# A Theory of Structured Propositions*

Andrew Bacon

August 24, 2022

**Abstract**

This paper argues that the theory of structured propositions is not undermined by the Russell-Myhill paradox. I develop a theory of structured propositions in which the Russell-Myhill paradox doesn't arise: the theory does not involve ramification or compromises to the underlying logic, but rather rejects common assumptions, encoded in the notation of the $\lambda$-calculus, about what properties and relations can be built. I argue that the structuralist had independent reasons to reject these underlying assumptions. The theory is given both a diagrammatic representation, and a logical representation in a novel language. In the latter half of the paper I turn to some technical questions concerning the treatment of quantification, and demonstrate various equivalences between the diagrammatic and logical representations, and a fragment of the $\lambda$-calculus.

Metaphysicians have often entertained the idea that reality has structure of some sort: that propositions, properties, and relations might have constituents that are structured in some way or other, and that this structure might do metaphysical work, for instance, by clarifying the relationship between the fundamental properties and relations and everything else, or the relationship between facts and their grounds.[1] In light of this, there has been a renewed interest in the 'Russell-Myhill paradox', an argument that purports to reduce the structural conception of reality to absurdity.[2] While I myself have doubts about whether a highly structured view is really necessary to do the sort of work that metaphysics needs, we cannot properly make this assessment without a clear view of the philosophical landscape.[3] The field of higher-order metaphysics is young—the program of mapping out the attractive and consistent hypotheses about the structure of reality is still in its early stages, and a paradox-free theory of structure seems key to completing this picture.

Extant theories of structured propositions typically take a "quasi-syntactic" approach to structure. The structure of propositions and properties are modeled on the structure of sentences and predicates in a language.[4] This paper provides a different model of structured

*Many thanks to two anonymous referees at this journal for many helpful suggestions about presentation and content, and to Arthur Wu for spotting several typos.

[1]In the analytic tradition, this idea of structure plays a prominent role in the thinking of the logical atomists (Russell (1940), Wittgenstein (1961)), and many versions of this idea have been articulated since; for two recent and different appeals to some sort of structure in reality for metaphysical theorizing see Sider (2011) and Fine (2012).

[2]See Dorr (2016), Goodman (2017), Fritz (forthcoming) for recent papers on the Russell-Myhill paradox that draw out these implications for metaphysical theorizing. The paradox traces back to Russell (1937) appendix B and Myhill (1958).

[3]I have laid out my preferred account of fundamentality and metaphysical definability in this journal, within a framework that posits far less structure in reality than posited by the structuralist (Bacon (2020)).

[4]See King (2008) for a survey of such approaches. Even those sensitive enough to the threat of paradox to offer consistency proofs will often construct 'quasi-syntactic' models of their theories; see for instance Fritz et al. (2021).

propositions, a fundamentally pictorial one. A relation is represented by a diagram containing holes into which diagrams representing other entities can be slotted. Among other things, the pictorial view provides a very simple diagnosis of the Russell-Myhill paradox: the principles refuted by that paradox, while a commitment of the quasi-syntactic viewpoint, are simply unsound in the pictorial theory.

Section 1 begins by distinguishing our investigation of structured theories of reality from adjacent theories concerning structured representational objects which may very well be best modeled syntactically. In section 2, I introduce the pictorial theory in terms of 'relational diagrams' and use it to give a preliminary diagnosis of the Russell-Myhill paradox in section 3. Sections 4, 5 and 6 concern more traditional logical representations of propositions using $\lambda$-notation. It is argued that while the standard theories governing $\lambda$ due to Church and Curry build in substantive metaphysical assumptions that make it unsuitable for theorizing about structured relations, one can obtain a structure friendly formalism by restricting the possible $\lambda$-terms in a natural way. The resulting theory trivially validates the standard rules governing $\lambda$: $\lambda$-abstraction corresponds to poking a hole where a constituent used to be in a relational diagram, and application corresponds to filling that hole again with a constituent. By contrast, the standard rules are invalid on the quasi-syntactic model leaving the interpretation of $\lambda$-terms highly unconstrained.[5] In section 7, I turn to the treatment of quantification and identity in this theory, and section 8 explains how the theory avoids the Russell-Myhill paradox. Some conjectures and theorems concerning the consistency of this theory are formulated.

# 1 Metaphysical and Representational Structuralism

Theories of structured propositions have been brought to bear on a variety of issues in philosophy, and the motivations and goals of these theories can be materially different. So it is prudent to begin by distinguishing these motivations and goals. I will use the term *proposition* to mean, roughly, whatever stands to a sentence or a thought as a person might stand to a name or concept for that person (e.g. Cicero to the name 'Tully'). Putting it somewhat glibly, a proposition, as I will use the term, is a part of reality, not merely a way of representing reality.[6] We may distinguish, then, between *representational structuralism* and *metaphysical structuralism*. Representational structuralism is the view that representational objects, including sentences, thoughts, modes of presentation, concepts and so on, are structured.[7] Metaphysical structuralism, by contrast, is a metaphysical view which maintains, superficially, that reality itself —

[5]Some problems concerning $\lambda$ on the quasi-syntactic model are also discussed in Dorr (2016) p58 and Bacon (2020) p562 (although see Hodes (2015) p392 for an alternative understanding of the syntactic view which keeps the standard rule governing $\lambda$). The pictorial view will also avoid awkward questions that arise for the quasi-syntactic view, such as 'what are the propositional constituents corresponding to bound variables on the quasi-syntactic view?'

[6]I am not entirely happy with this gloss. On the most flatfooted way of understanding what it is to be a 'part of reality', sentences and other representational objects are just as much parts of reality as propositions are. I think the distinction has more to do with which things are doing the representing and which things are being represented, but I think the present gloss does enough to make clear what the distinction I am talking about is.

[7]The thesis that sentences and other linguistic items are structured is not particularly controversial. However many theorists posit entities that mediate between language and reality — modes of presentation, mental representations, LFs and so on. Others theorize in terms of the word 'proposition', but treat them as ways of representing the world, as opposed to parts of the world itself. I count all of these entities as representational, and the view that they are structured as instances of representational structuralism. Thus many prominent defenders of structured "propositions" — including Soames (2013), or King (1996) — will count as representational structuralists by my accounting. Soames, for instance, talks about distinct but 'representationally identical' propositions: a contradiction in terms, according to my use of 'proposition'.

propositions, properties, individuals, and so on, are built up out of simpler constituents or are themselves simple.

Structuralism is typically invoked for two sorts of reasons. One motivation is to account for certain attitude reports. Perhaps Max is not very good at arithmetic, so that we might want to assert 1, but deny 2

1. Max believes that $117 = 117$.

2. Max believes that $117 = 98 + 19$.

The advocate of a structured theory of propositions might recommend their theory as way of accommodating these judgments. For if the proposition that $117 = 117$ and the proposition that $117 = 98 + 19$ are different — $98 + 19$ is structured while 117 is simple — then there is not even a *prima facie* obstacle to accepting 1 whilst rejecting 2.

The other is more theoretical in nature, and arises from the apparent need to be able to theorize about the hierarchical structure of reality. Several metaphysical frameworks call for very fine distinctions between properties and propositions. For instance, according to some philosophers, the property of being a vixen is less fundamental than the properties of being a fox and being female: the former is built, or metaphysically definable, from the latter two. The most fundamental properties and relations, according to this picture, are then the metaphysically simple entities, that are not built out of any constituents. Metaphysical structuralism provides a convenient model of propositions and properties in which to draw these sorts of distinctions.[8]

As I see it, the argument from attitude reports at best provides a motivation for representational structuralism. Representational structuralism can be invoked to resolve the problem of attitude reports in several ways. Perhaps the objects of belief are not propositions in my metaphysically loaded sense — the sort of things that stand to sentences as people stand to names. Or perhaps they are, but we only stand in attitudinal relations to them relative to another contextually salient parameter — a mode of presentation — which is structured.[9] One reason to resist the parallel argument for metaphysical structuralism is simply this: perhaps the concept, or mode of presentation '117' and '$98 + 19$' are different because the former is simple, while latter is not: it contains three constituents '98', '+', and '19. But the *number* $98 + 19$ is not structured: it does not contain 98, or the operation of addition as constituents. To say otherwise would be mathematically revisionary: if $98 + 19$ had three constituents, and 117 did not, they would be distinct by Leibniz's law — yet it is a mathematical fact that they are the very same number.

The key point here is that our purpose is to develop a version of metaphysical structuralism, and the result should not be held up to the demands or goals of a representational structuralist. In particular, a metaphysical structuralist has no special reason to think that a complex term (such as '$98 + 19$') corresponds to something complex in reality, that a simple term corresponds (such as 'vixen') to something simple in reality, or that the structure of reality is necessarily revealed by the structure of language at all.[10]

# 2 Relational Diagrams

In this section, I introduce a purely pictorial way to represent structured propositions, properties and relations: *relational diagrams*. The decision to begin this way is partly motivated by the

[8]Albeit not the only one. For non-structuralist accounts of reality that allow for similar theoretical work, see Dorr (2016) and Bacon (2020).

[9]See, for instance, Crimmins and Perry (1989), Richard (1990), Goodman and Lederman (2021).

[10]One might argue that the simple term 'vixen' expresses a metaphysically complex property involving the properties of *being female* and *being a fox*.

desire to cement intuitions early on, but also to emphasize that the theory I am developing is really a precisification of an already existing practice for picturing structured propositions.[11] Subsequent choices in the formalisation of the view, which might look arbitrary when stated in a linear written notation, will fall directly out of this picture. Secondly, relational diagrams enjoy one nice feature which more conventional written notations lack. There is a one-to-one correspondence between the relational diagrams and the structured propositions and relations they are intended to represent. Our written notations developed later, while more amenable to familiar logical analysis, will not have this feature — distinct representations will denote the same structured entity.

A structured entity is either simple, or made out of smaller immediate constituents, themselves structured entities. Only certain structured entities can be combined to form complex wholes. For instance, *is tall* and *Alex* may be combined to make the structured proposition *Alex is tall*, and *it's not the case that* can be combined with *snow is white* to make *it's not the case that snow is white*. But the operator *it's not the case that* and *Alex* cannot be combined, nor can *is tall* with *snow is white*. This fact is reflected in the language used to denote these entities — there is no way of grammatically combining the name 'Alex' with the operator expression 'it's not the case that', or 'is tall' with 'snow is white'. In both these cases the way in which the two entities are combined — a proposition with an operator, or an individual with a property — is the same, and we will call this mode of combination *application*. (We will encounter other modes of combination later.)

Type theory provides a convenient framework for systematizing facts about which entities can and cannot be combined by application. According to this framework, every entity is assigned a *type*. There are types $e$ and $t$ for entities denoted by names and sentences, respectively. For convenience we refer to type $e$ entities as individuals, and type $t$ entities as propositions.[12] Every other type has the form $\sigma \to \tau$, where $\sigma$ and $\tau$ are types. $\sigma$ is the type of entities that entities of type $(\sigma \to \tau)$ can combine with, and $\tau$ is the type of entity that would result from such a combination. We follow the standard convention of associating brackets to the right, writing $\sigma_1 \to \sigma_2 \to ... \to \sigma_n \to \tau$ instead of $(\sigma_1 \to (\sigma_2 \to ... \to (\sigma_n \to \tau)...))$. Thus for instance, we will call entities of type $t \to t$ *operators*, because when applied to a proposition they yield another proposition, and entities of type $e \to t$ will be called properties because they can be applied to individuals to form propositions. A binary relation, like *loves*, has type $e \to (e \to t)$ because it can be applied to two individuals in succession to form a proposition.[13] We shall focus on the *relational types*: $e$ is a relational type, and types of the form $\sigma_1 \to ... \to \sigma_n \to t$ where $\sigma_1 ... \sigma_n$ are also relational types are relational types (when $n = 0$ we see that $t$ is a relational type). All entities of relational type are thus (possibly nullary) relations between entities of types $\sigma_1, ..., \sigma_n$.[14] We will frequently write a colon, ':', to mean 'has type' — for instance '$F : e \to t$' means '$F$ has type $e \to t$'.

According to this theory, propositions contain constituents which appear a certain number of times and in a certain order within that proposition. We will depict propositions by completely

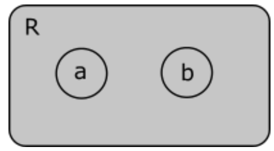
[11] See, for instance, Dixon (2018).

[12] One should be careful with these glosses: as Prior (1971) chapter 2 argues, quantification into sentence position is not accurately paraphrased by singular quantification over propositions. Prior himself, for instance, is a nominalist and rejects propositions, yet is still willing to quantify into sentence position.

[13] One quirk of this choice is that there is an asymmetry in the way that binary relations are treated. Plugging an individual $a$ into $R$s first argument place to produce a unary property is succinctly notated by $Ra$, whereas there is no direct way to notate the result of plugging $a$ in $R$s second argument place. Shortly we will see that it can be represented by the more complex $\lambda$ term $\lambda x.Rxa$. This asymmetry shouldn't be seen as reflecting an important metaphysical asymmetry.

[14] This restriction is not entirely unnatural given our earlier remark about non-relational operations like $+$ (which has the non-relational type $e \to e \to e$). We will for the most part assume that individuals are completely unstructured.

greyed-out boxes. If $R$ is a simple binary relation (i.e. a relation of type $e \to e \to t$) and $a$ and $b$ are simple individuals (i.e. entities of type $e$) then $Rab$ represents the proposition that $a$ $R$s $b$ (e.g. *Alice loves Bob*), and its relational diagram is:




It is tempting to say that the leftmost constituent is $a$ and the rightmost constituent is $b$. However *left* and *right* are clearly properties of physical inscriptions. Geometrical properties like left and right make no more sense applied to propositions than they do applied to numbers. A single proposition can be expressed in more than one language. Some languages use different written conventions, for instance in Hebrew one can express the same proposition one would express in English with the opposite ordering of words — thus the same proposition can be expressed by sentences in which the name for *Alice* appears to the left, or to the right of *Bob*. While the chirality of a name in a sentence is clearly an artefact of a languages particular conventions, we shall adopt the assumption — indeed, the defining assumption of structuralism — that there is a language independent ordering of the constituents $a$ and $b$. Given two propositions, $Rab$ and $Sdc$, we will assume that we can make interpropositional comparisons of order — for instance, that $a$ has the same position in $Rab$ as $d$ has in $Sdc$, even if that is depicted as 'left' in some representations and 'right' in others. Provided we are mindful of the fact that 'left' and 'right', and 'before' and 'after', are only meaningful relative to a language or pictorial representation, however, it is harmless and convenient to use chiral talk in relation to propositions.

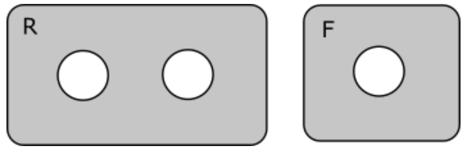
The idea that one can make interpropositional comparisons of order corresponds to what Kit Fine calls the 'standard' theory of structured propositions and relations. But it is worth noting that there are also less structured views that could be given diagrammatic representations. One could imagine a variant of our diagrams in which each is presented on a (separate) transparency instead of paper or a computer screen, with no instruction about which way up it is to be presented. In which case interpropositional notions of constituency, betweenness and adjacency would make sense, but order would become meaningless: the former relations are preserved after turning a transparency over, whereas the latter relation is not. One could perhaps envision higher-dimensional objects in which even these notions are meaningless.[15]

The relation $R$ is also a constituent of this proposition, but our pictorial representation does not depict it as being before, after or between $a$ and $b$. Evidently our chosen linear notation has to take a stand on this: we have adopted the prefix convention of writing $R$ before the arguments $a$ and $b$. But even in a language that reads left to right, other notational conventions are also possible, including infix notation, $aRb$, often used with symbols like $=$ and $\wedge$, or postfix notation $aF$, $baR$ (analogous to the conventions used by the mathematical collective Bourbaki). The fact that $R$ has a position relative to $a$ and $b$ could also be argued to be a notational artefact, forced by the typographic constraint that a relation symbol has to appear somewhere. I do, in fact, take this position, and will consider an argument that $R$ has has no straightforward position relative to $a$ and $b$ shortly.

Let us turn from structured propositions to structured relations and properties. I will adopt a broadly Fregean picture of properties and relations as unsaturated propositions: we may think of them as propositions with holes poked into some of the argument places. For instance, we

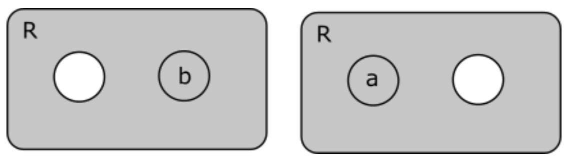
[15] A theory of structured propositions where none of these notions are meaningful is described in Fine (2000), and given mathematical models in Leo (2008) and Leo (2010).

can depict our simple binary relation $R$, and a simple unary property using relational diagrams as follows:



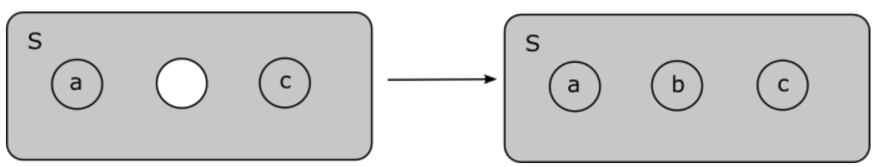

Holes, like constituents, have positions, and may be said to appear to the left or to the right of other holes and constituents. In the present example, there is a left and a right hole, and no other constituents or holes.[16]

Non-simple properties and relations can be built by plugging arguments in to holes. For instance, if $R$ is a simple relation, *loves*, then you can form a unary property by plugging *Alice* or *Bob* into either hole, creating the complex unary properties *Alice loves* and *loves Bob*:




Both are unary properties, and can be applied to an argument $c$ in only one way to make $Rac$ and $Rcb$ respectively.

It should now be clear why our theory does not assign properties and relations positions in predications like $Rab$ and $Fa$. Consider a complex unary property, $Sa \cdot c$, built from a ternary relation $S$, of type $e \to e \to e \to t$, and two individuals $a$ and $c$:




when a third individual $b$ is plugged into its only hole, some constituents of $Sa \cdot c$ appear to the left of $b$ and other constituents of $Sa \cdot c$ appear to the right. It seems just as wrong to say that $Sa \cdot c$ is to the left or right of $b$ as it does to say that California is to the east or west of Sacramento.

In fact, this feature of the view strikes me as hard to avoid in a structured theory of propositions. Consider a unary predication $Fa$. If $F$ has an order relative to $a$, then there's only two options: it appears before the argument, $a$, or after it. The view that properties appear before their arguments is inconsistent with some other natural assumptions about structure: we have the following inconsistent quartet (a completely parallel argument establishes the inconsistency of the view that unary properties appear after their arguments, so I will not consider it separately):

1. When you apply a unary property to an argument, the property appears before the argument in the resulting structured proposition.

[16] Like physical holes, however, it makes no sense to talk about swapping the two holes in this diagram to make a new one. This makes for an important disanalogy between holes and constituents, which can be swapped. If we wished, we could adopt a more cumbersome way of talking that didn't treat holes as entities in their own right.

2. If a constituent $c$ occurs before another constituent $c'$ in a proposition $p$ and they have no constituents in common, then every constituent of $c$ appears before every constituent of $c'$ in $p$.

3. (a) There is a unary property, $F$, whose only constituents are *loves* and *Bob*, which when applied to *Alice* yields the proposition that *Alice loves Bob*.

   (b) There is a unary property, $G$, whose only constituents *loves* and *Bob*, which when applied to *Alice* yields the proposition that *Bob loves Alice*.

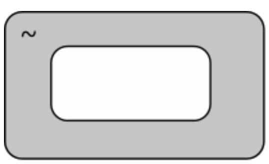
4. Either (i) for any $a$ and $b$, $a$ appears before $b$, but not conversely, in *a loves b*, or (ii) for any $a$ and $b$, $b$ appears before $a$, but not conversely, in *a loves b*.

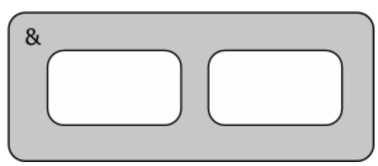
Here we are taking *before* as a primitive ordering of the constituents of a proposition; 4 is stated so as to be neutral about the relation between this ordering and left-to-right order of written English. I understand 'constituent' to mean an improper constituent, so that every entity counts as a constituent of itself.

4 presents two cases. Start by assuming 4i, and thus that *Alice* appears before *Bob* in *Alice loves Bob*. By 1, the property $F$ appears before *Alice* in the proposition that results from applying $F$ to *Alice* which, by 3a, is the proposition that *Alice loves Bob*. By 2 every constituent of $F$ appears before *Alice* in *Alice loves Bob*. 3a states the constituents of $F$ are *Bob* and *loves*, so *Bob* appears before *Alice* in *Alice loves Bob*. But this contradicts 4i, which states that *Bob* does not appear before *Alice* in *Alice loves Bob*. If 4ii is true, a completely a parallel argument, involving the property $G$, can be given.

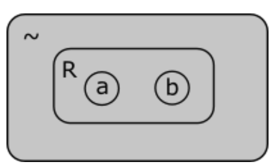
I take 4 to simply be an articulation of the linear structured view. 3a and 3b state the existence of the unary properties *Bob loves* and *loves Bob*, respectively. And 2 is simply a plausible principle about constituent order. Indeed, without 2, there is no well-defined notion of order by which one can compare the immediate constituents of immediate constituents.[17]

Until now we have only depicted first-order properties and relations. Operators, like negation, can also be represented using the same sorts of pictures: if a unary property is a proposition with an individual shaped hole, an operator is a proposition with a proposition-shaped hole. Here are relational diagrams for negation and conjunction:



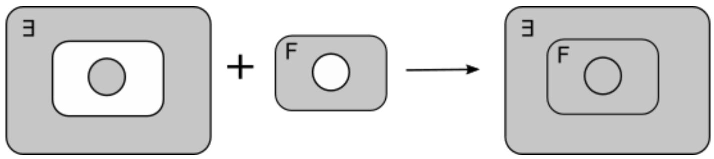

Plugging the proposition, *Rab* into negations hole, for instance, will yield:




More generally, we can think of an $n$-ary relation of type $\sigma_1 \to ... \to \sigma_n \to t$ as a proposition with $\sigma_1 ... \sigma_n$ shaped holes. To keep our diagrams clean, we will indicate the type of a hole in the surrounding text rather than writing it explicitly into the diagram.

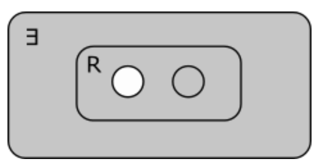
[17] The notion of a constituent appearing *before* another in a proposition is well-defined provided the proposition contains at most one occurrence of each constituent, as in the above argument. Otherwise one needs to invoke the notion of an *occurrence* of a constituent.

Diagrams involving higher-order properties applied to properties require special note. Since the first-order existential quantifiers are often represented as a higher-order property — the property a property has when it is instantiated — I shall use that as my running example. The existential quantifier *something* thus has type $(e \to t) \to t$, and we depict it as a proposition with an $(e \to t)$ shaped hole. In general, a hole of shape $\sigma$ is what you get from taking a diagram for a simple entity of type $\sigma$ and inverting grey and white (and, of course, removing any identifying labels like $R$ or $F$). The result of plugging the hole of $\exists$ with a unary property $F$ of type $e \to t$, e.g. *walks*, results in the following:



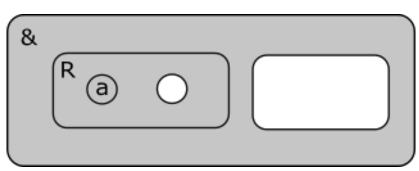

Note that the hole associated with $F$ is not filled with a constituent, but is still greyed out. While it is a hole of the constituent property *walks*, it is not a hole in the resulting proposition, *something walks*. Evidently, *something walks* is a completely saturated proposition: it shouldn't have any holes. Informally, we can think of the greyed out hole as being 'bound' by the higher-order property.[18] If this feature of our diagrams seems unfamiliar, compared to the use of similar diagrams in metaphysics, it is because higher-order properties are rarely depicted in this fashion.

It is not possible to apply the higher-order property of existence $\exists_e : (e \to t) \to t$ to a binary relation *loves*, $R : e \to e \to t$, since the former only accepts unary properties. But they can be combined in another way distinct from application that yields a *property* (as opposed to a proposition) as output: the property of loving something. Grey always fills the rightmost holes:




More generally, if you have a higher-order relation $R$ between things of type $\sigma_1 ... \sigma_k ... \sigma_n$, and another relation $S$ of type $\rho_1 \to ... \to \rho_i \to \sigma_k$, you can plug $S$ into the $k$th argument hole of $R$ to form a relation between things of type $\sigma_1, ..., \sigma_{k-1}, \rho_1, ..., \rho_i, \sigma_{k+1}, ..., \sigma_n$. You depict this by plugging the picture of $S$ into the $k$th hole in $R$, and greying out all but the first $i$ holes appearing in $S$.

Here is a more complicated example:




This can be made by plugging $a$ into $R$s first hole, making the unary property $Ra$. Conjunction accepts things of type $t$ as arguments for application, but we can use our more general mode of combination to plug $Ra$ into conjunctions first slot, yielding the $e \to t \to t$ relation of being an $x$ and a $p$ such that Alice loves $x$ and $p$.

[18] Note that all higher-order properties 'bind' holes in this way; it is not special to quantifiers.

The collection of diagrams representing individuals, propositions, properties and relations can be more precisely characterized as follows. First we shall define the sorts of diagrams that are associated with *simple* entities: individuals and simple propositions, properties and relations. Below *the hole of shape* $\sigma$ refers to the result of switching the colours grey and white in the diagram of a simple entity of type $\sigma$ and removing the label.

1. The diagram for an individual (entity of type $e$) is a grey circle, labeled by a name for the individual within the circle.

2. The diagram for a simple proposition (entity of type $t$) is a grey rectangle, labeled by a name for the proposition within the rectangle.

3. The diagram for a simple relation of type $\sigma_1 \to ... \to \sigma_n \to t$ consists of a grey rectangle with holes of shapes $\sigma_1 ... \sigma_n$, in that order, labeled by a name for that relation in the outermost connected region of grey.

The last clause is inductive because we may assume we know what diagrams, and thus also holes, are associated with $\sigma_1 ... \sigma_n$. Labels will typically be drawn from a signature of non-logical constants of some language.

Once each simple proposition, property, relation and individual has a relational diagram associated with it of the appropriate type, we have one rule for constructing diagrams for complex propositions, properties and relations.[19]

4. If you have a diagram $d$ which has holes of shape $\sigma_1 ... \sigma_k ... \sigma_n$ in that order, and another diagram $d'$ that has holes of shape $\rho_1 ... \rho_i ... \rho_j$ where $\rho_{i+1} ... \rho_j$ are the types of the holes in a relation of type $\sigma_k$, then you can plug $d'$ into $d$s $k$th hole, greying out the holes corresponding to $\rho_{i+1} ... \rho_j$ to form a relation with holes of type $\sigma_1 ... \sigma_{k-1}, \rho_1 ... \rho_i, \sigma_{k+1} ... \sigma_n$ in that order.

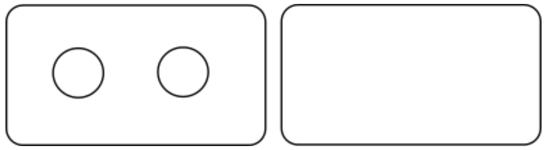
The holes *of* a relational diagram $d$ (or, the holes the diagram *has*) are the holes of shape $\sigma$ for some type $\sigma$ that are (i) contained in $d$ and (ii) are maximal in the sense that they are not properly contained in any other holes appearing in $d$.[20] The holes of a relational diagram thus do not overlap and are linearly ordered by the left-to-right order of appearance in the relational diagram, so we are able to unambiguously talk about the 'first hole', 'second hole', etc. of a relational diagram. Note that the operation of simply plugging an argument into the $k$th hole of a relation (without introducing any new holes) is a special case of our rule, where $i = 0$.

Relational diagrams make the structure of propositions and relations especially clear. There is always a *main* relation, corresponding to the outermost greyed out box: a metaphysically simple relation, analogous to the main connective of a propositional formula, which may have some of its holes filled with other structured entities, possibly contributing further holes. These entities may be called the *immediate* constituents of the proposition. These further entities, if they are not individuals, will similarly have a main relation and immediate constituents, which appear nested in the overall diagram, and so on.

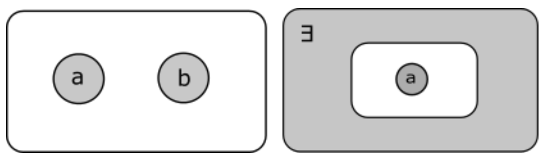
[19] Thus we obtain a general correspondence between relational diagrams and relations of the same type. The correspondence clearly depends on the simple relations we have associated with each simple relational diagram. Rule 4 extends this association to complex relational diagrams and complex relations, but to do this we have need to pick one of the two ways to align the ordering that orders holes and constituents in relations with the left-to-right ordering in our pictorial representations. Thus each relational diagram corresponds to a unique relation of matching type relative to a (type preserving) assignment of simple relations to simple relational diagrams, *and* a choice of aligning the propositional constituent ordering with the left-right order on diagrams.

[20] Thus, for instance, a simple entity of of type $((e \to t) \to t) \to t$ will contain a hole of shape $(e \to t) \to t$ (see the image associated with $\exists$ above, but with grey and white inverted), and this hole itself contains a hole of shape $e$. Only the former is a hole of the entire entity, since it is a unary property.

It's instructive also to consider diagrams that cannot be constructed from the above rules. For instance, it simply isn't possible to construct a diagram that doesn't have any constituents, or is made entirely from holes, as illustrated below.



Similarly, it's impossible to construct a diagram that doesn't have a main relation — for instance, one cannot have a hole where the main relation ought to be, as illustrated in the first diagram below. Since entities are constructed by plugging things into holes, to construct such an entity, one would have to start with the illegitimate entity made entirely out of holes depicted above and fill the two individual holes with $a$ and $b$ respectively. The second diagram has a main relation, but given its hole has shape $e \to t$, there should not be a label on the innermost grey circle.




This concludes our informal description of relational diagrams. There is clearly room for more precision in the description of these entities; later we will present a more conventional logical notation for denoting structured entities and establish a many-one correspondence between them and relational diagrams that clears up any remaining vagueness. However, despite being less amenable to the usual methods of the philosophical logician, the relational diagrams enjoy one advantage over standard logical notations: according to the structured view under consideration, there is a one-to-one correspondence between relational diagrams and relations[21], whereas there will in general be many different ways to express the same entity in logical notation. Intuitively, a logical expression may be thought of as a set of instructions telling you how to build a structured entity, and two instructions can differ over the order in which it tells you to put pieces together whilst still yielding the same result (just as, say, different lego instructions can produce the same lego construction). This represents a fundamental divergence between the present pictorial theory and structured theories of propositions modeled on language: the structure of reality is not directly reflected by the structure of linguistic expressions, but only by the linguistic expressions up to a notion of equivalence. Linguistic representations will thus tend to obscure the true structure of reality. We illustrate this point next by giving a straightforward diagnosis of the 'Russell-Myhill paradox'.

# 3 The Russell-Myhill Paradox: A Preliminary Diagnosis

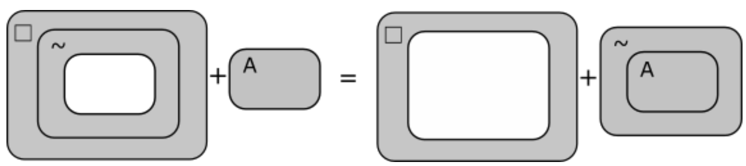
Several philosophers have recently argued that the theory of structured propositions is untenable on the grounds that it is inconsistent. For example, Dorr (2016) points out that the following principle, formulated in a standard higher-order language with '$\lambda$-terms' (to be described in the next section), is inconsistent with very minimal logical assumptions:

[21]Relative to a way of associating simple diagrams to simple relations of matching type, and a choice about which way we associate constituent order with the left-to-right ordering on diagrams. See footnote 19.

**Predicate Argument Structure** $Fa = Gb \to F = G \wedge a = b$

Here $F$ and $G$ can be any predicative terms (possibly open) expressing relations with type $\sigma \to \tau$ and the arguments $a$ and $b$ must express entities with type $\sigma$.[22] Similar arguments against structured principles are put forward in Hodes (2015), Uzquiano (2015), and Goodman (2017); all these arguments trace back to Russell (1937) appendix B.

According to many theories of structured propositions, propositions have a quasi-syntactic structure, with the structure of properties and propositions being modeled on the structure of predicates and sentences of a language. Predicate Argument Structure is a commitment of the most straightforward implementation of that idea: two subject-predicate sentences '$Fa$' and '$Gb$' cannot be the very same sentence unless the predicates '$F$' and '$G$', and the names '$a$' and '$b$' are the same. However we have already said enough to see that Predicate Argument Structure is not a commitment of our pictorial theory of structured propositions. Consider the proposition $\Box(\neg A)$. We have two distinct ways of building this proposition. We can plug the proposition $A$ into negation's single hole, and then plug the resulting proposition, $\neg A$, into necessity's single hole. Alternatively we can put $\Box$ and $\neg$ together to get the structured operator $\Box\neg$, obtained by plugging $\neg$'s diagram into the only hole in $\Box$'s diagram leaving a diagram with a single hole. And we can then plug the proposition $A$ into that final hole. Either way we get the same result:




But Predicate Argument Structure says that there's only one way to make a proposition by applying an operator to a proposition. It entails the following pair of identities: The composite operator $(\Box\neg)$ is identical to $\Box$, and $A$ is identical to $\neg A$. Neither identity holds on the present pictorial theory of structured propositions (the latter is simply inconsistent). So this structuralist should not accept Predicate Argument Structure.

There are other routes to paradox, for instance in Bacon (2020) the following weaker principle is considered:[23]

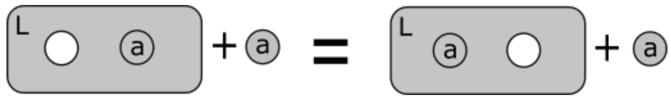
**Predicate Structure** $Fa = Ga \to F = G$

Again, understanding this as schematic in both terms $F, G : \sigma \to \tau$ and $a : \sigma$, and in the types $\sigma$ and $\tau$. This principle does not imply the aforementioned identities, since in this principle the argument of the two predications in the antecedent, $Fa$ and $Ga$, are the same, so it is not susceptible to our counterexample to Predicate Argument Structure for there we exploited the fact that $\Box\neg A$ could be decomposed into an application in two different different ways with $\neg$ appearing as a consistuent of the predicate and of the argument respectively. Unfortunately Predicate Structure is also inconsistent by a Russell-Myhill style argument.

But the structuralist also has independently rejected Predicate Structure. I can make the proposition that *Alice loves Alice* in two ways: I can plug *Alice* into the first argument hole, yielding *loves Alice*, and apply the resulting unary property to *Alice*. Or I can plug *Alice* into the second argument hole, yield *Alice loves*, and apply the resulting unary property to *Alice*. Thus:

[22] In the systems in which this schema is formulated one may infer universal generalizations like $\forall XY \forall xy(Xx = Yy \to X = Y \wedge x = y)$, quantifying into both predicative position $\sigma \to \tau$ as well as the position of the argument of type $\sigma$.

[23] There it is called simply Structure, but this is in conflict with Dorr's naming conventions.




But we should not conclude that the unary properties expressed by *loves Alice* and *Alice loves* are identical: perhaps Alice loves people who don't love her back, or there are people she doesn't love who love her. Moreover, the diagrams for both relations are clearly different, having an empty hole in different positions.[24]

So, already, we are in a position to see that the present theory has the means to resist Russell-Myhill style reasoning in a principled manner. This is not the end of the matter, of course: we have not said enough to even raise the question of consistency of the present theory. First we will describe the type of system that Dorr, Goodman and others use to formulate their inconsistency results; it is based on Church and Curry's work on type theory, and we will see that it encodes a different, more liberal theory of propositions and relations which the present structuralist rejects. Once this is done we will be able to give a more fine-grained analysis of the inconsistency results of Dorr (2016) and Goodman (2017), which are formulated in these languages.

# 4 Church's system of higher-order logic

There is a now standard formalism for representing properties and relations in a typed language. The $\lambda$-calculus, originally developed by Church, is formulated in terms of variables and a syncategorematic device, $\lambda$, for forming expressions denoting properties and relations. Less widely used, but of equal expressive power, is the combinatory calculus of Curry (1930), which eschews variables altogether and achieves the effect of Church's $\lambda$ by other means. Underpinning both of these formalisms, however, is a substantive metaphysics of properties and relations: a metaphysics, I shall argue, that the present structuralist does not accept. In this section I will outline Church's framework. In the next I will show that this framework, and the related framework of Curry, are simply not suitable for theorizing about structured propositions: many of the expressions that appear in these formalisms are meaningless by the pictorial structuralist's lights — they do not correspond to properties or relations.

Church's $\lambda$-calculus is a typed language: it is comprised of a collection of expressions each possessing one of the types described in section 2. The basic expressions consist of a collection of logical and non-logical constants, and then there are various rules telling us how to create complex terms. These complex terms are constructed from the constants using variables and the device $\lambda$, to be explained detail below. In an interpreted language of this kind, each constant will have an entity of the corresponding type as its meaning – a sentence a proposition, a predicate a property, and so on. And, provided reality cooperates, each way of constructing complex terms from simpler terms corresponds to a way of putting together entities, so that every expression of the $\lambda$-calculus is meaningful. The rules for creating complex expressions from simpler ones have been broken down into a collection of primitive rules described in table 1.[25] These rules deliver statements of the form 'the term $M$ of type $\tau$ may be made from free

[24] Another variant of Structure, suggested to me by Lavinia Piccollo, replaces Dorr's conjunction with a biconditional:

$Fa = Gb \rightarrow (F = G \leftrightarrow a = b)$

While this principle is compatible with the two decompositions of $\Box(\neg p)$, it similarly implies that *loves Alice* and *Alice loves* are identical.

[25] We use $\Gamma$ and $\Delta$ to stand for arbitrary sequences of type assignments, $x_1 : \sigma_1, ..., x_n : \sigma_n$ where $x_1...x_n$ are a finite sequence of pairwise distinct variables. To preserve the pairwise distinctness required of type assignments,

| | | | |
|---|---|---|---|
| $\dfrac{}{x{:}\sigma \vdash x{:}\sigma}$ | **Identity** | $\dfrac{\Gamma \vdash M{:}\sigma \to \tau \qquad \Delta \vdash N{:}\sigma}{\Gamma,\Delta \vdash (MN){:}\tau}$ | **Application** |
| $\dfrac{\Gamma, x{:}\sigma, y{:}\tau, \Delta \vdash M{:}\rho}{\Gamma, y{:}\tau, x{:}\sigma, \Delta \vdash M{:}\rho}$ | **Exchange** | $\dfrac{\Gamma, x{:}\sigma \vdash M{:}\tau}{\Gamma \vdash \lambda x M{:}\sigma \to \tau}$ | **Abstraction** |
| $\dfrac{\Gamma \vdash M{:}\tau}{x{:}\sigma, \Gamma \vdash M{:}\tau}$ | **Weakening** | $\dfrac{}{\vdash c{:}\tau}$ | **Constants** |
| $\dfrac{\Gamma, x{:}\sigma, y{:}\sigma, \Delta \vdash M{:}\tau}{\Gamma, z{:}\sigma, \Delta \vdash M[z/x][z/y]{:}\tau}$ | **Contraction** | | |

Table 1: Natural deduction for Curry typing

variables $x_1...x_n$ of types $\sigma_1...\sigma_n$'. Statements of this form will be called *type assertions* and will be abbreviated $x_1 : \sigma_1, ..., x_n : \sigma_n \vdash M : \tau$.[26]

A complete understanding of the finer mechanics of this system will not be that important to the reader. The following examples illustrate the primary function of each rule, by showing what further terms each rule allows one to create from a given binary relation $R : e \to e \to t$, interpreted as *loves*.

1. The rule Exchange plays a crucial role in forming 'converses' like $\lambda xy.Ryx$ — *is loved by* — in which $\lambda$ binds variables in a different order to the order in which they appear in the body of the term.[27]

2. The rule Contraction allows one to form reflexivized properties like $\lambda x.Rxx$ — *loves oneself* – where $\lambda$ binds a variable that appears multiple times in the body of the term.[28]

3. The rule Weakening allows one to form vacuous properties like $\lambda x.Rab$ — *being an x such that John loves Mary* — where $\lambda$ abstracts a variable that doesn't appear in the body of the term.[29]

4. The rule Identity allows one to form 'combinators': terms made entirely out of bound variables and $\lambda$ — for instance $\lambda p.p$. Combined with Exchange, Contraction or Weakening respectively one can form combinators corresponding to each of the above examples: $C = \lambda X \lambda y \lambda z.Xzy$, $W = \lambda X \lambda y.Xyy$, and $K = \lambda p \lambda y.p$, where $X : e \to e \to t$, $p : t$,

the rule Application is thus understood so as to only apply when $\Gamma$ and $\Delta$ have no variables in common, Weakening when $x : \sigma$ does not appear in $\Gamma$, and Contraction when $z : \sigma$ does not appear in $\Gamma$ or $\Delta$.

[26] This system for typing terms originated with Curry in the 1930s, and following Milner (1978) is often used by computer scientists who want more flexibility in assigning types to variables. Our variant does *not* have this flexibility: we have built the types into the variables: for us $x_1 : \sigma_1, ..., x_n : \sigma_n$ is only a legitimate type context if those variables in fact have those types. This means two things: (i) we do not have to annotate variables when we $\lambda$-abstract, and (ii) an unannotated $\lambda$-term has at most one type. The reasons we are using Curry's system, as opposed to Church's, is that it will later make available clean ways to restrict the legitimate $\lambda$-terms.

[27] An instance of Identity and Constants, $x : e \vdash x : e$ and $\vdash R : e \to e \to t$, lets you infer $x : e \vdash Rx : e \to t$ using Application. Using this and another instance of Identity ($y : e \vdash y : e$) we can infer $x : e, y : e \vdash Rxy : t$ again by Application. Exchange lets you move to $y : e, x : e \vdash Rxy : t$, and two applications of Abstraction let you get to $\vdash \lambda y \lambda x.Rxy : e \to e \to t$.

[28] Assuming we have already derived $x : e, y : e \vdash Rxy : t$ as above, Contraction lets you infer $z : e \vdash Rzz : t$, and an application of Abstraction let's you get to $\vdash \lambda z.Rzz : e \to t$.

[29] Assuming we have derived $\vdash Rab : t$ using Application and Constants, Weakening lets us infer $x : e \vdash Rab : t$, and an application of Abstraction let's you get to $\vdash \lambda x.Rab : e \to t$.

$y, z : e$. Informally, $C$ takes a relation and outputs its converse, $W$ takes a relation and outputs its reflexization, and $K$ takes a proposition and outputs the vacuous property *is such that p*. (There are variants of $C, W$ and $K$ for other types as well.)

5. The rules of Abstraction and Application let you introduce $\lambda$s into terms and apply terms to one another, and are essential if we are to construct any interesting terms involving $\lambda$. The reader should consult the above footnotes to get a feel for the role they play in the construction of the terms mentioned.

Within this language one can state various theories of properties and relations. In order to do this we need some logical apparatus: we assume that among the logical constants we have at least the classical conditional (a binary connective, $\to$ of type $t \to t \to t$), and for each type $\sigma$ a quantifier $\forall_\sigma$ of type $(\sigma \to t) \to t$ which expresses generality into the position of type $\sigma$, just as the first-order quantifier express generality in the position of a name. Applying $\forall_\sigma$ to a predicate $F : \sigma \to t$ forms a sentence $\forall_\sigma F$ of type $t$, saying that $F$ is universally satisfied. A minimal theory must at least include sentences that follow from the classical propositional and quantificational principles, understanding the latter to encompass the higher-order as well as the first-order quantifiers: I put these in a footnote.[30] (In this axiomatization, and henceforth, I employ several standard abbreviations: I will write $M_0M_1M_2...M_n$ instead of $(...((M_0M_1)M_2)...M_n)$, $A \to B$ instead of $\to AB$ and I will suppress $\lambda$s appearing immediately after quantifiers, writing $\forall_\sigma x.A$ instead of $\forall_\sigma \lambda x.A$. The remaining truth functional connectives can be introduced by definition, e.g. $\forall_{t\to t}\forall_t$ for $\bot$, $\lambda p.(p \to \bot)$ for $\neg$ and so on. Significantly, it is possible to define an identity operation, $=_\sigma$, at each type by the definition $\lambda xy.\forall_{\sigma\to t}(Xx \to Xy)$.)

The above axioms say very little about Church's special device for denoting properties and relations: $\lambda$. Further axioms governing $\lambda$, then, must be added that pin down the intended meaning of $\lambda$. The primary purpose of $\lambda$, of course, is to allow one to form complex predications. For instance, given predicates $F, G : e \to t$, meaning *is wise* and *is old* respectively, one can construct, using these rules the complex predicate $\lambda x.(Fx \wedge Gx) : e \to t$, meaning *is wise and old*. The principle $\beta$ essentially ensures that our complex predicates involving $\lambda$s behave as expected: for instance, it will tell us that $\lambda x.(Fx \wedge Gx)a$ is the same as $Fa \wedge Ga$ — for Socrates to be wise and old just is for Socrates to be wise and Socrates to be old. The principle $\eta$ lets you eliminate inessential uses of $\lambda$-abstraction: $\lambda x.Fx$ and $F$ are the same — i.e. *to be an x such that x is wise just is to be wise.*

To be more precise, a pair of terms of the form $(\lambda x.M)N$ and $M[N/x]$ are said to be *immediately $\beta$-equivalent*, with the caveat that $N$ is free for $x$ in $M$.[31] Two terms of the form $M$ and $\lambda x.Mx$ are immediately $\eta$ equivalent, when $x$ isn't free in $M$. A minimal theory, which I shall simply call $\mathsf{H}$, adds the further principles $\beta$ and $\eta$ to the above axioms, capturing the idea that immediate $\beta$ and $\eta$-equivalents are synonymous and thus intersubstitutable *salve veritate*.[32].

---

[30]

PC Every instance of a propositional tautology.

UI $\forall_\sigma F \to Fa$

MP From $A \to B$ and $A$ infer $B$

Gen From $A \to Fx$ infer $A \to \forall F$, provided $x$ doesn't appear free in $A$

[31]Informally, this qualification simply means that $N$ doesn't contain any free variables that get bound once $N$ is replaced with $x$. More precise definitions may be found, e.g., in Hindley and Seldin (2008).

[32]Note that these principles license the substitution of immediate $\eta$ and $\beta$-equivalents even in the scope of $\lambda$s binding variables appearing free in the terms being substituted.

$\beta$ $A \leftrightarrow A'$ where $A'$ is obtained from $A$ by substituting an occurence of $(\lambda x.M)N$ with $M[N/x]$ or conversely, provided $N$ is free for $x$ in $M$.

$\eta$ $A \leftrightarrow A'$ where $A'$ is obtained from $A$ by substituting an occurrence of $M$ with $\lambda x.Mx$ or conversely, provided $x$ isn't free in $M$.

$\lambda$ is a useful technical device, but unlike other useful philosophical notions — such as knowledge, disjunction, chance, and so on — it is not obviously something we have pretheoretic grip on. We appear to learn its meaning solely through the logical principles that relate it to antecedently understood expressions that don't involve it. A nice feature of $\beta$ and $\eta$ is that they pin down the meaning of the $\lambda$-terms uniquely.[33] Suppose that $\lambda'$ is another abstraction device which satisfies the analogue of $\beta$, which we'll call $\beta'$: $(\lambda' x.M)N$ may be substituted for $M[N/x]$ in any sentence *salve veritate*. Given $\beta'$ and $\eta$, one can derive the identity of $\lambda'$-terms and $\lambda$-terms. In particular, when $N = x$, $M[x/x] = M$, so $\beta'$ tells us that $(\lambda' x.M)x$ and $M$ are intersubstitutable. Thus we can infer from $\lambda x.M = \lambda x.M$ that:

$$\lambda x.(\lambda' x.M)x = \lambda x.M$$

Given $\eta$ (for $\lambda$) we can substitute $\lambda' x.M$ for $\lambda x.(\lambda' x.M)x$ in the above giving us:

$$\lambda' x.M = \lambda x.M$$

Our principles thus pin down the meaning of $\lambda$-terms entirely, given the background theory $\mathsf{H}$. (A completely parallel argument establishes the same conclusion from $\beta$ and $\eta'$ instead of $\beta'$ and $\eta$.[34])

From the self-identity statements $(\lambda x.M)N = (\lambda x.M)N$ and $\lambda x.Mx = \lambda x.Mx$ you can derive from $\beta$ and $\eta$ the following two identities:

$\beta^=$ $(\lambda x.M)N =_\tau M[N/x]$ provided $N$ is free for $x$ in $M$.

$\eta^=$ $\lambda x.Mx =_{\sigma\to\tau} M$ provided $x$ isn't free in $M$.

These will be useful subsequently.

# 5 The metaphysics of Church's system

It is implicit in Church's formalism that if $M$ is a closed expression built from constants that denote properties and relations — i.e. an expression built from denoting constants, bound variables and $\lambda$ — then $M$ also denotes a property, relation, or what have you. These ontological commitments correspond to theorems of our minimal Churchian theory of properties and relations $\mathsf{H}$. E.g. one can formalise and prove that for every binary relation, $R$, there is another unary property of $R$ing oneself, $\lambda x.Rxx$, that for every pair of properties $F$ and $G$, there is also the property of being $F$ and $G$, $\lambda x.(Fx \wedge Gx)$, and so on. It also tells us how these complex properties and relations behave, by entailing various identities: the proposition that $a$ $R$s itself is the proposition that $a$ $R$s $a$, $(\lambda x.Rxx)a = Raa$, the proposition that $a$ is $F$ and $G$ is the proposition that $a$ is $F$ and $a$ is $G$, $\lambda x.(Fx \wedge Gx)a = Fa \wedge Ga)$), and so on. Spare as it is, the Churchian theory of properties and relations makes some substantive assertions.

Evidently this is a formalism that heavily involves bound variables. Now, here is a long-standing challenge for classical theories of structured propositions in which the structure of reality is modeled on the structure of language:

[33] My argument here is inspired by similar arguments found in Harris (1982), that the axioms for the quantifiers and connective pin their meanings uniquely. Dorr (2014) contains further discussion of Harris's argument for the logical connectives.

[34] A very similar argument establishes the more general intersubstitutivity of $\lambda'$ for $\lambda$ *salve veritate*.

What is the structure of propositions corresponding to sentences involving bound variables?

For instance, a textbook first-order formalization of a non-quantified sentence, like *Alice loves Bob*, and a quantified one, like *everyone loves someone*, might be *Rab* and $\forall x \exists y Rxy$ respectively. The structure of the former proposition is perfectly reflected by the structure of the sentence that expresses it, and so some structured proposition theorists have thought to count bound variables among the constituents of the structured proposition expressed by the latter.[35]

For the representational structuralist, this is a contentious but not implausible assumption to make. After all, many of the most convenient representations of reality involve bound variables. But for the metaphysical structuralist, this is a quite radical thesis. It implies, for instance, that there is a distinction in reality — not just in their representations — between the following properties:

1. *is an $x$ such that $x$ is old and $x$ is wise* (i.e. $\lambda x.Fx \wedge Gx$)
2. *is a $y$ such that $y$ is old and $y$ is wise* (i.e. $\lambda y.Fy \wedge Gy$)

For the variable $x$ is a constituent of the first property, but not the second.

Bound variables solve a technical problem: they allow us to form complex predicates and to express quantificational claims. If we simply removed them from Church's formalism we wouldn't have enough representations around to express all the things we want to be able to express. The structure of reality itself, however, need not answer to our expressive needs. Some ways of generating enough representations of things we might need to express also generate more than we need: the distinguishing features of the proliferated representations are merely *representational artefacts*, that do not reflect anything in reality. Indeed, one could argue the pervasiveness of bound variables in philosophy and linguistics is an accident in the history of logic. Other solutions to the expressive problem are possible which do not overgenerate representations to the same degree. De Bruijn (1972) developed a system for introducing bound variables in which it is impossible to form distinct predicates like 1 and 2 above, and there are entirely variable free approaches to type theory, two of which we will consider shortly (one due to Curry, another novel to this theory).

Apart from the strong sense that bound variables are a representational artefact, the radical structured view is inconsistent with Church's principle $\beta$. For if the properties 1 and 2 are distinct then so are the propositions $\lambda x.(Fx \wedge Gx)a$ and $\lambda y.(Fy \wedge Gy)a$, yet by $\beta$ they are both identical to $Fa \wedge Ga$.

Luckily, one does not need to accept the radically fine-grained theory of properties and relations in order to adopt the Churchian representations of them. The mapping from Church's representations of properties and relations need not be one-one: two representations differing only over which bound variables are employed can pick out the same property or relation. A more general principle, that predicts all this and more, asserts that that bound variables do not contribute constituents to the properties they help denote:

**Moderate Structuralism** An expression built from a closed expression $M$ using only bound variables and $\lambda$ denotes an entity with the same simple constituents (in kind and number of occurrences) as the entity $M$ denotes.

Thus for instance, $R$, $\lambda xy.Rxy$, $\lambda zw.Rzw$, $\lambda x.Rxx$, $\lambda xy.Ryx$, and so on, will all denote entities with the same simple constituents. (The parenthetical about number is there to make sure

[35] This is clear in, for instance, King (1996), although note he repudiates this aspect of the view in King (2007), p41.

that we can register the difference between the number of constituent occurrences in things like *Rabb* and *Raab*. If we wanted to be more pedantic, we could say that two entities have the same simple constituents only when the *multiset* of their simple constituent occurrences is the same.)

Many instances of Moderate Structuralism are, in fact, a consequence of Church's principle $\beta$ and some informal principles about constituent count.

**Constituent Count** The number of times a simple entity occurs as a constituent in $MN$ is the sum of the number of occurrences in $M$ and in $N$ respectively.

Why the restriction to simple entities? Assuming the well-foundedness of constituenthood, $Fa$ is an (improper) constituent of $Fa$, but is not a constituent of $F$ or $a$. By contrast, a *simple* entity cannot be a constituent of $Fa$ without being a constituent of $F$ or of $a$, and this is a straightforward consequence of our principle.

Constituent Count tells us that the constituent occurrences of simple entities in $(\lambda xy.Ryx)ab$ is the sum of the constituent occurrences in $\lambda xy.Ryx$, $a$ and $b$. By $\beta$, $(\lambda xy.Ryx)ab = Rba$, whose simple constituent occurrences may be calculated by summing the simple constituent occurrences in $R$, $b$ and $a$ (again, by Constituent Count). The only way these occurrences of simple constituents can be the same is if $\lambda xy.Ryx$ has the same occurrences of simple entities as $R$. Similar justifications are possible for the other instances of Moderate Structuralism discussed above.

We are now in a position to note two ways in which Churchian theories of properties and relations differ from the one we have outlined in section 2. There are roughly two reasons this is. Once we have conceded that bound variables do not contribute constituents, Churchian theories posit (i) the existence of entities that do not have any constituents at all, and (ii) entities that have no well-defined notion of constituent occurrence count and order, and no well-defined main relation. (i) and (ii) thus mark two ways in which Church's framework is ill-suited for the job of formulating a theory of structured propositions guided by the relational diagrams of section 2.[36]

Let's start with (i). Church's theory allows one to construct terms that are *entirely* constructed from bound variables and $\lambda$. Terms like this are called *combinators*, and examples include $\lambda p.p$, and the $C$, $W$ and $K$ combinators constructed above ($\lambda X\lambda y\lambda z.Xzy$, $\lambda X\lambda y.Xyy$, and $\lambda px.p$). The idea that combinators are constituentless is suggested directly by our commitment to Moderate Structuralism: if bound variables do not contribute constituents then terms made entirely out of bound variables and $\lambda$ must not contain any constituents. There is also an independent argument for this from $\beta$ and Constituent Count. For instance, given $\beta$, $(\lambda p.p)$ maps every proposition to itself, and so $\lambda p.p$ cannot add any occurrences of simple constituents to the proposition it is applied to. It follows that $\lambda p.p$ itself must not have any simple constituents (and so cannot be simple itself, either).[37] Similarly, Constituent Count

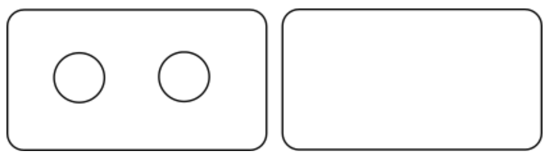
[36] Of course, it is conceivable that there are more liberal structured views that allow for constituentless entities, or that employ revised notions of constituent occurrence count and order. The question of what exactly counts as a 'structured theory' doesn't seem like a fruitful one to me: There is a vast array of possibilities that seem worthy of investigation here, but the structured theory of section 2 is reasonably concrete, and strikes me as being of sufficient interest to justify the search for alternatives to Church's framework.

[37] There is a caveat in this argument: what if every proposition $Q$ has infinitely many constituent occurrences of $\lambda p.p$ in it. Then the constituent occurrences of $\lambda p.p$ in $(\lambda p.p)Q$ and $Q$ could be the same — infinite — in accordance with Constituent Count. (Appealing to constituent order similarly doesn't help, for one could similarly maintain that every proposition begins with an $\omega$-sequence of $\lambda p.p$ constituent occurrences; adding another $\lambda p.p$ to the beginning of an $\omega$-sequence will plausibly not make a difference to the proposition.) The gap can be filled by making the further assumption that propositions are well-founded, or even the stronger assumption that propositions have finitely many constituent occurrences, both of which seem like reasonable assumptions.

tells us that $(\lambda X \lambda y \lambda z.Xzy)Rab$ has the same number of occurrences of simple constituents as $(\lambda X \lambda y \lambda z.Xzy)$, $R$, $a$ and $b$ together, and also tells us that $Rba$ has the simple constituents appearing in $R$, $a$ and $b$. By $\beta$, $(\lambda X \lambda y \lambda z.Xzy)Rab = Rba$, and so $\lambda X \lambda y \lambda z.Xzy$ must contribute no occurrences of simple constituents, in order for the number of constituent occurrences to add up.

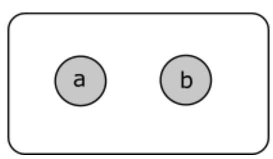
Now I think there is an important choicepoint in a theory of properties and relations that trades in notions like 'constituenthood' over whether it should admit completely constituentless properties and relations or not. Indeed, the thesis that there are constituentless entities is entirely consistent, and there is a lot of unexplored territory to be mapped out here.[38] I do not think it is productive to quibble over what counts as a 'structured theory', nonetheless the idea of a constituentless entity is certainly at odds with the naïve structured picture of entities being built from simple constituents that we have attempted to capture pictorially in section 2.

Our diagrammatic representation of structured entities is suggestive in this respect: there are no 'spectral' entities, that consist entirely of holes (i.e. that have no grey in them whatsoever). For instance, the combinators $\lambda X \lambda y \lambda z.Xyz$ and $\lambda p.p$ would naturally be depicted in the following ways:



As we noted, neither of these are relational diagrams. I have rendered the relational diagrams with borders for visual clarity, but we could equally have rendered them without. Doing so dramatizes the absurdity of an operation that is entirely made of holes, given the structured picture: one would simply have empty space.

Which of our typing rules are to blame for the construction of combinators? Since a combinator is made of entirely of $\lambda$s and variables, every way of building a combinator will involve at least the rules Identity and Abstraction — the only rules that introduce variables and $\lambda$s respectively. As a limiting case of this, $\vdash \lambda p.p$ follows by Abstraction from the $p : t \vdash p : t$ instance of Identity. Thus we could avoid terms denoting constituentless entities by restricting these rules somehow. We will later consider a system that keeps Abstraction but replaces Identity with something else. Related to the rejection of constituentless entities in our structured picture is the rejection of entities like $\lambda X.Xab$ that have no main relation, also depicted in section 2:




the same sorts of rules that allow us to build combinators allow us to build terms like $\lambda X.Xab$ that correspond to illegitimate relational diagrams like the above. These terms have bound variables in 'predicating position', where the variable appears applied to some other term (e.g. $Xa$ where $X : e \to t$), as opposed to 'argument position' where the variable appears as the

[38] My own attempt at this can be found in Bacon (2019), Bacon (2020); see especially the discussion of *pure* entities. See also the related notion of *logicality* in Dorr (2016).

argument to some outer term (e.g. $\exists_e X$).[39]

According to the present structured theory, every structured proposition can be decomposed into a unique *main*, or *outermost* relation, whose argument places have been filled with entities of the appropriate types—the proposition's *immediate constituents*. These arguments are either simple or can themselves be decomposed into an outermost relation with immediate constituents; this decomposition continues until we eventually reach simple entities which have no immediate constituents. (I use the word 'relation' here broadly so as to include relations between entities of non-$e$ types, such as connectives.) This unique decomposition idea is widely assumed in the literature on structured propositions; for quasi-syntactic views it corresponds to a metaphysical analogue of unique readability for sentences.

But Church's theory of properties and relations is inconsistent with this idea. Firstly, it posits the existence of *converses*. $S$ is a converse of the relation $R$ iff for any individuals $a$ and $b$, $Rab$ is the very same proposition as $Sba$. It is easily seen, using $\beta$, that $\lambda xy.Ryx$—$R^c$ for short—is a converse of $R$: $Rab =_t R^c ba$.[40] The existence of converses contradicts the unique decomposition idea: a single proposition $Rab$ — *that John loves Mary* say — can be decomposed as $R$ applied $a$ and then $b$, or a different relation, the converse of $R$, $R^c$, applied to $b$ and then $a$ (*Mary is loved by John*). It also posits *reflexizations*. A reflexization of a binary relation $R$ is a unary property $F$ such that for any $a$, $Fa$ is the very same proposition as $Raa$. It's easy to see that $\lambda x.Rxx$—$R^w$ for short—is a reflexization of $R$ using $\beta$: $R^w a = Raa$. But reflexizations also contradict the idea of a unique decomposition of a relational proposition: *John loves John* can be decomposed with the binary relation $R$ with $a$ put in both argument places, $Raa$, or as the unary property $R^w$ applied to $a$, $R^w a$. Similar problems arise with vacuous $\lambda$-abstraction: a unary predication $Fa$, say *John is tall*, might be thought to have a unique property, in this case *being tall*, as the main relation of the proposition. Yet it also has a decomposition as $(\lambda x.Fa)b$, where $b$ is an arbitrary individual and $\lambda x.Fa$ (*being such that John is tall*) is the main relation.

These different decompositions also apparently contradict the naïve notion of order and number as it applies to constituent occurrences. $a$ and $b$ seem to occur in different orders in the two decompositions of *John loves Mary*, $Rab$ and $R^c ba$. And $a$ seems to occur respectively once and twice in the two decompositions of *John loves John*, $(\lambda x.Rxx)a$ and $Raa$ (*John loves himself* and *John loves John*). With vacuous $\lambda$-abstraction any individual whatsoever can occur in some decomposition of any proposition $p$ (for instance, *snow is white* can also be decomposed as the unary property of *being such that snow is white* applied to *John*).

There has been much suspicion in metaphysics concerning the existence of converse relations that is quite independent of our particular theory, so it is an advantage that the theory does not posit them (see Williamson (1985), Fine (2000), Dorr (2004)). It should be stressed, however, that in denying the existence of converses the structuralist isn't rejecting the intelligibility of the active and passive voice as in, for instance, *the boy kicked the ball* and *the ball was kicked by the boy*. The mere existence of this grammatical distinction is not a decisive reason to think there is a corresponding distinction in reality. As argued in Williamson (1985), for instance, perhaps these are just two ways of expressing the same proposition. Similarly, the active and passive versions of the transitive verb, *kicked* and *was kicked by*, may simply be two different ways of picking out a single relation, whilst being subject to special grammatical rules about which order the arguments must appear in the surface structure of a sentence.[41] So it is possible

[39] Note that it's possible that a predicate variable be in argument position, as in $\exists_e X$. Thus bound predicate variables — i.e. a bound variable with a predicate type (e.g $e \to t$) — are fine, it is only bound variables in *predicating* position that do not correspond to relational diagrams.

[40] Proof: using $\beta$ twice, $(\lambda x \lambda y.Ryx)ba = (\lambda y.Ryb)a = Rab$.

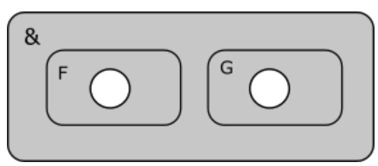
[41] There is another way to deny the existence of converses that is consistent with the present theory: one could maintain that *kicked* and *was kicked by* express distinct simple relations, $R$ and $S$. $Rab$ and $Sba$ should not be

to maintain that converses exist in language without existing in reality.[42] Similar strategies can be applied to English expressions like *loves oneself* and *is such that John is tall* that on the surface appear to be referring to reflexizations and vacuous properties.

The rejection of converse relations is straightforwardly reflected in our relational diagrams. It is simply impossible to draw a diagram corresponding to $\lambda xy.Ryx$. For, in our diagrams holes have no distinguishing features apart from their positions. It simply isn't possible to switch the positions of the two holes represented in our diagram $R$ to produce a new diagram. The culprit here is the rule Exchange, which allows one to abstract on variables in a term $M$ in a different order than those variables appear in the body of $M$. Terms in which the variables are abstracted in the same order as they appear in the body, like $\lambda xyzw.(Rxy \wedge Szw)$ receive straightforward diagrammatic representations, whereas putative relations like $\lambda xyzw.(Rzy \wedge Swx)$, in which they are not, do not. If we wanted to reject terms like this, then one could simply remove the rule of Exchange from our typing rules.

Similar things can be said about reflexizations and vacuous properties. There are no relational diagrams for $\lambda x.Rxx$ and $\lambda x.Fa$. To represent the former we would need to take a relational diagram with two holes, like our diagram for $R$ above, and merge the two holes into one. We similarly cannot represent the latter property: it is a unary property so it must have a hole somewhere. But there is nowhere for the hole to be, given that $F$s only hole is already occupied by $a$. Just as the rule of Exchange lets us build expressions for converses, the rules of Contraction and Weakening let us build expressions for reflexizations and vacuous properties respectively. A structuralist might therefore wish to remove these rules from their typing system if they wanted generate only terms that could be represented by relational diagrams.

It is worth remarking that in rejecting these features of Church's theory, we are also rejecting the legitimacy of many common operations on properties and relations. For instance, if you have two properties, $F$ and $G$, one might also postulate the existence of the property conjunction of them, $\lambda x.(Fx \wedge Gx)$. But this is expressed by a term in which $\lambda$ binds a variable occurring twice. The closest we get to property conjunction is the following:




But this has two holes, and is thus a binary relation, represented by the Church term $\lambda xy.(Fx \wedge Gy)$, sometimes called Quinean conjunction.

One might be tempted to diagnose the forgoing problems as being inherently linked to the use of $\lambda$ and bound variables in our representations of properties and relations. Curry (1930), drawing on Schönfinkel (1924), introduced a formulation of type theory that does not involve variables at all: the need for principles like Moderate Structuralism and the puzzles surrounding the status of bound variables as constituents, simply do not arise in this framework. Like Church, one has logical and non-logical constants of various types, and a rule for applying a term $M$ of type $\sigma \to \tau$ to another term $N$ of type $\sigma$ to form $(MN)$ of type $\tau$. To do the jobs that $\lambda$ normally deals with, Curry instead takes as primitive expressions that correspond to

identified, since they differ both in the order of their individual constituents and in their main relations, and so behave unlike converses. However, as Bader (2021) points out, this sort of theory seems to posit unexplained necessary connections between simple relations — namely $\Box(Rab \leftrightarrow Sba)$.

[42] The reader might wonder whether one could find an interpretation of the Churchian theory in which converses exist only 'in language'. It is hard to see how this is possible in light of the following theorem of the Churchian theory: for every asymmetric relation there exists another relation distinct from it that is its converse $\forall_{e \to e \to t} X(\exists xy(Xxy \wedge \neg Xyx) \to \exists Y(Y \neq_{e \to e \to t} X \wedge Y = X^c)$.

the combinators in Church's theory. In Church's theory, recall, a combinator is a closed term involving only bound variables and $\lambda$. Curry observed that one doesn't need to take all the combinators as primitive: one can take smaller sets of combinators as primitive, and obtain the others by definition, much like one may define the truth functional connectives from disjunction and negation. Here is one particularly sparse collection of combinators that will suffice: for each type $\sigma$, $\tau$ and $\rho$ one introduces a primitive combinator constants $I^{\sigma}$, $K^{\sigma\tau}$ and $S^{\sigma\tau\rho}$, denoting respectively the operations $\lambda x.x$, $\lambda xy.x$ and $\lambda XYz.Xz(Yz)$ denote in Church's theory where $x, z : \sigma, y : \tau$, $X : \sigma \to \tau \to \rho$, and $Y : \sigma \to \tau$.

This reformulation, however is no more friendly to a theory of structured propositions than Church's is: indeed the theories can be shown, in certain technical senses, to be equivalent.[43] For in Curry's combinatory logic, $S$, $K$ and $I$ are required to satisfy certain identities — identities that would be required by $\beta$ if instead of being taken as primitive they were take to be defined in terms of $\lambda$s as above.[44] In order to satisfy these identities, combinators must be constituentless for the same reason that they must be in Church's theory. And the problematic expressions $\lambda xy.Ryx$, $\lambda x.Fa$ and $\lambda x.Rxx$ have correlates in Curry's system: $K^{et}(Fa)$, $S^{eet}RI^{e}$, and $S(K(SR))K$. (I have omitted the relevant type superscripts from the final paraphrase because they are long and not necessary for my point.) Given the aforementioned identities we can derive correlates of the identities $(\lambda xy.Ryx)ab = Rba$, $(\lambda x.Rxx)a = Raa$ and $(\lambda x.Fa)b = Fa$ that appeared in our earlier discussion.[45] This concludes our case that the formalisms of Church and Curry build in metaphysical assumptions that the structuralist does not accept.

Recent authors arguing against structured propositions, such as Dorr (2016) and Goodman (2017), are well aware of these implications of the standard formalism. The moral we have drawn from this discussion is that certain $\lambda$-terms are not meaningful for the present structuralist, and should be excised from their language. Dorr and Goodman, by contrast, freely use the objectionable $\lambda$-terms in their case against the structuralist: their concession to the structuralist is that they do not assume that these $\lambda$-terms satisfy $\beta$, a crucial ingredient in some of the problems we discussed above (for instance, we appealed to it in our proof that $\lambda xy.Ryx$ is a converse of $R$). Instead of assuming that $(\lambda x_1...x_n.M)N_1...N_n$ and $M[N_1/x_1...N_n/x_n]$ are identical, as required by $\beta$ (see $\beta^{=}$ above), they merely require a biconditional:

**Extensionsal** $\beta^{=}$ $(\lambda x_1...x_n.M)N_1...N_n \leftrightarrow M[N_1/x_1...N_n/x_n]$

Similar weakenings of combinatory logic are also possible.

We will return to the arguments of Dorr and Goodman later. For now I will just remark that that the structuralist has no reason to concede even Extensional $\beta^{=}$, when the relevant instances involve $\lambda$-terms they find unintelligible. Indeed, there are serious concerns about adopting Church's $\lambda$-notation without also accepting his defining principles, $\beta$ and $\eta$. $\lambda$ is not a device we were familiar with pretheoretically: it is a technical device, introduced by Church

[43]Some care is needed in setting this correspondence up. Unlike the axioms $\beta$ and $\eta$ for the $\lambda$-terms, the standard axioms governing combinatory terms *do not* pin down the combinatory terms uniquely, and as a result the standard translations between the $\lambda$-calculus and combinatory logic are not exact inverses of each other. The correspondence is only exact when we strengthen combinatory logic with certain extensionality principles, which also pin down the combinatory terms in the sense we demonstrated for $\lambda$. See Hindley and Seldin (2008) p84 for more details.

[44]Specifically, the following identities:

**S** $S^{\sigma\tau\rho}FGa = Fa(Ga)$

**K** $K^{\sigma\tau}ab = a$

**I** $I^{\sigma}a = a$

[45]For instance, the first identity can be derived as follows: $(S(K(SR))K)ab = (K(SR))a(Ka)b$ by **S**. Since $K(SR)a = SR$, we may further reduce it to $SR(Ka)b$. And since $(Ka)b = a$, we finally get $Rba$.

to do a certain technical job (one which can be done in several other ways). The only thing he told us about the meaning of sentences involving his $\lambda$-notation was that they satisfied these two principles of $\beta$ and $\eta$. Luckily, as we showed earlier, those principles pin the meaning of $\lambda$-terms down uniquely. Without them, we have nothing to go on. Extensional $\beta$, for instance, falls dramatically short of pinning down their meanings: if it just happened that everyone loves people who hate them back and hate people who love them back, then it's consistent with extensional $\beta$ that $\lambda xy.Ryx$ denotes the hating relation, even when $R$ denotes the loving relation.[46] In short, Church's $\lambda$-terms were introduced in the context of particular metaphysical assumptions, and don't obviously have any meaning outside of that theory. (There is also an apparent *practical* reason to keep all the problematic $\lambda$-terms in the language, since they are also employed in the Church-Curry formalism in the expression of complex quantificational claims. But the presence of the $\lambda$-terms alone does not solve this expressive challenge: without $\beta$ it is far from clear that a sentence like $\exists_\sigma \lambda x.A$ bears any relation to the claim we wanted to express.[47] We answer the expressive challenge in section 7.)

The corresponding weakening of Curry's theory, in which certain identities between propositions are replaced with biconditionals, is also consistent with a highly structured picture.[48] For instance it is consistent in this weaker picture to suppose that the combinators $I^\sigma$, $K^{\sigma\tau}$ and $S^{\sigma\tau\rho}$ are metaphysically simple (as opposed to being constituentless, as $\beta$ seems to require). However, in addition to analogous worries about pinning down the meaning of combinators, the resulting view is subject to even more problems. The structure of apparently simple statements involving bound variables end up expressing long and complicated structured propositions according to the combinatory view, due to the way that bound variables get translated away into combinatorese.[49] More seriously, the standard ways of translating quantified statements into combinatory logic do not even deliver the right *truth value* when the theory is weakened in this way, which suggests this version of combinatory logic is particularly ill-suited to this form of weakening.[50]

---

[46]Nor does it help to prefix these principles with an operator denoting metaphysical necessity, since it seems just as bad to interpret $R$ to mean *actually loves* and $\lambda xy.Ryx$ to mean *actually hates*.

[47]Extensional $\beta^=$ ensures that $\exists_\sigma \lambda x.A$ has the right *truth value*, but this is hardly the same as solving the expressive challenge, since one can always come up with 'paraphrases' that get truth values right as long as there is a true and a false proposition.

[48]The weakened principles governing $S$, $K$ and $I$ are respectively:

**Extensional S** $\forall x_1...x_n(S^{\sigma\tau\rho}FGax_1...x_n \leftrightarrow Fa(Ga)x_1...x_n)$

**Extensionsal K** $\forall x_1...x_n(K^{\sigma\tau}ab \leftrightarrow ax_1...x_n)$

**Extensional I** $\forall x_1...x_n(I^\sigma Rx_1...x_n \leftrightarrow Rx_1...x_n$

[49]For instance you might expect *everybody loves somebody*, $\forall_e x\exists_e y.Ryx$, to denote a proposition that only contain three constituents corresponding to the two quantifiers and $R$. However this sentence becomes $\forall_e(S(K\exists_e)S(K(SR))K)$ under a standard translation from $\lambda$ languages to combinatorese. The complexity of the resulting translation depends both on what combinators you take a primitive, and the translation, but the result of applying these translations will invariably be more structured than the formulas they are paraphrasing.

[50]The problem arises when one wants to paraphrase statements that quantify into contexts that are sensitive to structure. Consider the following sentence where we are quantifying into operator position in the context of a belief operator, $B$: $\exists_{t\to t}X.B(XA)$, or $\exists_{t\to t}\lambda X.B(XA)$, where $B$ means *I believe that* and $A$ means *it's raining*. We may suppose this is true because I believe that it's not raining, so the existential is witnessed by the negation operator (and the truth of this requires only extensional $\beta$). However a standard translation of this quantified statement into combinatorese is $\exists_{t\to t}(S(KB)(SI(KA)))$, which (given the background logic of extensional **K**, **I** and **S**) is satisfied only if I believe a structured proposition of the form $SI(KA)X$ for some operator $X$. But I might believe it's not raining without believing any proposition structured like this. This above depended on a particular way of translating away bound variables in combinatorese, but when $S$, $K$ and $I$ are the only primitive combinators available all translations will disturb the structure of the open sentence in the scope of the belief operator.

# 6 The Structural Calculus

Let's return to our structuralist metaphors: there are some basic building blocks — simple properties, relations and individuals — and complex propositions, properties and relations can be made by combining these basic building blocks in various ways. The theories of Church and Curry posit ways of building new things from old that simply outstrip the sorts of building operations that are legitimate by the lights of the present structuralist. For instance, we have denied that one can build a unary property, like $\lambda x.Rxx$, from a simple binary relation $R$ alone.

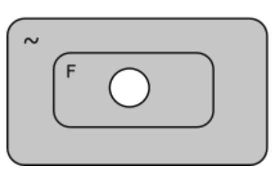
Putting it like this makes the following questions all the more urgent:

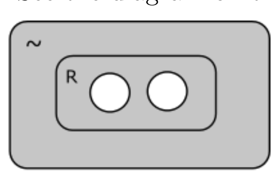
> Which ways of building new entities from old *are* legitimate by structuralist lights?
>
> Which ways of combining typed expressions with $\lambda$s (or combinators) *do* correspond to genuine ways of combining entities?

With satisfactory answers to these questions we will be in a position to make our theory of structured propositions more precise. But also we use this discussion to motivate a more perspicuous notation — the *structural calculus* — in which you simply cannot form expressions which do not denote by structuralist lights. Indeed, the syntactic operations by which one can combine two expressions will correspond exactly to the possible ways of combining two entities that make a structured entity. The system will also be related to a fragment of Church's system in which one can accept $\beta$ and $\eta$ in full generality.

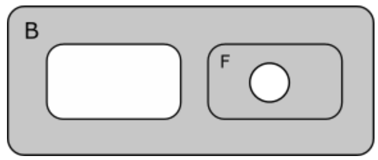
The most general way of combining entities according to this structuralist is by an operation that has application and composition as special cases; as a result I will call in 'complication'. Let's start with application. Given an $n$-ary relation, $R$, we can plug an individual into any of its argument slots. This is represented by simply filling the corresponding hole in its relational diagram by a relational diagram for an individual. In Church's notation, we notated the result of filling the first (leftmost in our pictorial representation) hole of a relation $R$ with $a$ as $(Ra)$. In order to represent filling $R$s second slot, third slot, fourth slot, and so on, with $a$ in Church's language we need to start using the $\lambda$-notation: they are respectively $\lambda x.Rxa$, $\lambda xy.Rxya$, $\lambda xyz.Rxyza$, and so on. Instead of introducing $\lambda$s we will generalize the simple notation of application. It will be convenient to number the holes in a relation from 0: we shall write $(Ra)^0_0$ to denote the result of plugging $a$ into $R$s 0th slot (i.e. $(Ra)$ in Church's notation), and $(Ra)^0_1$ for the result of plugging $a$ into $R$s 1st slot, $(Ra)^0_2$ for the 2nd slot, and so on. (For the moment, the reader should ignore the superscripts.)

This can be generalized to any relations between entities of any type: this general form of application corresponds to the result of plugging a hole in a relational diagram with another diagram that fits it perfectly, leaving no holes behind. We also said that one can slot a relational diagram into a hole leaving one more holes behind, thus contributing new holes to the result of the combination. This is composition. Take, for example, the diagram for $\neg$ of type $t \to t$. It has one proposition shaped hole. Suppose that, instead of filling that hole with a proposition, we filled it with a unary property $F : e \to t$ (say, *being tall*). Since this diagram has a single individual shaped hole, the result would have a hole remaining (see the diagram on the left below). This is the composition of *not* and *being tall*: *being not tall*. If we instead filled negation's hole with a binary relation, *loves*, we would get their composition *doesn't love*, which would leave two individual shaped holes behind. See the diagram on the right below.



We can of course compose negation with ternary and higher-arity relations in exactly the same way. In Church's notation we would represent these compositions using $\lambda$-terms: $\lambda x.\neg(Fx)$, $\lambda xy.\neg(Rxy)$, $\lambda xyz.\neg(Sxyz)$, and so on. Instead we will write $(\neg F)^1_0$, $(\neg R)^2_0$, $(\neg S)^3_0$, and so on. The superscript indicates that we are leaving one, two or three holes behind, and the subscript still indicates that we are performing this in the 0th slot of $\neg$. Of course, whatever we can do in the 0th slot, we can also do in the 1st, 2nd, 3rd slots. For instance, given a binary connective $B$, we can compose it with the unary property $F$ in the 1st slot to obtain $(BF)^1_1$:

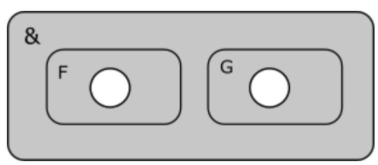



So in general we can combine two relations, $M$ and $N$, provided they have the right number and type of holes, and create another entity $(MN)^n_m$ by plugging $N$ into the $m$th hole of $M$, thereby adding $N$s $n$ holes where $M$s $m$th hole used to be. Clearly this corresponds exactly to the rule for creating complex relational diagrams from simpler ones in section 2:[51]

**Complication** Given $M : \sigma_1 \to ... \to \sigma_{m+1} \to \tau$ and $N : \rho_1 \to ...\rho_n \to \sigma_{m+1}$ we write $(MN)^n_m : \sigma_1 \to ... \to \sigma_m \to \rho_1 \to ... \to \rho_n \to \tau$ for the result of plugging $N$ into the $m$th hole in $M$, leaving all but the first $n$ holes in $N$s unfilled (see the rule for constructing relational diagrams). In the $\lambda$ notation :

$$(MN)^n_m \qquad \equiv \qquad \lambda x_1...x_m y_1...y_n.Mx_1...x_m(Ny_1...y_n)$$

We can see that this subsumes our previous two operations by setting $n$ or $m$ to 0 respectively. We will refer to brackets indexed by two 0s as application brackets. We will follow our previous convention of associating application brackets to the left: thus $M_0M_1...M_k$ is short for $(M_0M_1)^0_0M_2)^0_0...M_k)^0_0$. So, for instance, the Quinean conjunction of $F$ and $G$, $\lambda xy.((\wedge Fx)Gx)$ described earlier, can now be represented by the term $(\wedge F)^1_0G)^1_1$.




Notice that we have associated every term in our new language with a $\lambda$-term: constants can be mapped to themselves, and $(MN)^n_m$ can be associated with the corresponding $\lambda$-term. Formally the translation $(-)^\lambda$ from our variable free calculus to the $\lambda$-calculus may be given as follows:

- $(c)^\lambda = c$ when $c$ is a constant.
- $((MN)^n_m)^\lambda = \lambda x_1...x_m y_1...y_n.M^\lambda x_1...x_m(N^\lambda y_1...y_n)$ where $x_1...x_m$ and $y_1...y_n$ are the first variables not appearing free in $M$ and $N$ respectively.

Just by observing the form of this second case, it's also easy to see that the $\lambda$-terms in the image of this translation have none of the features of the $\lambda$-terms we discussed above: (i) variables always appear in the body of a term in the same order that they are abstracted, (ii) every

[51] See [ANONYMIZED] for more details.

| | | | |
|---|---|---|---|
| $\dfrac{\Gamma\vdash M{:}\sigma\to\tau}{\Gamma,x{:}\sigma\vdash Mx{:}\tau}$ | **Concretion** | $\dfrac{\Gamma\vdash M{:}\sigma\to\tau \qquad \Delta\vdash N{:}\sigma}{\Gamma,\Delta\vdash(MN){:}\tau}$ | **Application** |
| $\dfrac{}{\vdash c{:}\tau}$ | **Constants** | $\dfrac{\Gamma,x{:}\sigma\vdash M{:}\tau}{\Gamma\vdash\lambda xM{:}\sigma\to\tau}$ | **Abstraction** |

Table 2: Natural deduction for typing the structural $\lambda$-terms

variable that is abstracted appears exactly once in the body, and (iii) no variables appear in a predicating position.[52] This is in accordance with the underlying structural principle that you can't reorder, duplicate or throw away constituents within a proposition without changing the proposition, and that every proposition has a main relation.

In fact, there is a fairly simple way to revise our rules for constructing $\lambda$-terms in such a way that we can only construct the 'good' $\lambda$-terms. We can straightforwardly drop the rules Exchange, Contraction and Weakening, which allows one to reorder, duplicate and throw away variables in the body of a term. The rule Identity allows one to construct terms like $\lambda X.Xa$, which bind variables in a predicating position.[53] Simply dropping Identity would prevent one from ever introducing variables, even in argument position, and consequently prevent $\lambda$s not only from binding variables in predicating position, but also in argument position. The rule Concretion, from figure 6, however allows us to introduce free variables into argument position without introducing them into predicating position. Thus we can simply replace Identity with Concretion, yielding the system in figure 6.[54] (Since type assignments must always consist of sequences of distinct variables, the only well-formed instances of Concretion are those where $x : \sigma$ does not appear in $\Gamma$.) It is now possible to provide a reverse translation from $\lambda$-terms belonging to our restricted version of the $\lambda$-calculus to our novel language; this is slightly more technical and is relegated to an appendix. (The key idea is that we do not define the translation by induction on the structure of the $\lambda$-terms themselves but by induction on their proofs.)

We have observed already that there is not a one-to-one correlation between $\lambda$-terms and the properties and relations they denote. Simply relabeling bound variables will produce different representations for the same property. But also the equations $\beta$ and $\eta$, which can be accepted once we have rid ourselves of the problematic $\lambda$-terms, will generate further multiplicity in way we represent reality: for instance, $\beta$ tells us that the sentence $(\lambda xy.(Fx \wedge Gy))ab$ and $Fa \wedge Gb$, despite being different, both denote the same proposition. The same is true even for our new language. For instance we can make the proposition that it's not necessary that $A$ in two ways: we can glue $\neg$ and $\Box$ together by composition, in the way depicted in an earlier relational diagram, to make $(\neg\Box)^1_0$, and then apply it to $A$: $((\neg\Box)^1_0A)^0_0$. Or we can apply $\Box$ to $A$ and then $\neg$ to the result: $(\neg(\Box A)^0_0)^0_0)$. Thus we have the equation:

$$((\neg\Box)^1_0A)^0_0 = (\neg(\Box A)^0_0)^0_0)$$

[52] In particular, in the term $\lambda x_1...x_ny_1...y_k.Mx_1...x_n(Ny_1...y_k)$, the order of the free variables in the body of $Mx_1...x_n(Ny_1...y_k)$ is the same as in the $\lambda$ abstract, $\lambda x_1...x_ny_1...y_k$, and every variable in the string of variables abstracted appears at least and at most once in the body. Note that these three features or closed under '$\beta$-reduction', in the sense that if $(\lambda x.M)N$ has them, so does $M[N/x]$, so that we can accept as meaningful $\beta$-reductions of terms formed by complication.

[53] To construct this term, for instance, one can use instances of Identity and Constants, $X : e \to t \vdash X : e \to t$ and $\vdash a : e$, to derive $X : e \to t \vdash Xa : t$ by Application. Then by abstraction one can get $\vdash \lambda X.Xa : (e \to t) \to t$.

[54] Notice that Concretion bears a nice formal relationship to Abstraction, and if you are familiar with the Curry-Howard isomorphism you'll notice they correspond to both directions of the propositional metainference $\Gamma, A \vdash B$ iff $\Gamma \vdash A \to B$. So I think the system is highly natural from a mathematical perspective.

We can verify these identities by translating these two expressions into relational diagrams, and seeing that they are the same, as we did in the section 3. The differences in the above expressions corresponds to two different orders in which you could construct the diagram for $\neg\Box A$ — combine the diagrams for $\neg$ and $\Box$ first, or the diagrams for $\Box$ and $A$ first — even though the diagram you reach at the end is the same whichever way you did it.

It would be nice to have something general to say about equations like this one. Let us say that a *structural equation* is an equation between terms in the structural calculus that have identical relational diagrams. Ideally, we would like a general set of rules for generating equations between terms of the structural language — an *equational theory* — which implies all and only the structural equations. Something analogous to the principles $\beta$ and $\eta$ for the $\lambda$ calculus, or the corresponding principles for combinatory logic. Indeed it is possible to do this: in the appendix I present such a theory, which I call *the theory of structural equations*. In the appendix I sketch a proof that two terms of the structural calculus are equivalent in the theory of structural equations *if and only if* their translations into the restricted $\lambda$-calculus are $\beta\eta$-equivalent *if and only if* they are associated with the same relational diagram.

Since I have emphasized the importance of $\beta$ and $\eta$ in particular, let me briefly present the informal reasoning behind their soundness with respect to our diagrammatic representations. The idea is to think of free variables as constituents, and assign them the simple relational diagram you would assign a constant of the same type (except labeled with the variable in question). $\lambda$-abstraction on $x$ corresponds to punching a hole where the simple constituent $x$ appears in the relational diagram, and application corresponds to filling that hole with an argument. (The restriction on which legitimate $\lambda$-terms exist ensures that $\lambda$-abstraction is only ever be performed to a variable that appears in argument position, occurs exactly once in the diagram, and is the rightmost variable in the diagram.) Thus $(\lambda x.M)a$ should denote the result of plugging $a$ into the hole punched out where $x$ is in $M$, i.e $M[a/x]$, which is just what $\beta$ says. $\eta$ is justified in the exact same manner: $\lambda x.Fx$ corresponds to inserting a variable diagram into a hole in $F$ and then punching it out again, which leaves you with $F$ again. Theorem 1 of the appendix ensures that $\beta\eta$ is not only sound, but also *complete* for the structural equations, in the sense that two terms are mapped to the same relational diagram only if they are $\beta\eta$-equivalent.

# 7 Quantifiers and Identity

The systems of Frege, Church and Curry allow one to make all sorts of quantificational claims. We earlier cautioned against tailoring our metaphysics merely to meet an expressive challenge. But we should be able to meet that challenge all the same: there are many quantificational claims that, *prima facie*, appear to require bound variables appearing more than once, or in a different order to the order in which they are bound. Consider the following sentences, and their formalizations:[55]

Someone loves himself: $\exists x.Lxx$.

Everybody is loved by somebody: $\forall x\exists y.Lyx$

Everyone loves someone they are hated by: $\forall x\exists y.(Lxy \wedge Hyx)$

[55] There are also cases where the variable being bound does not appear at all in the scope of the quantifier:

Something is such that snow is white: $\exists x.A$

However they seem like edge cases. When $x$ does not appear free in $A$, $\exists x.A$ and $\forall x.A$ are logically equivalent to $A$. So it seems like there wouldn't be anything we couldn't express, up to logical equivalence, if we simply did not allow these cases.

To analyze the first case, Church and Curry postulate a special property of loving oneself, $\lambda x.Lxx$ or $WL$, to which they ascribe the higher-order property of being instantiated. But do we really need to postulate the existence of a property, *loves oneself*, in addition to the relation *loves*, in order for there to be the proposition that *someone loves himself*? I think not, as evidenced by Frege's original formalism, where quantifiers, not $\lambda$s, bind variables: one can directly make statements like $\exists x.Lxx$ without invoking $\lambda$s or combinators. Of course, Frege himself had a fairly strong theory of grain in which there were only two propositions — but one can easily remove these special extensionalist principles from Frege's system to obtain a theory that is neutral about the existence of this property (i.e. it is neutral about the sentence $\exists X \forall y(Xy = Lyy)$, for instance, while of course still implying $\exists p.(p = \exists y.Lyy)$).

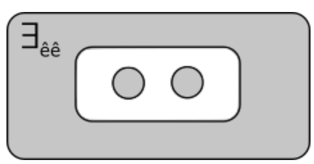
We do not need to reintroduce bound variables in order to say that someone loves himself. In addition to the higher-order property that unary properties have when they are instantiated by something, there is another higher-property that relations have when they are instantiated by the same thing in both arguments; that $R$ has when something $R$s itself. If the former is a legitimate higher-order property, then so is the latter. Moreover, we have no special reason to think that the latter higher-order property should be reducible to the former. It turns out there *is* a reduction when you accept the metaphysics implicit in the systems of Church and Curry. This is certainly a nice feature, but not one that we would have expected to hold antecedently. We shall notate these two higher-order properties as follows:

- $\exists_{\hat{\sigma}} : (\sigma \to t) \to t$
- $\exists_{\hat{\sigma}\hat{\sigma}} : (\sigma \to \sigma \to t) \to t$

Adopting the obvious parallel convention for the universal quantifier. We can now formalise our first example

Someone loves himself: $\exists_{\hat{e}\hat{e}} L$.

Here recalling our earlier convention of writing $\exists_{\hat{e}\hat{e}} L$ for $(\exists_{\hat{e}\hat{e}} L)^0_0$. As a relational diagram, $\exists_{\hat{e}\hat{e}}$ is represented as follows:



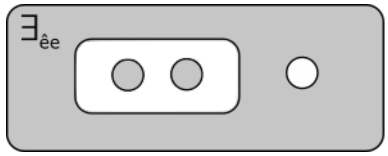

There are also two higher-order relations which take a binary relation, $R$, in its first argument place, and an individual $a$ in the second argument place to yield the proposition that something $R$s $a$, and something is $R$ed by $a$ respectively.

- $\exists_{\sigma\hat{\tau}} : (\sigma \to \tau \to t) \to \sigma \to t$.
- $\exists_{\hat{\sigma}\tau} : (\sigma \to \tau \to t) \to \tau \to t$.

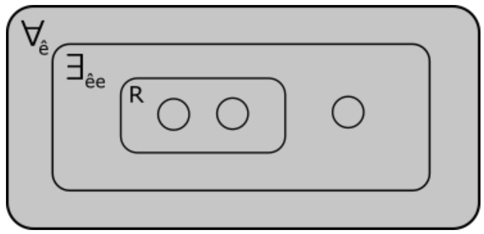
Thus we can formalise the second example:

Everyone is loved by somebody: $\forall_{\hat{e}}(\exists_{\hat{e}e} R)$

In terms of relational diagrams $\forall_{\hat{e}}$ has the same relational diagram as the ordinary existential quantifier (see section 2). $\exists_{\hat{e}e}$ takes a binary relation in its first argument and an individual in its second, so its relational diagram is:




so that $\forall_{\hat{e}}(\exists_{\hat{e}e}R)$ is represented like this:




There is clearly a general pattern here. Suppose that $R$ is an $n$-ary relation between entities of types $\sigma_1, ..., \sigma_n$, and that we have chosen to quantify into one or more of these argument places at the same time; lets say exactly $k$ argument places have been chosen (each of these argument places must, of course, be of the same type). Then there is a higher-order relation whose first hole is $R$ shaped, and its remaining $n-k$ holes are the shapes of whichever types remain of $\sigma_1...\sigma_n$. This is the coordinated existential quantifier that existentially quantifies into the $k$ positions indicated, and leaves the remaining $n-k$ holes.

A *hatting* of a sequence of types, $\sigma_1, ..., \sigma_n$, is the result of putting hats on top of some of the types in this sequence, provided the hats only appear above at most one sort of type: for instance $e\hat{e}t\hat{e}$, or $\hat{t}(e \to t)\hat{t}e$. We write $\boldsymbol{\sigma}$ to denote a hatted sequence of types, $\sigma_1...\sigma_n$. We write $\check{\boldsymbol{\sigma}}$ for the result of *unhatting* the hatted sequence $\boldsymbol{\sigma}$: that is deleting all the types from the sequence that are hatted. Unhatting the two previous examples yields $et$ and $(e \to t)e$ respectively. Finally, given a hatted sequence of types $\boldsymbol{\sigma}$ consisting of the types $\sigma_1, ..., \sigma_n$ we write $\boldsymbol{\sigma} \to \tau$ to mean the type $\sigma_1 \to ... \to \sigma_n \to \tau$, ignoring any hats appearing in $\boldsymbol{\sigma}$. So for every hatted sequence of types $\boldsymbol{\sigma}$ we posit a pair of quantifiers:

- $\forall_{\boldsymbol{\sigma}} : (\boldsymbol{\sigma} \to t) \to \check{\boldsymbol{\sigma}} \to t$
- $\exists_{\boldsymbol{\sigma}} : (\boldsymbol{\sigma} \to t) \to \check{\boldsymbol{\sigma}} \to t$

Now we can paraphrase our last example as $\forall_{\hat{e}\hat{e}}(\exists_{e\hat{e}\hat{e}e}((\wedge L)^2_0 H)^2_2)$.

Given that these particular examples are easily translated, one should expect a more general fact to hold: that anything you can express with quantifiers that bind vacuously, multiple times, or out of order, can be translated into the structural calculus using coordinated quantifiers. One way to precisify this conjecture is to formulate a variant logical system which has the ordinary connectives and quantifiers of higher-order logic, and whose terms are obtained by the typing system you get by adding Exchange, Weakening and Contraction to the structural type system (but still letting Concretion replace Identity), and showing that one can translate it into the structural calculus.[56]

One might object to the structural theory of quantification on the grounds that it is unparsimonious: we have posited infinitely many different primitive quantificational expressions. Let me offer two brief responses. First, the theories of Frege, Church and Curry are all in the same position: they posit infinitely many primitive quantificational expressions, $\forall_\sigma$ and $\exists_\sigma$, for each

[56]Not every term should receive a translation: for instance the term $\lambda x.Rxx$ doesn't correspond to a structural term. However every term in which every $\lambda$ is immediately preceded by a quantifier can be translated $\forall_\sigma \lambda x.A$ .

type $\sigma$. Second, the structuralist should generally resist reducing quantifiers to the higher-order quantifiers $\forall_\sigma$ and $\exists_\sigma$ even when this is possible. Quantifiers like *exactly three*, *many*, *few*, *most*, *just as many*, for instance, have such definitions, but they are logically complex and therefore denote structured entities, whereas the listed quantifiers appear to be simple.

By the same lights, the theory of Church and Curry allow one to provide reductive definitions of identity in terms of quantification. Two things of type $\sigma$ are identical iff they share exactly the same properties:

$$=_\sigma := \lambda xy.\forall_{\sigma\to t}X(Xy \leftrightarrow Xz)$$

The relation of sharing the same properties is often call *Leibniz equivalence*. One can certainly prove in a system containing $\mathsf{H}$ and the usual axioms for identity that $a$ and $b$ are identical iff they share the same properties: $a =_\sigma b \leftrightarrow (\lambda xy.\forall_{\sigma\to t}X(Xy \leftrightarrow Xz))ab$. But, as with the other first-order quantifiers, this is not to say that the identity relation is the same as the thing denoted by the logically complex expression on the right. For one might insist that identity is metaphysically simple, while Leibniz equivalence is a structured entity that has quantifiers and the biconditional as constituents.

So it seems independently desirable for a structuralist to take the identity symbol $=_\sigma$ at each type as primitive. However there is a deeper reason why the present structuralist cannot simply adopt Leibniz equivalence as their definition of identity. While one can paraphrase away cases in which a variable appears more than once or which they are bound in a jumbled order using our coordinated quantifiers, we cannot paraphrase terms in which the variables appear in predicating position, as the variable $X$ does twice in the definition of Leibniz equivalence. Other definitions of identity which are acceptable in orthodox type theory often fail for similar reasons: for instance, Andrews' definition of identity in Andrews (2002), $\lambda xy.\forall Z(\forall w.Zww \to Zxy)$, also has a variable $Z$ appearing in predicating position.

Is it possible to come up with a more cunning definition of identity that doesn't involve variables appearing in predicating position at all? The answer — *no* — follows from a more general feature of the structural calculus that is worth examining further. In orthodox versions of higher-order logic (with $\lambda$ or combinators) it is possible to define non-extensional operations from extensional primitives.[57] We saw above that propositional identity, $=_t$, can be defined in terms of quantification into operator position and the biconditional alone, each of which are extensional operations. Yet on the plausible assumption that there are more than two propositions, propositional identity is not an extensional connective: propositions with the same truth value are not substitutable *salve veritate* in the context of a propositional identity symbol, because propositions with the same truth value needn't be identical. By contrast, in the structural calculus, one can prove that it is only possible to define extensional operations from extensional primitives.[58]

[57] See §4 of Bacon (2018) or §9.5 of Bacon (Forthcoming).

[58] In general, we say that $m$-ary relations $R$ and $R'$ of the same type are coextensive when $\forall x_1..x_m(Rx_1...x_m \leftrightarrow R'x_1...x_m)$, and that individuals are coextensive when they are identical. $R$ is *extensional* iff whenever $x_1...x_n$ are coextensive with $y_1...y_n$, then $Rx_1...x_n$ is coextensive with $Ry_1...y_n$ (provided this is well typed). Suppose for induction that $R$ and $S$ are extensional. We wish to show that $(RS)^n_m$ is extensional if $x_1...x_n, x_{n+1}...x_{n+m}$ are respectively coextensive with $y_1...y_n, y_{n+1}...y_{n+m}$. By the extensionality of $S$ we have that $Sx_{n+1}...x_{n+m}$ is coextensive with $Sy_{n+1}...y_{n+m}$, and by the extensionality of $R$ we then get that $Rx_1...x_n(Sx_{n+1}...x_{n+m})$ is coextensive with $Ry_1...y_n(Sy_{n+1}...y_{n+m})$, which is just to say that $(RS)^n_m x_1...x_{n+m}$ is coextensive with $(RS)^n_m y_1...y_{n+m}$ as required.

## 8 Paradox

Let us return now to the Russell-Myhill paradox. As before, we noted that it reduced the following schemas, formulated in the system $\mathsf{H}_0$ (i.e. $\mathsf{H}$ with $\eta$ removed, and $\beta$ replaced with extensional $\beta$), to absurdity:

**Predicate Argument Structure** $Fa = Gb \to F = G \wedge a = b$

**Predicate Structure** $Fa = Ga \to F = G$

We can now describe our counterexamples more explicitly in the structural calculus. A single proposition, $\Box(\neg A)$ may obtained both by applying the composite operator $(\Box\neg)^1_0$ to the simple proposition $A$, or by applying the simple operator $\Box$ to the composite propostion $(\neg A)^0_0$. Predicate Structure is also rejected, since one can make the proposition $Laa$ by applying $(La)^0_1$ to $a$ or applying $(La)^0_0$ to $a$, (i.e. *Alice loves Alice* can be made by applying *loves Alice* to *Alice* or *Alice loves* to *Alice*). The moral is that either of these principles would be plausible if reality were structured as the predicates and sentences of the structural calculus, clearly the *sentences* $((\Box\neg)^1_0 A)^0_0$ and $(\Box(\neg A)^0_0)^0_0$ are different in virtue of their having the form of predication with different predicates and arguments (and similarly the sentences $((Ra)^0_0 a)^0_0$ and $((Ra)^0_1)^0_0$). The problem is that the logical notation is obscuring the true structure of the propositions, which are accurately represented by the relational diagrams depicted in section 3.

But apart from simply rejecting the premises of the Russell-Myhill paradox, the present view has no reason to accept the background logic in which this result is formulated: specifically, they have no reason to accept Extensional-$\beta$. As we argued in section 5, certain $\lambda$-terms in Church's calculus are simply not meaningful by the lights of the structured theorist. Once we have rejected the meaningfulness of these $\lambda$-terms, what reason do we have to accept the instances of Extensional-$\beta$ that involve them? One might point out that in particular cases we have candidate meanings for $\lambda$-terms that would suffice to make the relevant instance of extensional-$\beta$ true. We could interpret $\lambda xy.Lyx$ as *hates*, provided everyone hates people that love them back, and conversely. But while interpretations of particular $\lambda$-terms can be introduced in this *ad hoc* way, we do not have a general guarantee that there are enough interpretations to make all instances of Extensional-$\beta$ true. Consider the following term $\lambda X.Xa$ — illegitimate by the structuralist's lights — of applying to Socrates: one needs a property of properties $H$, that applies to exactly those properties that apply to Socrates. But the structural picture provides no guarantee that there is a such a property. Properties are relatively sparse on this view: you have some fundamental properties and relations — things like *is an electron*, *is a space-time point*, and so on — and then you have the things you can define from them using our building operations. But there is no guarantee that one can build properties with arbitrary extensions.[59] (The analogy with languages is instructive here: one can define many properties in, say, the language of arithmetic, like *being even*, *being prime*, and so on, but there are simply more collections of numbers than formulas of the language so not every collection of numbers is the extension of some predicate.) As we showed in section 6, there are *some* $\lambda$-terms that are meaningful by structuralist lights: for these terms one can even accept the full principle $\beta$, where the biconditional is replaced with an identity. But outside this limited domain, the structuralist has no reason to accept Extensional-$\beta$.

It's also worth remembering that the derivation of the Russell-Myhill paradox turns essentially on the use of a $\lambda$-term that our structuralist rejects. I won't recount the argument here,

[59]Uzquiano (2015), for instance, does not appeal to Extensional-$\beta$ in his derivation of an inconsistency, but rather a comprehension principle that guarantees the existence of properties that have the extension of any open formula. But this comprehension principle is spurious for the same reasons.

but it involves the property of being a proposition which ascribes some property to the proposition that snow is white that it doesn't itself have: $\lambda p \exists X.(p = Xs \wedge \neg Xp)$. While the two occurrences of the bound variable $X$ can be paraphrased away using our coordinated quantifiers (see footnote 68 below), there is no similar trick for achieving quantification into predicating position.[60]

While the view described here is not susceptible to paradoxes that rest on Structure and its variants, the question of consistency of the structuralist theory is highly non-trivial. Let's get a bit more precise about what that theory is. Firstly, there are some logical axioms that correspond to the principles PC, UI and MP of $\mathsf{H}$. Because we are using more general coordinated quantifiers the axiom of UI is slightly more involved. Let $\forall_{\boldsymbol{\sigma}}$ be one of these quantifiers with type $(\boldsymbol{\sigma} \to t) \to \check{\boldsymbol{\sigma}} \to t$. If $\boldsymbol{\sigma}$ is a hatted sequence of types containing the types $\sigma_1 ... \sigma_n$, then we will write $\boldsymbol{a} : \boldsymbol{\sigma}$ for a hatted sequence of terms $a_1 : \sigma_1 ... a_n : \sigma$, when $a_i$ is hatted iff $\sigma_i$ is hatted in $\boldsymbol{\sigma}$, and all the hatted terms are the same. As before, we write $\check{\boldsymbol{a}}$ for the result of deleting all the hatted elements of the sequence $\boldsymbol{a}$. Finally $F\boldsymbol{a}$ is just the term $Fa_1 ... a_n$, ignoring any hats appearing above the $a_i$s. UI may then be reformulated as below, where $\boldsymbol{a} : \boldsymbol{\sigma}$.

**PC** Every instance of a propositional tautology.

**UI** $(\forall_{\boldsymbol{\sigma}} F)\check{\boldsymbol{a}} \to F\boldsymbol{a}$

**MP** From $A \to B$ and $A$ infer $B$

To illustrate an instance of UI where $\boldsymbol{a} = \hat{a}b\hat{a} : \hat{e}e\hat{e}$, we have $(\forall_{\hat{e}e\hat{e}} R)b \to Raba$. It corresponds to the this instance of UI in Church's system: $\forall_e x.Rxbx \to Raba$. Since we have taken identity as primitive we need some axioms that govern identity:

**Identity** $a =_\sigma a$

**Substitution** $a =_\sigma b \to A \to A[b/a]$

In place of $\beta$ and $\eta$ we have provided an equational theory that tells us when two terms $a$ and $b$ denote the same entity:

**Structural Equations** $a =_\sigma b$ when $a = b$ is a structural equation (i.e. $a = b$ is a theorem of the system in figure 3).

Since our language is variable free, we do not have an analogue of the rule Gen. We could introduce variables into the deductive system purely for the purpose of making proofs easier. However our positive theory of structure has a stronger rule that replaces this rule.

The positive theory of structure will be formulated in what Russell called a *logically perfect language*: in a logically perfect language there will be one (logical or non-logical) constant, and no more, for each logically simple entity, and no constants denoting logically complex entities.[61] (This will mean, in line with our previous discussion, that each logical constant of the language, such as $\forall_{\boldsymbol{\sigma}}$ and $\to$, expresses a logically simple entity of the appropriate type.) In such a language we can formulate two principles capturing the structured vision:

[60] A more complicated trick is available for the two $p$s. Interestingly there is a precedent to the ban on $\lambda$ binding into predicating position in relation to the intensional paradoxes: Menzel (1986) describes an untyped first-order theory of properties that has $\lambda$ expressions subject to a similar ban, which is intended to avoid Russell's paradox by preventing the construction the property of not applying to oneself, $\lambda x(\neg xx)$. (Of course, such a property would be ill-typed in a typed setting, whether or not we imposed a ban on bound predicating variables.)

[61] Russell (1940), p25.'In a logically perfect language, there will be one word and no more for every simple object, and everything that is not simple will be expressed by a combination of words, by a combination derived, of course, from the words for the simple things that enter in, one word for each simple component.'

**Distinctness** $a \neq_\sigma b$ provided $a = b$ is not a structural equation.

**Completeness** $F[\boldsymbol{a}/c_1], F[\boldsymbol{a}/c_2], ... \vdash \forall_{\boldsymbol{\sigma}} F\check{\boldsymbol{a}}$

here $[\boldsymbol{a}/c]$ refers to the result of substituting all the hatted elements of $\boldsymbol{a} : \boldsymbol{\sigma}$ with $c$, and $c_1, c_2, ...$ range over all the possible terms in the language of the relevant type.

The intuition behind Distinctness is straightforward: we previously argued, by the correspondence with relational diagrams, that our equational theory contained all and only the true equations between structured entities; so we should include not only the identities the equational theory proves, but also the distinctness claims the theory doesn't prove. There is also an equivalent way of stating Distinctness in which it's relationship with Predicate Argument Structure is clearer: it roughly states that if $A$ is identical to $B$, then the main relations of $A$ and $B$ are the same, and the first arguments, second arguments, and so on, are all the same. (Given any term $A$ of the structural calculus, it is possible to identify a unique logical or non-logical constant $A^{\text{main}}$ appearing in $A$ corresponding to the main relation of the entity expressed by $A$, and unique (possibly complex) terms $A^{\text{arg1}}$, $A^{\text{arg2}}$, ... expressing the first constituent, second constituent, and so on, of the entity denoted by $A$. See the definition of the normal form of $A$ in the appendix. So the revised principle reads:

$$A = B \rightarrow A^{\mathsf{main}} = B^{\mathsf{main}} \wedge A^{\mathsf{arg1}} = B^{\mathsf{arg1}} \wedge ... \wedge A^{\mathsf{arg}n} = B^{\mathsf{arg}n}$$

Evidently Distinctness should not be expected to hold in an arbitrary language: for suppose that we had a terms $F : (e \rightarrow t) \rightarrow (e \rightarrow t)$, $G : e \rightarrow t$ and $H : e \rightarrow t$ — 'female', 'fox' and 'vixen' — that pick out the property modifier of being a female, the property of being a fox, and the structured property of being a female fox respectively. Then the identity $(FG)^0_0 = H$ would be true, even though our equational theory evidently cannot prove this identity — our equational theory contains nothing that could distinguish one constant, like $H$, from another.

Completeness is also only plausible on the assumption that we are theorizing in a logically perfect language. Consider the instance of Completeness for the familiar universal quantifier $\forall_{\hat{\sigma}}$:

$$Fc_1, Fc_2, ... \vdash \forall_{\hat{\sigma}} F$$

The idea here is that we have an expression in the language for every single entity of a given type. Every logically simple entity has a name. And every structured entity is built from the simple ones, using our general way of gluing entities together: complication. Since our language has a device for expressing this too, every entity is denoted by an expression. Thus if for each expression $c : \sigma$, $Fc$ is true, the universal generalization $\forall_{\hat{\sigma}} F$ is true.[62]

The consistency of the structural calculus is a delicate matter. A simple way to establish consistency would be to find a *valuation*: a certain sort of function, $v$, that maps each sentence of the language to a 1 or a 0, such that the axioms are assigned value 1, the rules preserve value 1, and not every sentence is mapped to value 1. Consistency follows from the fact that every theorem, but not every sentence, gets assigned value 1.

This task can be simplified by appealing to a fact about terms of type $t$ in the structural calculus. Say that two terms $M$ and $N$ are *equivalent* if $M = N$ is a structural equation. The fact in question is that every term of type $t$ is equivalent to an expression of the form $Ra_1...a_n$ where $R$ is a constant.[63] So the possible forms that a term of type $t$ can take are $\wedge AB$, $\neg A$,

[62] Completeness is related to the principle Fundamental Completeness in Bacon (2019) and Bacon (2020): it is supposed to capture the idea that everything can be built from the fundamental. Fundamental Completeness is formulated in terms of the notion of 'metaphysical definability' (explained there) without having to use an infinitary rule. Our infinitary rule allows us avoid invoking the special vocabulary of 'metaphysical definability'.

[63] The analogous fact about the $\lambda$-calculus is well-known (see, e.g., Hindley and Seldin (2008)), and can be transferred to the structural calculus using theorem 3.

$=_\sigma ab$, $\forall_{\boldsymbol{\sigma}} Fa_1...a_n$, and $Ra_1...a_n$ where $R$ is a non-logical constant. If the theory is consistent, a function defined on sentences of this form can be extended to a valuation of the language by choosing, for an arbitrary sentence, an equivalent with the desired form. We may finally define a *valuation* as a function $v$ mapping each sentence to 1 or 0 satisfying the following conditions:

- If $A$ and $B$ are equivalent then $v(A) = v(B)$.
- $v(a =_\sigma b) = 1$ iff $a$ and $b$ are equivalent.
- $v(A \wedge B) = \min(v(A), v(B))$
- $v(\neg A) = 1 - v(A)$
- $v(\forall_{\boldsymbol{\sigma}} F\check{\boldsymbol{a}}) = \min_{c:\tau} v(F[\boldsymbol{a}/c])$

Note that the last clause gives the quantifier a substitutional interpretation. This ought to be equivalent to a standard interpretation, in a logically perfect language, because every entity is denoted by an expression.

One naïve strategy for constructing a valuation is to assign a truth value $|Ra_1...a_n|$ arbitrarily to atomic formulae[64] where $R$ is non-logical, and then proceed to extend it to other sentences (either inductively, or some other way). Inductive definitions of this form are familiar from definitions of truth valuations for propositional languages, or first-order languages with the quantifiers interpreted substitutionally.

Unfortunately, the naïve strategy does not work: indeed, there will be some extensions we could assign to the non-logical relation constants for which these constraints simply cannot be satisfied. The reasons have nothing to do with the Russell-Myhill paradox, but relate to another paradox in higher-order logic due to Arthur Prior (Prior (1961)). Unlike the Russell-Myhill paradox, Prior's theorem does not rest on any assumptions about propositional granularity: it applies equally to coarse grained theories, such as the Fregean theory, in which there are only two propositions. In orthodox notation it states:

**Prior's Theorem** $Q\forall p(Qp \to \neg p) \to \exists p(Qp \wedge p) \wedge \exists p(Qp \wedge \neg p)$

Prior's theorem can be formulated and proved in our system as well. I will not revisit here the reasons this result is puzzling; for now it suffices to say that it is simply a commitment of the classical logical assumptions we have made so far.[65]

The naïve strategy cannot work, then, for the following reason: suppose that $Q$ was a non-logical constant, and I were simply to set its extension to contain all equivalents of $(\vdash \forall_{\hat{t}\hat{t}} \lambda p \lambda q(Qp \to \neg q))^s$ — the translation of Prior's problematic sentence into the structural calculus — and nothing else. If the clauses for assigning truth values to propositions were satisfied Prior's theorem would get a value of 0, yet if our truth assignment is sound for our logic it must assign Prior's theorem a value of 1.

The naïve strategy might have looked plausible if one thought that you could *inductively* extend any assignment of truth values to the atomic sentences to a valuation, as one does in the propositional calculus, or first-order logic with substitutional quantifiers. But this is simply not possible in propositionally quantified logic, or its extensions like higher-order logic. The problem is this: the clause for the quantifier is not well-founded. We have defined the truth value of $(\forall_{\boldsymbol{\sigma}} F)\boldsymbol{a}$ in terms of the truth value of $F[\boldsymbol{a}/c]$ for arbitrary $c$. But the term $c$ might itself

[64] At least, arbitrarily subject to the constraint that if $a_1...a_n$ are respectively equivalent to $a'_1...a'_n$ then $|Ra_1...a_n| = |Ra'_1...a'_n|$.

[65] If you want to revisit these assumptions, see for instance Tucker and Thomason (2011), Tucker (2018), Kaplan (1995), Deutsch (2014), Bacon et al. (2016), Bacon and Russell (2019).

be highly logically complex: it might introduce new quantifiers, which have to be evaluated themselves, ad infinitum. For concreteness, note that to evaluate $\exists_{\hat{t}}\neg$ we must first evaluate $\neg A$ for every sentence $A$: one such sentence is $\exists_{\hat{t}}\neg$ itself! So to evaluate $\exists_{\hat{t}}\neg$ we have to evaluate $\neg\exists_{\hat{t}}\neg$, which involves figuring out the truth value of the sentence you wanted to calculate in the first place. In the present case this circularity doesn't matter because the existential has a true witness, $\neg\bot$, for which the evaluation procedure does terminate. But the circularity makes special trouble in cases like Prior's paradox, where the candidate witnesses of an existential, or counterexamples to a universal, are circular in this way. See also Kripke (1976) p331-332 for some relevant discussion.

Prior's theorem tells us that not every assignment of extensions to the non-logical constants extends to a valuation. Moreover, despite initial appearances, our constraints on $v$ do not have the form of an inductive definition of truth. None of this is to say, however, that there can't be any valuations satisfying the constraints. For instance, in Kripke's discussion of substitutional quantification, he notes that 'even for some substitution classes violating the condition that quantifiers not occur in terms, there may be another proof, different from the [inductive] one just given, of the existence and uniqueness of a set [valuation, in this case] with the desired properties. Though in this case the set would probably not be said to be inductively defined, we could still say that substitutional truth was uniquely characterized and thus well defined.' He then goes on to describe a special case in which valuations of this sort do exist. I thus put forward the following conjecture:

**Conjecture 1.** *There exists a valuation on the structural calculus over any signature of non-logical constants.*

While the naïve strategy of defining a truth valuation inductively on formula complexity doesn't work, there are other measures of complexity which do decrease when moving from a generalization to an instance, even when the instantiating term is very complex in the same sense. For example, we could treat a formula of the form $A =_\sigma B$ as having complexity 0, irrespective of the complexity of $A$ and $B$. Then a quantified claim whose $\lambda$-translation binds variables only appearing under the scope of $=$ will be such that its instances have lesser complexity in the new sense, and this can be leveraged to prove a restricted version of our conjecture, theorem 2 below.[66] Indeed, I suspect the conjecture can be proved by choosing a sufficiently crafty measure of complexity, although my attempts so far have failed.

However we can use this strategy, including the new measure of complexity, to prove various limited consistency results. In particular, one can prove that, provided we quarantine the issues to do with propositional quantification and Prior's paradox, the theory is consistent. A simple way to achieve this is to remove the propositional quantifiers from the language: the quantifiers $\forall_{\boldsymbol{\sigma}}$ where the hatted types in $\boldsymbol{\sigma}$ are $t$. This shows that the theory is, in a natural sense, immune to the Russell-Myhill paradox, as that is formulated in terms of quantification into *operator* position.

**Theorem 2.** *Any assignment of truth values to atomic formulae $|Ra_1...a_n|$ can be extended to a valuation on the restricted language that contains no propositional quantifiers.*

This is not the venue to give a detailed proof of this theorem, although I put an informal explanation as to why it is true in a footnote.[67] It's worth noting that this consistency argument

[66] And since the value of $A =_\sigma B$ depends entirely on $A$ and $B$ (for instance, it does not depend on $v(A)$ or $v(B)$), the truth values of complexity 0 formulae are completely determined.

[67] It is instructive to see why the analogous theorem would be false in the case of the full $\lambda$-calculus. In $\mathsf{H}$, every closed sentence $A$ is logically equivalent to a sentence in $\beta\eta$ prenex normal form: $\forall x_1 \exists x_2 .... \forall x_{n-1} \exists x_n \bigvee_i \bigwedge_j A_{ij}(x_1, ..., x_n)$ where each $A_{ij}(x_1, ..., x_n)$ is either an atomic formula or the negation

works in fragments of the $\lambda$-calculus that go beyond the structural $\lambda$-terms. For instance system you get from Church's by replacing Identity with Concretion (or, equivalently, augmenting the structural type system with Exchange, Contraction and Weakening).

It is also worth mentioning that there is a closely related theory that contains the propositional quantifiers and which is provably consistent. In this theory the logical expressions are not treated as constants but as syncategorematic; the details are presented in Bacon (Forthcoming). The picture behind this theory is that the logical operations are not entities in their own right but merely further ways of combining entities to make structured entities, like application and complication in the present theory. (There are no entities corresponding to application or complication combinators because $\lambda Xy.Xy$ and $\lambda XY.(XY)^n_m$ are not structural terms.) The idea that logical words do not stand for anything in reality, they are just further modes of combination, was a central thesis of the logical atomists (see Russell (1940) p39 and Wittgenstein (1961) §5.4).

Lastly, let me mention some limitative results. Evidently the Russell-Myhill paradox tells us that the structural calculus cannot be extended with arbitrary $\lambda$-terms satisfying extensional $\beta$, or combinatory terms satisfying extensional $\mathsf{I}$, $\mathsf{K}$ and $\mathsf{S}$. But one might wonder which terms, and corresponding conversion principles, in particular reduce the theory to inconsistency. A complete survey would take us too far afield, but here is a central limitative result: there cannot be an extensional notion of application for operators. More precisely: one cannot introduce an operation $App^{t\to t} : (t \to t) \to t \to t$ that satisfies.

**Extensional Application** $App^{t\to t}MA \leftrightarrow MA$, where $M : t \to t$ and $A : t$.

The reason is simply this: one can simulate quantification into the position of a predicating operator since any case where an operator variable appears in predicate position, like $XA$, is extensionally equivalent to a case where it is in argument position, $App^{t\to t}XA$. One is then in a position to define the Russell-Myhill operator, and generate a paradox in the usual way.[68]

---

of one, in the variables $x_1, ..., x_n$. The only atomic formulas are of the form:

- $Ra_1...a_n$ where $R$ is a non-logical constant.
- $M =_\sigma N$
- $XN_1...N_n$ where $X$ is a relational variable.
- $p$ where $p$ is a propositional variable.

To evaluate an arbitrary quantified claim with the substitutional interpretation it thus suffices to be able to evaluate the truth value of any of the above, after replacing the free variables $x_1, ..., x_n$ in those formulae with closed terms. The first two cases are evaluated immediately: we have directly said whether atomic claims of the form $Ra_1...a_n$ and identities $M =_\sigma N$ are true or false. The third case at least contains the predicating variable $X$ free, so we need to know the truth value of $MN_1...N_k$ for any relational expression $M$ of any complexity. In the fourth case, $p$ is the only variable, so we need to know the truth value of $A$ for any sentence $A$. Since $M$ and $A$ can be arbitrarily complex, the naïve evaluation procedure will never terminate. But if we were restricting attention to formulas that are in the range of a translation from the structural calculus, the third case couldn't arise, because no bound variables can appear in predicating position. Moreover, if we have banned propositional quantifiers the last case doesn't arise either, so the evaluation procedure always terminates.

[68] Recall that the Russell-Myhill paradox traded essentially on the operator expression $\lambda p\exists X(p = Xs \wedge \neg Xp)$, which we have seen the structuralist has independent reason to believe does not denote anything. This $\lambda$-term fails to be structural for two reasons: the bound variables $p$ and $X$ appear twice in the body, and the bound variable $X$ appears in predicating position. We can solve the first sort of problem as follows. Suppose that $y : \sigma, z : \sigma \vdash A : t$ is a structural sequent. Then the term $\lambda x.(A[x/y, x/z])$ is a legitimate structural term except for the fact that the bound variable $x$ appears twice in $A$. On the other hand, the term $\lambda x\exists_{\hat{\sigma}\hat{\sigma}\hat{\sigma}}\lambda wyz.(x = w \wedge A)$ can be shown to be structural: it is the property of being an $x$ such that there are $w$, $y$ and $z$, all identical, such that $x = w$ and $A(y, z)$. This is clearly ought to be coextensive with the property of being an $x$ such that $A(x, x)$ (i.e. the illegitimate term $\lambda x.(A[x/y, x/z])$).

The other issue is that $X$ appears in predicating position, for instance in the subformula $Xp$. But if we had an extensional application combinator, $App^{t\to t}$, we could replace these with the materially equivalent $App^{t\to t}Xp$.

Why should we have *expected* there to be a term that satisfies Extensional Application? The $\lambda$ and combinatory calculi posit entities, $\lambda X\lambda p.Xp$ and $I^{t\to t}$, that satisfy stronger propositional identities, such as $I^{t\to t}MA =_t MA$, that imply the biconditional in Extensional Application. But these stronger equations are unacceptable to this structuralist, and so cannot serve as an argument that there ought to be entities satisfying the weaker biconditional. A more promising line of argument starts from the assumption that there could be *simple* binary relations between operators and propositions (i.e. of type $(t\to t)\to t\to t)$, and that their extensions ought not to be unconstrained — God should be free to set the extensions of the simple properties and relations as he pleases, as it were. This draws on a different analogy between reality and language: the extensions of the non-logical predicates and relations in a language are completely unconstrained in the usual sorts of model theories for first and higher-order logics. Prior's paradox teaches us that the present structuralist must reject this part of the analogy between reality and language — one cannot introduce a simple operator that applies only to Prior's proposition. But even if God could set the extensions of relations between operators and propositions however he liked, that would still not guarantee that he could stipulate that such a relation hold between an operator and a proposition if and only if the operator applies to the proposition: whether a structured operator applies to a structured proposition will depend on the extensions of the simple propositions, properties and relations, so such a stipulation would be circular and so not guaranteed to be satisfiable.[69][70]

We mentioned in section 5 that certain operations definable in the $\lambda$-calculus, like predicate conjunction, cannot be be defined in structural logic. This is true als of predicate and operator negation — $\lambda X\lambda y.\neg(Xy)$ and $\lambda X\lambda p.\neg(Xp)$. The above suggests that they cannot even be introduced as new primitives. For instance, if we introduced an operator negation $\neg_{t\to t} : (t\to t)\to t\to t$, subject to the law $\neg_{t\to t}Xp \leftrightarrow \neg(Xp)$, we could define another operation, $(\neg_{t\to t}\neg_{t\to t})^1_0$ that obeys Extensional Application.

Let me conclude the paper by raising some questions for further investigation (in addition, of course, to the presently open question of the consistency of the structural calculus). While we have focused here on a particular diagrammatic theory of structured propositions, there are many other theories of structured propositions I have not considered, and one might wonder how general the approach to blocking paradox explored in this paper is. There are several variants (seven to be exact) of the present theory that share its paradox blocking aspects, which can be developed easily in the present framework. They are obtained by adding any of the seven remaining combinations of converses, reflexizations and vacuous abstraction to the structural calculus.[71] (We observed, already, that introducing the ability to quantify into predicating position, as permitted by the typing rule Identity, reintroduces paradox.) Another question, then, is whether natural diagrammatic representations of these other type systems

Indeed, using these two tricks we can construct a term — namely, $\lambda p\exists_{\hat{t}\hat{t}\hat{t}}\exists_{t(t\hat{\to}t)t(t\hat{\to}t)t}\lambda qXq'X'q''(p = q \wedge App^{t\to t}Xs = q' \wedge \neg(App^{t\to t}X'q''))$ — which can be shown to generate the original form of the Russell-Myhill paradox. .

[69] Indeed, by theorem 2 we know that any extension assigned to the non-logical constants in a language without propositional quantifiers can be extended to a valuation, and yet the paradox from Extensional Application can be formulated in such a language so we know that no extensions can be assigned to a constant of type $(t\to t)\to t\to t$ that would extend to a valuation under which Extensional Application would hold.

[70] There is a more superficial reason the paradox from Extensional Application appears surprising: it seems (at first glance) as though you couldn't do anything with the claim $App^{t\to t}Xp$ that you couldn't simply do with $Xp$, suggesting that an operation like $App^{t\to t}$ is redundant. But this is, of course, wrong: $App^{t\to t}$ allows one to do exactly the sorts of things that lead to the Russell-Myhill paradox, like simulate quantification into predicating position.

[71] In terms of the $\lambda$-rules they can be described by adding any non-empty subset of {Weakening, Exchange, Contraction} to the $\lambda$-rules for the structural calculus. The most liberal of these is equivalently obtained by replacing Identity with Concretion in Church's theory.

can be presented.[72] And finally, one question I have not touched on is to what extent present work in semantics would have to be revised in light of the metaphysics postulated by such a theory. The influence of the theories of Church and Curry is so pervasive in semantics that it would be hard to separate the field from the theory.[73] The structural calculus is more restrictive than the theories of Church and Curry, and so much work remains to show that the discipline of semantics can be sustained within that framework.[74]

# 9 Appendix

## 9.1 Translating between the structural calculus, the $\lambda$-calculus and relational diagrams

In the text we defined a translation, $(-)^\lambda$,taking terms of the structural calculus to the $\lambda$-calculus, defined by setting $(c)^\lambda = c$ when $c$ is a constant, and setting $((MN)^n_m)^\lambda = \lambda x_1...x_m y_1...y_n.M^\lambda x_1...x_m(N^\lambda y_1...y_n)$ for terms formed by complication. It is possible to define a reverse translation from $\lambda$-terms of our restricted $\lambda$-calculus, given by the rules in figure 6, into the structural calculus . We do this by way of a more general translation that maps typing statements of the form $\Gamma \vdash M : \sigma$ to terms of the variable free language. Instead of defining the translation inductively on the structure of the $\lambda$-terms, the trick is to define it inductively on the derivations of the typing judgments. In this system every typing judgment, $\Gamma \vdash M : \tau$, has a unique derivation (unlike the full system), so it is in fact possible to define the translation directly on the typing judgments themselves.[75] For example, we know that any judgment of the form $\Gamma \vdash \lambda x.M : \sigma \to \tau$ must have been derived by Abstraction from $\Gamma, x : \sigma \vdash M : \tau$, so that we can define the translation of the former typing judgment in terms of the translation of the latter, which we may assume has already been assigned an interpretation inductively.

- Constants: $(\vdash c : \sigma)^s = c$

[72]For instance, if the holes could be distuinguished, by colors say, one could represent $R$ and its converse by two binary relational diagrams with the same label '$R$', but with its two holes in different orders. This might correspond to adding Exchange to the typing system. One could also imagine having two holes of the same type having the same color, which means they always get filled simultaneously.

[73]The $\lambda$-calculus has been a central tool in semantics since Montague (1973). Combinatory logic, by contrast, plays a foundational role in the direct compositionality program, tracing back to Jacobson (1999).

[74]Some methodological considerations in favour of orthodox higher-order logics can be found in Williamson (2013).

[75]First, observe that it is not possible to derive a typing statement of the form $\Gamma \vdash x : \sigma$ where $x$ is a variable, since none of the rules introduce a lone variable on the right. We can then prove by induction on $n$ that every term that has a derivation of length $\leq n$ has a unique derivation of length $\leq n$. Since this holds for every $n$, each term that has a derivation must have a unique derivation of any length whatsoever. Derivations of length one are instances of Constants, and so are uniquely determined by the constant. Since we have ruled out variables, all derivable typing statements have either constants, $\lambda$-terms (have the form $\lambda x.M$) or application terms (have the form $(MN)$) on the right of the turnstyle. If it is a constant, then it could only have been produced by the rule Constants. Any two derivations of length $\leq n+1$ of a $\lambda$-term $\lambda x.M$ must end with an application of Abstraction, preceded by an $\leq n$-length derivation of $M$. Both of these shorter derivations must be identical by the inductive hypothesis, so the derivations of $\lambda x.M$ must be identical. Any two derivations of length $\leq n+1$ of an application term $(MN)$ where $N$ is a variable, end with an application of Concretion preceded by $\leq n$-length derivations of $M$, which are the same by the IH (it cannot be derived from Application, since that would require a derivation of $N$ which we observed is impossible when $N$ is a variable). Finally, if $N$ is not a variable, then $(MN)$ could only have been derived by an application of Application preceded by $\leq n$-length derivations of $M$ and of $N$, which are unique for $M$ and $N$ among those lengths by the IH. Either way derivations of length $\leq n+1$ of $MN$ are unique. (Here derivation length is defined in the obvious way: a single application of Constants has length 1, derivations ending in Concretion or Abstraction have length 1 plus the length of their subderivation, a derivation ending in Application has length 1 plus the maximum of the lengths of the two subderivations.)

| | | | |
|---|---|---|---|
| $\overline{(MN)^n_m a_1...a_m b_1...b_n = Ma_1...a_m(Nb_1...b_n)}$ | $\gamma$ | $\overline{M=M}$ | $\iota$ |
| $\dfrac{M=N}{N=M}$ | $\sigma$ | $\dfrac{M=N \qquad N=P}{M=P}$ | $\tau$ |
| $\dfrac{M=M'}{MN=M'N}$ | $\mu$ | $\dfrac{N=N'}{MN=MN'}$ | $\nu$ |
| $\dfrac{Mx=Nx}{M=N}$ | $\zeta$ | (provided $x$ is not free in $M, N$) | |

Table 3: Equational Theory A

- Concretion: $(\Gamma, x : \sigma \vdash Mx : \tau)^s = (\Gamma \vdash M : \sigma \to \tau)^s$
- Abstraction: $(\Gamma \vdash \lambda x.M : \sigma \to \tau)^s = (\Gamma, x : \sigma \vdash M : \tau)^s$
- Application: $(\Gamma, \Delta \vdash MN : \tau)^s = ((\Gamma \vdash M : \sigma \to \tau)^s(\Delta \vdash N : \sigma)^s)^n_m$, where $m$ and $n$ are the length of $\Gamma$ and $\Delta$ respectively.

Now as a special case, any closed term $M$ that can be constructed from our rules will have a derivation ending in $\vdash M : \sigma$, and so may be converted into a term of our new language by the above translation.[76] Last, but not least, we can associate expressions of the structural calculus with diagrams: an individual constant gets associated with a labelled grey circle, a propositional constant with a labelled grey rectangle, and an $n$-ary relational constant $R$ corresponds to a simple relational diagram with $n$ holes, as in rules 1,2 and 3. Assuming we have assigned relational diagrams to $M$ and $N$, the term $(MN)^n_m$ corresponds to the relational diagram obtained from our rule 4 applied to the diagrams associated with $M$ and $N$.

## 9.2 The theory of structural equations

The goal here is to define an equational theory for the structural calculus such that an equation between structural terms in derivable if and only if the structural terms are associated with the same relational diagram. One axiomatization of these identities is offered in table 3. We call this Equational Theory A. There is a subtlety here, namely that the rule $\zeta$ is formulated in terms of a variable parameter $x$. To reason in this theory we therefore need to add infinitely many variables to the language at each type. While our notation is variable free — in the sense that there are no *bound* variables, and our expressive needs can be met without them — we have adopted *free* variables so as to simplify our equational theory. For this reason I will call them 'deductive' variables. In fact, it is possible to eliminate variables altogether, and reaxiomatize the theory in an entirely variable free way by replacing the conclusions of $\mu$ and $\nu$ with complications (e.g. $\mu$ becomes $\mu^n_m$, where the conclusion is replaced by $(MN)^n_m = (M'N)^n_m$) and replacing the rules $\gamma$ and $\zeta$ with the following pair of closed equations:

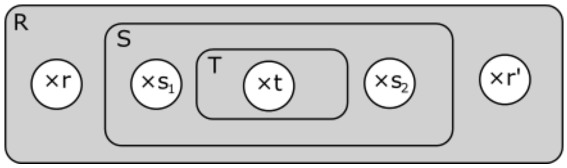
**Recompose** $(R(ST)^t_{s_1})^{s_1+t+s_2}_r = ((RS)^{s_1+1+s_2}_r T)^t_{r+s_1}$

**Rearrange** $((RS)^s_{r_1} T)^t_{r_1+s+r_2} = ((RT)^t_{r_1+1+r_2} S)^s_{r_1}$

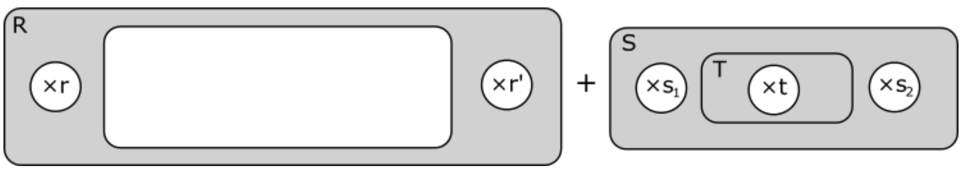
[76] In general, one would also need to show that this translation is consistent: that two different derivations of the same typing judgment couldn't yield different translates. But, unlike the full system in figure 1, it's actually impossible to find two derivations of the same sequent in the pared down system.

We call this theory Equational Theory B.

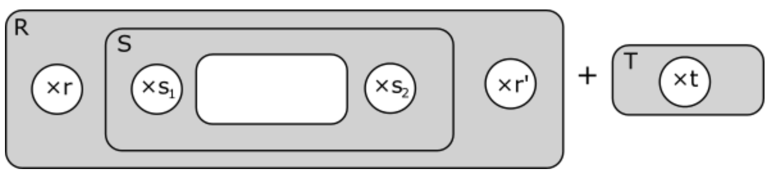
Let us show that these two new equations of Theory B are sound with respect to the diagrammatic reasoning. For simplicity we depict the case where $R$, $S$ and $T$ are themselves unstructured and their holes are individual or proposition shaped (nothing important rests on this). The first equation gives us two representations of the following structured relation, writing '$\times r$' over a hole indicate that this stands for a sequence of $r$ holes:



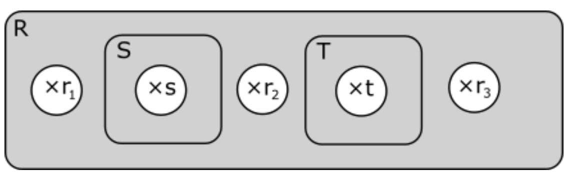

We can construct it firstly by plugging $T$ into $S$ after the $s_1$th hole to make $(ST)^t_{s_1}$ (see the diagram on the right below), and the plugging the result into the $r$th hole of $R$ (on the left).



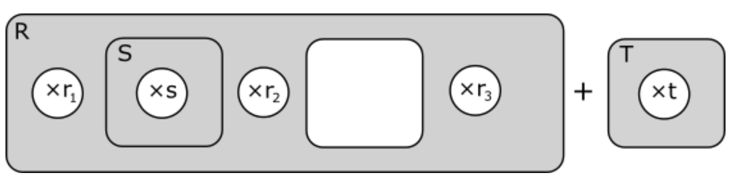

Alternatively, we could plug $S$ after the $r$th hole of $R$ (left), and $T$ after the $r + s_1$th hole (right):



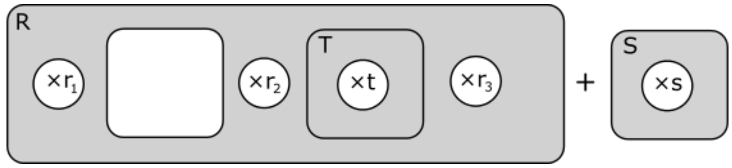

Rearrange concerns the following sort of structured relation:




You can make it by plugging in $S$ after the $r_1$th hole in $R$ (left) and then plugging $T$ (right) into the $r_1 + s + r_2$th hole of the result:




remember that the $r_1 + 1 + r_2$th hole in $R$ is the same as the $r_1 + s + r_2$th hole, once $S$ has be inserted in $R$. Thus we could achieve the same result by first plugging $T$ immediately after the $r_1 + 1 + r_2$th hole in $R$ (left) and then plugging $S$ after the $r_1$th hole.




That these equations suffice to prove all the structural equations, however, is more involved and treated in the next appendix.

## 9.3 Structural equations, $\beta\eta$-equivalence and sameness of relational diagrams

Now we will sketch a proof of the following theorem.

**Theorem 3.** *The following are equivalent, where $M$ and $N$ are terms of the structural calculus:*

1. *$M$ and $N$ are associated with identical relational diagrams.*
2. *$M^\lambda = N^\lambda$ is derivable in the theory of $\eta\beta$-equivalence (see e.g. Hindley and Seldin (2008) chapters 6 and 7).*
3. *$M = N$ is derivable in Equational Theory A (see figure 3).*
4. *$M = N$ is derivable in Equational Theory B (without deductive variables).*

The full proof of this theorem is technical and not suitable for this venue, however the argumentative strategy may be presented reasonably informally. I'll begin with equivalences between 1, 2, and 3: there will be some redundancy here, but the proof strategies for each implication shed light on the connections between the first three criteria of identity. The equivalence of 2 and 3 are the least illuminating, but can be shown by a routine induction on the length of derivations. The implication $3 \Rightarrow 1$ can also be shown by induction on derivations, and has an instructive visualization: the idea is to think of deductive variables as constituents, and assign them the simple relational diagram you would assign a constant of the same type (except labeled with the deductive variable in question). The claim that the rule $\zeta$ preserves sameness of associated diagram amount to saying that if $Mx$ and $Nx$ correspond to the same diagram, then the result of poking a hole where $x$ is in both diagrams will be the same. The remaining rules clearly preserve the sameness of the associated diagram. The implication $1 \Rightarrow 3$ is a little more involved. It is proved by assigning to each relational diagram a 'normal form' term of the structural calculus, which have the property that no two normal form terms of the structural calculus are provably equivalent. The terms in *normal form* are defined inductively as follows:

- Constants are in normal form.
- If $N_1 ... N_k$ are in normal form, and $R$ is a relational constant, then

$$(...(RN_1)^{n_1}_{r_1} N_2)^{n_2}_{r_1+n_1+r_2} N_3)^{n_3}_{r_1+n_1+r_2+n_2+r_3} ... N_k)^{n_k}_{r_1+n_1+...+r_{k+1}}$$

  is in normal form (provided the result is well-typed).

We assign simple relational diagrams their corresponding constants. Now every other relational diagram will consist of an outer grey box, the *main* relation, labeled by a constant $R$, with a number of holes, some of which may be filled with further relational diagrams, $d_1 ... d_k$. We'll suppose there is a stretch of $r_1$ unfilled holes, followed by a hole filled with $d_1$ (which itself contributes $n_1$ further holes), then a stretch of $r_2$ unfilled holes, and then a hole filled with $d_2$ (contributing $n_2$ further holes), and so on. Assuming, for induction, that we have assigned normal forms to $d_1 ... d_k$, $N_1 ... N_k$, we may assign the entire diagram the normal form $(...(RN_1)^{n_1}_{r_1} N_2)^{n_2}_{r_1+n_1+r_2} ... N_k)^{n_k}_{r_1+n_1+...+r_{k+1}}$. $2 \Rightarrow 1$ follows by showing that every term of the structural calculus is equivalent to the normal form associated with its relational diagram.

To complete the theorem we will show the equivalence of 3 and 4, thus connecting the final variable free theory, Equational Theory B, to our other criteria of identity. This is crucial if we care about a notation that completely dispenses with variables.

Recall that we have added deductive variables to the structural calculus, treating them as if they were new constants. Given a structural term $M$, in a deductive variable $x : \sigma$, we will define another structural term, $[x]M$, that doesn't contain $x$. In terms of relational diagrams, we can think of $[x]M$ as the result of poking holes in all the places that $x$ appears in the relational diagram in $M$. This operation is well-defined provided $x$ does not appear in predicating position, and if $M$ contains $k$ occurrences of $x$, $[x]M$ will have $k$ more holes than $M$ does. Thus

If $M : \tau$, $[x]M : \sigma^k \to \tau$

where $\sigma^0 \to \tau = \tau$ and $\sigma^{n+1} \to \tau = \sigma \to \sigma^n \to \tau$. It will be convenient to write $c(M)$ for the number of occurrences of $x$ in $M$ ($x$ will not vary throughout this argument).

For any term $M$ we define $c(M)$ to be the total number of occurrences of the variable $x$ in $M$.

**Definition 4** (Multi-abstraction)**.** *Suppose that $M$ is a structural term. We define $[x]M$ as follows.*

- *$[x].M$ is undefined when $M$ is a variable or constant.*
- *$[x].(Mx)^0_m = [x]M$.*
- *$[x].(MN)^n_m = (([x]M)([x]N))^{n+c(N)}_{m+c(M)}$ when the above case doesn't apply, and $[x]M$ and $[x]N$ are defined.*

We begin by showing that the following useful rule is admissible in Equational Theory B:

$$\frac{M = N}{[x].M = [x].N}$$

Where the instances include all an only the cases where $[x]M$ and $[x]N$ are defined.

Although we ultimately want to use Theory B to dispense with deductive variables, we can still apply the rules of Theory B to languages that contain deductive variables. Note that in a language with no deductive variables $[x]M$ is always undefined. We'll say that a proof of $M = N$ in Theory B is *nice* if it only involves free variables appearing in $M$ and $N$.

**Proposition 5.** *If there is a nice proof of $M = N$ in Equational Theory B, then $[x]M$ is defined iff $[x]N$ is defined, and if they are defined there is a nice proof of $[x]M = [x]N$ in Equational Theory B.*

*Proof.* We prove this by induction on the length of the deduction in this theory. Let's start with Recompose. The two sides are defined in exactly the same conditions: (i) $[x]R$ and $[x]S$ are defined, that $T$ is the variable $x$ and $t = 0$, (ii) $T$ is not $x$ and $[x]R$, $[x]S$, $[x]T$ are defined.

(i) Suppose $[x]R$ and $[x]S$ are defined, that $T$ is the variable $x$ and $t = 0$. Then $[x](R(ST)^0_{s_1})^{s_1+s_2}_r$ and $[x]((RS)^{s_1+1+s_2}_r T)^0_{r+s_1}$ amount to the very same term: $([x]R[x]S)^{s_1+c(S)+1+s_2}_{r+c(R)}$, and so the identity is an instance of the self-identity axiom $\iota$.

(ii) Suppose that $T$ is not $x$ and $[x]R$, $[x]S$, $[x]T$ are defined. Then $[x](R(ST)^t_{s_1})^{s_1+t+s_2}_r$ and $[x]((RS)^{s_1+1+s_2}_r T)^t_{r+s_1}$ are defined and, moreover, form another instance of Recompose, where $r$ is $r + c(R)$, $t$ is $t + c(T)$ and $s_1$ is $s_1 + c(S)$.

In either case $[x](R(ST)^t_{s_1})^{s_1+t+s_2}_r = [x]((RS)^{s_1+1+s_2}_r T)^t_{r+s_1}$ is a theorem of Theory B. Clearly $x$ doesn't appear in any of these derivations, so they are nice.

Rearrange is proved similarly. The two sides are defined in exactly the same conditions: (i) $[x]R$, $[x]S$ and $[x]T$ are defined, (ii) $[x]R$ and $[x]S$ are defined, $t = 0$ and $T$ is $x$, (iii) $[x]R$ and $[x]T$ are defined, $s = 0$ and $S$ is $x$, (iv) $[x]R$ is defined and $t = s = 0$ and $T$ and $S$ are $x$. They are similarly proved by suitably chosen instances of of the axioms.

The rule $\mu^n_m$: Assume for the inductive hypothesis that $M = M'$ is a theorem. By the inductive hypothesis, $[x]M$ is defined iff $[x]M'$ is, so that $[x](MN)^n_m$ is defined iff $[x](M'N)^n_m$ is. Moreover, $[x](MN)^n_m$ reduces to $[x]M$ or $([x]M[x]N)^{n+c(N)}_{m+c(M)}$ and $[x](M'N)^n_m$ reduces to $[x]M'$ or $([x]M'[x]N)^{n+c(N)}_{m+c(M')}$ depending on whether $N$ is a variable. Either way the resulting equation may be derived without using $x$ by the inductive hypothesis (that $[x]M = [x]M'$ can be proved without using $x$), or the inductive hypothesis with another application of $\mu^n_m$.

For the rule $\nu^n_m$ we first observe that we cannot prove an equation $N = N'$ unless $N$ and $N'$ have the same number of occurrences of $x$ (as clearly each of the equational rules preserve this property). This means in particular that you cannot derive $N = x$ or $x = N'$ unless $N$ or $N'$ is $x$ respectively. In the case that they are both $x$, then $[x](Mx)^n_m$ and $[x](Mx)^n_m$ in the same case: $n = 0$ and $[x]M$ is defined. Moreover, the equation is an instance of self identity $\iota$. In the other case, the left-hand-side is defined exactly when both $[x]M$ and $[x]N$ are defined, and the right-hand-side iff $[x]M$ and $[x]N'$ are defined, so given the inductive hypothesis they are defined in the same circumstances. Moreover, the equation $[x](MN)^n_m = [x](MN')^n_m$ amounts to $([x]M[x]N)^{n+c(N)}_{m+c(M)} = ([x]M[x]N')^{n+c(N')}_{m+c(M)}$ which can be derive (without anywhere using $x$) from an instance of $\nu^n_m$.

The cases of transitivity ($\tau$), symmetry ($\sigma$) and identity ($\iota$) are all straightforward. □

**Theorem 6.** *Suppose $M$ and $N$ are terms of the structural calculus that do not contain deductive variables. Then $M = N$ is derivable in Theory A (possibly using deductive variables) iff it is derivable in Theory B without using deductive variables.*

*Proof.* First, we prove a lemma: if $M = N$ is an equation of an arbitrary language and is derivable in Theory A, it has a nice derivation in Theory B.

This is proved by induction on the length of proofs in theory A. The only non-trivial cases are $\zeta$ and $\tau$. Suppose that $x$ is not free in $M$ and $N$, and $M = N$ was proved by an instance $\zeta$ from $Mx = Nx$. By the inductive hypothesis there is a nice derivation of $Mx = Nx$ in Theory B. By proposition 5, we have a nice derivation of $[x]Mx = [x]Nx$ in Theory B, which is, by definition of $[x]$, a nice derivation of $M = N$ as required. Now suppose $M = N$ was derived in Theory A by $\tau$ from $M = P$ and $P = N$. So by the inductive hypothesis we have nice derivations of $M = P$ and $P = N$ in theory B, so we can conclude (possibly non-nicely) that $M = N$ in theory $B$ by applying $\tau$. However, an easy induction shows that if $Q = R$ is derivable in Theory A, $Q$ and $R$ contain the same variables, so $M, N$ and $P$ contain the same free variables and this derivation is nice after all.

It follows that every theorem, $M = N$, of Theory A, where $M = N$ doesn't contain deductive variables, is a theorem of Theory B. (Note the proof of $M = N$ in Theory A might involve deductive variables even when $M$ and $N$ don't, but the proof in Theory B should be carried out in a language without deductive variables.)

Conversely, every equation of Theory B can be derived in Theory A, because you can derive Rearrange, Recompose, $\mu_n^m$ and $\nu_n^m$ in Theory A (essentially using $\gamma$ and $\zeta$). $\square$

# References


P. B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof*. Kluwer Academic Publishers, 2002.

Andrew Bacon. The broadest necessity. *Journal of Philosophical Logic*, 47(5):733–783, 2018. doi: 10.1007/s10992-017-9447-9.

Andrew Bacon. Substitution structures. *Journal of Philosophical Logic*, 48(6):1017–1075, 2019. doi: 10.1007/s10992-019-09505-z.

Andrew Bacon. Logical combinatorialism. *Philosophical Review*, 2020.

Andrew Bacon. *A Philosophical Introduction to Higher-order Logics*. Routledge, Forthcoming.

Andrew Bacon and Jeffrey Sanford Russell. The logic of opacity. *Philosophy and Phenomenological Research*, 99(1):81–114, 2019.

Andrew Bacon, John Hawthorne, and Gabriel Uzquiano. Higher-order free logic and the prior-kaplan paradox. *Canadian Journal of Philosophy*, 46(4-5):493–541, 2016. doi: 10.1080/00455091.2016.1201387.

Ralf Bader. The fundamental and the brute. *Philosophical Studies*, 178(4):1121–1142, 2021. doi: 10.1007/s11098-020-01486-z.

Mark Crimmins and John Perry. The prince and the phone booth: Reporting puzzling beliefs. *Journal of Philosophy*, 86(12):685, 1989. doi: 10.2307/2027014.

H. B. Curry. Grundlagen der kombinatorischen logik. *American Journal of Mathematics*, 52 (3):509–536, 1930.

Nicolaas Govert De Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the church-rosser theorem. In *Indagationes Mathematicae (Proceedings)*, volume 75, pages 381–392. North-Holland, 1972.

Harry Deutsch. Resolution of some paradoxes of propositions. *Analysis*, 74(1):26–34, 2014. doi: 10.1093/analys/ant089.

T. Scott Dixon. Plural slot theory. In Karen Bennett and Dean Zimmerman, editors, *Oxford Studies in Metaphysics Volume 11*, pages 193–223. Oxford: Oxford University Press, 2018.

Cian Dorr. Non-symmetric relations. In Dean Zimmerman, editor, *Oxford Studies in Metaphysics Volume 1*, pages 155–92. Oxford University Press, 2004.

Cian Dorr. Quantifier variance and the collapse theorems. *The Monist*, 97(4):503–570, 2014. doi: 10.1093/monist/97.4.503.

Cian Dorr. To be f is to be g. *Philosophical Perspectives*, 30(1):39–134, 2016. doi: 10.1111/phpe.12079.

Kit Fine. Neutral relations. *Philosophical Review*, 109(1):1–33, 2000. doi: 10.1215/00318108-109-1-1.

Kit Fine. A guide to ground. In Fabrice Correia and Benjamin Schnieder, editors, *Metaphysical Grounding*, pages 37–80. Cambridge University Press, 2012.

Peter Fritz. Ground and grain. *Philosophy and Phenomenological Research*, forthcoming. doi: 10.1111/phpr.12822.

Peter Fritz, Harvey Lederman, and Gabriel Uzquiano. Closed structure. *Journal of Philosophical Logic*, 50(6):1249–1291, 2021. doi: 10.1007/s10992-021-09598-5.

Jeremy Goodman. Reality is not structured. *Analysis*, 77(1):43–53, 2017. doi: 10.1093/analys/anw002.

Jeremy Goodman and Harvey Lederman. Perspectivism. *Noûs*, 55(3):623–648, 2021. doi: 10.1111/nous.12322.

J. H. Harris. What's so logical about the "logical" axioms? *Studia Logica*, 41(2-3):159–171, 1982. doi: 10.1007/BF00370342.

J Roger Hindley and Jonathan P Seldin. *Lambda-calculus and Combinators, an Introduction*, volume 13. Cambridge University Press Cambridge, 2008.

Harold T. Hodes. Why ramify? *Notre Dame Journal of Formal Logic*, 56(2):379–415, 2015. doi: 10.1215/00294527-2864352.

Pauline Jacobson. Towards a variable-free semantics. *Linguistics and Philosophy*, 22(2):117–185, 1999. doi: 10.1023/A:1005464228727.

David Kaplan. A problem in possible worlds semantics. In Walter Sinnott-Armstrong, Diana Raffman, and Nicholas Asher, editors, *Modality, Morality and Belief: Essays in Honor of Ruth Barcan Marcus*, pages 41–52. Cambridge University Press, 1995.

Jeffrey King. Structured propositions and sentence structure. *Journal of Philosophical Logic*, 25(5):495–521, 1996. doi: 10.1007/BF00257383.

Jeffrey C. King. *The Nature and Structure of Content*. Oxford University Press, 2007.

Jeffrey C. King. Structured propositions. In *Stanford Encyclopedia of Philosophy*. 2008.

Saul A. Kripke. Is there a problem about substitutional quantification? In Gareth Evans and John McDowell, editors, *Truth and Meaning*, pages 324–419. Oxford University Press, 1976.

Joop Leo. Modeling relations. *Journal of Philosophical Logic*, 37(4):353–385, 2008. doi: 10.1007/s10992-007-9076-9.

Joop Leo. Modeling occurrences of objects in relations. *Review of Symbolic Logic*, 3(1):145–174, 2010. doi: 10.1017/s1755020309990347.

Christopher Menzel. *A Complete, Type-Free "Second-Order" Logic and its Philosophical Foundations*. CSLI Publications, 1986.

Robin Milner. A theory of type polymorphism in programming. *Journal of computer and system sciences*, 17(3):348–375, 1978.

Richard Montague. The proper treatment of quantification in ordinary english. In Patrick Suppes, Julius Moravcsik, and Jaakko Hintikka, editors, *Approaches to Natural Language*, pages 221–242. Dordrecht, 1973.

John Myhill. Problems arising in the formalization of intensional logic. *Logique Et Analyse*, 1 (1):78–83, 1958. doi: 10.2307/2272577.

A. N. Prior. On a family of paradoxes. *Notre Dame Journal of Formal Logic*, 2(1):16–32, 1961. doi: 10.1305/ndjfl/1093956750.

Arthur Prior. *Objects of Thought*. Clarendon Press, 1971.

Mark Richard. *Propositional Attitudes: An Essay on Thoughts and How We Ascribe Them*. Cambridge University Press, 1990.

Bertrand Russell. *Principles of Mathematics*. Routledge, 1937.

Bertrand Russell. *The Philosophy of Logical Atomism*. Routledge, 1940.

Moses Schönfinkel. Über die bausteine der mathematischen logik. *Mathematische annalen*, 92 (3):305–316, 1924.

Theodore Sider. *Writing the Book of the World*. Oxford, England: Oxford University Press, 2011.

Scott Soames. Cognitive propositions. *Philosophical Perspectives*, 27(1):479–501, 2013. doi: 10.1111/phpe.12030.

Dustin Tucker. Paradoxes and restricted quantification: A nonhierarchical approach. *Thought: A Journal of Philosophy*, 7(3):190–199, 2018. doi: 10.1002/tht3.383.

Dustin Tucker and Richmond H. Thomason. Paradoxes of intensionality. *Review of Symbolic Logic*, 4(3):394–411, 2011. doi: 10.1017/s1755020311000128.

Gabriel Uzquiano. A neglected resolution of russell's paradox of propositions. *Review of Symbolic Logic*, 8(2):328–344, 2015. doi: 10.1017/s1755020315000106.

Timothy Williamson. Converse relations. *Philosophical Review*, 94(2):249–262, 1985. doi: 10.2307/2185430.

Timothy Williamson. *Modal Logic as Metaphysics*. Oxford University Press, 2013.

Ludwig Wittgenstein. *Tractatus Logico-Philosophicus (Trans. Pears and McGuinness)*. Routledge, 1961.