A Theory of Structured Propositions

Andrew Bacon

May 22, 2020

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 λ-calculus, about what properties and relations can be built out of others. 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 special purpose language.

Section 1 begins by distinguishing our investigation of structured theories of reality from adjacent theories concerning structured representational objects. In section 2 I introduce *relational diagrams* — a precisification of common picture thinking in the metaphysics of structured propositions and relations — which are claimed by the structuralist under consideration to correspond with reality isomorphically. Section 3 compares the structured theory to the theory of propositions and relations implicit in the theories of Church and Curry, and submits that their formalisms are not suitable for representing structured theories of propositions and relations. In section 4 a new formalism is introduced, the structural calculus, for representing structured entities. Unlike the relational diagrams, distinct terms in this language can represent the same thing: some equivalences between the structural calculus, relational diagrams and a fragment of the λ-calculus are explored that clarify the situation. In section 5, I turn to the treatment of quantification and identity in this theory, and section 6 explains how the theory avoids the Russell-Myhill paradox. Some conjectures and theorems concerning the consistency of this theory are formulate, and the paper concludes in section 7.

1 Metaphysical and Representational Structuralism

Let's draw some preliminary distinctions. Firstly, I shall 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.[1] I will also distinguish between representational structuralism and metaphysical

[1] 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.

structuralism. Representational structuralism is the view that representational objects, including sentences, thoughts, modes of presentation, concepts and so on, are structured.[2] Metaphysical structuralism, by contrast, is a metaphysical view which maintains, superficially, that reality itself — propositions, properties, individuals, and so on, are built up out of smaller constituents or are simple.

Although the distinction between reality and representations seems clear cut, some care is needed, especially when attitude verbs are involved. Indeed, Frege [15] argued that in the presence of attitude verbs there is widespread equivocation between individuals and their names or their concepts. But even if you are not a Fregean, it is common to somehow involve fine-grained representational objects in the semantics and pragmatics of attitude reports. The moral, whatever your theory of attitudes, is that it is easy to conflate the two.

One way to avoid this sort of equivocation, and bare down on the metaphysical issues is to concentrate on theses that do not involve names. Generalizations formulated with first-order quantifiers and bound variables in place of names tend to have a less contentious status in these debates. Many Fregeans, for instance, posit no similar equivocation between the sense and reference of a bound variable, and so draw a distinction between quantified formulations of Leibniz's law and the corresponding instances involving proper names. The running together of representational and metaphysical questions is even more common with words like 'proposition' due to its varying uses in philosophy. One suitable framework in which to study these metaphysical questions in full generality and without ambiguity eschews words like 'proposition' and 'property' altogether, and quantifies directly into the position that predicates and sentences occupy. Just as we need no special word such as 'individual' when we avail ourselves to first-order quantification, we need no qualifications when quantifying into sentence or predicate position. We may thereby avoid the ambiguities in words like 'property' and 'proposition'.[3]

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

[2]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 [32], or King [23] — 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'.

[3]As advocated in Prior [29], Williamson [37], etc.

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.[4]

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.[5] One reason to resist the a 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.

There is also a moral to be found in this example for the metaphysical structuralist. Even when structure in reality is posited, we shouldn't expect it to perfectly reflect the structure of the language expressing it. We have just given an argument that suggests that numbers are metaphysically simple, even if they can be expressed with compound expressions like '$98 + 19$'. Conversely, the metaphysical structuralist might allow one to introduce simple expressions for metaphysically complex properties, like the word 'vixen' for the complex property of being a female fox.

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 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.[6] Subsequent choices in the formalisation of the view will fall directly out of this picture.

Lastly, relational diagrams enjoy one nice feature which more conventional written notations lack, namely: 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

[4]Albeit not the only one. For non-structuralist accounts of reality that allow for similar theoretical work, see Dorr [13] and Bacon [4].

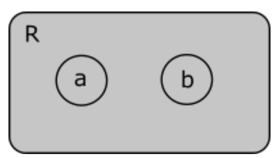
[5]Crimmins and Perry [6], Richard [30], Goodman and Lederman [17] etc.

[6]See, for instance, Dixon [10].

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.[7] Every other type has the form $\sigma \to \tau$, where σ and τ are types. σ is the type of entities that entities of type $(\sigma \to \tau)$ can combine with, and τ is the type of entity that would result from such a combination. 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.[8] We shall focus on the *relational types*: types of the form $\sigma_1 \to ... \to \sigma_n \to t$ where $\sigma_1 ... \sigma_n$ are also 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 $\sigma_1 ... \sigma_n$.[9] 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 amount of times and in a certain order within that proposition. We will depict propositions by completely 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 Rs 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. We must also be conscious of the fact that 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

[7]One should be careful with these glosses: as Prior [29] 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.

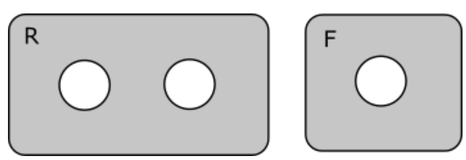
[8]One quirk of this choice is that there is an asymmetry in the way that binary relations are treated. Plugging an individual a into Rs first argument place to produce a unary property is succinctly notated by Ra, whereas there is no direct way to notate the result of plugging a in Rs second argument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

[9]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.

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.

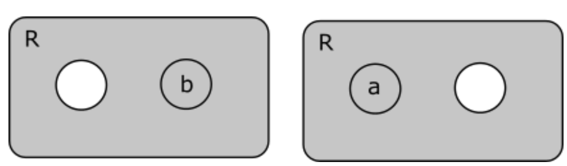
The relation R is also a constituent of this proposition, but our pictorial representation does not depict it as being before, after or between a and b. Evidently our chosen linear notation has to take a stand on this: we have adopted the prefix convention of writing R before the arguments a and b. But 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 the a relation symbol has to appear somewhere. We 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 can depict our simple binary relation R, and a simple unary property using relational diagrams as follows:



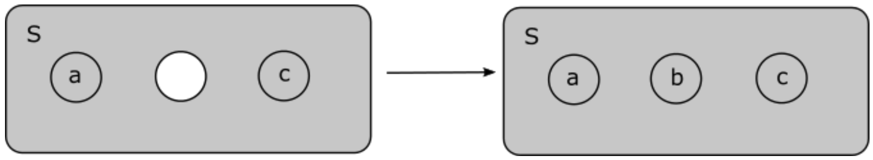
Holes, like constituents, have positions, and may be said to appear to the left or to the right of other holes and constituents. In the present example, there is a left and a right hole, and no other constituents or holes.

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.

2. If a constituent c occurs before another constituent c' in a proposition p, then every constituent of c appears before every constituent of c'.

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.*

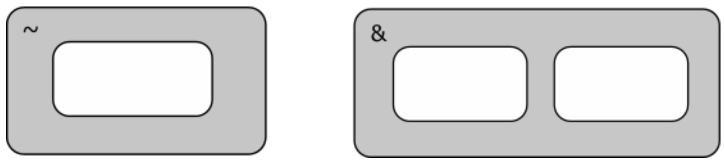
4. Either (i) for any a and b, a appears before b, but not conversely, in *a loves b*, or (ii) for any a and b, b appears before a, but not conversely, in *a loves b.*

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 a improper constituent, so that every entity counts as a constituent of itself.

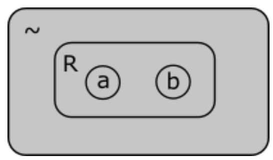
4 presents two cases. Start by assuming 4i, and thus that *Alice* appears before *Bob* in *Alice loves Bob.* By 1, the property F appears before *Alice* in the proposition that results from applying F to *Alice* which, by 3a, is the proposition that *Alice loves Bob.* By 2 every constituent of F appears 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.[10]

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:

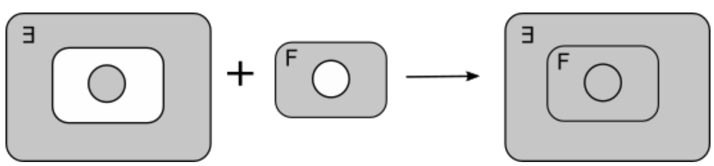


Plugging the proposition, *Rab* into negations hole, for instance, will yield:



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 be writing it explicitly into the diagram.

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 σ is what you get from taking a diagram for an entity of type σ 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:

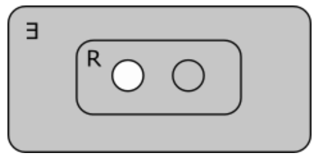


Note that the hole associated with F is not filled with a constituent, but is still greyed out. While it is a hole of the constituent property *walks*, it is not a hole in the resulting proposition, *something walks*. Evidently, *something walks* is a completely saturated proposition: it shouldn't have any holes. Informally, we can think of the greyed out hole as being 'bound' by the higher-order property.[11] 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.

[10]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.

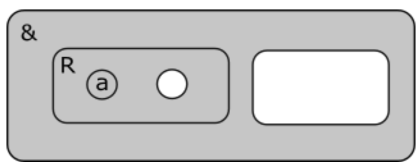
[11]Note that all higher-order properties 'bind' holes in this way; it is not special to quantifiers.

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 kth 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 kth 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 Rs 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..

The collection of diagrams representing individuals, propositions, properties and relations can thus be summarized as follows.

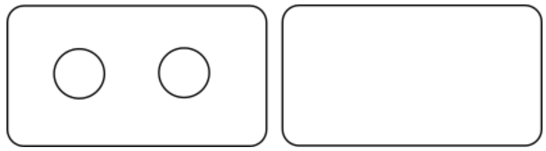
1. Each simple property, relation and individual has a relational diagram associated with it (a grey box containing holes of the relevant shapes, according to our above discussion).

2. If you have a diagram d which contains holes of type $\sigma_1 ... \sigma_k ... \sigma_n$ in that order, and another diagram d' that has holes of type $\rho_1 ... \rho_i ... \rho_j$ where $\rho_{i+1} ... \rho_j$ are the types of the holes in a relation of type σ_k, then you can plug d' into ds kth 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.

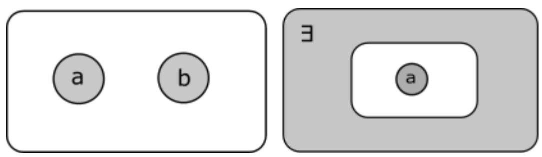
Note that the operation of simply plugging an argument into the kth hole of an relation (without introducing any new holes) is a special case of our second rule, where $i = 0$.

Relational diagrams make the structure of propositions and relations especially clear. There 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.

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. The second diagram has a main relation, but its first immediate constituent does not, and also cannot be constructed from our two rules.



Our informal descriptions of the diagrams and the operations on them could be made more precise if we wanted to. And there are still some fiddly things to check: for instance, whether our diagrams can all be consistently assigned types. But I shan't do any of that: instead I will develop a more conventional notation, comparing it, along the way, with some other less structured theories of properties and relations inspired by Church and Curry's work on type theory.

3 The theories of Church and Curry

There is a now standard formalism for representing properties and relations in a typed language. The λ-calculus, originally developed by Church, is formulated in terms of variables and a syncategorematic device, λ, for forming expressions denoting properties and relations. Less widely used, but of equal expressive power, is the combinatory calculus of Curry [7], which eschews variables altogether and achieves the effect of Church's λ by other means. Underpinning both of these formalisms, however, is a substantive metaphysics of properties and relations: a metaphysics, I shall argue, that the structuralist just described does not accept. In this section I will argue that the framework of Church and Curry are simply not suitable for theorizing about structured propositions: many of the expressions that appear in these formalisms are meaningless by the structuralists lights — they do not correspond to properties or relations. The case against the λ and combinatory calculi presented here will become especially relevant when we examine the inconsistency results of Dorr [13] and Goodman [16], which are formulated in these languages.

Church's theory is formulated in a language that contains various logical and non-logical constants. We assume that the logical constants contain at least a binary connective, $\rightarrow$ of type $t \rightarrow t \rightarrow t$, and for each type σ a quantifier $\forall_\sigma$ of type $(\sigma \rightarrow t) \rightarrow t$ which, when applied to a predicate $F : \sigma \rightarrow t$ forms a sentence $\forall_\sigma F$ of type t, roughly saying that F is universally satisfied. There is also, for each type σ, an infinity of variables of that type. Expressions of various types may be formed by following the rules in table 3. We write

$\dfrac{}{x{:}\sigma \vdash x{:}\sigma}$	**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

$x_1 : \sigma_1, ..., x_n : \sigma_n \vdash M : \tau$ to mean that M is term of type τ that can be constructed from free variables $x_1...x_n$ of types $\sigma_1...\sigma_n$.[12]

We use Γ and Δ to stand for arbitrary sequences of type assignments, $x_1 : \sigma_1, ..., x_n : \sigma_n$. The rule Application is understood so as to only apply when Γ and Δ have no variables in common. The system is best illustrated with a couple of examples. I will describe how to type the so-called C, W and K combinators, which will play a special role in the following discussion

- Given two instances of Identity — $X : e \to e \to t \vdash X : e \to e \to t$ and $y : e \vdash y : e$ — we can apply Application to two instances to get $X : e \to e \to t, y : e \vdash Xy : e \to t$. Using Application to combine this with another instance of Identity, $z : e \vdash z : e$ yields $X : e \to e \to t, y : e, z : e \vdash Xyz : t$.

- The C combinator: From 1 we can apply Exchange to get yields $X : e \to e \to t, z : e, y : e \vdash Xyz : t$, and three applications of Abstraction gives us the C combinator $\vdash \lambda X \lambda z \lambda y.Xyz : (e \to e \to t) \to (e \to e \to t)$ (for any types σ, τ and ρ there is a corresponding C combinator of type $(\sigma \to \tau \to \rho) \to (\tau \to \sigma \to \rho)$.)

- The W combinator: Alternatively, from 1 we could apply Contraction to yield $X : e \to e \to t, w : e \vdash Xww : t$, and Abstraction twice gives the W combinator: $\vdash \lambda X \lambda w.Xww : (e \to e \to t) \to e \to t$

- The K combinator: By applying Weakening to an instance of Identity, we get $x : e, p : t \vdash p : t$, and by Abstraction twice we get $\vdash \lambda p \lambda x.p : t \to e \to t$.

We will also appeal to the following terms in what follows. Given a constant $R : e \to e \to t$ one can formulate the terms $\lambda z \lambda y.Ryz$, $\lambda w.Rww$ and given $a : e, b : e$, we can also make $\lambda x.Rab$. They closely follow the derivations for C, W and K respectively, except that initial uses of Identity for X, are replaced with an instance of Constrants involving R.

[12] This system for typing terms originated with Curry in the 1930s, and following Milner [26] 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 λ-abstract, and (ii) an unannotated λ-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 λ-terms.

I will 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 we suppress λs appearing immediately after quantifiers, writing $\forall_\sigma x.A$ instead of $\forall_\sigma \lambda x.A$. The remaing 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 last rule Abstraction is crucial to Church's system, and is what allows one to make complex predications. In first-order logic we typically use open formulae to represent complex predicates. A unary predicate can be represented by an open formula in one variable, like $Fx \wedge Gx$, and the result of 'predicating' an open sentence to an argument a, simply involves substituting x with a: in this case $Fa \wedge Ga$. But an open sentence involving a free variable x of type e, like $Fx \wedge Gx$, has type t, and so in accordance with the rule Application, cannot be applied to a name. λ is a device that allows us to turn the open sentence like $Fx \wedge Gx$ into a closed predicate $\lambda x.(Fx \wedge Gx)$; here we say that x in $Fx \wedge Gx$ is *bound* by λx. We may pronounce this predicate in English as 'is an x such that x is F and x is G'. The idea here is that the predicate $\lambda x.(Fx \wedge Gx)$ behaves in the way that you would expect, given the way that open formulae are used to express predicates and relations in first order logic: that applying a λ-term to an argument should be the same as substituting the argument into the body of the term (see the principal β discussed below).

Within this language one can state various theories of properties and relations. 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:

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

The above axioms say very little about Church's special device for denoting properties and relations: λ. Further axioms governing λ, then, must be added that pin down the intended meaning of λ. A pair of terms of the form $(\lambda x.M)N$ and $M[N/x]$ are said to be *immediately β-equivalent*, with the caveat that N is free for x in M.[13] Two terms of the form M and $\lambda x.Mx$ are immediately η equivalent, when x isn't free in M. A minimal theory, which I shall simply call H, adds the further principles β andη to the above axioms.[14].

β $A \leftrightarrow A'$ 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.

η $A \leftrightarrow A'$ 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.

In other words, immediate β and η-equivalents are intersubstitutable *salve veritate*.

[13] 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 [19].

[14] Note that these principles license the substitution of immediate η and β-equivalents even in the scope of λs binding variables appearing free in the terms being substituted.

A nice feature of β and η is that they pin down the meaning of the λ-terms uniquely.[15] In fact, given β and η, one can derive the identity of any other term that satisfies β with the corresponding λ-term. Suppose $\lambda x.M$ is a λ-term of type $\sigma \to \tau$ and that $F : \sigma \to \tau$ is another expression which satisfies the corresponding instance of β: FN may also be substituted for $M[N/x]$ in any sentence *salve veritate*. In particular, when $N = x$, $M[x/x] = M$, so β tells us that Fx and M are intersubstitutable. Thus we can infer from $\lambda x.M = \lambda x.M$ that:

$$\lambda x.Fx = \lambda x.M$$

Given η we can substitute F for $\lambda x.Fx$ in the above giving us:

$$F = \lambda x.M$$

β thus pins down the meaning of λ-terms entirely, given the background theory H.

From the self-identity statements $(\lambda x.M)N = (\lambda x.M)N$ and $\lambda x.Mx = \lambda x.Mx$ you can derive from β and η 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.

The second consequence, $\eta^{=}$, simply identifies properties like *is wise* and *is an x such that x is wise*, where the λ abstract doesn't seem necessary. The truth status of this principle won't matter much for our discussion.[16] The first principal, β, ensures that λ terms really do fulfill the job description of making complex predicates from open sentences: that, for instance, the result of applying $\lambda x.(Fx \wedge Gx)$ to an argument a is the same as substituting a into the open sentence $Fx \wedge Gx$. That is to say, $\lambda x.(Fx \wedge Gx)a = Fa \wedge Ga$ — or in words, that *Socrates is an x such that x is wise and x is old* is the same proposition as *Socrates is wise and Socrates is old.*

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 expressions built from denoting constants, bound variables and λ — 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 H. E.g. one can formalise and prove that for every binary relation, R, there is another unary property of Ring 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 a Rs itself is the proposition a Rs 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 longstanding challenge for classical theories of structured propositions:

> What is the structure of propositions expressed by sentences involving bound variables?

[15] My argument here is inspired by similar arguments found in Harris [18], that the axioms for the quantifiers and connective pin their meanings uniquely. See also Dorr [12] for more discussion.

[16] We will later assuming it, when providing a representation of the structured theory using λs.

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.[17]

For the representational structuralist, this is a reasonable 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.

Let us briefly recap why bound variables are necessary. Variables were originally introduced by Frege in the Begriffsschrift — his version of higher-order logic — where they were exclusively bound by quantifiers. Modern treatment of quantifiers, such as those in the type theories of Church and Curry, can actually be treated in a variable free manner: $\exists_e$ is an expression of type $(e \to t) \to t$ that can combine directly with a predicate to form a variable free sentence $\exists_e F$. If we left it at that, then there would be no proliferation of properties like the ones posited in 1 and 2. But without some other device, such as λ-abstraction, there simply aren't enough predicates to form all the quantificational claims one wants to express. We can't quantify into embedded contexts as I might want to if I wanted to say 'something is both F and G' — something Frege could have formulated with $\exists x.(Fx \wedge Gx)$ — unless there is a predicate corresponding to the open formula $Fx \wedge Gx$ with which the existential quantifier can combine. Turning an open sentence into a predicate is exactly what λ allow us to do. In Frege's system quantifiers are variable binders, whereas in the Church system, all the variable binding jobs are delegated to λ. Either way, we use bound variables.

Bound variables thus solve a technical problem: we introduced them so that we have enough representations around to express all the things we want to be able to express. The structure of reality itself, however, is completely independent of our expressive needs. Some ways of generating enough representations of things we might need to express, generated more than we need: the distinguishing features of the proliferated representations are merely *representational artefacts*, that do not reflect anything in reality. Indeed, the difference in the two representations above 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 [8] 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 feeling that bound variables are a representational artefact, the radical structured view is inconsistent with Church's principal β. 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 β they are both identical to $Fa \wedge Ga$.

[17] This is clear in, for instance, King [23], although note he repudiates this aspect of the view in [24], p41.

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 the bound variables can pick out the same property or relation. More generally, we should simply reject the view that bound variables contribute constituents to the properties they help denote:

Moderate Structuralism An expression built from a closed expression M using only bound variables and λ denotes an entity with the same constituents (in kind and number) 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 constituents. (The parenthetical about number is there to make sure that we can register the difference between the constituents in things like $Rabb$ and $Raab$. If we wanted to be more pedantic, we could say that two entities have the same constituents only when the *multiset* of their constituents is the same.)

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

Constituent Count MN has the same constituents (in kind and number) as M and N have together.

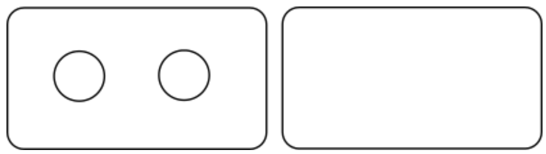
By this principle, for instance, the constituents in $(\lambda xy.Ryx)ab$ are the union of constituents of $\lambda xy.Ryx$, a and b. By β, $(\lambda xy.Ryx)ab = Rba$, whose constituents are the sum of the constituents of R, b and a (again, by Constituent Count). The only way these constituents can be the same is if $\lambda xy.Ryx$ has the same constituents 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 entities that do not have any constituents at all, and (ii) entities that have no well-defined notion of constituent count and order. (i) and (ii) also mark two ways in which Church's framework is an ill-suited for the job of formulating a theory of structured propositions.

Let's start with (i). Church's theory allows one to construct terms that are *entirely* constructed from bound variables and λ. Terms like this are called *combinators*, and examples include $\lambda p.p$, and the C, W and K combinators constructed above ($\lambda X\lambda y\lambda z.Xzy$, $\lambda X\lambda y.Xyy$, $\lambda xy.x$). 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 λ must not contain any constituents. We can verify this more formally by appealing to β and Constituent Count. For instance, given β, $(\lambda p.p)$ maps every proposition to itself, and so $\lambda p.p$ cannot add any constituents to the proposition it is applied to. $\lambda p.p$ must therefore have no constituents. Similarly, Constituent Count tells us that $(\lambda X\lambda y\lambda z.Xzy)Rab$ has the constituents of $(\lambda X\lambda y\lambda z.Xzy)$, R, a and b, and also tell us that Rba has the constituents of just R, a and b. By β, $(\lambda X\lambda y\lambda z.Xzy)Rab = Rba$, and so $\lambda X\lambda y\lambda z.Xzy$ must contribute no constituents, in order for the constituents to add up.

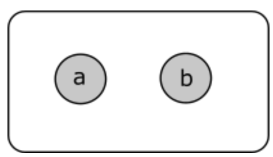
Now I think there is an important choicepoint in a theory of properties and relations that trades in notions like 'constituenthood' over whether it should admit completely constituentless properties and relations or not. Indeed, the thesis that there are constituentless entities is entirely consistent, and there is a lot of unexplored territory to be mapped out here.[18] But I take it to be pretty uncontentious that constituentless properties and relations are not part of anything resembling the orthodox theory of structured propositions.

Our diagrammatic representation 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 λs and variables, every way of building a combinator will involve at least the rules Identity and Abstraction — the only rules that introduce variables and λs 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.

Another way in which this theory of properties and relations conflicts with a structured one is that it seems to render our notion of constituent order ill-defined. For any binary relation, R, there is another relation $\lambda x \lambda y.Ryx$ — the *converse* of R, which we shall simply calle R^c. By β it is easily seen that $Rab = R^c ba$.[19] Of course, given Moderate Structuralism, R and R^c have exactly the same constituents. There is something already mereologically puzzling about this situation, since if R is simple, then it seems as though we have two distinct relations, R and R^c, that have exactly one constituent each, and that constituent is the same.[20]

[18]My own attempt at this can be found in Bacon [3]; see especially the discussion of *pure* entities. See also the related notion of *logicality* in Dorr [13].

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

[20]Perhaps $\lambda xy.Ryx$ has a R as proper constituent, while R does not. But this is still mereologically puzzling, for one might expect that something can have proper constituent only if it has some other constituent disjoint from the first (see the mereological principal of Strong Supplementation for analogy).

The existence of converse relations problematizes the notion of constituent order, introduced in section 2. Recall that operation of application, which we write (Ra) when $R : e \to e \to t$ and $a : e$, corresponds in our relational diagrams to inserting as diagram into the leftmost hole of R. Similar points would apply to the relation R^c, if it had a diagram. Thus, if we plug a into the leftmost hole of R and b into the right we get Rab, and a occurs before b in the resulting proposition. Similarly if we plug b into the left hole of R^c and a into the right hole we get $R^c ba$, and b occurs before a in this proposition. But by β, Rab and $R^c ba$ are the very same proposition, yielding the result that a appears before b and after it in a single proposition. A structured theorist should thus deny the existence of a distinct relation $\lambda xy.Ryx$ (and thus Church's notation, in which its existence is implicitly accepted).

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 [36], Fine [14], Dorr [11]). We may still, however, acknowledge the fact that transitive verbs in English can appear in both the active and passive voice: for instance *the boy kicked the ball* and *the ball was kicked by the boy*. The mere existence of this grammatical distinction is not decisive reason to think there is a corresponding distinction in reality. As argued in Williamson [36], 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 denote the same relation, whilst being subject to special grammatical rules about which order the arguments must appear in the surface structure of a sentence.

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.

Another feature of Church's notation is that a variable in M that is bound by a λ may occur in M multiple times or not at all, as permitted by the rules of Contraction and Weakening respectively. Where Exchange previously created trouble for the notion of constituent order, both of these rules seem to cause trouble for the notion of constituent *count*. For instance, using Weakening, one can form a unary property from the proposition Rab by *vacuous* λ-abstraction: $\lambda x.Rab$.[21] As above, our diagrams do not permit us to represent such a property: it is a unary property so it must have a hole somewhere. But there is nowhere for the hole to be, given that Rs only two holes are already occupied by a and b.

The property is also problematic for a structured theorist. For if we plug a constituent c into the only hole in $\lambda x.Rab$ — assuming it were to somehow have a hole — to form the proposition $(\lambda x.Rab)c$, we should obtain something that has the constituents of R, a, b and c (by Moderate Structuralism, and Constituent Count). Yet by β this proposition is

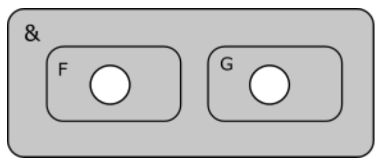
[21] Given a derivation $\vdash Rab : t$ one can apply Weakening to obtain $x : e \vdash Rab : t$, and then abstraction to obtain $\vdash \lambda x.Rab : e \to t$.

identical to Rab, which given Constituent Count does not contain c among its constituents. In short, vacuous λ-abstraction allows one to 'forget' constituents.

The other case of a bound variable occurring twice is exemplified by the term $\lambda x.Rxx$, denoting the unary property of Ring oneself. This is licensed by the rule Contraction.[22] By Moderate Structuralism this denotes a unary property that has only R as a constituent, assuming R is simple (and so, like $\lambda xy.Ryx$, is mereologically peculiar). If you plug a simple constituent, a, into the only hole in $\lambda x.Rxx$ you should, by Constituent Count, have something that has a once and R. But by β, $(\lambda x.Rxx)a = Raa$, which by Constituent Count has the constituent a appearing twice.

Our notation also prevents us from creating such properties: it would effectively require us to be able to take a relational diagram with two holes, like our diagram for R above, and merge the two holes into one.

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 λ 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 λ and bound variables in our representations of properties and relations. Curry [7] 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 σ to form (MN) of type τ. To do the jobs that λ normally deals with, Curry instead takes as primitive expressions that correspond to the combinators in Church's theory. In Church's theory, recall, a combinator is a closed term involving only bound variables and λ. 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 σ, τ and ρ one introduces a primitive combinator constants I^{σ}, $K^{\sigma\tau}$ and $S^{\sigma\tau\rho}$, denoting respectively the operations $\lambda x.x$, $\lambda xy.x$ and $\lambda XYz.Xz(Yz)$ denotes in Church's theory where $x, z : \sigma, y : \tau$, $X : \sigma \to \tau \to \rho$, and $Y : \sigma \to \tau$.

The minimal theory of properties can be reformulated in Curry's notation. One has the logical principles, PC, UI, MP and Gen.[23] In place of β and η one has instead:

[22]From a derivation of $x : e, y : e \vdash Rxy : t$ one can get $z : e \vdash Rzz : t$ by Contraction, and $\vdash \lambda z.Rzz : e \to t$ by Abstraction.

[23]Even though Curry's formalism contains no bound variables, in formulating logical systems in which to reason, it is technically convenient to augment Curry's language with *free* variables. These play a useful role when reasoning about arbitrary objects; see the principle Gen.

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

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

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

Notice that these identities would be instances of β in Church's theory if $S^{\sigma\tau\rho}$ and $K^{\sigma\tau}$ were replaced with the corresponding defined combinators in Church's theory.[24]

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.[25] In order to satisfy the identities **S**, **K** and **I**, combinators must be constituentless for the same reason that they must be in Church's theory. And and the problematic expressions $\lambda xy.Ryx$, $\lambda x.Rab$ and $\lambda x.Rxx$ have correlates in Curry's system

$$\lambda x.Rab \equiv K^{et}(Rab)$$

$$\lambda x.Rxx \equiv S^{eet}RI^{e}$$

$$\lambda xy.Ryx \equiv S(K(SR))K$$

(I have omitted the relevant type superscripts from the final paraphrase because they are long and not necessary for my point.) Given the identities **S**,**K** and **I**, we can derive correlates of the identities $(\lambda xy.Ryx)ab = Rba$, $(\lambda x.Rxx)a = Raa$ and $(\lambda x.Rab)c = Rab$ that appeared in our earlier discussion.[26]

How should the structured proposition theorist respond, once it is realised that many of the expressions in the language of Church and Curry are not meaningful according to their theory? One option is to stop using them altogether. Another is to keep reasoning with the problematic expressions, but simply weaken the axioms governing them (i.e. β or the axioms S, K and I), so that they no longer implied problematic identities, like $(\lambda xy.Ryx)ab = Rba$.

I think latter option is a bad idea. It was once believed that there was an intra-mercurial planet, *Vulcan*. When it was discovered that no such planet existed, astronomers stopped using the word 'Vulcan'. I submit that it would have been a bad idea to have continued to use 'Vulcan' as though it still referred to *something*, but weaken the astronomical theory so that it no longer implied that Vulcan was an intra-mercurial planet. If the weakened theory leaves it open whether 'Vulcan' now refers to Leonard Nimoy, or some other existing individual, one has the right to complain that we don't have a clear idea what or who we are now theorizing about when we make assertions using the word 'Vulcan'. I think that it would be similarly bizarre to respond to the above by keeping the usual λ or combinatory notation, but weakening the principles governing them, like β, or S, K and I. For what would such a weakening even be saying? What properties and relations would these weakened principles even concern, once we have rejected the existence of the entities the relevant λ and combinatory terms purported to denote? The λ notation was an innovation of Church,

---

[24] The principles **S**, **K** and **I** do not uniquely pin down the interpretation of the S, K and I combinators, in the way that β does for λ-terms. However a slight strengthening of these principles, comprising the *extensional* combinatory logic, does. This can be obtained by introducing **S**, **K** and **I** into an equational theory (the principles ν, μ, σ, τ and ι from table 4), and adding the extensionality axiom ζ (see table 4 below). For instance, if two expressions S and S' satisfy **S**, then one can show that $SXYz = Xz(Yz) = S'XYz$. Applying ζ thrice allows one to derive $S = S'$. Equivalently, one can replace ζ with some further equational axioms; see Hindley and Seldin [19] p84 for more details.

[25] The equivalence holds for the extensional combinatory calculus described in footnote 24.

[26] 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.

and is legitimate if one has accepted the underlying metaphysical assumptions. But it was not like we had an independent grip on the notion before Church, and he came along and just gave a name to it: Church's λ-terms don't obviously have any meaning outside of that theory.

The argument against structured propositions found in Dorr [13] and Goodman [16] is sensitive to the sorts of objections a structuralist might have against principles like β, S, K and I. Their arguments are instead targeted at a structuralist who has done exactly what we have cautioned against: retained the meaningless expressions but weakened β. We'll have more to say about this argument shortly. But for the moment, it will be instructive to examine these weakened principles to see to what extent they pin the meanings of λ-terms down. The idea is to replace the identity in $\beta^=$ with a biconditional, in the cases where such a replacement makes sense (i.e. the things flanking the identity have type t). We can also provide similar weakenings of S, K and I:

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

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 $\forall x_1...x_n(I^{\sigma}Rx_1...x_n \leftrightarrow Rx_1...x_n$

Let's start with extensional β. There is an apparent *practical* reason to keep λ-terms in the language, for we needed λ-terms in order to express complex quantificational claims. However, without any constraints on the interpretation of the λ-term $\lambda x.A$, it is far from clear a corresponding quantified claim, $\exists_\sigma \lambda x.A$, bears any relation to the claim we wanted to express. Extensional β apparently helps with this expressive challenge. If you have an open formula $A(x)$, expressing a collection of propositions parameterized by the value of x, extensional β guarantees that there is a property that has the same *truth value*, for each value of x, as the proposition expressed by $A(x)$. And thus one can always create a proposition that has the same *truth value* as 'for some x, $A(x)$'. But this hardly the same as solving the expressive challenge, since one can meet this demand simply by postulating a true and a false proposition.

Underlying this issue is the problem that extensional β really does nothing to pin down the role of the λ-terms. To illustrate, extensional β is consistent with a λ-term denoting some arbitrary relation that just happens to be extensionally correct. Suppose everyone loves people who hate them back and hate people who love them back: then for all extensional β tells us, $\lambda xy.Ryx$ might denote the hating relation, even when R denotes the loving relation.[27] The situation here is only slightly better than the theory that tells you that Vulcan is something or other, but doesn't tell you what.

Of course, one might attempt to posit a more intimate relationship between the relation expressed by R and the relation expressed by $\lambda xy.Ryx$, but it is hard to say what that relationship is without reinstating the above worries. If one doesn't demand a more intimate relationship, then it is natural to wonder why we should bother with the special notation? A more perspicuous alternative would be to just ban certain λ terms — the problematic ones identified earlier — which have the pretense of being intimately related to the denotations of their linguistic constituents, and adopt some other solution to the problem of quantification.

[27]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*.

The corresponding weakening of Curry's theory is also consistent with a highly structured picture. For instance it is consistent to suppose that the combinators I^{σ}, $K^{\sigma\tau}$ and $S^{\sigma\tau\rho}$ are metaphysically simple (as opposed to being constituentless). This is a view in which, while both reality and language are structured, the structure of reality does not reflect the structure of language in any straightforward sense. What in language appears to be a fairly simple statement, like $\forall_e x \exists_e y.Ryx$, which you might expect to denote a proposition that only contain three constituents corresponding to the two quantifiers and R, becomes much more complex when translated into combinators. On one flatfooted translation it becomes (again, omitting type superscripts from the combinators for readability):[28]

$$\forall_e(S(K\exists_e)S(K(SR))K)$$

which denotes something containing nine constituents using the same method of counting. Other translations of λ-terms into combinators may produce different constituent counts, all of them delivering more than three constituents. (This multiplicity of translations is in itself puzzling, and just further highlights how the structure of reality and of language have become uncoordinated.)

More serious, however, is that it is not clear that these weakened principles will always do the jobs required of them. One ought to be able to infer from *Alex believes that it's not the case that snow is white*, represented as $B(\neg A)$ in the λ-calculus and combinatory logic alike, that *for some X, Alex believes that it's X that snow is white*, which can be represented in the λ-calculus by the formula $\exists_{t\to t}X.B(XA)$, or $\exists_{t\to t}\lambda X.B(XA)$ leaving the λ unsuppressed for clarity. That is, ought to be able to make the following inference:

$$B(\neg A) \vdash \exists_{t\to t}\lambda X.B(XA)$$

Extensional β licenses this inference: $B(\neg A)$ entails by extensional β $(\lambda X.B(XA))\neg$. From this one can infer $\exists_{t\to t}\lambda X.B(XA)$ by an instance of existential generalization: $Fa \to \exists_{\sigma}F$, where a is $\neg$ and F is $\lambda X.B(XA)$. The problem is that without something as strong as extensional β we don't have a good translation of *for some X, Alex believes that it's X that snow is white* into combinatory logic. This is what we get if we apply the standard translation of the above inference into combinatory logic, described in footnote 28:

$$B(\neg A) \vdash \exists_{t\to t}(S(KB)(SI(KA)))$$

(Actually, other translations of $\exists_{t\to t}\lambda X.B(XA)$) are possible, but it will not affect the following discussion). This translation is not adequate, since on a structured picture the premise leaves no guarantee that the conclusion is true. For every operator X, $S(KB)(SI(KA))X$ is equivalent to $(KB)X(SI(KA)X)$ by extensional **S**, which, in turn is equivalent to $B(SI(KA)X)$ by extensional **K**. So the only way the conclusion could be true, given Extensional **S** and **K**, is if there is some operator X such that Alex believes the logically complex claim $(SI(KA)X)$. This could well be false, even when Alex believes that snow is not white, so the inference above is not valid. Note that when identity strength **S**, **K** and **I** are assumed, the standard translation delivers the right results, but at the cost of giving up the structured view of propositions: an apparently highly structured proposition, like $(SI(KA)\neg)$ will be identical to a less structured proposition like $\neg A$, which Alex does believe.

[28] The translation from λ-terms to combinatory terms may be described as follows: $M^c = M$ when M is a variable or a constant, $(MN)^c = M^cN^c$, and $(\lambda x.M)^c = [x].M^c$, where $[x]$ is an operation on combinatory terms defined as follows: $[x]y = K^{\sigma\tau}y$, $[x]c = K^{\sigma\tau}$, $[x]x = I^{\sigma}$, $[x].(MN) = S^{\sigma\tau\rho}([x]M)([x]N)$, $x:\sigma, y:\tau$ and $M:\tau\to\rho, N:\tau$, and c is a constant of type τ.

4 The Structural Calculus

Let's return to our structuralist metaphors: there are some basic building blocks — fundamental 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 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 λs (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 β and η in full generality.

Let's begin with the aspect of the Church and Curry notation that is entirely unproblematic: application. For instance, given a binary relation $R : e \to e \to t$ and $a : e$ you can apply R to a to form the unary property $Ra : e \to t$. Metaphysically this is a perfectly well defined operation, and just corresponds to filling the left hole in the relation denoted by R with the individual denoted by a. But clearly there is another perfectly good way of gluing R and a together, namely plugging a in to Rs right hole. Since application corresponds to filling the leftmost hole of a relation with an argument, this is not something one can define from the ordinary notion of application alone. There is, of course, a λ expression we could employ to denote this property, namely $\lambda x.Rxa$. But instead of adopting the λ-notation in a piecemeal way, we will develop our own notation. (A precise characterization of the λ expressions that are legitimate by this theories lights will be provided shortly.) 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 Rs 0th slot, and $(Ra)^0_1$ for the result of plugging a into Rs 1st slot, $(Ra)^0_2$ for the 2nd slot, and so on. (For the moment, the reader should ignore the superscripts.) More generally, we have:

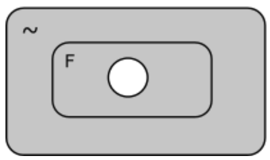
> Given a term $M : \sigma_1 \to ... \to \sigma_k \to \tau$, $1 \leq m \leq k$ and $N : \sigma_m$, $(MN)^0_m$ is a term of type $\sigma_1 \to ... \to \sigma_{m-1} \to \sigma_{m+1} \to ... \to \sigma_k \to \tau$, which denotes the result of plugging N into the mth hole of M. In the λ notation, it corresponds to the expression.[29]
>
> $$(MN)^0_m \qquad \equiv \qquad \lambda x_1..x_{m-1}x_{m+1}..x_k.Mx_1...x_{m-1}Nx_{m+1}..x_k$$

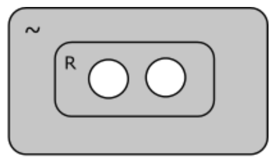
There is also another way one can glue two entities together to form a complex entity: composition. Suppose I have a pair of operators, such as *it's not the case that* and *it's necessarily the case that*, represented by terms $\neg : t \to t$ and $\Box : t \to t$ respectively. Then,

[29] Note that, assuming η, the λ-term corresponding to $(MN)^0_m$ can written simply $\lambda x_1..x_{m-1}.Mx_1...x_{m-1}N$.

as in our diagram [REF], these operators can be glued together to form a complex operator, which we shall represent with the term $(\neg\Box)^1_0 : t \rightarrow t$, meaning *it's not necessarily the case that*, that has both negation and necessity as constituents. Similarly, suppose you have a predicate modifier $Q : (e \rightarrow t) \rightarrow (e \rightarrow t)$, meaning *quickly*, and a predicate T, meaning *talks*, you can form a compound predicate *quickly walks*, $(QT)^1_0$ that denotes a composite property with two constituents denoted by Q and T respectively. Both examples have a similar form: given a term $N : \sigma \rightarrow \tau$ and $M : \tau \rightarrow \rho$, there is another term $(MN)^1_0 : \sigma \rightarrow \rho$.

Unary composition is actually one of a wider range of operations. Just as I can compose a predicate $F : e \rightarrow t$ with negation $\neg : t \rightarrow t$ to form the complex predicate $(\neg F)^1_0 : e \rightarrow t$ (i.e. *is not* F), depicted by the first relational diagram below, one can do the same things with relations of arity ≥ 2. If $R : e \rightarrow e \rightarrow t$ represents *loves*, and $\neg : t \rightarrow t$ is the negation operator, then we represent the binary relation *doesn't love* with the expression $(\neg R)^2_0 : e \rightarrow e \rightarrow t$, depicted in the second relational diagram below:



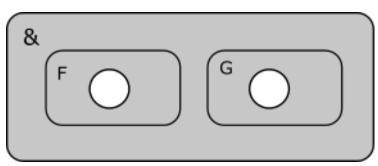


Thus in general we have

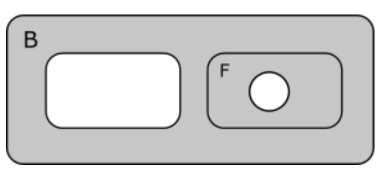
If $M : \sigma \rightarrow \tau$ and $N : \rho_1 \rightarrow ... \rightarrow \rho_n \rightarrow \sigma$ then $(MN)^n_0 : \rho_1 \rightarrow ... \rightarrow \rho_n \rightarrow \tau$ is the result of composing M and N. It corresponds to the λ expression

$$(MN)^n_0 \qquad \equiv \qquad \lambda x_1...x_n.M(Nx_1...x_n)$$

$(MN)^n_0$ corresponds, Unfortunately, these two operations are not sufficient. There are also many other ways to glue entities together in this structural picture. Given conjunction, and two unary predicates F and G, I can make the binary relation of being and x and y such that x is F and y is G $(\lambda xy. \wedge (Fx)(Gy))$.



or given an operation $B : e \rightarrow t \rightarrow t$, such as *believes*, and a predicate $F : e \rightarrow t$, meaning walks, one can form the relation $\lambda xy.Bx(Fy)$ (i.e. x *believes* y *walks*):



The operations of plugging an argument into a hole, and of composing two functional entities are special cases of a more general operation, which I'll call generalized application:[30]

[30]See [ANONYMIZED] for more details.

Generalized Application Given $M : \sigma_1 \to ... \to \sigma_{m+1} \to \tau$ and $N : \rho_1 \to ...\rho\rho_n \to \sigma_{m+1}$ 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 mth hole in M, leaving all but the first n holes in Ns unfilled (see the second rule for constructing relational diagrams). In the λ 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. Our two miscellaneous examples can now be represented as follows: $(\wedge F)^1_0 G)^1_1$ and $(BF)^1_1$. It's a useful exercise to work out why these definition suffice. 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 $(M_0M_1)^0_0M_2)^0_0...M_k)^0_0$.

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

- $(c)^\lambda = c$ when c is a constant.
- $((MN)^n_m)^\lambda = \lambda x_1...x_m y_1...y_n.M^\lambda x_1...x_m(N^\lambda y_1...y_n)$

Just by observing the form of this second case, it's also easy to see that the λ-terms in the image of this translation have none of the features of the λ-terms we discussed above: (i) variables always appear in the body of a term in the same order that they are abstracted, (ii) every variable that is abstracted appears exactly once in the body, and (iii) no variables appear in a predicating position.[31] 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.

In fact, there is a fairly simple way to revise our rules for constructing λ-terms in such a way that we can only construct the 'good' λ-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.[32] Simply dropping Identity would prevent one from ever introducing variables, even in argument position, and consequently prevent λs not only from binding variables in predicating position, but also in argument position. The rule Concretion, from table 4, 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 4.

(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.)

[31] In particular, in the term $\lambda x_1...x_n y_1...y_k.Mx_1...x_n(Ny_1...y_k)$, the order of the free variables in the body of $Mx_1...x_n(Ny_1...y_k)$ is the same as in the λ abstract, $\lambda x_1...x_n y_1...y_k$, and every variable in the string of variables abstracted appears at least and at most once in the body.

[32] 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 $\vdash \lambda X.Xa : (e \to t) \to t$.

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

Table 2: Natural deduction for typing the structural λ-terms

It is now possible to provide a reverse translation from λ-terms to our novel language. 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 λ-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. 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: $(\vdash c : \sigma)^s = c$
- 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 Γ and Δ 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.[33] Last, but not least, we can associate expressions of the structural calculus with diagrams: an n-ary relational constant R corresponds to a simple relational diagram with n holes, as in rule 1, an individual to a filled circle, and $(MN)^n_m$ corresponds to our rule 2 for creating complex diagrams from simple ones.

We have observed already that there is not a one-to-one correlation between λ-terms and the properties and relations they denote. Simply relabeling bound variables will produce different representations for the same property. But also the equations β and η, which can be accepted once we have rid ourselves of the problematic λ-terms, will generate further multiplicity in way we represent reality: for instance, β 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: $((\neg\Box)^1_0 A)^0_0$. 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:

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

[33] 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 table 3, it's actually impossible to find two derivations of the same sequent in the pared down system.

$\overline{(MN)^n_m a_1...a_m b_1...b_n = M a_1...a_m (N b_1...b_n)}$	γ	$\overline{M=M}$	ι
$\dfrac{M=N}{N=M}$	σ	$\dfrac{M=N \qquad N=P}{M=P}$	τ
$\dfrac{M=M'}{MN=M'N}$	μ	$\dfrac{N=N'}{MN=MN'}$	ν
$\dfrac{Mx=Nx}{M=N}$	ζ		

Table 3: The theory of structural equations with deductive variables

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 previous section.

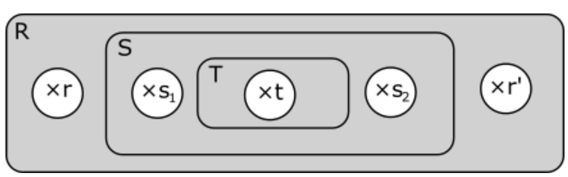
It would be nice to have something general to say about equations like this one. Let us say that a *structural equation* is an equation between terms in the structural calculus that have identical diagrams. Ideally, we would like a general set of rules for generating equations between terms of the structural language — an *equational theory* — which implies all and only the structural equations. Something analogous to the principles β and η for the λ calculus, or **S**, **K** and **I** for combinatory logic.

One axiomatization of these identities is offered in figure 4. There is a subtlety here, namely that the rule ζ 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 μ and ν with generalized applications (e.g. μ becomes $(MN)^n_m = (M'N)^n_m$) and replacing the rules γ and ζ with the following pair of closed equations:

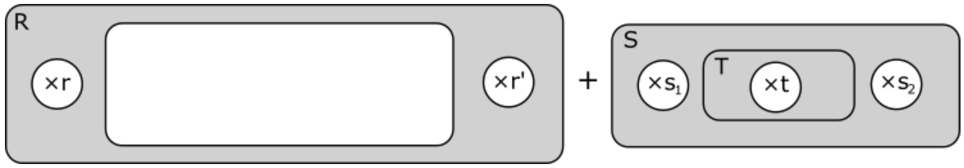
Recompose $(R(ST)^t_{s_1})^{s_1+t+s_2}_r = ((RS)^{s_1+1+s_2}_r T)^t_{r+s_1}$

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

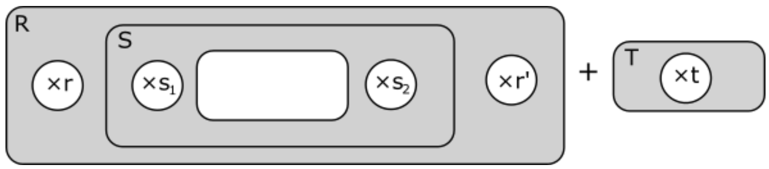
The easiest way to see why these equations are true is to draw the corresponding relational diagrams. For simplicity we depict the case where R, S and T are themselves unstructured. The first equation gives us two representations of the following structured relation:



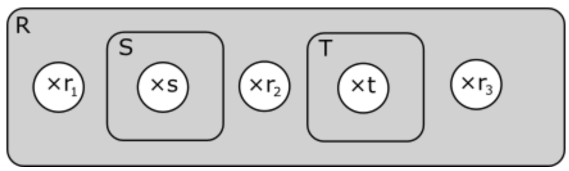
We can construct it firstly by plugging T into S in after the s_1th hole to make $(ST)^t_{s_1}$ (see the diagram on the right below), and the plugging the result into the rth hole of R (on the left).



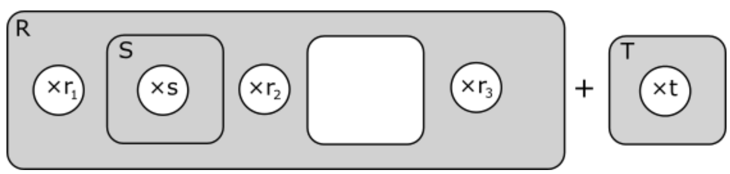
Alternatively, we could plug S after the rth hole of R (left), and T after the $r+s_1$th hole (right):



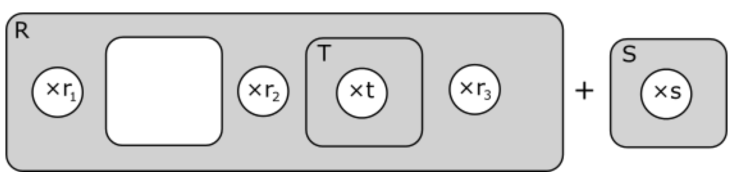
Rearrange concerns the following sort of structured relation:



You can make it by plugging in S after the r_1th hole in R (left) and then plugging T (right) into the r_1+s+r_2th hole of the result:



remember that the r_1+1+r_2th hole in R is the same as the r_1+s+r_2th hole, once S has be inserted in R. Thus te could achieve the same result by first plugging T immediately after the r_1+1+r_2th hole in R (left) and then plugging S after the r_1th hole.



That these equations suffice to prove all the structural equations, however, is a much more subtle matter.

Theorem 1. ?? *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. [19] chapters 6 and 7).*

3. $M = N$ is derivable in the theory of structural equations with variables (see figure 4).

4. $M = N$ is derivable in the theory of structural equations without variables.

The equivalence of 2 and 3 can be shown by an induction on the length of derivations. The implication $3 \Rightarrow 1$ can also be shown by induction on derivations, and is quite easy to visualize: the idea is to think of free variables as constituents, and assign them the same sorts of relational diagrams you would assign a constant of the same type. The claim that the rule ζ 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. $2 \Rightarrow 1$ follows from the equivalence of 2 and 3. The implication $1 \Rightarrow 3$ is a little more involved. It is proved by assigning to each relational diagram a canonical term of the structural calculus. 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. $3 \Rightarrow 4$ is the most involved: a sketch of proof is offered in the appendix. (It involves introducing, for a special class of structural terms with free variables, a substitute for λ-abstraction, and showing that another rule is admissible in that theory.)

5 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 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:[34]

[34]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.

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

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

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

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 λs, bind variables: one can directly make statements like $\exists x.Lxx$ without invoking λs 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 Rs 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 $(\exists_{\hat{e}\hat{e}} L)^0_0$.

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 Rs a, and something is Red 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: $\forall_{\hat{e}}(\exists_{\hat{e}e} R)$

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 $\boldsymbol{\sigma}$.

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

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 co-ordinated 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 the it into the structural calculus.[35]

One might object that the resulting theory is unparsimonious, since we have posited infinitely many different primitive quantificational expressions. However the type theories of Frege, Church and Curry are all in the same position: for each type σ there is a different quantificational expression, $\exists_\sigma$ and $\forall_\sigma$, for expressing generalizations at that type. One might think the situation here is worse, since one needs infinitely many primitive operations to express all the sorts of quantificational claims at a given type: there are infinitely many first-order quantificational expressions, for example. But even this is not that different from the situation in orthodox type theory. There are infinitely many first-order quantifiers apart from the existential and universal quantifiers: *exactly three*, *many*, *few*, *most*, *just as many* and so on. In orthodox type theories one can supply logically complex definitions of these quantifiers: for instance Frege gave us an analysis of *there are just as many Fs and Gs* in terms of higher-order quantification over binary relations that are one-to-one correlations. But many philosophers — even those skeptical of higher-order quantification — have found the listed expressions to be perfectly intelligible: they just express other kinds of first-order quantification, alongside *all* and *some*. A second-order analysis of them is just as inappropriate as a second-order analysis of first-order existential quantification. A structuralist might therefore insist that, like 'some' and 'all', the expressions 'many', 'most', 'just as many' and so on express metaphysically simple, unstructured entities, not structured entities involving higher-order quantifiers. The expectation that we should, for instance, be able to define the binary existential quantifier, $\exists_{\hat{e}\hat{e}}$, the property a relation has when a single thing instantiates both its arguments, in terms of the unary existential quantifier, originates from the fact that we are able to offer this reduction in the case of Church and Curry; I do

[35]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 λ is immediately preceded by a quantifier can be translated $\forall_\sigma \lambda x.A$.

not believe it is something we should expect to be possible antecedently, without making any special metaphysical assumptions about properties and relations.

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 σ 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 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 structuralist cannot simply adopt Leibniz equivalence as their definition of identity. While one can paraphrase away cases of in which a variable appears more than once or which 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 [1], $\lambda xy.\forall Z(\forall w.Zww \to Zxy)$, 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 it is possible to define non-extensional operations from extensional primitives.[36] We saw above that propositional identity, $=_t$, can be defined in terms of the operator quantifiers 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, it is only possible to define extensional operations from extensional primitives.

In general, we say that F and F' when coextensive when $\forall x_1..x_m(Fx_1...x_m \leftrightarrow F'x_1...x_m)$ is true, and that a context is extensional when coextensive expressions are intersubstitutable *salve veritate*. Now suppose that F and F' are coextensive, and G and G' are coextensive. This means that $\forall x_1..x_m(Fx_1...x_m \leftrightarrow F'x_1...x_m)$ and $\forall y_1...y_n(Gy_1...y_n \leftrightarrow G'y_1...y_n)$ are both true. Then it follows that $(FG)^m_n$ and $(F'G')^m_n$ are coextensive: rewriting it in the λ-notation this simply amounts to $\forall x_1...x_my_1...y_n(Fx_1...x_m(Gy_1...y_n) \leftrightarrow F'x_1...x_m(G'y_1...y_n))$, which clearly follows from our two assumptions. It follows that if a sentence A is built entirely from extensional constants, and that F and F' are coextensive expressions, then $A \leftrightarrow A[F/F']$ is true: coextensive properties intersubstitutable *salve veritate*.[37]

[36] See Bacon [2].

[37] A straightforward induction shows that for an arbitrary expression H and $H[F/F']$ are coextensive, which implies the special case where H has type T.

6 Paradox

Several philosophers have recently argued that the theory of structured propositions is untenable on the grounds that it is inconsistent. For example, Dorr [13] points out that the following principal, formulated in Church's system with extensional β, is inconsistent:

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

where this schema can be instantiated by any terms $F, G : \sigma \to \tau$ and $a, b : \sigma$. In Church's system, there is an instance of this schema where F, G, a and b are variables, from which can universally generalize to get the quantified claim $\forall XYzw(Xz = Yw \to X = Y \wedge z = w)$. One can derive an inconsistency from this schema using only principles of H. Indeed, Dorr notes that even if you replace β with extensional-β the inconsistency still arises. Similar arguments against structured principles are put forward in Hodes [20], Uzquanio [35], and Goodman [16]; all these arguments trace back to Russell [31] appendix B.

The options for resisting this result are pretty limited. A couple of well-worn responses are to weaken the logical principles of H in various ways. One can weaken the propositional logic, following the familiar strategies for dealing with the liar and other paradoxes. Another option is to weaken the quantificational logic: one can, for instance, relinquish universal instantiation — the sort of move that lies behind free-logical and ramified responses to these paradoxes. But these responses often involve draconian restrictions on the type system, or severely limit our ability to reason in these systems (or both).[38]

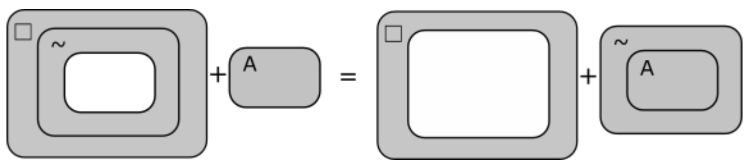
The principle Extensional-β, by contrast, is an obvious point to get off the boat. As we argued in section 3, certain λ-terms in Church's calculus are simply not meaningful by the lights of the structured theorist. Once we have rejected the meaningfulness of these λ-terms, what reason do we have to accept the instances of Extensional-β that involve them? One might point out that in particular cases we have candidate meanings for λ-terms that would suffice to make the relevant instance of extensional-β true. We could interpret $\lambda xy.Lyx$ as *hates*, provided everyone hates people that love them back, and conversely. But while interpretations of particular λ-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-β true. Consider the following term $\lambda X.Xa$ — illegitimate by the structuralists 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.[39] (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 4, there are *some* λ-terms that are meaningful by structuralist lights: for these terms one can even accept the full principle β, where the biconditional is replaced with an identity. But outside this limited domain, the structuralist has no reason to accept Extensional-β.

[38] My own view of these limitations can be found in Bacon et al. [5].

[39] Uzquiano [35], for instance, does not appeal to Extensional-β 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.

It's worth emphasizing that the derivation of the Russell-Myhill paradox turns essentially on the use of a λ-term that our structuralist rejects. I won't recount the argument here, 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 the bound variable X can be paraphrased away using our coordinated quantifiers, there is no similar trick for achieving quantification into predicating position.[40]

Now let us turn to Structure. Would a refutation of Structure count as a refutation of the theory of structured propositions? No: even before learning of its inconsistency in some versions of higher-order logic, a structuralist already had good reason to reject Structure. Consider the proposition $\Box(\neg A)$, considered earlier. We have two distinct ways of building this proposition. We can put negation and A together by application — using our notation $(\neg A)^0_0$. We can then apply $\Box$ to the result, to get $(\Box(\neg A)^0_0)^0_0$. Alternatively we can put $\Box$ and $\neg$ together to get the structured operator $(\Box\neg)^1_0$, and we can then apply that to A, $((\Box\neg)^1_0 A)^0_0$. Either way we get the same result:



But the principle Structure says that there's only one way to make a proposition by applying an operator to a proposition: Structure would thus imply the following pair of identities:

$$(\Box\neg)^1_0 = \Box \text{ and } A = (\neg A)^0_0$$

Both identities are clearly false — the latter inconsistent.

There are other routes to paradox, for instance in Bacon [4] the following weaker principle is considered:[41]

Predicate Structure $Fa = Ga \rightarrow F = G$

Again, understanding this as schematic in both $F, G : \sigma \rightarrow \tau$ and $a : \sigma$, and in the types σ and τ. This principle does not imply, given the identity $(\Box(\neg A)^0_0)^0_0 = ((\Box\neg)^1_0 A)^0_0$, that $\Box = (\Box\neg)^1_0$ and $(\neg A)^0_0 = A$. 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 an plug *Alice* into the second argument hole, yield *Alice loves*, and apply the resulting unary property to *Alice*. Thus:

$$((La)^0_0 a)^0_0 = ((La)^0_1) a)^0_0$$

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.[42]

[40] A more complicated trick is available for the two ps.

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

[42] 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 $(La)^0_0$ and $(La)^0_1$ are identical.

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 H. Because we are using more general co-ordinated 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 σ_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_is. 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 β and η 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 table 4).

Since our language is variable free, we do not have an analogue of the rule Gen. One could go the way we went with the rule ζ, where we introduced 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 I shall call a *fundamental language*: the idea here is that every fundamental entity is denoted by a unique constant of the language, and no constant of the language denotes a complex entity. In such a language we can formulate two principles capturing the structured vision:

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: our equational theory tells us exactly when two terms of our language denote the same structured entity, so we must assert not only the identities it proves, but the distinctness for equations it does not prove. Evidently Distinctness should not be expected to hold in an arbitrary language: for suppose that we had a terms $F : (e \to t) \to (e \to t)$, $G : e \to t$ and $H : e \to t$ — '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 $(FG)^1_0 = H$ 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 fundamental 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 fundamental entity has a name. And every structured entity is built from the fundamental guys, using our general way of gluing entities together: generalized application. 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.[43]

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.[44] So the possible forms that a term of type t can take are $\wedge AB$, $\neg A$, $=_\sigma ab$, $\forall_{\boldsymbol{\sigma}} Fa_1...a_n$, 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 $a = b$ is a structural equation
- $v(A \wedge B) = \min(v(A), v(B))$
- $v(\neg A) = 1 - v(A)$
- $v(\forall_{\boldsymbol{\sigma}} F\check{\boldsymbol{a}}) = \min_{c:\tau} v(F[\boldsymbol{a}/c])$

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

[43] Completeness is related to the principle Fundamental Completeness in Bacon [3] and Bacon [4]: 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'.

[44] The analogous fact about the λ-calculus is well-known (see, e.g., Hindley and Seldin [19]), and can be transferred to the structural calculus using theorem [REF].

One naïve strategy for constructing a valuation is to assign a truth value $|Ra_1...a_n|$ arbitrarily to atomic formulae[45] 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 fundamental 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 [28]. 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.[46]

The naïve strategy cannot work, then, for the following reason: suppose that Q was a fundamental constant, and I were simply to set its extension to contain all equivalents of $(\vdash \forall p(Qp \to \neg p))^s$ — 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 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 extistential, or counterexamples to a universal, are circular in this way. See also Kripke [25] p331-332 for some relevant discussion.

Prior's theorem tells us that not every assignment of extensions to the fundamental 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

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

[46] If you want to revisit these assumptions, see for instance Tucker and Thomason [34], Tucker [33], Kaplan [22], Deutsch [9], Bacon et al. [5], Bacon [**?**].

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 2. *There exists a valuation on the structural calculus over any signature of fundamental 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 λ-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 3 below.[47] 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 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 3. *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. However, it is instructive to see why the analogous theorem would be false in the case of the full λ-calculus. In H, or 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 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,

[47]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.

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. It's worth noting that this consistency argument works in fragments of the λ-calculus that go beyond the structural λ-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).

Lastly, let me mention some limitative results. Evidently the Russell-Myhill paradox tells us that structural calculus cannot be extended with λ-terms satisfying extensional β, or combinatory terms satisfying extensional I, K and 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 identity combinator for operators.

> One cannot introduce an operation $I^{t\to t} : (t \to t) \to t \to t$ that satisfies Extensional I.

Extensional $I^{t\to t}$ $I^{t\to t}MA \leftrightarrow MA$, 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, $I^{t\to t}XA$. One is then in a position to define the Russell-Myhill operator, and generate a paradox in the usual way.[48]

Given our earlier remarks about the meaningfulness of terms like $\lambda X.X$ and $I^{t\to t}$ in the λ and combinatory calculi, we shouldn't have *expected* there to be a term that satisfies Extensional $I^{t\to t}$. If it is at all surprising, it is for a superficial reason: it seems (at first glance) as though you couldn't do anything with the claim $I^{t\to t}Xp$ that you couldn't simply do with Xp, suggesting that an operation like $I^{t\to t}$ is redundant. But this is, of course, wrong: $I^{t\to t}$ allows one to do exactly the sorts of things that lead to the Russell-Myhill paradox, like simulate quantification into predicating position.

[To do: be explicit about what I mean by quantification into predicating position. A variable in predicating position Xa versus in argument position Fx. Note that it's possible that a predicate variable be in argument position, as in $\exists_e X$: quantification into predicate position and predicating position must be distinguished.]

We mentioned earlier that certain operations definable in the λ-calculus, like predicate and operator negation — $\lambda X\lambda y.\neg(Xy)$ and $\lambda X\lambda p.\neg(Xp)$ — cannot be defined structural logic. 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 $I^{t\to t}$.

[48] TO DO: Write out full proof. Appendix.

7 Conclusion

In this paper I have argued that the theory of structured propositions is not directly undermined by the Russell-Myhill paradox, and have developed a theory of structured propositions that does not fall afoul of the Russell-Myhill paradox. But if there is a larger moral to be taken away from all of this, it should not be that we can continue to reason naïvely about structured propositions. 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.[49] 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. Similar remarks apply to the use of the theory of structured propositions in metaphysics. Secondly, the consistency of the theory presented here is still an open question. If it is indeed a faithful precisification of the structured theory employed informally in metaphysics and elsewhere, its consistency is then an urgent matter.

8 Appendix

In this appendix we prove that if M and N do not contain deductive variables, $M = N$ is derivable in the theory of structural equations with deductive variables iff it is derivable in the variable free equational theory. This forms the centerpiece for theorem **??**.

We shall add deductive variables to the structural calculus by simply treating them as new constants. Given a structural term M, in a variable $x : \sigma$, we will define another structural term, $[x]M$. 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.

For any term M we define $c(M)$ to be the total number of occurences 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.*

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

[49] The λ-calculus has been a central tool in semantics since Montague [27]. Combinatory logic, by contrast, plays a foundational role in the direct compositionality program, tracing back to Jacobson [21].

Proposition 5. *If the equation $M = N$ is a theorem of the theory of structural equations without variables [REF], then $[x]M$ is defined iff $[x]N$ is defined, and $[x]M = [x]N$ is also a theorem of [REF].*

Proof. We prove this by induction on the length of deduction in [REF]. 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 ι.

(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 [REF]

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 μ: Assume for the inductive hypothesis that $M = M'$ is a theorem. Then $[x](MN)^n_m$ and $[x](M'N)^n_m$ are defined in exactly when $[x]M$ and $[x]M'$ are defined respectively, so by inductive hypothesis they are defined in exactly the same conditions. 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 using the inductive hypothesis, or the inductive hypothesis with another application of μ.

For the rule ν 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 the 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 ι. 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 from an instance of ν^n_m.

The cases of transitivity (τ), symmetry (σ) and identity (ι) are all straightforward. $\square$

As a corollary of this proposition, the following rule is admissible in [REF]:

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

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

Corollary 6. *The rule ζ is admissible in the theory rearrange.*

Proof. Suppose that $Mx = Nx$ is derivable in theory [REF] and x isn't free in M or N. Thus both $[x].Mx$ and $[x].Nx$ are defined and identical to M and N respectively. By theorem [REF], we know that there exists a derivation of $[x].Mx = [x].Nx$, which given the definition of $[x]$ is a derivation of $M = N$. □

Theorem 7. *Suppose M and N are terms of the structural calculus that do not contain deductive variables. Then $M = N$ is derivable in the theory of structural equations with deductive variables iff it is derivable in the theory of structural equations without deductive variables.*

Proof. Firstly we show that the two equational theories are equivalent with respect to a common language including deductive variables.

The first inclusion follows by observing that you can derive Rearrange, Recompose, μ_n^m and ν_n^m in the ζ theory with deductive variables (essentially using γ and ζ). In the other direction, we notice that all of the rules in table [REF] are admissible in the ζ free theory. So they prove the same equations in the common language.

In particular, if $M = N$ does not include deductive variables and has a derivation in the theory with ζ, then it has a derivation in the ζ free theory. We know that this derivation will not involve any variables, as apart from τ, none of the rules allow a subterm to disappear between the premise and conclusion equations (i.e. a subterm of a rule can only). Moreover, it's easy to see that if $M = N$ is derivable and M contains a particular variable so does N, thus even τ cannot allow a variable to disappear. Thus if $M = N$ can be derived in the ζ theory in a language with deductive variables, then it can be derived in the ζ-free theory in the language without deductive variables.

If $M = N$ is derivable in the ζ-free theory without deductive variables then our above reasoning shows that it can be derived in the ζ theory with deductive variables.

□

References

[1] Peter B Andrews. *An introduction to mathematical logic and type theory*, volume 27. Springer Science & Business Media, 2002.

[2] Andrew Bacon. The broadest necessity. *Journal of Philosophical Logic*, 47(5):733–783, 2018.

[3] Andrew Bacon. Substitution structures. *Journal of Philosophical Logic*, 48(6):1017–1075, 2019.

[4] Andrew Bacon. Logical combinatorialism. *Philosophical Review*, forthcoming.

[5] 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.

[6] Mark Crimmins and John Perry. The prince and the phone booth: Reporting puzzling beliefs. *Journal of Philosophy*, 86(12):685, 1989.

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

[8] 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.

[9] Harry Deutsch. Resolution of some paradoxes of propositions. *Analysis*, 74(1):26–34, 2014.

[10] 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.

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

[12] Cian Dorr. Quantifier variance and the collapse theorems. *The Monist*, 97(4):503–570, 2014.

[13] Cian Dorr. To be f is to be g. *Philosophical Perspectives*, 30(1):39–134, 2016.

[14] Kit Fine. Neutral relations. *Philosophical Review*, 109(1):1–33, 2000.

[15] Gottlob Frege. On sense and reference. In Darragh Byrne and Max Kölbel, editors, *Arguing About Language*, pages 36–56. Routledge, 2010.

[16] Jeremy Goodman. Reality is not structured. *Analysis*, 77(1):43–53, 2017.

[17] Jeremy Goodman and Harvey Lederman. Perspectivism. *Noûs*, forthcoming.

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

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

[20] Harold T. Hodes. Why ramify? *Notre Dame Journal of Formal Logic*, 56(2):379–415, 2015.

[21] Pauline Jacobson. Towards a variable-free semantics. *Linguistics and Philosophy*, 22(2):117–185, 1999.

[22] 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.

[23] Jeffrey King. Structured propositions and sentence structure. *Journal of Philosophical Logic*, 25(5):495–521, 1996.

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

[25] 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.

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

[27] 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.

[28] A. N. Prior. On a family of paradoxes. *Notre Dame Journal of Formal Logic*, 2(1):16–32, 1961.

[29] A. N. PRIOR. *Objects of Thought*. Clarendon Press, 1971.

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

[31] Bertrand Russell. *Principles of Mathematics*. Routledge, 1937.

[32] Scott Soames. Cognitive propositions. *Philosophical Perspectives*, 27(1):479–501, 2013.

[33] Dustin Tucker. Paradoxes and restricted quantification: A nonhierarchical approach. *Thought: A Journal of Philosophy*, 7(3):190–199, 2018.

[34] Dustin Tucker and Richmond H. Thomason. Paradoxes of intensionality. *Review of Symbolic Logic*, 4(3):394–411, 2011.

[35] Gabriel Uzquiano. A neglected resolution of russell's paradox of propositions. *Review of Symbolic Logic*, 8(2):328–344, 2015.

[36] Timothy Williamson. Converse relations. *Philosophical Review*, 94(2):249–262, 1985.

[37] Timothy Williamson. Everything. *Philosophical Perspectives*, 17(1):415–465, 2003.