Higher-order logic as metaphysics*

Jeremy Goodman

September 20, 2023

This chapter offers an opinionated introduction to higher-order formal languages with an eye towards their applications in metaphysics. These languages go beyond the first-order predicate calculus in two ways: they allow for predicate abstraction and also for quantification into sentence and predicate positions. Together these two innovations allow us to formulate deep questions about the structure of reality using purely logical vocabulary.

Much of the literature on the interpretation of higher-order formal languages focuses on how different they are from natural languages. While these differences are real, focussing on them can give the misleading impression that the intelligibility of higher-order languages is more vexed than it really is. For the first-order predicate calculus also departs from natural language in many of the same ways, and its intelligibility should not be in question. This chapter therefore proceeds by explaining how readers fluent in first-order logic can leverage that fluency to become comfortable theorizing in higher-order languages.

Section 1 explains the operative understanding of first-order languages, while sections 2 and 3 advance interpretations of predicate abstraction and of higher-order quantification as natural outgrowths of that understanding. Section 4 explains how these logical resources allow us to ask significant metaphysical questions. Section 5 explores the metaphysical implication of the claims about predicate abstraction advanced in section 2. I argue that we must reject the popular idea that structural differences between sentences correspond to parallel distinctions in the logical structure of extra-linguistic reality. But it may still be possible to give a purely logical characterization of objectual aboutness and related notions.

1 First-order preliminaries

The aim of this section is to make explicit a certain understanding of the first-order predicate calculus. I will argue, first, that predicate letters are meaningful expressions that can, and often are, introduced using explicit definitions. I will

*Forthcoming in *Higher-Order Metaphysics*, edited by Peter Fritz and Nicholas Jones. An ancestor of this paper was presented at the 2014 CRNAP workshop in Oxford, and in graduate seminars at USC in 2017 and 2020. Thanks to Andrew Bacon, Michael Caie, Cian Dorr, Ibrahim Haydar, Harvey Lederman, and Jake Nebel for comments on earlier drafts, and especially to Peter and Nick for their encouragement, advice, and patience.

then argue that our ability to use first-order formal languages to theorize about the world is not underwritten by any explicit or tacit translation of its formulas into some independently understood natural language. These claims are not intended to be particularly controversial or exciting on their own, but they will take on new significance in the context of higher-order languages.

1.1 Defining predicates

When we argue, reason, and theorize using the first-order predicate calculus, we are not typically using formulas as mere abbreviations or shorthand. Rather, we are doing the same kind of thing that physicists do when they use symbolic notation to argue, reason, and theorize about physical systems. The equations bearing Maxwell's and Einstein's names aren't mere abbreviations for sentences of English or of German. They have an intended interpretation of their own. The same is true when philosophers use the first-order predicate calculus. This is true not only when metaphysicians engage in the kind of systematic theorizing that most closely resembles theoretical physics, but also in the more common, pedestrian, *ad hoc* uses of logical formalism that one finds in all areas of contemporary analytic philosophy.

When we so philosophize, the predicate letters we use are genuinely meaningful expressions with intended interpretations. This raises the question: how do these predicate letters get their meanings? Let's consider three ways. (Our focus will be on the third, but the first two provide a useful contrast.)

The first way is that we simply start using a new predicate symbol as part of an inquiry or attempt to communicate. If our use is sufficiently systematic and the world cooperates, this may be enough for the symbol (as we use it) to be meaningful. How this happens is a very hard question, since our understanding of 'meta-semantics' (i.e., that in virtue of which our words are meaningful and have the meanings that they do) leaves much to be desired. Yet since our *words* clearly have meanings, and largely as a result of how we use them, there is no clear reason to doubt that predicate letters can come to be meaningful in the very same way, through appropriately systematic use.

A second way that predicate letters might get their meanings is modeled on how names can get their referents through reference-fixing descriptions. Since we know that there was a unique person who first invented the zip, we might stipulate that the name "Julius" refer to them.[1] Similarly, we might introduce a meaningful predicate letter "F" by stipulating that it express whatever condition uniquely plays a given theoretical role.[2] That is, we uniquely characterize what a predicate is to mean and then stipulate that this be what it means.

While these are both legitimate ways of introducing meaningful predicate letters into our vocabulary, our focus will be on a third way: *explicit definition.* This is what we often teach students to do in beginning logic classes. Here are some examples (where "P" and "A" are for "parent" and "aunt"):

[1] The example is from Evans (1979), building on Kripke (1972).

[2] Lewis (1970) influentially explained how this might be done for many predicates simultaneously in terms of a theoretical role uniquely played by many conditions taken together.

$Pxy := y$ is a child of x

$Fz := z$ is female

$Mxy := x$ and y are married

$Axy := Fx \wedge \exists z_1 \exists z_2 \exists g((x = z_1 \vee Mxz_1) \wedge Pgz_1 \wedge Pgz_2 \wedge Pz_2y \wedge z_1 \neq z_2))$

These definitions all have the form:

$Rx_1 \ldots x_n := \varphi$

Here $x_1, \ldots, x_n$ are n distinct variables and φ is a formula in which $x_1, \ldots, x_n$ are all and only the variables that occur free. The predicate being defined is R, and it is an n-place predicate.[3]

These variables are 'individual' variables, in the sense that they occur in syntactic positions that can be occupied by proper names and similar referential expressions like numerals and singular pronouns. For any such expressions $a_1, \ldots a_n$, we let $\varphi[a_i/x_i]$ be the sentence that results from simultaneously replacing each free occurrence of any x_i in φ with an occurrence of the corresponding a_i (for all $i \leq n$). For example, replacing "z" with the name "Zephyr" in the right hand side of the definition of "F" above yields the sentence "z is female"["Zephyr"/"z"], which is simply "Zephyr is female". In this case, the replacement yields a sentence of English. But in other cases it could be a sentence of *Loglish*, a familiar chimera of English words and logical symbols. This is what happens when we substitute "Anne" and "Will" for "x" and "y" in the right hand side of the definition of "A", yielding:

1. $F\text{Anne} \wedge \exists z_1 \exists z_2 \exists g((\text{Anne} = z_1 \vee Mxz_1) \wedge Pgz_1 \wedge Pgz_2 \wedge Pz_2\text{Will} \wedge z_1 \neq z_2))$

This is a little hard to read, since it juxtaposes predicate letters with English names. This is one reason why we typically introduce lowercase letters ('individual constants') to refer to individuals, whose juxtaposition with predicate letters is easier on the eyes. But this aesthetic shortcoming is no deep challenge to the intelligibility of Loglish sentences. Consider, for example:

2. $\forall x(\text{Narcissus loves } x \leftrightarrow x = \text{Narcissus})$.

This sentence is clearly intelligible, and sentences like it are commonplace in contemporary philosophical writing.

I have claimed that predicate letters, used in a certain familiar way, are meaningful symbols, and then described three ways in which predicates can acquire their meanings. But in the case of the predicates introduced in the third way, using explicit definitions, it is instructive to consider a natural alternative hypothesis. Here is the idea. Consider:

3. $P\text{ElizabethCharles}$

[3] "Definition" of course has broader uses too; for example, inductive definitions in mathematics are (in the present terminology) cases of reference fixing rather than explicit definitions.

Given our explicit definition of "P", there is a way of understanding this sentence without treating that predicate letter as a meaningful in itself. Instead, we can treat its definition as a recipe for using the letter "P" to abbreviate sentences that we already understand; in this case, the sentence:

4. Charles is a child of Elizabeth.

This idea is reminiscent of Russell (1905), who argued, roughly, that descriptions are not meaningful on their own, but in any given context they provide a convenient shorthand for sentences not containing any such descriptions.

This proposal gets something important right. There is an intimate connection between 3 and 4 owing to how "P" was explicitly defined. We will return to this point shortly. Nevertheless, there is good reason to think that "P" is a meaningful symbol and not a mere device for abbreviation. For our practice of introducing predicate letters using explicit definitions is of a piece with our practice of introducing new nouns, verbs, and adjectives into English using explicit definitions. In the case of English, such new words are clearly meaningful – this is why they can later be acquired and used by other English speakers, even if those speakers don't know exactly how those words were initially defined. And having already granted the existence of *some* meaningful predicate letters, there is no good reason to deny that our stock of them can be extended using definitions in the same way that our stock of meaningful words can be.[4]

What is special about definitionally introduced predicates is that, just like definitionally introduced words, when we predicate them they are *as good as abbreviations*. In particular, 3 and 4 are *everywhere intersubstitutable*.[5] We can express this general feature of explicit definitions schematically as follows:

Definitional Equivalence
From $Rx_1 \dots x_n := \varphi$, infer: $\Phi[Ra_1 \dots a_n]$ if and only if $\Phi[\varphi[a_i/x_i]]$.

The notation $\Phi[A]$ stands for the formula that results from filling the gap "$\dots$" in Φ with the expression A. So if Φ is "Kurt plays $\dots$", then Φ["guitar"] is "Kurt plays guitar". The schematic biconditional $\ulcorner\Phi[A]$ if and only if $\Phi[B]\urcorner$ then says, in effect, that A and B are everywhere intersubstitutable. For example, from the definition of "P" above, we can infer:

[4]What would be an example of a non-meaningful predicate-like symbol? Imagine using V so that, for any name a, $\ulcorner Va \urcorner$ abbreviates $\ulcorner$"a" has five letters$\urcorner$. So VTully and not VCicero. But this doesn't mark a difference between Tully and Cicero – there isn't any, since Tully is Cicero. Moreover, there is no sensible question "What is it to be V?", in the way that "What is it to be round/good/human/interesting?" are sensible questions. "Representational" may be better than "meaningful" for the intended status here. But that status, whatever we call it, entails (i) that materially inequivalent predications of the same predicate mark a difference between the entities of which it is predicated, and (ii) it is a sensible question what this difference consists in. These informal ideas will be made precise in section 4.

[5]More precisely, 3 and 4 are as intersubstitutable as they would be if "P" were merely a device for abbreviation. This qualification accommodates the fact that they are not substitutable inside quotation marks. Whether they are substitutable in the complement clauses of propositional attitude reports is a delicate question beyond the scope of this paper. For a defense of an affirmative answer, see Goodman and Lederman (2021).

5. (Charles is heir to the throne partly in virtue of the fact that PElizabethCharles) if and only if (Charles is heir to the throne partly in virtue of the fact that Charles is a child of Elizabeth).

We will see below how higher-order languages allow us to get a surprising amount of metaphysical mileage out of the fact that we can introduce predicates that are both meaningful and governed by **Definitional Equivalence**.

1.2 First-order logic as metaphysics

Let us now turn to the second claim I made at the beginning of this section: that we have an understanding of sentences of first-order logic that does not go by way of (and is not hostage to the existence of) any translation into English or any other natural language. As Williamson (2003) puts it, we can make such formal languages the 'home language' in which we theorize.[6] I will now give a simple example to illustrate what I mean by this, and use it to make two points that will become important when we consider higher-order languages.

I assume that anyone reading this chapter can understand the following two sentences with little effort:

6. $\exists x \exists y (x \neq y)$
7. $\forall x \forall y (x = y)$

What do they mean? If you're like me, your first impulse is to say that the first means that there are (at least) two things, while the latter means that there is (at most) one thing. Or you might instead say that they respectively mean that some things are distinct from each other and that all things are identical to one another. And it isn't hard to come up with further not unreasonable glosses.

The structural discrepancy between the above formulas and the English sentences we naturally use to gloss them, reinforced by the diversity of these glosses, is revealing. It illustrates that we do not understand 6 and 7 by exploiting any mechanical translation of the symbolism of first-order logic into English. To the contrary: it is because we understand what 6 and 7 mean, considered on their own, that we can recognize that they mean the same as (or near enough) the sentences that come to mind when asked to say in English what they mean.

It is important to distinguish the above use of English sentences to gloss sentences of formal languages from another superficially similar practice of using English words in connection to logical formulas. Consider:

8. There is an x such that there is a y such that x is distinct from y.
9. For all x and y, x is identical to y.

[6] I am not claiming that the human language faculty plays no role in how we intelligently manipulate sentences of formal languages. See Pietroski (2018, chapter 3), which emphasizes the differences between human language and artificial formalisms without casting doubt either on the intelligibility of the latter or on the idea that the human language faculty plays an important cognitive role when we deploy formal languages in intellectual projects.

Philosophers frequently use 'sentences' like 8 and 9 as a way of writing and pronouncing formulas like 6 and 7. This is especially important in speech, since listing symbols would be intolerable.[7] But not only are 8 and 9 clearly not sentences of English, they are also quite unlike the Loglish sentences 1-3 above. Those Loglish sentences involved extending the rules for well-formed formulas of the predicate calculus to subsume a fragment of English: names can be replaced by variables to form formulas; two formulas can be combined to form a third by flanking "$\leftrightarrow$"; one formula can be enclosed in parentheses and prefixed by "$\forall x$" to form another formula; and so on. Nothing like this is going on in 8, in which the initial occurrences of "x" and "y" take the place not of proper names but of singular common nouns like "dog'; nor in 9, where they take the place of plural common nouns like "cats".

8 and 9, then, are much closer to abbreviations. Their interpretation is parasitic on the interpretation of the sentences 6 and 7 that they stand for. The best way to think of 8, for example, is simply as a convenient way of pronouncing 6. This pronunciation scheme is useful owing to the many ways in which quantification in first-order logic differs structurally from quantification in natural languages. For example, English grammar gives rise to scope ambiguities that are avoided in the predicate calculus through the use of distinct variables. Quantifiers in English are also always syntactically restricted, in the sense that they combine first with a noun before they combine with a verb phrase (as "dog" restricts "every" in "every dog barks"), whereas quantifier prefixes in the predicate calculus combine directly with open sentences.[8] The convention of using variables in the place of nouns that would restrict quantifiers is an ingenious way of exploiting the grammar of English to smoothly pronounce formulas of formal languages in which quantification works quite differently.

I am belaboring this point not because it is of great significance in the case of first-order languages, where it is unlikely to cause any confusion, but because a parallel point is significant when we consider higher-order languages. To preview: there are standard ways of using English words like "property", "proposition", and "type" to pronounce formulas of higher-order logic. But, at least in the higher-order metaphysics literature, this is typically *merely* a pronunciation scheme, which we deploy out of convenience, and the utility of which is easily understood by reflecting on the differences between the syntax of quantification in higher-order formal languages and the syntax of quantification in English. While there are substantive metaphysical questions about what properties and propositions there are, no such questions are begged by merely using "property" and "proposition" as part of a scheme for pronouncing formulas, just as it would clearly be confused to think that the use of 8 commits one to two categories of things, the xs and the ys.

A second important point illustrated by 6 and 7 is that sometimes logic is metaphysics. That is to say: we can use the symbolism distinctive of formal logic

[7] I learned this from experience, when some friends had an unflattering impression of me circa 2013 that involved saying "Box phi horseshoe psi".

[8] Even in "everything exists" there are good reasons to think that, at the level of logical form, there are distinct constituents corresponding, roughly, to "every" and "thing".

to formulate competing claims about the world, in the same way that physicists use their proprietary notation to formulate competing claims about physical reality. And debates over such claims belong to metaphysics (as opposed to physics, history, ethics, or some other discipline). The competing claims that, on the one hand, there are at least two things, and, on the other hand, that there is at most one thing, have in fact been debated by metaphysicians. And these claims can be articulated using only first-order quantification, identity, and negation – that is, using only logical vocabulary.

The shortcoming of first-order logic as metaphysics isn't any tension between the sometimes supposed neutrality of logic and speculativeness of metaphysics. If there is a sense of 'logic' in which it must be uncontroversial, or a sense of 'metaphysics' where its claims must be tendentious, it is not my sense here. The problem with first-order logic as metaphysics is rather that it is *boring*. For 7 is clearly false, the illustrious pedigree of ontological monism notwithstanding. (You and I are a counterexample.) There are at least two things, and indeed arguably at least n things for every finite number n. And from these truths we can derive every truth that can be formulated using only the logical resources of first-order logic with identity; cf. Rayo and Williamson (2003).

By contrast, higher-order logic as metaphysics is not boring. In sections 4 and 5 we will see how higher-order languages allow us to ask a host of new interesting and non-obvious questions, with the same starkness and precision exhibited by purely logical formulas like 6 and 7.

Let me summarize what I have argued so far. I have argued that predicates introduced by explicit definition satisfy **Definitional Equivalence** and are meaningful expressions in their own right. I then argued that the symbolism of first-order logic is a meaningful representational system, which we can directly interpret without translating it into any natural language. In fact, some uses of natural language that one might initially mistake for partial translations of symbolism into English are in fact the reverse: we sometimes use English words and syntactic structures to pronounce formulas, but when we do so their interpretation is parasitic on that of the formulas in question, not the other way around. That we do this is understandable given the syntactic differences between first-order logic and English, especially when it comes to quantification.

The reason for being explicit about these points is to preempt doubts about the intelligibility of higher-order formal languages based on the extent to which they depart from natural language. Some proponents of higher-order metaphysics have responded to such misgivings by trying to minimize the extent of these departures, most heroically Prior (1971); cf. Boolos (1984) and Rayo and Yablo (2001). Here I have adopted a different strategy. The departures between higher-order logic and natural language are many, and if there were no precedent of understanding similarly alien notational systems, then that would indeed be grounds for caution. But there are many such precedents, one of which is first-order logic. We have become so used to it that we easily forget how innovative it is from the perspective of natural language. It is no accident that contemporary quantification theory had to wait over two millennia from Aristotle to Frege. The question of how it is that we can manage to use and understand artificial symbolisms like first-order logic is a deep question in cognitive psychology, but we know that it has an answer because understand them we do.

Of course, higher-order logic does involve some innovations beyond first-order logic. Let us now turn to these.

2 Predicate abstraction

This section explains the first innovation of higher-order languages: predicate abstraction. Section 2.1 explains how predicate abstraction can be understood as an outgrowth of our practice of introducing predicate letters using explicit definitions, discussed in section 1.1. Section 2.2 then explains two ways of using predicate abstraction that extend beyond what can be done by defining predicate letters in first-order languages, and introduces the type distinctions that are central to theorizing in higher-order languages.

2.1 Abstraction as definition: motivating λ-conversion

Instead of introducing a new predicate letter R using an explicit definition $\ulcorner Rx_1 \ldots x_n := \varphi \urcorner$, and then going on to use R, we can achieve the same effect in a single step, by using a complex predicate that wears its definition on its sleeve: $\ulcorner(\lambda x_1 \ldots x_n.\varphi)\urcorner$. This subsection makes precise the sense in which these λ-terms are equivalent to explicitly defined predicates, and shows how this equivalence motivates certain general principles governing such terms.

λ-terms are equivalent to corresponding explicitly defined predicates in the sense that the two are *everywhere intersubstitutable*. More precisely:

> **Abstraction as Definition**
> From $Rx_1 \ldots x_n := \varphi$, infer: $\Phi[R]$ if and only if $\Phi[(\lambda x_1 \ldots x_n.\varphi)]$.

This basic equivalence underpins the three central principle governing λ-terms: **β-conversion**, **α-conversion**, and **η-conversion**. Let me explain.

First, consider what happens when we combine **Abstraction as Definition** (linking λ-terms and defined predicates) with **Definitional Equivalence** (linking predications of defined predicates with their definitions). Taken together, these principles imply that, from an explicit definition $\ulcorner Rx_1 \ldots x_n := \varphi \urcorner$, we may infer $\ulcorner \Phi[(\lambda x_1 \ldots x_n.\varphi)a_1 \ldots a_n] \leftrightarrow \Phi[\varphi[a_i/x_i]] \urcorner$. Notice that the defined predicate R does not occur in this schematic biconditional. And any schema in which R does not occur clearly doesn't depend for its acceptability on the definition of R. So such biconditionals should be acceptable on their own. This motivates the central logical principle governing λ-terms:[9]

> **β-conversion**
> $\Phi[(\lambda x_1 \ldots x_n.\varphi)a_1 \ldots a_n] \leftrightarrow \Phi[\varphi[a_i/x_i]]$, where each a_i is free for x_i in φ

[9] Earlier versions of this argument can be found in Dorr (2016) and Goodman (2016).

In other words, every predication of a λ-term is eliminable in terms of the formula that it *immediately β-reduces* to (in which that λ-term does not occur).[10] For example, the following is instance of **β-conversion**:

10. $\neg((\lambda xy.y$ is a child of $x)$AliceBob$) \leftrightarrow \neg($Bob is a child of Alice$)$.

This is because the right-hand-side results from the left-hand-side by replacing "$(\lambda xy.y$ is a child of $x)$AliceBob" with the sentence "Bob is a child of Alice" that it immediately β-reduces to.

Notice that the above argument from **Abstraction as Definition** and **Definitional Equivalence** only directly motivates **β-conversion** for λ-terms $\ulcorner(\lambda x_1 \ldots x_n.\varphi)\urcorner$ that are of the right form to correspond to explicit definitions of predicates in the first-order predicate calculus: $x_1, \ldots, x_n$ be must distinct *individual* variables, and they must be *all* and *only* the variables free in φ. The next subsection argues that we can relax the requirements that $x_1, \ldots, x_n$ must be individual variables and that they must include all variables free in φ; section 5.2 considers the implications of relaxing the requirement that $x_1, \ldots, x_n$ must include only variables free in φ.

The connection between predicate abstraction and explicit definition motivates a second logical principle governing λ-terms:

α-conversion
$\Phi[(\lambda x_1 \ldots x_n.\varphi)] \leftrightarrow \Phi[(\lambda y_1 \ldots y_n.\varphi[y_i/x_i])]$, where no y_i are free in φ

Intuitively, this says that all that matters in λ-terms is the pattern of free variables in the *prefix* (part before the ".") and *matrix* (part after the "."). λ-terms agreeing on this pattern but using different variables are everywhere intersubstitutable. For example, the following is an instance of α-conversion:

11. Definitely $(\lambda x.x$ is guilty$)$Holmes $\leftrightarrow$ definitely $(\lambda y.y$ is guilty$)$Holmes.

Just as **Abstraction as Definition** motivates **β-conversion** via **Definitional Equivalence**, it also motivates **α-conversion** via a parallel principle about explicit definitions, namely:

Alphabetic Irrelevance
From $Rx_1 \ldots x_n := \varphi$ and $Sy_1 \ldots y_n := \varphi[y_i/x_i]$, infer: $\Phi[R] \leftrightarrow \Phi[S]$.

This principle is hopefully obvious. It can also be motivated by noting the equivalence of our practice of explicitly defining predicate letters using arbitrarily chosen variables and other ways of defining predicate letters in which variables play no role. Consider, for example, the numeral-based notation in

[10] The proviso that each a_i is *free for* x_i in φ means that, if a_i is a variable, then it doesn't become bound when it is substituted for any free occurrence of x_i in φ. (Recall that $\varphi[a_i/x_i]$ is the result of simultaneously replacing each free occurrence of any x_i in φ with an occurrence of the corresponding a_i.) For example, x is not free for y in $\ulcorner\forall x(x = y)\urcorner$; so the (false) sentence $\ulcorner\exists x\forall x(x = x) \leftrightarrow \exists x(\lambda y.\forall x(x = y))x\urcorner$ is not an instance of **β-conversion**. Wherever I write (or have written) $\varphi[a_i/x_i]$, the proviso that each a_i is free for x_i in φ should be taken as read. Where a_i is a complex term containing free variables, this means that none of those variables become bound when it is substituted for any free occurrence of x_i in φ.

Goldfarb (2003), where instead of writing $\ulcorner Pxy := y$ is a child of $x\urcorner$ we would write $\ulcorner P :=$ 2 is a child of 1$\urcorner$ – clearly both definitions are equivalent.

A different way of motivating **α-conversion** is as a consequence of **β-conversion** together with the following schema:

> **η-conversion**
> $\Phi[R] \leftrightarrow \Phi[(\lambda x_1 \dots x_n.Rx_1 \dots x_n)]$, where no x_i are free in R

To establish a given instance of **α-conversion**, note that $\ulcorner(\lambda x_1 \dots x_n.\varphi)\urcorner$ is intersubstitutable with $\ulcorner(\lambda y_1 \dots y_n.(\lambda x_1 \dots x_n.\varphi)y_1 \dots y_n)\urcorner$ (by **η-conversion**), which is then intersubstitutable with $\ulcorner(\lambda y_1 \dots y_n.\varphi[y_i/x_i])\urcorner$ (by **β-conversion**).

Like **β-conversion** and **α-conversion**, **η-conversion** can be motivated by **Abstraction as Definition** together with an independently plausible principle about explicit definitions. The principle in this case is:

> **Defining Synonyms**
> From $Sx_1 \dots x_n := Rx_1 \dots x_n$, infer: $\Phi[S]$ if and only if $\Phi[R]$.

Here is how it motivates **η-conversion**. Having issued a definition $\ulcorner Sx_1 \dots x_n := Rx_1 \dots x_n\urcorner$, S is then intersubstitutable with R (by **Defining Synonyms**) and also with $\ulcorner(\lambda x_1 \dots x_n.Rx_1 \dots x_n)\urcorner$ (by **Abstraction as Definition**). Cutting out the intermediary S, we have the relevant instance of **η-conversion**, licensing the intersubsitution of R and $\ulcorner(\lambda x_1 \dots x_n.Rx_1 \dots x_n)\urcorner$.

Defining Synonyms can in turn be motivated as follows. We can clearly introduce new words as synonyms for old ones. Sometimes we do this to guard against equivocation, as when the original word is ambiguous or context-sensitive. Other times we do so for expediency – for example, in introducing "T-Rex" as a shorter synonym for "*tyranosaurus rex*". These new words are meaningful in their own right. That is why speakers can become competent with "T-Rex" without even knowing the word "*tyranosaurus rex*". And the introduction of such words is of a piece with the practice of explicitly defining predicates. Given this parallelism, and the fact that stipulated synonyms like "T-Rex" and "*tyranosaurus rex*" are everywhere intersubstitutable, we should say the same about explicitly defined predicates. And that is what **Defining Synonyms** ensures: having explicitly defined S in terms of R, *à la* defining "T-Rex" in terms of "*tyranosaurus rex*", it says that S and R should likewise be everywhere intersubstitutable.[11]

[11] In first-order languages, **β-conversion** makes **η-conversion** (and hence **α-conversion**) redundant, since every instance of **η-conversion** is also an instance of **β-conversion**. This is because the only way that a predicate R can occur in a sentence of a first-order language is as the main predicate of a subformula $\ulcorner Ra_1 \dots a_n\urcorner$. In such contexts, intersubstituting R and $\ulcorner(\lambda x_1 \dots x_n.Rx_1 \dots x_n)\urcorner$, as licensed by **$\eta$-conversion**, is tantamount to intersubstituting $\ulcorner Ra_1 \dots a_n\urcorner$ and $\ulcorner(\lambda x_1 \dots x_n.Rx_1 \dots x_n)a_1 \dots a_n\urcorner$, as already licensed by **β-conversion**. But **η-conversion** is not redundant in higher-order languages, because in such languages predicates can occur in other syntactic positions. In particular, they can occur as arguments to higher-level predicates, as explained below.

2.2 Higher-order abstraction and open λ-terms

I will now consider two ways in which we can extend the use of predicate abstraction beyond what we already do in the predicate calculus. The first is to allow abstraction using non-individual variables. The second is to allow λ-terms in which not all variables free in the matrix clause are bound in the prefix. Anyone who is comfortable with the foregoing account of predicate abstraction should, I believe, have little trouble extending the practice in either of these ways.

Let's begin with an example. It is common to define the material conditional in terms of negation and conjunction. While sometimes $\ulcorner\varphi \supset \psi\urcorner$ is said to be mere shorthand for $\ulcorner\neg(\varphi \wedge \neg\psi)\urcorner$, often $\supset$ is introduced as a meaningful connective in its own right, defined in terms of $\neg$ and $\wedge$ but no less a genuinely meaningful expression for that. Similarly we might define possibility as the dual of necessity. We can do so using explicit definitions of the same general shape as in the predicate calculus (using infix notation where convenient):

$$p \supset q := \neg(p \wedge \neg q)$$

$$\Diamond p := \neg\Box\neg p$$

The only difference is that the variables occurring in these definitions are not individual variables (those taking the place of names), but rather sentential variables (those taking the place of sentences).

The point generalizes. For example, we can use variables occupying the position of binary predicate letters to explicitly define higher-order predicates that take binary predicate letters as arguments, as below:

$$\mathcal{S}R := \forall x\forall y(Rxy \leftrightarrow Ryx)$$

$$\mathcal{C}RS := \forall x\forall y(Rxy \leftrightarrow Syx)$$

Here "$\mathcal{S}$" and "$\mathcal{C}$" (for having "symmetric" and "converse" extensions) are the defined predicates, and "R" and "S" are variables in the syntactic position of binary first-order predicates (like "P" and "A" above).

The same considerations given in the last section with respect to first-order predicates apply here: so defined, "$\supset$", "$\Diamond$", "$\mathcal{S}$", and "$\mathcal{C}$" are meaningful symbols governed by **Definitional Equivalence**, **Alphabetic Irrelevance**, and **Defining Synonyms**. So by **Abstraction as Definition**, the corresponding λ-terms – "$(\lambda pq.\neg(p \wedge \neg q))$", and so on – obey β**-**, α**-**, and η**-conversion**.

So far I have been using some standard typographical conventions to signal the syntactic category of various symbols: "p" and "q" for sentential variables, "a" and "b" and "x" and "y" for individual constants and variables, "R" and "S" for variables in the position of binary first-order predicates, and fancier typefaces for terms taking such expressions as arguments. It is time to introduce a more systematic way of marking these syntactic categories.

We do so using the system of *simple relational types*. These types allow us to classify symbols according to how they can be grammatically combined. There are two basic types: the type of proper names, individual constants, and individual variables, which we call e, and the type of sentences and formulas, which we call t. Every non-empty sequence of types $\langle\tau_1,\ldots,\tau_n\rangle$ is also a type.[12] An expression of type $\langle\tau_1,\ldots,\tau_n\rangle$ can be thought of as an n-place predicate, which requires exactly n arguments respectively of types $\tau_1,\ldots,\tau_n$. So the predicate "F" defined in the last section has type $\langle e\rangle$, since predicating it consists in applying it to a single argument of type e; "$=$" has type $\langle e,e\rangle$, since it combines with two expressions of type e to form a formula; "$\mathcal{S}$" has type $\langle\langle e,e\rangle\rangle$; "$\mathcal{C}$" has type $\langle\langle e,e\rangle,\langle e,e\rangle\rangle$; "$\Diamond$" has type $\langle t\rangle$; "$\wedge$" has type $\langle t,t\rangle$; and so on. In general, the type of an expression introduced by explicit definition is the sequence of the types of the variables occurring in the definition.

So understood, type distinctions are both innocuous and familiar. When in first-order logic we distinguish the syntactic behavior of individual constants and predicates, or unary and binary predicates, or binary predicates and binary sentential connectives, we are already displaying a sensitivity to the distinctions between expressions of types e, $\langle e\rangle$, $\langle e,e\rangle$, and $\langle t,t\rangle$. Making these distinctions explicit obviates the need for conventions like using "p" for variables in sentence position and"x" for variable in name position. We instead indicate the type of a symbol with a superscript on its first occurrence, when it matters and it is not otherwise clear from context.

The second way in which predicate abstraction goes beyond our practice of explicitly defining predicates in first-order logic is allowing *open* λ-terms in which not all variables occurring free in the matrix are bound in the prefix, as in the following instance of **β-conversion**:

12. $\forall xRxx \leftrightarrow \forall x(\lambda y.Rxy)x$

Unlike the use of non-individual variables, allowing open λ-terms isn't easily motivated by pointing to a natural corresponding extension in our practice of explicit definition.[13] We allow such terms instead on the grounds that, once one gets used to the notation, they seem perfectly intelligible. Although open λ-terms are not essential to the topic of this chapter, as many of the metaphysical questions discussed in sections 4 and 5 can be posed without them, we allow

[12] Most work in higher-order logic operates instead with a system of *functional types*. In such systems, the relational type $\langle e,e\rangle$ corresponds to the functional type $(e\to(e\to t))$, $\langle\langle t\rangle\rangle$ corresponds to $((t\to t)\to t)$, and so on; see Fritz and Jones (this volume) for the details. The syntax of functionally typed languages is different from relationally typed languages in that polyadic application and abstraction happen one argument or variable at a time rather than all at once. Fortunately, this difference doesn't matter for present purposes. This is because we can treat one-argument-at-a-time application and one-variable-at-a-time abstraction as shorthand for the sequential formation of certain complex λ-terms; given **β-conversion** and **α-conversion**, the result of doing this is everywhere intersubstitutable with application and abstraction as defined here; see Dorr (2016, appendices 1 and 2) for the technical details.

[13] This is because doing so would require issuing definitions like $\ulcorner Gx := x = y\urcorner$. If we maintained **Definitional Equivalence** for G, then $\ulcorner Ga\urcorner$ should be everywhere intersubstitutable with $\ulcorner a = y\urcorner$, even within the scope of $\forall y$. Such substitutivity would be as if y was covertly free in the atomic symbol G, despite not occurring in it. There is no parallel awkwardness with λ-terms and **β-conversion**.

such terms because forbidding them is awkward and unnatural.[14] Below I will assume that open λ-terms are in as good standing as closed ones, and similarly obey β**-**, α**-**, and η**-conversion**.

We can now be more precise and systematic about how terms can be combined to form more complex ones:

> **Application**: If R and $a_1, \ldots, a_n$ are respectively terms of types $\langle \tau_1, \ldots, \tau_n \rangle$ and $\tau_1, \ldots, \tau_n$, then $\ulcorner Ra_1 \ldots, a_n \urcorner$ is a term of type t.
>
> **Abstraction**: If φ is a term of type t and $x_1, \ldots, x_n$ are n distinct variables free in φ respectively of types $\tau_1, \ldots, \tau_n$, then $\ulcorner(\lambda x_1 \ldots x_n.\varphi)\urcorner$ is a term of type $\langle \tau_1, \ldots, \tau_n \rangle$ in which none of $x_1, \ldots, x_n$ is free.

I have explained this use of λ-abstraction as an outgrowth of our familiar practice of explicit definition in order to explain how such λ-terms can be understood and why they should be governed by β**-**, α**-**, and η**-conversion**. But nothing I have said so far has suggested any *point* in introducing λ-terms. Indeed, the kinship with definitions might make one suspicious that they could serve any purpose. Applying defined predicates is always in principle eliminable in favor of those predicates' definitions. When this is impossible in practice, it is because a hierarchy of definitions has allowed for the compact expression of something that would be too cumbersome to write down were all the definitions unpacked. Yet since λ-terms are longer than their matrix clauses, they seem to sacrifice the main benefit of definitions – namely, economy of symbolism.

λ-terms are indeed rather pointless as an addition to first-order languages. But the situation is different in higher-order languages, in which we can quantify into predicate positions, and hence both instantiate generalizations with λ-terms and have λ-terms occur as arguments to bound higher-order variables. This is the topic of the next section.

3 Higher-order quantification

This section first introduces higher-order quantification. It then considers how higher-order quantification is related to the existence of such abstract objects as propositions, properties, and relations. I argue, first, that the intelligibility of higher-order quantification does not presuppose the existence of abstract objects; second, that such quantification is not equivalent to claims about abstract objects; and, third, that higher-order quantification gives us some reason to doubt the existence of such abstract objects.

[14] Such terms are needed for a Fregean treatment of quantifiers as higher-order predicates (explained in section 3.4) in order to handle nested quantification. But if we instead opt for an orthodox treatment of quantification using variable-binding operators, then the most important arguments in higher-order metaphysics, even those such as the Russell-Myhill argument that essentially involve λ-terms, do not feature open λ-terms; see Goodman (2017).

3.1 Universal Instantiation and Comprehension

In first-order logic, the orthodox notation for expressing generality involves two infinite sets of expressions: individual variables $x_1, x_2, \ldots$, and a stock of corresponding *quantifier prefixes* $\forall x_1, \forall x_2$, etc. If φ is a formula, then $\ulcorner \forall x_i \varphi \urcorner$ is another formula, in which all free occurrence of x_i in φ are bound by $\forall x_i$.

The central principle of quantificational logic is the following axiom schema (which corresponds to the rule of $\forall$-elimination in natural deduction systems):

> **Universal Instantiation**
> $\forall x \varphi \to \varphi[a/x]$

While what counts as an instance of this schema depends on what formulas, variables, and terms are included in our language, a commitment to such schemas is typically taken to be *open ended*, in the sense that we do not need to revisit it when new vocabulary is introduced into our language and the schema acquires new instances as a result; see McGee (1997).

Let us now move from first-order to higher-order languages. In introducing predicate abstraction in the last section, we have already tacitly helped ourselves to an infinite stock of variables $x_1^\tau, x_2^\tau, \ldots$ for every type τ. As in first-order logic, let us now extend our language with corresponding quantifier prefixes $\forall x_1^\tau, \forall x_2^\tau$, etc. **Universal Instantiation** thereby gains new instances, not only because the language now contains new formulas which can be substituted for φ, but also because variables and terms of types other than e can now be substituted for x and a. Here are two examples:

13. $\forall p^t (p \to p) \to (Rab \to Rab)$
14. $\forall R^{\langle e,e \rangle} (\mathcal{C}R(\lambda xy.Ryx) \leftrightarrow \mathcal{S}R) \to$
$$(\mathcal{C}(\lambda xy.Zyyx)(\lambda xy.(\lambda xy.Zyyx)yx) \leftrightarrow \mathcal{S}(\lambda xy.Zyyx))$$

Crucially, higher-order variables can be instantiated with complex terms, both sentences and λ-terms.

To readers new to higher-order quantification and wondering how to understand it, my advice is to leverage your antecedent understanding of first-order quantification. Formally, higher-order quantification is of a piece with first-order quantification – e.g., in the centrality of **Universal Instantiation** to quantificational reasoning. It is no accident that modern quantification theory, as first developed by Frege (1879) and as deployed by many of his contemporaries, was as part of a higher-order formal language. It is also no accident that we have to explicitly instruct beginning logic students *not* to quantify into sentence and predicate positions – the impulse to do so comes very naturally once one has understood first-order quantification.

I will not give any direct argument for the intelligibility of higher-order quantification – not because I have any doubts about its good standing, but because intelligibility is, in general, a hard thing to be convinced of by direct argument.[15] To readers who feel unsure whether higher-order quantification makes sense, I suggest simply trying it on for size. (Later I will explain why some common reservations about its good standing are misguided when higher-order languages are understood in the way I have motivated here.)

We can define existential quantification to be the dual of universal quantification, so that $\ulcorner \exists X \varphi \urcorner$ abbreviates $\ulcorner \neg \forall X \neg \varphi \urcorner$. **Universal Instantiation** is then equivalent to:

> **Existential Generalization**
> $\varphi[a/x] \to \exists x \varphi$

Together with **β-conversion**, we can then derive:

> **Higher-Order Comprehension**
> $\exists Z \forall x_1 \ldots \forall x_n (Z x_1 \ldots x_n \leftrightarrow \varphi)$, where $x_1, \ldots, x_n$ but not Z are free in φ

The derivation is simple, and highlights the importance of being able to instantiate higher-order generalizations with λ-terms:

1. $\forall x_1 \ldots \forall x_n (\varphi \leftrightarrow \varphi)$ [uncontroversial quantification theory]
2. $\forall x_1 \ldots \forall x_n ((\lambda x_1 \ldots x_n . \varphi) x_1 \ldots x_n \leftrightarrow \varphi)$ [1, by **β-conversion**]
3. $\exists Z \forall x_1 \ldots \forall x_n (Z x_1 \ldots x_n \leftrightarrow \varphi)$ [2, **Existential Generalization**]

Comprehension plays a starring role in many presentations of higher-order logic.[16] In fact, many systems of higher-order logic omit λ-terms altogether in favor of comprehension principles. We will see in the next two sections that this is sufficient for some applications and insufficient for others. But even in applications where we can get by with only **Higher-Order Comprehension**, λ-terms are still important. That is because the above argument provides the firmest justification for accepting **Higher-Order Comprehension** to begin with, and λ-terms feature essentially in that derivation.

The next two subsections addressed the question of how higher-order quantification is related to first-order quantification over propositions, properties, and relations. Section 3.2 shows that the most obvious reason to think the former somehow involves the latter in fact shows nothing of the kind. Section 3.3 argues that, whatever connections there may be between higher-order quantification and the existence of properties, those connections in no way call into question the good standing of higher-order quantification on the understanding advanced above; if anything, higher-order quantification undermines the primary consideration in favor of positing any such abstract objects to begin with.

[15] "We must learn to use higher-order languages as our home language. [...But w]hat we are willing to take as our home language is partly a matter of what we feel comfortable with; unfortunately, it can be hard to argue someone into feeling comfortable." (Williamson, 2003)

[16] This is usually in the form of the stronger principle:

> **Full Higher-Order Comprehension**
> $\forall y_1 \ldots y_m \exists Z \forall x_1 \ldots \forall x_n (Z x_1 \ldots x_n \leftrightarrow \varphi)$, where Z is not free in φ

To derive this one needs the other standard axioms of quantification theory, such as the distribution of universal quantification over material implication and the closure of one's axioms under the rule of universal generalization.

3.2 "Propositions", "properties", and pronunciation

As explained in section 2.2, type distinctions are nothing new to higher-order languages. Already in first-order logic we need to distinguish expressions on the basis of both the number of terms they take as arguments and the syntactic categories of those arguments. What is new in higher-order languages is simply that it becomes more urgent to keep track of these type distinctions explicitly, especially once we start deploying higher-order quantification. We have seen how types can be indicated in formulas by superscripts. But in speech and prose, something else is called for.

We can take our cue from quantification in first order languages, where we use glosses like 8 and 9 to fluently but unambiguously pronounce formulas. We can extend this pronunciation scheme to type distinctions as follows. We will continue to pronounce quantifier prefixes using "for all ... such that"/"there is a ... such that", but rather than pronouncing the variable in the position of a noun restricting the determiners "all" and "some", as in 8 and 9, it will instead accompany a noun indicating the variable's type: the case of type e, "individual"; in the case of type t, "proposition"; in the case of type $\langle\langle e\rangle\rangle$, "property of properties of individuals"; in the case of type $\langle e, t\rangle$, "relation between individuals and propositions"; and so on. Here are some examples:

15. There is a proposition p such that necessarily p.

    $\exists p \Box p$

16. For all individuals x there is a binary relation between individuals R such that Rxx.

    $\forall x^e \exists R(Rxx)$

We can extend this pronunciation strategy to variables bound by λ-terms. This is best illustrated by example:

17. a is an individual x such that x is identical to x.

    $(\lambda x.x = x)a$

18. Being a proposition p such that p and possibly not-p is a property of propositions O such that some proposition q is such that Oq.

    $(\lambda O.\exists qOq)(\lambda p^t.p \land \Diamond\neg p)$

19. There is an entity y of type τ such that y and P are an x and F such that Fx.

    $\exists y^\tau((\lambda xF.Fx)yP)$

Where λ-terms occur in *predicate position* – that is, as the main predicate of a (sub)formula – we use "is(/are) an ... (and ...) such that", where the gaps are filled by the relevant variables accompanied by the nouns indicating their types. If the type of an expression is unspecified, we may simply speak of "entities" of that type, as in 19. Where λ-terms occur in *argument position* – that is, as arguments to predicates of higher types, as in 18 – we use "being" rather than "is(/are)". Just as we may refrain from writing type annotations when they can be inferred from context, we do the same in these pronunciations, as in 19.

This strategy for articulating formulas in speech and prose should be sharply distinguished from postulating an ontology of properties, propositions, relations, and other (presumably abstract) individuals of different kinds. The aforementioned pronunciations carry no such commitment for the same reason that 8 and 9 carry no commitment to special kinds of things, the xs and the ys.

3.3 Properties and paradox

Quine (1970) influentially disparaged second-order quantification (and, by extension, higher-order quantification) as at best "set theory in sheep's clothing". If he had believed in properties, he would have called it property theory in sheep's clothing. What could he have meant by this?

The most natural way of understanding his suggestion is that formulas of the form $\ulcorner \exists X^{\langle e \rangle} \Phi[Xa] \urcorner$ are equivalent to corresponding formulas of the form $\ulcorner \exists x^e (x \text{ is a property } \wedge\ \Phi[a_i \text{ instantiates } x] \urcorner$. The main problem with this idea has to do with Russell's paradox: there are obvious truths involving quantification into predicate position which would be equivalent to contradictions if higher-order quantification were equivalent to first-order quantification over properties in the aforementioned way. It is worth rehearsing this argument, since it allows us to see many of the principles mentioned above in action.

$Ixy := y$ instantiates x

$Px := x$ is a property

1. $\forall x^e(\neg Ixx \leftrightarrow \neg Ixx)$ [first-order logic]
2. $\forall x(\neg Ixx \leftrightarrow (\lambda y.\neg Iyy)x)$ [1, **β-conversion**]
3. $\exists F^{\langle e \rangle} \forall x(\neg Ixx \leftrightarrow Fx)$ [2, **Existential Generalization**]
4. $\exists f^e(Pf \wedge \forall x(\neg Ixx \leftrightarrow Ifx))$ [3, Quinean 'sheep's clothing' thesis]
5. $\neg \exists f^e \forall x(\neg Ixx \leftrightarrow Ifx)$ [first-order logic (Russell's theorem)]
6. $\bot$ [4,5]

So higher-order quantification cannot be disguised first-order quantification over properties, on pain of inconsistency. If there were powerful arguments for Quine's sheep's clothing thesis, then we might consider giving up **Universal Instantiation**, **β-conversion**, or even classical first-order logic in order to accommodate it. But in the absence of such arguments, we should simply see the inconsistency as a reductio of Quine's contention.

Let us now consider some further connections between higher-order logic and Russell's paradox. Russell showed that the following schema is inconsistent:

Naïve Comprehension
$\exists x \forall y (Ixy \leftrightarrow \varphi)$, where y is not free in φ

Now consider the following argument against the existence of properties:

P1 If there are properties, then every instance of **Naïve Comprehension** is true.

P2 $\ulcorner \exists x \forall y (Ixy \leftrightarrow \neg Ixx) \urcorner$ is not true, since it is classically inconsistent.

C Therefore, there are no properties.

This is one of the strongest arguments against the existence of properties. It is much stronger than the parallel argument against the existence of sets. For we have a highly developed, extremely fruitful, and presumably consistent theory of sets, namely Zermelo-Fraenkel set theory. This theory can in turn be motivated by the iterative conception of sets, which provides a principled basis for denying that sets are ever members of themselves, and hence that there is a set of all sets, and hence against accepting the set-theoretic analogue of **Naïve Comprehension** (with $\in$ in place of I); cf. Boolos (1971). The situation with properties is completely different. No analogue of Zermelo-Fraenkel set theory has even achieved widespread attention, let alone consensus. This is in part because we seem to lack any intuitive conception of properties that either prohibits them from instantiating themselves or restricts self-application in a principled yet consistent way.

This point is important because higher-order quantification arguably allows us to accomplish most of the work that first-order quantification over properties is typically invoked to do in philosophy, but without an explicit commitment to any such ontology; cf. Prior (1971) and Bacon (this volume). This in turn gives us reason to take the above argument against the existence of properties more seriously, since **Higher-Order Comprehension**, unlike **Naïve Comprehension**, is perfectly consistent and is not threatened by Russell-style reasoning.[17]

To be clear, I am not claiming that type distinctions are motivated by the paradoxes of self-application. Many writers on higher-order logic (both its proponents and detractors) have claimed this, and I think it is a mistake. The canonical case is Russell (1908), who proposed syntactically stratifying expressions standing for properties in such a way that claims of self-instantiation are deemed ungrammatical. The inconsistency of **Naïve Comprehension** would then be avoided by rendering its problematic instances ill-formed.[18] Like many others, including Button and Trueman (this volume, §2.2), I find this purported resolution to the paradoxes rather galling: when faced with an argument from

[17]There remain hard questions about how English nominalizing devices like "the property of being" and generalizations of the form "for every property . . . " are related to λ-terms and higher-order quantification; see Button and Trueman (this volume). But the intelligibility of the latter is not hostage to an account of how they are related to the former. For λ-terms and higher-order quantification have been explained here as an extension of our understanding of first-order logic, and not in terms of English constructions using the word "property".

[18]Formulas and complex predicates are also stratified into levels to block related 'intensional' paradoxes such as the Russell-Myhill argument; see Klement (this volume) for discussion.

plausible premises to a contradiction, we don't normally think that one available response is to conclude that some of the argument's premises are *ungrammatical*.

Fortunately, we needn't settle this issue here. For whatever we think about the legitimacy of Russell's type-theoretic response to the paradoxes, it has no bearing on the status of type distinctions as *we* have been deploying them. Our type distinctions were motivated independently of the paradoxes, simply to keep track of syntactic distinctions that arise organically and inevitably from the predicate calculus if we allow ourselves to iteratively introduce new expressions using explicit definitions. Russell's type distinctions, by contrast, amount to inventing syntactic distinctions and prohibitions that we have no antecedent reason to believe in.

The independence of type distinctions from the paradoxes, and more generally the independence of higher-order quantification from Platonist ontology, is also reflected in the historical development of higher-order logic. For Frege (1879, §11) was already sensitive to the importance of type distinctions (although he didn't discuss them as explicitly as we have here), decades before the paradoxes were even discovered. And his quantification theory was higher-order from the start; only later, especially in Frege (1951), did he wrestle in earnest with how it related to his preferred Platonist ontology.

3.4 Separating variable binding and generalization

This subsection describes two other ways of formalizing quantification in higher-order languages, which are common in the literature and which illuminate some of what can be done with λ-terms. It can be skipped without loss of continuity.

While the notation of quantifier prefixes may be the most readily intelligible, at least for those of us raised on first-order logic, more elegant options are available in languages with predicate abstraction. Here is one: rather than having infinitely many symbols $\forall x_i^\tau$ that can combine with formulas, we can instead have a single symbol $\forall$ that combines with monadic predicates. The canonical case will be predicates formed by predicate abstraction. So instead of $\ulcorner \forall x_i^\tau \varphi \urcorner$, we first form the λ-term $\ulcorner (\lambda x_i^\tau . \varphi) \urcorner$, which is a term of type $\langle \tau \rangle$, and then prefix it with $\forall$ to form the formula $\ulcorner \forall (\lambda x_i^\tau . \varphi) \urcorner$. Since this is more cumbersome to read (especially because embedded quantification proliferates parentheses), it is convenient to maintain the notation of quantifier prefixes as convenient shorthand, even if we officially adopt the notation just described.

There are a few ways in which this new notation is more elegant. One is that it involves only one primitive quantifier symbol, where before (even in first-order logic) we had infinitely many. A second inelegance of quantifier prefixes is that, although they have similar syntactic behavior to terms like $\neg$ of type $\langle t \rangle$, they cannot be classified as terms. This is because doing so would lead to violations of **Universal Instantiation**, as the following false sentence illustrates:

20. $\forall Z^{\langle t \rangle} \forall x \forall y (x = y \to Z(x = x) \to Z(x = y)) \to$
    $\forall x \forall y (x = y \to \forall x (x = x) \to \forall x (x = y))$

In the new notation, there is less temptation to treat $\forall$ as a term, since there is no type that it could be a term of. Its syntactic behavior is more promiscuous than that of any term, since it can combine with terms of infinitely many different types to form a formula (i.e., with terms of type $\langle\tau\rangle$ for every type τ).

A third advantage of the new notation is that it symbolically separates what are naturally seen, already by Frege (1879, §§10-11), as distinct ideas: variable binding, in the form of the formation of complex predicates, and generalization, in the sense of forming formulas using quantifiers.[19] This separation allow us to replace **Universal Instantiation** with a simpler axiom schema:

UI: $\forall F \to Fa$

Universal Instantiation follows from **UI** given **β-conversion**: $\ulcorner\forall x\varphi\urcorner$ abbreviates $\ulcorner\forall(\lambda x.\varphi)\urcorner$, which together with the relevant instance of **UI** yields $\ulcorner(\lambda x.\varphi)a\urcorner$ (by modus ponens), which is materially equivalent to $\varphi[a/x]$, given the relevant instance of **β-conversion**, provided a is free for x in φ.

Let us now consider a third notation for higher-order quantification. Like the one just introduced, it hives off the role of variable binding to predicate abstraction. But it is also impressed by the idea that generalization is not a *sui generis* syntactic operation for making formulas from predicates, but is rather a kind of *predication*. This idea, too, goes back to Frege (2013, §31) – in particular, to his claim that quantifiers are higher-order predicates.

Formally, the proposal introduces a typed family of quantifiers: for every type τ, there is a universal quantifier $\forall_\tau$ of type $\langle\langle\tau\rangle\rangle$. **UI** must then be modified accordingly, so that the relevant quantifier is schematic on the type of F:

UI: $\forall_\tau F^{\langle\tau\rangle} \to Fa$

This allows us to have a language with quantification in which all formulas are built from basic terms using only **Application** and **Abstraction**.

We can compare the three notations as follows:

Quantifier prefixes: $\forall x^\tau \forall y^\sigma \varphi$

Syncategorematic quantifier: $\forall(\lambda x^\tau.\forall(\lambda y^\sigma.\varphi))$

Categorematic quantifiers: $\forall_\tau(\lambda x^\tau.\forall_\sigma(\lambda y^\sigma.\varphi))$

It is fairly straightforward to move back and forth between these notations, and the differences between them won't matter in what follows.[20] While I will use the notation of quantifier prefixes below, in doing so I remain neutral on whether that notation is basic or abbreviates a quantifier combining (either syncategorematically or by **Application**) with a λ-term.

[19] On the evolution of Frege's conception of this separation, see Heck and May (2013).

[20] For example, we can simulate the third approach within the second by defining $\forall_\tau$ as $(\lambda X^{\langle\tau\rangle}.\forall X)$. Two small differences between the first approach and the second and third are worth noting: (i) we have required that x occur free in φ for any λ-term $\ulcorner(\lambda x.\varphi)\urcorner$ (a point we will return to in section 5.2); but it it not standardly required that a quantifier prefix attach only to formulas in which the corresponding variable is free (although Frege (1879, §11) did require this); (ii) while non-vacuous generalization using quantifier prefixes always requires binding free variables, as in $\ulcorner\forall xFx\urcorner$, on the second and third proposals quantifiers can combine directly with monadic predicate constants and variables, as in $\ulcorner\forall_\tau F^{\langle\tau\rangle}\urcorner$.

4 Grain science

Higher-order quantification both allows us to ask interesting new metaphysical questions and to ask old questions better than before. Many of these new questions can be asked in *pure higher-order logic*, using only negation, conjunction, quantification, and predicate abstraction. In this way pure higher-order logic is unlike first-order logic with identity (since, as discussed in section 1, the questions that can be asked using only negation, conjunction, first-order quantification, and identity are limited and boring).

This is true already in the *second-order fragment* of pure higher-order logic, in which we only use variables of types e, $\langle e\rangle$, $\langle e, e\rangle$, $\langle e, e, e\rangle$, etc. For example, we can ask whether any binary relation among individuals well-orders all individuals, whether analogues of the generalized continuum hypothesis are true, and similar questions; cf. Shapiro (1991).

Higher-order quantification is also invaluable in metaphysics more broadly. To take one example, many questions about modal ontology are best adjudicated in a higher-order setting, as in Fine (1977), Williamson (2013), Fritz and Goodman (2016), Bacon (2018), Dorr et al. (2021), and Fairchild (this volume).

The aim of this section is to introduce a family of questions in pure higher-order logic that illustrate the fruitfulness of higher-order quantification at all types, and not merely the second-order fragment. These are questions about the granularity of reality – for short, questions in *grain science*.

Here is one way of explaining grain science, largely following Dorr (2016). Recall that the identity predicate of first-order logic is closely related to certain uses of "is" in English, as in "Mark Twain is Samuel Clemens". Now consider the related use of "is" where it combines with infinitival verb phrases rather than with proper names, as in "to be a bachelor is to be an unmarried man", "to be made of water is to be made of H_2O molecules", and "to walk is to stride while keeping one foot on the ground". Following Dorr (2016), I will call these sentences *identifications*. The "is" in these identifications feels a lot like the "is" of ascriptions of numerical identity. But the fact that it combines with a pair of verb phrases rather than with a pair of names, and verb phrases are naturally regimented as terms of type $\langle e\rangle$, means that it is better regimented as a term of type $\langle\langle e\rangle, \langle e\rangle\rangle$, and so not as the identity sign "=", which has type $\langle e, e\rangle$.

We can also make sense of identifications involving expressions of other types. In "for one thing to be a subset of another is for it to be a set all of whose members are members of the other", the part before the "is" is naturally regimented as the symbol $\subseteq$, which has type $\langle e, e\rangle$. And in "for it to be not not raining is for it to be raining", the part before the "is" is naturally regimented as "it is not not raining", which has type t. So regimenting these sentences seems to call for identification predicates respectively of types $\langle\langle e, e\rangle, \langle e, e\rangle\rangle$ and $\langle t, t\rangle$.

While it is not clear how far we can stretch English to extend this practice, it is natural, reflecting on these examples, to posit a typed family of identification predicates $\equiv_\tau$ of type $\langle\tau, \tau\rangle$, one for each type τ. Dorr argues that we can understand this family of predicates well enough to be confident that each is governed by principles analogous to those governing numerical identity, namely:[21]

Reflexivity$_\equiv$: $a \equiv_\tau a$

Substitution$_\equiv$: $a \equiv_\tau b \to (\varphi \leftrightarrow \varphi[b/a])$

Grain science is the study of further principles governing identifications: *what can we say in general about which identifications hold and which do not?*

Here is a second, complementary way into the topic of grain science, which shows both how it can be understood by anyone who accepts **Higher-Order Comprehension** and that it should not be conflated with a Platonist investigation into the individuation of propositions, properties, and relations, understood as a kind of abstract individuals. Rather than positing a family of identification predicates $\equiv_\tau$ whose interpretation is anchored in part by identifications in English and in part by a formal analogy with the identity sign from first-order logic, we can use pure higher-order logic to explicitly define a typed family of 'Leibniz equivalence' predicates $\approx_\tau$ as follows:

$$x \approx_\tau y := \forall Z^{\langle\tau\rangle}(Zx \leftrightarrow Zy)$$

Informally, $x \approx y$ just in case *there is no difference* between x and y – i.e., there is no Z that differentiates them.

So defined, $\approx$ provably has the logical behavior of identity:

Reflexivity: $a \approx_\tau a$

Substitution: $a \approx_\tau b \to (\varphi \leftrightarrow \varphi[b/a])$

Reflexivity should be obvious, and is derivable in any higher-order quantification theory that includes classical logic and is closed under the standard rule of universal generalization (if $\vdash \varphi[a/x]$, then $\vdash \forall x\varphi$, where a is free for x in φ). **Substitution** is derivable from **Higher-Order Comprehension** and **Universal Instantiation**.[22]

Grain science can then be understood as the study of Leibniz equivalence: *what can we say in general about which Leibniz equivalences hold and which do not?* This is not to say that identification and Leibniz's equivalence are the same thing, in the sense of the following principle from Bacon and Russell (2019):

The Identity Identity: $\equiv_\tau \equiv_{\langle\tau,\tau\rangle} \approx_\tau$

[21] He also argues, convincingly in my view, that many central questions about the nature or analysis of notions of interest in philosophy and beyond (goodness, causation, meaning, etc.) are best understood as questions whose candidate answers are identifications.

[22] *Proof sketch*:

1. $\exists Z\forall x(Zx \leftrightarrow \varphi)$ [**Higher-Order Comprehension**]
2. $\forall x\forall y(\forall Z(Zx \leftrightarrow Zy) \to (\varphi \leftrightarrow \varphi[y/x]))$ [1, uncontroversial quantificational reasoning]
3. $a \approx b \to (\varphi \leftrightarrow \varphi[b/a])$ [2, **Universal Instantiation**, definition of $\approx$]

It is sufficient that $\approx$ and $\equiv$ be coextensive, since the point of grain science, as conceived by Dorr and others, is to generalize about what identifications/Leibniz equivalences are true, not about the nature of identification itself.[23] I will use $\approx$ hereafter simply to highlight the power of *pure* higher-order logic as metaphysics.

Given **Reflexivity** and **Substitution**, the logical behavior of $\approx_\tau$ is sufficiently identity-like that those working in grain science typically pronounce it "is", or even "is the same ... as" (with the gap filled by a noun indicating the type τ). But this innocent use of "is" among the cognoscenti simply to pronounce $\approx$ can lead to unfortunate misunderstandings. Consider, for illustration, the following controversial principle about reality's granularity:

Involution: $\forall p(p \approx_t \neg\neg p)$

Someone who overhears "Every proposition p is such that p is not not p" or "For all p, p is the same proposition as not not p" would be forgiven for thinking that what was at issue was a question about the individuation of *propositions*, understood as a certain kind of abstract object. But they would be mistaken. As discussed in the last section, this use of "proposition" is not to specify a kind of abstract individuals, but merely to fluently convey type distinctions in English rather than in symbols (in this case, the fact that "p" has type t).

Here is one reason why it matters that questions in grain science aren't questions of how certain kinds of abstract individuals are individuated. A natural reaction to the question "How are propositions individuated?" is that it depends what we mean by "proposition". For a common view among Platonists is that there are different families of abstract objects one might call "propositions", each individuated in different ways, and that what kind of "proposition" (or, if understood in one of the coarser ways, "states of affairs", "intensions", or even "truth values") is worth theorizing about is largely a pragmatic matter, which will differ depending on one's projects.

Once we distinguish grain science, in the sense of questions formulated in terms of $\approx$, from similar sounding Platonist investigations, the apparent option of pluralism owing to an ambiguity in "proposition" evaporates. For the operative use of "proposition" is, in effect, a parenthetical clarification about the syntactic category of a variable in a sentence of a formal language. Such sentences contain neither the word "proposition" nor any analogue thereof, since they are sentences of pure higher-order logic.[24]

To give a sense of grain science's potentially widespread metaphysical implications, here is a simple example. In the last decade there has been an explosion of work on the idea that some truths are true in virtue of others. Following Fine (2012), it is common to formalize this notion of 'grounding' using a binary sentential connective "$\prec$", and to endorse the following two principles:

[23] Moreover, even if **Substitution**$_\equiv$ is rejected, this does not threaten the interest or implications of grain science understood in terms of Leibniz equivalence; see Caie et al. (2020).

[24] See Dorr (2014) for discussion of whether grain scientific questions might be ambiguous in a different way, owing to different possible interpretations of higher-order quantification. For more on how to isolate the intended 'unrestricted' reading of both first-order and higher-order quantification, see Williamson (2003) and Goodman (2016, §5).

21. $\varphi \to (\varphi \prec \neg\neg\varphi)$

22. $\neg(\varphi \prec \varphi)$

Informally: every truth is something in virtue of which its double negation is true, and nothing is true in virtue of itself. Taken together, these two principles are inconsistent with **Involution**: if $p \prec \neg\neg p$ but $\neg(p \prec p)$, then there is not *no* difference between p and $\neg\neg p$, because being a q such that p grounds q differentiates $\neg\neg p$ from p. So if **Involution** is accepted, grounding orthodoxy must be rejected.[25]

5 λ-terms in higher-order metaphysics

This section explores the implications of **β-conversion** for questions in grain science. I begin by arguing that we must reject the most natural higher-order articulation of the widely held doctrine that propositions have something like syntactic structure. I then argue, more constructively, that doing makes room for an attractive account of the idea that propositions are nevertheless singular, in the sense of having certain individuals and not others as essential ingredients. However, the tenability of this account turns on subtle questions to do with the logic and interpretation of vacuous λ-terms, such as $\ulcorner(\lambda x.p)\urcorner$, if we enrich our language with such terms.

5.1 Against Structure

Many philosophers are attracted to the view that propositions are structured in the image of the sentences that express them. We can understand this structured picture as a kind of grain science, so that its implications can be formulated in pure higher-order logic. In particular, consider:

Structure

$$\forall R \forall S \forall x_1 \dots \forall x_n \forall y_1 \dots \forall y_n (Rx_1 \dots x_n \approx Sy_1 \dots y_n \to R \approx S \wedge x_1 \approx y_1 \wedge \dots \wedge x_n \approx y_n)$$

The idea is that, just as the terms $R, a_1, \dots, a_n$ can be recovered by decomposing the sentence $\ulcorner Ra_1 \dots a_n \urcorner$ into its immediate constituents, reality has a kind of 'logical structure' whereby 'propositions' have similarly unique decompositions.

Despite its considerable appeal, **Structure** is inconsistent. In particular, an argument essentially due to Russell (1903) shows that the instance of **Structure** where R and S have type $\langle t \rangle$ is inconsistent given classical propositional logic, **Universal Instantiation**, and the following weakening of **β-conversion**:[26]

[25] Dorr (2016) argues that even defenders of otherwise 'fine-grained' views should accept **Involution**. Fritz (forthcoming) explores other tensions between grounding orthodoxy and granularity considerations.

[26] This discovery is sometimes called the 'Russell-Myhill paradox', due to its independent discovery by Myhill (1958). Although Russell was thinking of propositions as abstract individuals, his argument is easily transposed to a higher-order setting (unlike 'Russell's paradox', establishing the inconsistency of **Naïve Comprehension**, which dissolves when reformulated in higher-order terms); see Uzquiano (2015), Dorr (2016), and Goodman (2017) for discussion.

β-equivalence $(\lambda x_1 \ldots x_n.\varphi)a_1 \ldots a_n \leftrightarrow \varphi[a_i/x_i]$

This argument is dialectically important because many philosophers attracted to **Structure** are skeptical of **β-conversion** but not of **β-equivalence**.

But given **β-conversion**, there are simpler and stronger arguments against **Structure**. Here is one: for a to be identical-to-a is for a to be self-identical, but it is not the case that to be identical-to-a is to be self-identical. More formally:

1. $a = a \approx a = a$ [**Reflexivity**]
2. $(\lambda x.x = x)a \approx (\lambda x.a = x)a$ [1, **β-conversion**]
3. $b = b \wedge \neg(a = b)$ [premise – a and b may be any distinct individuals]
4. $(\lambda F.Fb)(\lambda x.x = x) \wedge \neg(\lambda F.Fb)(\lambda x.a = x)$ [3, **β-conversion**]
5. $\neg((\lambda x.x = x) \approx (\lambda x.a = x))$ [4, **Existential Generalization**, def. $\approx$]
6. $\exists R \exists S \exists x^e(Rx \approx Sx \wedge \neg(R \approx S))$ [2, 5, **Existential Generalization**]
7. $\neg\forall R \forall S \forall x^e \forall y(Rx \approx Sx \rightarrow R \approx S \wedge x \approx y)$ [6]

The conclusion of this argument is the negation of the instance of **Structure** where R and S have type $\langle e \rangle$: one and the same proposition can be decomposed into the application of a property to an individual in more than one way.[27] Assuming the earlier case for **β-conversion** and **Universal Instantiation** was successful, this argument is sound and we must reject **Structure**.

This doesn't mean that reality cannot have language-like logical structure in any sense. After all, there are formal languages in which sentences can be decomposed into applications of predicates to arguments in more than one way.[28] That being said, the above mode of argument is quite general. It remains to be seen whether a consistent and principled articulation of a structured vision that respects **β-conversion** can be developed.

5.2 Objectual aboutness: singularity without structure

This section considers the prospects of using pure higher-order logic to make sense of the idea that propositions have particular ingredients or constituents.

While this idea is intuitive, it also tends to be lumped together with the kind of structured picture discussed in the last section. So the question is whether,

[27] This argument generalizes to all instances of **Structure**. By contrast, the Russell-Myhill argument only generalizes to certain types, since the instances of **Structure** where the type of R and S nowhere involves t are consistent with **β-equivalence** and **Universal Instantiation** in the absence of **β-conversion**; see Goodman (forthcoming).

[28] These include Pierce's (1931-1958) graphical notation for first-order logic, the Lambek (1958) calculus, and the 'structural calculus' of Bacon (forthcoming). Inspired by the latter, Bacon argues that the best strategy for defending a structured picture of reality is to accept **Structure**, **β-conversion** and **Universal Instantiation** but reject as meaningless a wide swath of λ-terms, including $\ulcorner(\lambda F.Fa)\urcorner$ and $\ulcorner(\lambda x.x = x)\urcorner$. In my view, the strongest consideration against this strategy is that, given **Abstraction as Definition**, it requires positing an implausible parallel barrier to introducing meaningful predicates by explicit definition.

having rejected **Structure**, we can still make sense of a discriminating and useful notion of logical constituents/ingredients.

The answer isn't obvious. At a purely metaphorical level, there are all sorts of ways of talking about constituents and ingredients that don't involve quasi-syntactic structure: these nails but not those screws are constituents of this table; eggs are ingredients of challah but not of sorbet; etc. Moreover, the bearing of **β-conversion** on the tenability of a robust notion of constituents/ingredients turns out to be subtler than in the case of the structure. On the one hand, it suggests a natural account of those notions in purely logical terms, which wouldn't be available on a more 'structured' picture; on the other hand, a natural extension of **β-conversion** threatens to trivialize that account.

I will begin by motivating a simple, grain-theoretic necessary and sufficient condition for a given individual to be a constituent of a given proposition. I will then explain how the non-triviality of this account depends on accepting as much **β-conversion** as we have so far accepted *and no more*. In particular, the account presupposes the *illegitimacy* of both enriching our higher-order language with vacuous λ-terms and accepting **β-conversion** for such terms. I will argue finally that this intermediate amount of **β-conversion** is not unprincipled, because it is all that is clearly motivated by **Abstraction as Definition**.

To start, here is a familiar and popular idea, advocated most influentially by Russell. Although some propositions are purely 'general' or 'qualitative', others are 'singular', 'haecceitistic', or 'object-involving', by having certain particular individuals as 'constituents'. In short, propositions are *about* some individuals but not others.

An implication of this idea, as it is usually understood, is that the proposition expressed by a sentence is at least about every individual referred to by some name in that sentence. We can regiment this claim schematically as follows:

Named Aboutness: $\mathcal{A}(\varphi, a)$, where a is an individual constant in φ

Here $\mathcal{A}$ ("is about") is a constant of of type $\langle t, e\rangle$; we write $\ulcorner\mathcal{A}(p, x)\urcorner$ instead of $\ulcorner\mathcal{A}px\urcorner$ for readability. Now, the thought behind **Named Aboutness** is meant to apply generally, to all individuals, whether or nor we have names for them. This motivates the following more general schema:

***De re* Aboutness**: $\forall x\mathcal{A}(\varphi, x)$, where x is free in φ

Given **β-conversion**, this is equivalent to: $\forall x\mathcal{A}((\lambda x.\varphi)x, x)$, with x free in φ. This fact suggests the following purely logical account of aboutness:

Aboutness: $\forall p\forall x(\mathcal{A}(p, x) \leftrightarrow \exists F^e(p \approx Fx))$

Informally, p is about an individual x just in case, for some property of individuals F, p *is* Fx (i.e., there is no difference whatsoever between p and Fx). As with **Structure**, heuristic talk of 'propositions' and 'properties' has been replaced with quantification into sentence and predicate positions.

If this simple and economical account succeeds in capturing a robust notion of aboutness that can be put to work where metaphysicians have wanted to deploy such a notion, that would be a strong advertisement for higher-order logic as metaphysics. Defending the claim that it does capture such a notion is far beyond the scope of this paper.[29] My goal here is simply to argue that the account is at least worth exploring and is not obviously untenable.

Although the idea of singular propositions is often run together with the structured picture discussed in the last section, note that **Aboutness** is only tenable if that picture is rejected. For example, **Named Aboutness** requires that $\mathcal{A}((\lambda x.a = x)b, a)$; **Aboutness** then entails $\exists F(Fa \approx (\lambda x.a = x)b)$, which is inconsistent with **Structure** for $a \neq b$. This is the sense in which β**-conversion** makes available a simple, purely logical theory of aboutness. If β**-conversion** were rejected in favor of a more structured picture, it is unclear how any such account could be given.

Another attraction of **Aboutness** is that it easily generalizes to other types, both in the sense that entities of types other than t can have ingredients, and in the sense that entities of types other than e can be ingredients.[30] The details are in a footnote, since they are orthogonal to most pressing challenge for all such accounts. As with structure, the challenge turns on β**-conversion**.

Up to this point we have required that every variable in the prefix of a λ-term occur free in the matrix of that λ-term. So, in particular, $\ulcorner(\lambda x^e.\varphi)\urcorner$ is not well-formed if x does not occur free in φ. This makes our language what is known as a *λI-language*, as in the original presentation of the simply typed λ-calculus in Church (1940), rather than a λK-language, which has become more standard, in which there is no prohibition on vacuous predicate abstraction.

Anyone who understands the λI-language should agree that there are *some* perfectly intelligible ways of understanding the vacuous λ-terms of λK-languages. For example, one could treat $\ulcorner(\lambda x^e.\varphi)\urcorner$ as an abbreviation for the λI-term $\ulcorner(\lambda x.\varphi \wedge x \approx x)\urcorner$ when x is not free in φ. The extent to which this is a principled interpretation depends on background granularity-theoretic issues, as discussed in Dorr (2016). But since such interpretations are not impossible, it is best to not ask whether vacuous λ-terms are 'intelligible' or 'legitimate', but rather whether there is an interpretation of such terms on which they are both meaningful and obey β**-conversion**.

The reason this question matters is that, if there is such an interpretation of vacuous λ-terms, then **Aboutness** trivializes $\mathcal{A}$, since every proposition will be about every individual. This is because β**-conversion** for vacuous λ-terms implies that $\forall p \forall x(p \approx (\lambda y.p)x)$, and hence $\forall p \forall x \exists F(p \approx Fx)$.

[29] In unpublished work I show how **Aboutness** can form the core of a simple and strong theory in pure higher-order logic of how individuals figure in propositions, properties, and relations, extending the view of propositional granularity outlined in Goodman (2019, §7.2).

[30] The former kind of generalization is helpful to capture distinctions among properties and relations – for example, to characterize those that are *qualitative* as those not about any individuals. The second kind of generalization has been suggested by Dorr (2016) to develop a notion of 'metaphysical priority', the idea being, for example, that blue and green are both metaphysically prior to Goodman's (1955) grue and bleen (and not vice versa). Here is the general definition. For convenience, we identify t with $\langle\rangle$ (as is commonly done) and let $\ulcorner(\lambda.\varphi)\urcorner$ abbreviate φ. We can then define a typed family of "is an ingredient of" predicates as follows: $\mathcal{I}^{\langle\tau_1,\langle\tau_2,\ldots,\tau_n\rangle\rangle}(x, F) := \exists G^{\langle\tau_1,\ldots,\tau_n\rangle}(F \approx (\lambda x_2 \ldots x_n.Gxx_2 \ldots x_n))$.

As someone attracted to **Aboutness**, I am inclined to deny that there is any use of predicate abstraction on which vacuous λ-terms are both meaningful (in any sense that would license existentially generalizing on them) and governed by β**-conversion**. This denial is tenable, in my view, because vacuous β**-conversion** cannot be directly motivated by **Abstraction as Definition**. Our preexisting practice of introducing predicates by explicit definition does not involve anything like vacuous abstraction: vacuous definition is not something we do in the wild. Indeed, a particularly elegant regimentation of our practice of explicit definition – the numeral-based notation mentioned in section 2, used by Goldfarb (2003), Smith (2020), and others – is incapable of even formulating vacuous definitions of monadic predicates.

To be clear, I am not claiming that it is *unprincipled* to hold that there is a way of understanding vacuous predicate abstraction that obeys β**-conversion**. Formally, it is not an *un*natural generalization of the use of λI-terms. More importantly, the existence of an interpretation of vacuous λ-terms on which they are meaningful and obey β**-conversion** turns out to be a consequence of views in grain science that we should take very seriously. For example, *Classicism*, as developed by Bacon and Dorr (this volume), entails that β**-conversion** holds on the above interpretation of vacuous λ-terms as short for non-vacuous λI-terms with extra $\ulcorner x \approx x \urcorner$ conjuncts in their matrix clauses. This pits **Aboutness** against Classicism and other similarly coarse-grained views. It thereby demands an intermediate theory of granularity that involves rejecting **Structure** without collapsing all distinctions among logical equivalents; see Goodman (2019).

6 Conclusion

The aim of this chapter has been to advance a particular interpretation of a simply relationally typed higher-order language. This was done in four stages: starting with first-order logic, adding first-order predicate abstraction, generalizing to higher-order predicate abstraction, and finally adding higher-order quantification. I argued that β**-conversion** and **Universal Instantiation** are valid on this interpretation. I then explained how this fact allows us to use pure higher-order logic to ask, and begin to answer, questions about the logical structure of reality, and gestured at some of these questions' non-trivial implications. I hope this incremental approach helps to persuade those new to higher-order languages that the formalism makes sense, and also helps to advance the debate by challenging those skeptical of higher-order metaphysics to say at what stage they think it falls into disrepute.

References

Andrew Bacon. The broadest necessity. *Journal of Philosophical Logic*, 47(5):733–83, 2018.
Andrew Bacon. A theory of structured propositions. *Philosophical Review*, forthcoming.
Andrew Bacon. A case for higher-order metaphysics. this volume.
Andrew Bacon and Cian Dorr. Classicism. this volume.

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

George Boolos. The iterative conception of set. *The Journal of Philosophy*, 68(8):215–231, 1971.

George Boolos. To be is to be the value of a variable (or to be some values of some variables). *Journal of Philosophy*, 81:430–49, 1984.

Tim Button and Robert Trueman. A fictionalist theory of universals. this volume.

Michael Caie, Jeremy Goodman, and Harvey Lederman. Classical opacity. *Philosophy and Phenomenological Research*, 101(3):524–66, 2020.

Alonzo Church. A formulation of the simple theory of types. *The Journal of Symbolic Logic*, 5(2):56–68, 1940.

Cian Dorr. Quantifier variance and the collapse theorems. *The Monist*, 97(4):503–70, 2014.

Cian Dorr. To be F is to be G. *Philosophical Perspectives*, 30(1):39–134, 2016.

Cian Dorr, John Hawthorne, and Juhani Yli-Vakkuri. *The Bounds of Possibility*. Oxford University Press, 2021.

Gareth Evans. Reference and contingency. *Monist*, 62:161–189, 1979.

Maegan Fairchild. Symmetry and hybrid contingentism. this volume.

Kit Fine. Properties, propositions and sets. *Journal of Philosophical Logic*, 6(1):135–191, 1977.

Kit Fine. Guide to ground. In Fabrice Correia and Benjamin Schnieder, editors, *Metaphysical Grounding: Understanding the Structure of Reality*, pages 37–80. Cambridge: Cambridge University Press, 2012.

Gottlob Frege. *Begriffsschrift, ein der arithmetischen nachgebildete Formelsprache des reinen Denkens*. Verlag L. Nebert, Halle/Saale, 1879. Translated as *Concept Script, A Formal Language of Pure Thought Modelled Upon That of Arithmetic*, by S. Bauer-Mengelberg in *From Frege to Godel: A Source Book in Mathematical Logic, 1879–1931* (ed. J. van Heijenoort, 1967), pp. 1–82. Harvard University Press, Cambridge, MA.

Gottlob Frege. On concept and object. *Mind*, 60(238):168–80, 1951. Translated by P. T. Geach and Max Black. First published in 1892 as "Über Befriff und Gegenstand".

Gottlob Frege. *Basic Laws of Arithmetic: Derived using concept-script*. Oxford University Press, 2013. Translated by Philip Ebert and Marcus Rossberg. First published in 1892 as *Grundgesetze der Arithmetik*.

Peter Fritz. Ground and grain. *Philosophy and Phenomenological Research*, forthcoming.

Peter Fritz and Jeremy Goodman. Higher-order contingentism, part 1: Closure and generation. *Journal of Philosophical Logic*, 45(6):645–695, 2016.

Peter Fritz and Nicholas Jones. Higher-order metaphysics: an introduction. this volume.

Warren Goldfarb. *Deductive Logic*. Indianapolis: Hacket, 2003.

Jeremy Goodman. An argument for necessitism. *Philosophical Perspectives*, 30:160–82, 2016.

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

Jeremy Goodman. Agglomerative algebras. *Journal of Philosophical Logic*, 48(4):631–48, 2019.

Jeremy Goodman. Grounding generalizations. *Journal of Philosophical Logic*, forthcoming.

Jeremy Goodman and Harvey Lederman. Perspectivism. *Noûs*, 55(3):623–48, 2021.

Nelson Goodman. *Fact, Fiction, and Forecast*. Cambridge, MA: Harvard University Press, 1955.

Richard Heck and Robert May. The function is unsaturated. In Michael Beaney, editor, *The Oxford Handbook of Analytic Philosophy*, pages 825–50. Oxford University Press, 2013.

Kevin Klement. Higher-order metaphysics in Frege and Russell. this volume.

Saul A. Kripke. Naming and necessity. In Donald Davidson and Gilbert Harman, editors, *Semantics of Natural Language*, pages 253–355, 763–769. Dordrecht: D. Reidel, 1972.

Joachim Lambek. The mathematics of sentence structure. *The American Mathematical Monthly*, 65(3):154–170, 1958.

David Lewis. How to define theoretical terms. *Journal of Philosophy*, 67:427–446, 1970.

Vann McGee. How we learn mathematical language. *Philosophical Review*, 106(1):35–68, 1997.

John Myhill. Problems arising in the formalization of intensional logic. *Logique et Analyse*, 1 (1):78–83, 1958.

Charles Sanders Peirce. On existential graphs, Euler's diagrams, and logical algebra. In Charles Hartshorne, Paul Weiss, and Arthur W. Burks, editors, *Collected Papers of Charles Sanders Peirce*, volume 4. Cambridge, MA: Harvard University Press, 1931-1958.

Paul Pietroski. *Conjoining Meanings*. Oxford University Press, 2018.

A. N. Prior. Platonism and quantification. In *Objects of Thought*, chapter 3, pages 31–47. Oxford: Oxford University Press, 1971.

W. V. O. Quine. *Philosophy of Logic*. Cambridge, MA: Harvard University Press, 1970.

Agustín Rayo and Timothy Williamson. A completeness theorem for unrestricted first-order languages. In JC Beall, editor, *Liars and Heaps: New Essays on Paradox*, pages 331–356. Oxford: Oxford University Press, 2003.

Agustín Rayo and Stephen Yablo. Nominalism through de-nominalization. *Noûs*, 35(1):74–92, 2001.

Bertrand Russell. *The Principles of Mathematics*. London: W. W. Norton & Company, 1903.

Bertrand Russell. On denoting. *Mind*, 14:479–493, 1905.

Bertrand Russell. Mathematical logic as based on the theory of types. *American Journal of Mathematics*, 30(3):222–62, 1908.

Stewart Shapiro. *Foundations without Foundationalism: A Case for Second-Order Logic*. Oxford: Clarendon Press, 1991.

Peter Smith. *An Introduction for Formal Logic*. Cambridge University Press, 2nd edition, 2020.

Gabriel Uzquiano. A neglected resolution of Russell's paradox of propositions. *Review of Symbolic Logic*, 8(2):328–44, 2015.

Timothy Williamson. Everything. *Philosophical Perspectives*, 17:415–465, 2003.

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