Entity Grounding, Structure and Ramification

Abstract

The literature on metaphysical ground often conceives the relation of grounding as only concerning facts or fact-like entities that hold 'in virtue of' other such entities. A few exceptions to this tradition stand out, however, according to which entities of all kinds, such as individuals, propositions, facts, properties and relations, are capable of entering into grounding relations—what is sometimes called 'entity grounding'. In this paper, I lay down and defend certain plausible principles of entity grounding along the lines of what is explicitly or implicitly entertained in the literature, and argue that capturing these principles requires relational entities such as propositions and properties to come in infinite levels—hierarchies that are best captured by a ramified theory of types.

1 Introduction

The literature on metaphysical ground often conceives the relation of grounding as only concerning facts or fact-like entities that hold 'in virtue of' other such entities, manifesting the idea that the latter 'explain' or are 'more fundamental' than the former (see, e.g., Audi 2012; Fine 2012; Rosen 2010, for such construals of ground). A few exceptions to this tradition stand out, however, according to which entities of all kinds, such as individuals, propositions, facts, properties and relations, are capable of entering into grounding relations (as in, e.g., deRosset 2013; Schaffer 2009; Wilhelm 2020a)—what is sometimes called 'entity grounding' (Wilhelm 2020a).

In this paper, I lay down and defend certain plausible principles of entity grounding, along the lines of what's been explicitly or implicitly entertained in the literature, and argue that they require propositions, properties and other types of relations each to come in infinite levels, where, roughly put, the inhabitants of higher levels are obtained through quantification over the ones from lower levels. I then rigorously propose certain ramified type systems that best capture the talk of entity grounding and the infinitary hierarchies it calls for.

This paper is the first of five broadly related papers in which I set out to explore a deep interconnection between structured views of reality, different notions of metaphysical priority, and ramified type systems. The general goal of the series is to show that these together bring about a uniform, elegant picture of reality within which a cluster of puzzles and paradoxes of ground and grain in contemporary metaphysics are settled while rejecting any of them will make the picture collapse in its entirety. The main task of this paper, in particular, is to argue that certain considerations regarding entity grounding and structure call for infinitary hierarchies of relational entities such as propositions and properties, and to rigorously devise a ramified type system that captures them. Anonymous (MS[c]) attends to issues regarding the consistency of ramified type theory and how, in particular, it secures highly structured accounts of propositions from paradoxes of grain, such as the Russell-Myhill paradox (see e.g., Goodman 2016; Hodes 2015; Myhill 1958; Russell 1903; Uzquiano 2015, for various versions of the paradox). Anonymous (MS[e]) uses highly structured propositions to lay down a novel and expressive semantics for unrestricted impure propositional logics of truth-functional, iterated and identity grounding (as explored in, e.g., Bennett 2011; Correia 2017; Fine 2012; Krämer 2018; Wilhelm 2020b), and Anonymous (MS[b]) leverages the ramified hierarchy to settle some of the prominent puzzles of quantificational ground (as explored in, e.g., Donaldson 2017; Fine 2010; Fritz 2021; Krämer 2013) in a unified way.[1] Finally, Anonymous (MS[d]) sheds light on the historical roots of the ideas explored in this paper. It is hoped that the findings of these papers contribute to promoting ramified type theory in the scene of contemporary metaphysics, alongside its main rivals, namely simple type theory and its coarse-grained metaphysical offspring.

Here's how the paper is organized. In §2 I introduce the notion of entity grounding and argue for some minimal principles that characterize it. In §3 I argue that attempts in capturing the propositional fragment of the talk of entity grounding naturally lead to fragmentation of the space of propositions into an infinite hierarchy of levels. §4 explicates notions of structure and constituency for relational entities and argues for stratification of all relational types. In §5 I introduce a general ramified type system motivated by discussions taking place in the preceding sections. The paper concludes in §6.

[1] In effect, our ramified approach can be said to constitute a 'predicative' solution to these puzzles which has long been speculated but remained fairly underexplored; see Fine (2010) and Krämer (2013).

2 Entity Grounding and Its Principles

In this section, I introduce the notion of entity grounding and lay down some plausible principles that characterize it. The next two sections will utilize these principles and argue for fragmentation of propositions, as well as other relational entities, into certain infinitary hierarchies of levels that are best captured by ramified type systems.

Entity grounding (hereafter: *e-grounding*) is a relation of metaphysical priority that can hold between entities of any type. An individual may e-ground a proposition or fact, a proposition may e-ground a property, a property may e-ground a relation or a proposition, and so on. To illustrate with examples along the lines of the literature: '[for any entity i,] $i = i$ is grounded in i' (Wilhelm 2020a), 'Obama, the man in full, grounds the fact that Obama exists; Obama grounds his singleton; the property *being white* grounds *being white or square*; England grounds (in part) the property of *being queen of England*' (deRosset 2013).[2]

The examples above, and many more in the literature, highlight an implicit or explicit sense of structural complexity that statements of e-grounding exhibit. Thus, correctly saying that a e-grounds b reflects the fact that a is, as it were, a 'building block' of b, or b is somehow 'constructed' in part from a.[3] It is this construction, it would seem, that puts a prior

[2] Traces of the idea that Russell's ramified type theory can be motivated by a sense of 'construction', and the relation of this to vicious circle principles, can be found in Gödel (1984 [1944]) and Quine (1963), though the resolution there is that only a *constructivist* worldview, according to which, e.g., mathematical entities are pure constructions of mind, can accommodate ramified types (see, e.g., Gödel 1984 [1944], p. 456). However, Jung (1999) and Hylton (2008) argue through textual evidence that Russell's notion of 'presupposition' does exactly carry the relevant sense of construction, though in a completely *realist* background. My notion of entity grounding is, in fact, motivated by and a close kin of Russell's 'presupposition', and most of the principles applying to the former (as introduced below) correspond to similar principles governing the latter (as laid down in the references above). However, unlike entity grounding, Russell's 'presuppositions' aren't given primitively, and in fact are closer in nature to naive, 'modal-existential' conceptions of ontological dependence which have been heavily criticized in the recent literature (see, e.g., Fine 1995). Another difference is our departure from Russell's rather contentious assumptions on the nature of propositional functions and how they're related to presuppositions (see Hylton 2008, for a discussion of the latter). In what follows I will not address these historical remarks due to the scope limits of the paper; the interested reader can see Anonymous (MS[d]) for a detailed discussion.

[3] One might think the talk of 'building blocks' presupposes some sense of unique decomposition of entities into parts—*the* parts that constitute them, as it were—and that there might be more than one way to decompose a relational entity into parts, much like there are many ways to cut a sphere into two hemispheres. While it well might be so, the assumption of either a unique or plural decomposition of entities into constituents would seem to serve our purposes in this paper equally well; in effect, we can run our arguments for those decompositions that are relevant to our purposes.

to b in a metaphysical sense. We will explicate the sense of construction at stake further, along the way.[4]

We mentioned that e-grounding is a relation of metaphysical priority. One might wonder at the outset whether the notion of e-grounding is the same as that of fact grounding (henceforth: f-grounding). But that can't be the case. The most straightforward reason for this is that, as was mentioned earlier, f-grounding, as opposed to e-grounding, is much more restricted in its scope, being only concerned with entities like propositions and facts.[5] Also, and as it will become clear along the way, even if one narrows down the scope of e-grounding to fact-like entities, the notion does not have anything to do with the truth of the entities involved, but rather, somehow with their structural complexity.

In what follows, I will argue for some plausible principles that characterize the notion of e-grounding. As we will see, most of these principles are either explicitly or implicitly, and in part or fully entertained by other authors in the literature, and otherwise quite naturally build upon them. Aside from the overall conformity to the literature and the intuitive appeal of the principles of e-grounding that are to be discussed in this section, it is important to also bear in mind that perhaps an even more substantial defense of these principles consists in the role they play in the larger scene of contemporary metaphysics, where, as the sequels to this paper show, a vast array of puzzles and paradoxes of ground and grain are uniformly and naturally resolved. One can perhaps compare this to the status of certain principles of set theory, e.g., as in ZFC, where, aside from their intuitive appeal, the rich and unified mathematics they bring about itself counts as an indirect, abductive augment for their truth and suitability, and even as a foundation for mathematics.

We start by taking, along with Schaffer (2009), the notion of e-grounding as a primitive, that is not analyzable in terms of any other notions. Also, following deRosset (2013) and Schaffer (2009), I take the relation of e-grounding to be transitive and irreflexive (hence a strict order).

Thus we have our first two principles:

- *Nothing e-grounds itself.* IR

[4] It is to be noted that while in this paper we mostly take interest in and focus on structurally related entities in our explication of e-grounding, some of the examples in the literature may not necessarily carry that sense. For instance, according to Schaffer (2009), a Swiss cheese grounds its holes, or natural properties ground moral properties. Whether or not such examples can count as instances of true e-grounding statements, we find the 'constructional' intuition behind e-grounding plausible and somewhat crucial in the discussions to follow.

[5] That said, however, it's not at all obvious whether or not f-grounding is a special kind of e-grounding. We will return to this point when we introduce the e-grounding principle S, blow.

- *If a e-grounds b and b e-grounds c, then a e-grounds c.* TR

These requirements are especially natural when, as suggested earlier, we come to think about the relation as somehow measuring the 'constructional' profile of entities. Surely nothing is a 'building block' of itself. Also, if a is a 'building block' of b, and b is itself a 'building block' of yet another thing c, then there is a sense in which a is a 'building block' of c.

Another principle that we would like to entertain is along the lines of this: propositions, properties and relations are e-grounded by their constituents, assuming that a suitable sense of relational constituency is in place. For instance, the proposition that Joe drinks soda is e-grounded by Joe, and the property of drinking soda, and the property of being friends with Gary is e-grounded by Gary and the relation of friendship. As the examples from the beginning of the section show, assumptions similar to this can also be found, in implicit or explicit forms, in the literature.

We would like to entertain a principle along these lines, but since one of the goals of this paper is to rigorously account for the talk of e-grounding in suitable formal languages, it would be desirable to recover the constituents of propositions, properties or relations, from the syntactic structure of the expressions that refer to them. That is, first we would like to find a suitable notion of constituency for relational entities in our formal language that captures our intuitions about the constituency of relational entities (which we rigorously will at §4); once we have such a notion pinned down, our principle will be as follows:

- *Entities picked by expressions are e-grounded by the things that are picked by the constituents of those expressions.* S

Thus, for instance, we would like to say that the proposition *Joe is sleeping* is e-grounded by Joe and the property of sleeping because the sentence expressing that proposition—'Joe is sleeping'—has as constituents the name 'Joe' and the predicate 'is sleeping'.

Some qualifications about S, and in particular its propositional fragment, are in order. First, notice that in S we're not taking the e-grounds of an entity to consist *only* of the things denoted by its constitutive expressions. (Call the version of S that does so the *strong* variant.) We're only *including* those things in the list of entities that e-ground the proposition, but we are open, and in many cases, obligated to, allowing for more things to count as e-grounds of the proposition. (Call the more liberal version of S the *weak* variant.) We will return to the importance of this choice after we introduce our next principle.

Second, consider the propositional instance of S, according to which propositions are e-grounded by the denotations of the expressions which constitute the sentences expressing them. This principle, in either of its strong or weak readings, clearly imposes *some* structure on propositions. But how much structure do we really need for this to go through? The most granular account of propositions available in the literature take propositions to almost exactly reflect the syntactic structure of the sentences that express them, in a way that two propositions $F(a)$ and $G(b)$ are the same only if F and G are the same, and a and b are the same; call this identity condition STRUCTURE (see, e.g., Kaplan 1989[1977]; J. King 1996; J. C. King 2009; Russell 1903; Soames 1987, for such structured accounts of propositions). But endorsing the weak variant of S doesn't necessarily lead to STRUCTURE. For consider the pair of propositions *Sarah lives in LA or John is happy* and *John is happy or Sarah lives in LA*. By STRUCTURE, these two are different propositions, but by S, they have the same grounds; there's nothing else that tells us whether or not they are identical. So as long as we adopt the weak variant of S, we don't need to endorse highly structured accounts of propositions along the lines of STRUCTURE.[6]

That said, however, even weak S would presumably require enough structure that coarse-grained views of propositions, such as Booleanism (Bacon 2018; Dorr 2016), would become difficult to maintain. For example, according to Booleanism, the sentence 'John is happy' and its self-conjunction, 'John is happy and John is happy', express the same proposition because they're provably equivalent, but under any plausible sense of syntactic constituency, the former sentence is a constituent of the latter, so, by S, the proposition expressed by the latter is e-grounded by the one expressed by the former, hence they have to be distinct due to IR. This is one of the major conflicts of our project and some of the rival views in recent metaphysics, where such coarse-grained accounts of propositions stand out.[7]

So, if the instances of S that concern propositions are true, then propo-

[6] On the other hand, it can be readily seen that the strong variant of S leads to the highly granular picture of propositions manifested by STRUCTURE. For by strong S, $F(a)$ is e-grounded *only* by a and F, and $G(b)$ is e-grounded *only* by b and G. Now, if $F(a)$ and $G(b)$ are the same, they should have the same grounds; in particular, $F(a)$ also *only* has b and G as its grounds. It then follows that $\{a, F\} = \{b, G\}$, which, only plausibly results in F being G, and a being b. (Other possibilities manifest type mismatch in any language where predicates and names are considered as members of different syntactic categories.) Now, it can be shown that STRUCTURE is inconsistent with simple type theory due to the Russell and Myhill paradox (Goodman 2016; Hodes 2015; Myhill 1958; Russell 1903; Uzquiano 2015). But even so, we can still admit STRUCTURE since it is consistent with the ramified type system which we will eventually adopt in this paper. This consistency is established in Anonymous (MS[c]).

[7] Another related conflict consists in the type systems that we use: while views like Booleanism extract metaphysics from simple type systems, we cannot capture the talk of entity grounding within such systems (as this paper shows) and have to deploy ramified type systems, in the end.

sitions cannot be too coarse-grained. That said, however, we will see in the next two sections that it's still possible to argue that propositions, properties and relations have to come in levels, *even if* propositions aren't structured at all. In fact, in §4 we will see that it's only essential for non-propositional types of relational entities to be structured in certain ways, for ramification to go through unless certain assumptions regarding quantificational statements are in place (more on this shortly). In any case, we find the propositional instances of S plausible, and in the rest of the paper, we put them out in the open and leave it to the reader to choose whether or not to accept it. We will revisit and discuss this choice and its implications later in the paper.

Finally, earlier we argued that due to scope differences, f-grounding and e-grounding cannot be identified. But we also brought upon the natural question of whether or not f-grounding can be considered as an instance of e-grounding. Our principle S answers to this negatively. Consider again the fact that John is happy. Assuming that *John is happy* is a constituent of *John is not happy*, from *S* it follows that the former e-grounds the latter. But of course, we can't say the same thing about f-grounding, not at least under a *factive* conception of f-grounding, according to which the relata of grounding statements should be true.[8]

The final assumption that we make about e-grounding is this:

- *Quantificational propositions are e-grounded by all the entities they non-vacuously quantify over.* Q

For example, consider the proposition *every individual is self-identical*. By Q this proposition is e-grounded by every individual. On the other hand, we don't want to say that *every property of individuals is such that Mike lives in Chicago* is e-grounded by every property of individuals. There's a sense in which *Mike lives in Chicago* in no way uses every property of individuals as a 'building block'. For instance, the property of jumping off a cliff seems to play no contribution in the construction of the proposition in question. This is why it's important to assure that Q concerns non-vacuous quantification.

A natural way to motivate Q is via construing universal and existential quantification as 'long', possibly infinite conjunctions and disjunctions, respectively. In that case, Q will become a special case of a more general version of S, where structured propositions with infinite constituents are allowed.[9]

[8] Accordingly, non-factive grounding might have a different status in this regard, and in fact, be a special kind of e-grounding. See Fine (2012) for more on the distinction between factive and none-factive grounding.

[9] The idea of reducing quantificational sentences to infinitary conjunctions or disjunctions goes back to the school of logical atomism (see, e.g., Russell 2009[1918], lecture 5,

Notice, however, that such construals aren't necessary for committing to Q. It just seems quite natural and intuitive to think of, e.g., universal quantification as somehow built out of the things it quantifies over, even if it's not construed as a conjunction. Phenomena like this, where an entity that uses up, as it were, all of a kind in its construction falls out of the range of the things it uses, aren't unheard of. Set theory is a good source of such examples. For instance assuming that A is a set, the singleton $\{A\}$, which, in a way is 'constructed' out of A with a set-formation operation, doesn't belong to A.[10]

As another example, consider the way ordinal numbers are constructed in set theory: ω is constructed through a union over *all* natural numbers, and itself falls out of their realm; $\omega + \omega$ is constructed by consuming *all* ordinals of the form $\omega + i$, for natural i, and itself falls out of them, and so on. This hierarchical construction of transfinite ordinals by unioning over all numbers beneath them needn't be cashed out in terms of 'infinitary sums'; it is an independently plausible and useful construction. Yet another rich source of such objects is category theory. In general *accessible categories* are (possibly) large categories that are in a certain way constructed by small categories; e.g., the objects of the former are colimits of small objects from the latter and fall out of their range.[11]

In any case, now that we have introduced Q, we're also in a position to see why we chose the weak variant of S over its strong variant: this is mandated upon us by Q. For example, by Q, every individual e-grounds the proposition *some individual is distinct from John*. Now, it follows, for instance, that Sarah e-grounds *some individual is distinct from John*, but Sarah is not picked by any of the syntactic constituents of the sentence, 'Some individual is distinct from John'. That said, however, if take Q as an instance of S, then we end up committing to strong S.

To conclude the section, here's a summary of the list of our principles of e-grounding:

- *Nothing e-grounds itself.* IR

- *If a e-grounds b and b e-grounds c, then a e-grounds c.* TR

for an early objection to the idea). For some recent discussions of the problems such construals face, in particular, what's known as the 'totality problem', see Fine (2012) and Fine (2017). Throughout the paper, we sometimes make such reductionist assumptions about quantification, but mainly heuristically; there are, however, ways to make rigorous these assumptions. See the discussion after Definition 1, for more on this comparison.

[10] This is a particularly suggestive example because singletons are canonically taken to be e-grounded by their elements (deRosset 2013; Schaffer 2009). See Fine (1995) for a similar view regarding the ontological dependence of singletons on their elements.

[11] See, e.g., Adamek and Rosicky (1994) for an introduction to accessible categories.

- *Entities denoted by expressions are e-grounded by the things that are picked by the constituents of those expressions.* S

- *Quantificational propositions are e-grounded by all the entities they non-vacuously quantify over.* Q

3 Ramifying Propositions

I now argue that for the principles of e-grounding to be accommodated we need an infinite hierarchy of propositions, where, roughly put, tenants of each level are obtained through quantification from those of the lower levels. In line with this, we develop a formal language and logic that capture such a stratified universe of propositions as well as the talk of e-grounding based upon it.

We start with a simple, informal argument showing that our principles, in fact, two of them, lead to an inconsistency. Then we argue that the particular inconsistency involved is most naturally and efficiently resolved if there are at least two kinds of propositions, where the members of the second kind are obtained through quantification over all the members of the first kind. Ensuing arguments suggest that there must at least three kinds of hierarchical propositions, at least four kinds, ..., and for any natural number n, at least n kinds of them. In general, we will soon realize that the principles of e-grounding are most naturally and efficiently captured, without running into those generic inconsistencies, if propositions come in an infinite hierarchy of levels as described above.

Throughout this section, we focus only on the propositional fragment of the principles above—instances that are concerned with the relation of e-grounding between propositions. By doing so we can pinpoint the issues at stake in a setting where there aren't many types of entities available. Not only this avoids certain heavy-handed metaphysical commitments to all sorts of higher-order entities, but it also helps us to see in the clearest way why we need propositions to come in a ramified hierarchy, without being distracted by other entities and the structural complexities associated with them. Moreover, as we will see, capturing the propositional fragment is considerably easier when it comes to developing formal languages and logics for e-grounding. The next section will look up to this section as a role model and argue for stratifying other relational types into infinitary levels.

Before we start, there's an important methodological remark that needs to be explicated. Throughout this section, as well as the rest of the paper, we assume that the principles of e-grounding from the previous section articulate substantive constraints on e-grounding that we aim not to abandon. In other words, we take these principles true by stipulation and set out to explore their implications as well as the formal systems that can capture them. Aside from the inert plausibility and naturality of the principles, which we discussed in the previous section, it is this loyalty that, as we will see shortly, quite naturally leads to hierarchical propositions, and later on, other relational entities in a way that is best captured by ramified type systems. On the other hand, and as has been advertised several times, the latter is what provides a unified and natural solution to a cluster of puzzles and paradoxes of ground and grain. The intuitive appeal of our principles, espoused with other abductive, large-scale considerations surrounding them, gives us enough confidence to hold onto these principles as much as possible.

With this methodological remark out of the way, we now offer a series of arguments to the effect that there must be infinitely many kinds of propositions that behave in the way explained earlier. The arguments here are informal; a formalization of the talk propositional e-grounding and the arguments here will follow later in the section.

We first start by giving an informal presentation of a paradox that arises from our principles of e-grounding. By Q, any proposition p which states that every proposition is such and such (e.g., is true or false) is e-grounded by *every* proposition. So p cannot be among the propositions in its range of quantification, otherwise, contradiction ensues by IR. So the range of quantification in p consists of *all* propositions, and yet p cannot belong to it, hence contradiction.

There are two choice options in response to this argument: (i) at least one of IR or Q is false, or (ii) there is no such proposition as *all propositions are such and such*, or rather, p doesn't express any proposition. In line with the methodological remark above, we avoid option (i). But option (ii), with no further explanation attached to it, doesn't sound satisfactory either: the sentence 'Every proposition is such and such' supposedly denotes *something*. In fact, anyone who commits to second-order quantification as a reliable source for doing metaphysics would admit that p expresses a proposition (e.g., Fine 1970; Kaplan 1970; Williamson 2013). So, something needs to fill in the gap if p is taken not to be express a proposition.

To get around this tension, we could posit certain new types of entities that behave very much like the good old propositions but aren't propositions, strictly speaking. That is, in line with option (ii), we take it that p doesn't denote to a *proposition*, but it does denote to something akin to a proposition, only more sensitive, in its nature, to quantification. So, there should be at least two kinds of proposition-like entities, one of which is obtained by quantification over all members of the other. Call the latter *level-1*, and the former *level-2* propositions. These namings seem appropriate: as we will see later, leveled propositions interact with each other, and exhibit truth-functional behaviors very similar to how the good old propositions do. In fact, leveled propositions together will play the theoretical roles that propositions are supposed to play alone, but (it would seem) also accommodate the talk of e-grounding, without running into inconsistencies. In the light of this, if one still insists on keeping the term 'proposition' in their metaphysical vocabulary, one can then take the term to ambiguously refer to either level-1 or level-2 propositions.

This segregation also gains support from our intuitions about the notion of e-grounding in terms of 'construction' and the constructional profile of entities: if a e-grounds b, then a is, in a sense, a 'building block' of b. In particular, we took quantificational propositions to be 'constructed' out of the things they quantify over. By considering $\forall qq$ as a member of the collection that it ranges over, however, we'd be treating $\forall qq$ as if it's one of its own 'building blocks,' which doesn't sit well with that intuition.[12]

But the satisfaction that a bi-level account of propositions brings is only temporary. For we can similarly argue that there are at least *three* kinds of propositions. Suppose, on the contrary, that there are exactly two kinds of propositions, level-1 and level-2 propositions, as described above. Suppose also that Q is naturally revised for the new propositions: propositions that quantify over level-i propositions are e-grounded by all level-i propositions (where $i = 1, 2$). Given our assumption, the proposition p that every level-2 proposition is such and such should be either of level 1 or 2. Also, by Q, p is e-grounded by all level-2 propositions. Now, suppose p is of level 1. But since at least one level-2 proposition q (e.g., the proposition that all level-1 propositions are true or false) is e-grounded by *all* level-1 propositions, the proposition q must also be e-grounded by p. By TR, p e-grounds itself, which goes against IR. And if p is of level 2, then again it e-grounds itself. So p must be denoting a *third* kind of entity that's akin to leveled propositions, but isn't one of them; though it's obtained from level-2 propositions via quantification. Call this new, proposition-like entity a *level-3* proposition.

This line of argument clearly can be generalized to the effect that there are at least four, five, ..., n levels of propositions, for any natural number n, that behave expectedly, as explained above. So there must be infinitely many levels of propositions with the expected hierarchical construction. Notice that, given the similar e-grounding behavior of existential and universal

[12] Another helpful way to see the issue at stake is to revisit our heuristic way of construing universal quantifications as infinitary conjunctions (similarly for existential quantifications construed as infinitary disjunctions). Suppose $\forall p\phi$ is just a 'long', infinitary conjunction $\phi(\psi) \wedge \phi(\gamma) \wedge ...$ in a language with infinitary conjunctions and enough constants for all propositions. Then, in general, instantiating a universal statement $\forall p\phi$ with itself will amount to considering the long conjunction $\phi(\psi) \wedge \phi(\gamma) \wedge ...$ as one of its own conjuncts. Under a sufficiently structured view of propositions, which is presupposed by our principle S, it's no more appropriate to consider the 'long' conjunction as one of its own conjuncts than it is to take the 'short' conjunction $\phi \wedge \psi$ as one of its own conjuncts.

propositions (both being e-grounded by the things they quantify over), similar arguments can be given, using existential propositions, to the effect that propositions have to come in levels.

In the rest of this section, we aim at crafting formal languages that rigorously capture the propositional fragment of the talk of e-grounding and the arguments above.[13] Our starting point is the simple language of second-order logic, where quantification over propositions (construed as nullary predicates) is permitted. Formal languages that allow for quantification into sentential position appear in a number of places in philosophy, none of which have anything to do with notions of metaphysical priority such as e-grounding (see Fine 1970; Kaplan 1970; Williamson 2013, for some works along these lines). We, in particular, will be working with a further impoverished language with propositional quantification, where no first-order entities (i.e., individuals) play any role. So the only instances of structured propositions are going to be Boolean combinations of propositions, or when we add an operator for e-grounding, propositions that say of e-grounding relations that hold between propositions; no (non-nullary) predicates are available in the present language.

So, let $\mathcal{L}_1$ be the language of propositional logic with the addition of propositional variables $p, q, r...$ and universal quantification over them. We assume that formulas are closed under Boolean connectives and, for the sake of simplicity, that our language doesn't have any non-logical constants. Here's the abstract syntax of $\mathcal{L}_1$ in Backus–Naur form:

$\mathcal{L}_1 ::= p \,|\, \neg\phi \,|\, \phi \circ \phi \,|\, \forall p\, \phi \,|\, \exists p\, \phi$, where $\circ \in \{\rightarrow, \leftrightarrow, \vee, \wedge\}$[14]

Each legal term of this language is called a *formula*. *Free* and *bound* variables are defined in the usual way, and represent the set of all free variables of a formula ϕ with $FV(\phi)$. A formula with no free variable is called a *sentence*.

[13] The approach employed here is somewhat similar in spirit to the way Fritz (MS) motivates, in a step-by-step manner, simple type theory as a way of capturing some plausible talk of properties in English. What we will be doing, instead, is to motivate certain formal languages, also in a step-by-step manner, that aim at capturing our talk of e-grounding. There's a slight difference in our approaches, though: Fritz (MS) works with a more abstract sense of language, where he fixates upon certain desiderata that his desired languages need to satisfy. We, on the other hand, start with some concrete examples of languages that already exist in the literature and have gained traction by some philosophers and start improving upon them, step by step.

[14] In this paper we assume that all Boolean and quantified statements come as primitives, and not interdefined in terms of the other ones (e.g., defining $\wedge$ in terms of $\vee$ and $\neg$). One reason for this is to remain maximally neutral about the nature and granularity of logical connectives and quantifiers, without committing to any prejudices about their granularity. Moreover, in later sections where we choose a certain approach to present the logical vocabulary (coming in the form of constants, instead of clauses), the principles of e-grounding stated for a set of primitive operators may not be generalized to the interdefined ones.

Proof System $\vdash^{\mathcal{L}_1}$:

Axioms:

- Axioms of propositional logic PC
- $\forall p\,\phi \to \phi[\psi/p]$ UI
- $\phi[\psi/p] \to \exists p\,\phi$ EG
- $\forall p\,(\phi \to \psi) \to (\phi \to \forall p\,\psi)$, where $p \notin FV(\phi)$ UD
- $\forall p\,(\phi \to \psi) \to (\exists p\,\phi \to \psi)$, where $p \notin FV(\psi)$ ED

Inference Rules:

- If $\vdash \phi$ and $\vdash \phi \to \psi$, then $\vdash \psi$ MP
- If $\vdash \phi$ then $\vdash \forall p\,\phi$ UG

We add an entity grounding operator $\ll$ to $\mathcal{L}_1$, to be able to express our desired principles of e-grounding in the extended language:

$\mathcal{L}_1^{\ll} ::= p \,|\, \neg\phi \,|\, \phi \circ \phi \,|\, \forall p\,\phi$, where $\circ \in \{\to, \leftrightarrow, \vee, \wedge, \ll\}$.

We can now express our informal principles of e-grounding from the previous section in the language $\mathcal{L}_1^{\ll}$.

Proof System $\vdash^{\mathcal{L}_1^{\ll}}$:

The extended proof system $\vdash^{\mathcal{L}_1^{\ll}}$ is just $\vdash^{\mathcal{L}_1}$ plus the following axioms:

- $(\phi \ll \psi \,\wedge\, \psi \ll \gamma) \to \phi \ll \gamma$ TR_p
- $\neg(\phi \ll \phi)$ IR_p
- $\phi \ll (\phi \circ \psi) \wedge \psi \ll (\phi \circ \psi) \wedge (\phi \ll \neg\phi)$, where $\circ \in \{\to, \leftrightarrow, \vee, \wedge, \ll\}$ S_p
- $\psi \ll \forall p\,\phi \,\wedge\, \psi \ll \exists p\,\phi$, where $p \in FV(\phi)$ Q_p

Notice that, in the statement of S_p, each choice of $\circ$ amounts to a separate schema of the logic; we have packed them all together only for convenience and higher readability. Notice also that in all of the principles above, ϕ, ψ and γ schematically stand for formulas.

We can now see rigorously where things go wrong in this system. (In what follows, we replace the schematic ϕ with $\forall q\,q$.)

Theorem 1. $\varnothing \vdash^{\mathcal{L}_1^{\ll}} \bot$

Proof.

(1) $\forall q\, q \ll \forall q\, q$ $\qquad$ Q_p
(2) $\neg(\forall q\, q \ll \forall q\, q)$ $\qquad$ IR_p
(3) $\bot$ $\qquad$ PC 1, 2 □

An immediate reaction to this contradiction is to undermine at least one of the two principles of e-grounding that led to it, that is, IR_p and Q_p. But remember the methodological remark from the beginning of the section: we take our principles of e-grounding to articulate substantive constraints on e-grounding that we should aim not to abandon. Instead, we try to find suitable formal languages and logics that can accommodate them. In the present case, we only started by *assuming* that $\mathcal{L}_1$, a relatively well-known and simple language that seems suitable for our purposes, can do the job when enriched with an e-grounding operator (hence the language $\mathcal{L}_1^{\ll}$), and we faced an inconsistency using our minimal background logic. So, we do not conclude that any of the principles of e-grounding involved are false; rather, we question our choice of language in modeling the informal, stipulatively endorsed IR and Q. But how do we improve on our languages?

Notice that Theorem 1 essentially formalizes the first informal paradox that we proposed at the beginning of this section, in response to which we posited two kinds of propositions—level-1 and level-2 propositions. An improvement of the language that goes hand in hand with this solution, therefore, is desirable. Since we have it that $\forall q\, q$ expresses a level-2 proposition (obtained by quantification over all level-1 propositions), we may impose a similar structure on the sentences of $\mathcal{L}_1$. More specifically, if we assume that the sentential variable q is of level 1, then we can take the level of the sentence $\forall q\, q$ to be 2. Logical rules such as UI will need to be revised accordingly, accommodating leveled sentences. As a result of such level assignments to our sentences, we no longer will be able to instantiate $\forall q\, q$, which ranges over all *level-1* sentences, with itself, as it is of level 2, and the proof of Theorem 1 breaks down in its second step.

To accommodate all of this more rigorously in a formal setting, we explicitly assign types to our sentences, along the lines of type theory. In simple type theory (higher-order logic), it is a common practice to distinguish different kinds of expressions by assigning to them *types*. For example, individual terms are assigned type e, propositional terms type $\langle\rangle$, and n-ary relational terms type $\langle t_1, ..., t_n\rangle$, where $t_1, ..., t_n$ are themselves types. For various reasons, however, we started our project with languages that only have sentential types. That is, so far we have only worked with terms of type $\langle\rangle$, so we didn't need to write down the types of our terms. But now we have found a basis for distinguishing two kinds of propositions and sentences that correspond to them. On the other hand, if we hold onto $\langle\rangle$ as the only symbol for types, we won't be able to syntactically distinguish sentences that stand for different levels of propositions.

So we extend our second-order language by adding a new sentential type. More specifically, we now index the old sentential type with numbers 1 and 2 to explicitly indicate which kind of propositions they stand for. We reserve the type $\langle\rangle/2$ for formulas that stand for level-2 propositions, i.e., ones that are 'constructed from' *all* members of the other kind propositions, and $\langle\rangle/1$ for formulas that stand for the 'building blocks' of the former kind of propositions. A corresponding revision of the proof system $\vdash^{\mathcal{L}_1}$ is also required. In particular, we replace UI_p with two similar principles UI_i, one for each $i, j \in \{1,2\}$: $\forall p^{\langle\rangle/i}\phi_j \rightarrow \phi_j[\psi_i/p]$, where ψ_i schematically stands for any formula of level i. The logic $\vdash^{\mathcal{L}_1^*}$ of e-grounding also needs to be revised in such a way that leveled formulas are accommodated. In particular Q_p needs to be replaced by two parallel principles Q_i, one for each $i, j \in \{1,2\}$: $\psi_j \ll \forall q^{\langle\rangle/i}\phi_j$, where ϕ_j and ψ_j schematically stand for formulas of levels j and i, respectively.

Now, since we're working with leveled formulas, it should be rigorously decided by the syntax how the levels of Boolean and quantificational formulas are determined by the level of their constituents. For example, what is the level of the negation of a level-1 sentence, or the conjunction of a level-1 and a level-2 sentence? We take any combination $\phi \circ \psi$ of two leveled sentences ϕ and ψ to be of the *maximum* level of them. The reason for this is that we motivated the talk of 'levels' directly via quantification: for example, we took $\forall p^{\langle\rangle/1}p$ to be of type $\langle\rangle/2$. So there's no other way for quantification to lift levels. As for quantified statements, we can say that the type of $\forall p^{\langle\rangle/1}\phi_j$, for a formula ϕ_j of level j, is of type $\langle\rangle/\max\{2, j\}$, which is just $\langle\rangle/2$, where $j \in \{1,2\}$.

But what about formulas of the form $\forall p^{\langle\rangle/2}\phi$, where we quantify over *level-2* sentences? For all we know at this stage, such formulas will have to be either of level 1 or 2. But it can be readily verified that either of these options leads to inconsistencies like the one above. Here's why. (What follows is a more rigorous reconstruction of the second informal argument that was given at the beginning of the section.) Suppose, say, $\forall p^{\langle\rangle/2}(p \vee \neg p)$ is of level 2. Then since it quantifies over all level-2 propositions, by Q_2 it should be e-grounded by itself, which contradicts irreflexivity. So $\forall p^{\langle\rangle/2}(p \vee \neg p)$ must be of level 1. But now on the one hand, according to Q_2, $\forall p^{\langle\rangle/2}(p \vee \neg p)$ is e-grounded by all level-2 propositions, and on the other hand, at least one of these propositions (e.g., $\forall p^{\langle\rangle/1}(p \vee \neg p)$) is, by Q_1, itself e-grounded by all level-1 propositions, including $\forall p^{\langle\rangle/2}(p \vee \neg p)$. By transitivity of e-grounding it follows that $\forall p^{\langle\rangle/2}(p \vee \neg p)$ e-grounds itself, which again contradicts irreflexivity.

In line with the resolution of the informal, corresponding argument at the beginning of the section, we posit a third kind of propositional type to avoid the present inconsistencies. The syntax of the resulting language, as well as the principles of the logic governing it, also need revisions similar to the ones offered at the previous stage. It can easily be seen that similar inconsistencies arise as before and that we need a fourth kind of proposition-like entity to avoid the ensuing inconsistencies.

As is readily verified, this process goes on and on *ad infinitum*. That is, at every level n, we are going to need to posit a sentential type $\langle\rangle/n{+}1$ for level-n sentences. In general, continuing the process of improving our languages and their logics leads to an infinite sequence of language pairs $\mathcal{L}_1, \mathcal{L}_1^{\ll}, \mathcal{L}_2, \mathcal{L}_2^{\ll}, \mathcal{L}_3, \mathcal{L}_3^{\ll}, \ldots$ and corresponding logic pairs $\vdash^{\mathcal{L}_1}, \vdash^{\mathcal{L}_1^{\ll}}, \vdash^{\mathcal{L}_2}, \vdash^{\mathcal{L}_2^{\ll}}, \vdash^{\mathcal{L}_3}$ $, \vdash^{\mathcal{L}_3^{\ll}}, \ldots$, all attempting to capture the notion of e-grounding and its principles at a certain stage, but facing a familiar inconsistency.

Here's a minimal language $\mathcal{L}_\infty^{\ll}$ and a corresponding proof system $\vdash^{\mathcal{L}_\infty^{\ll}}$ that encompasses all the useful type distinctions and rules that these languages had to offer, but without running into similar, generic inconsistencies. We assume that for any $i \geq 1$ we have a denumerably infinite set $\mathrm{Var}^{\langle\rangle/i}$ of variables of type $\langle\rangle/i$. Our formulas are recursively defined as follows:

Definition 1 ($\mathcal{L}_\infty^{\ll}$)**.** The formulas of $\mathcal{L}_\infty^{\ll}$ are defined as follows:

1. If $p^{\langle\rangle/i} \in \mathrm{Var}^{\langle\rangle/i}$, then $p^{\langle\rangle/i}$ is a formula of type $\langle\rangle/i$,
2. If ϕ is of type $\langle\rangle/i$, then $\neg\phi$ is also a formula and of type $\langle\rangle/i$,
3. If ϕ and ψ are respectively formulas of types $\langle\rangle/i$ and $\langle\rangle/j$, then $\phi \circ \psi$ is a formula and is of type $\langle\rangle/\max\{i, j\}$, where $\circ \in \{\rightarrow, \leftrightarrow, \vee, \wedge, \ll\}$,
4. If ϕ is a formula of type $\langle\rangle/j$, then $\forall p^{\langle\rangle/i}\phi$ is a formula of type $\langle\rangle/\max\{i{+}1, j\}$.

We can now see why the construal of quantification as conjunction or disjunction, in a literal sense, can break apart. Given Definition 1 and the related discussions, the statements of quantification but not conjunction shift levels of sentences. For reasons like this, in this paper we rely on the construal of Q as an instance of S mostly for heuristic purposes.[15]

Here's the proof system $\vdash^{\mathcal{L}_\infty^{\ll}}$.

Proof System $\vdash^{\mathcal{L}_\infty^{\ll}}$:

Axioms:

[15] That said, however, there are ways to make the analogy more appropriate, without running into such level-mismatches. For instance, instead of taking the level of, say, the conjunction $\phi_i \wedge \psi_j$ of a level-i formula ϕ_i and a level-j formula ψ_j to be $\max\{i, j\}$, we could take it to have the level $\max\{i, j\} + 1$. In that case, we can add an infinitary conjunction operation to our language and just generalize this level-assignment to it to get our desired level-assignment for quantified statements.

1. Leveled axioms of propositional logic[16] PC_{ijk}

2. $\forall p^{()/k}\phi_j \rightarrow \phi_j[\psi_i/p]$ UI_{ij}

3. $\phi_j[\psi_i/p] \rightarrow \exists p^{()/k}\phi_j$ EG_{ij}

4. $\forall p^{()/k}(\phi_j \rightarrow \psi_k) \rightarrow (\phi_j \rightarrow \forall p^{()/k}\psi_k)$, where $p \notin FV(\phi_i)$ UD_{ijk}

5. $\forall p^{()/k}(\phi_j \rightarrow \psi_k) \rightarrow (\exists p^{()/k}\phi_j \rightarrow \psi_k)$, where $p \notin FV(\psi_k)$ ED_{ijk}

6. $(\phi_i \ll \psi_j \wedge\ \psi_j \ll \gamma_k) \rightarrow \phi_i \ll \gamma_k$ TR_{ijk}

7. $\neg(\phi_i \ll \phi_i)$ IR_i

8. $\phi_i \ll (\phi_i \circ \psi_j)\ \wedge\ \psi_j \ll (\phi_i \circ \psi_j)\ \wedge\ (\phi_i \ll \neg\phi_i)$, where $\circ \in \{\rightarrow, \leftrightarrow, \vee, \wedge, \ll\}$ S_{ij}

9. $\psi_i \ll \forall p^{()/k}\phi_j \wedge \psi_i \ll \exists p^{()/k}\phi_j$, where $p^{()/k} \in FV(\phi)$ Q_{ij}

Inference Rules:

10. If $\vdash \phi_i$ and $\vdash \phi_i \rightarrow \psi_j$, then $\vdash \psi_j$ MP_{ij}

11. If $\vdash \phi_i[\gamma_j/p^{()/j}]$, then $\vdash \forall p^{()/j}\phi_i$ UG_{ij}

Where in all of the principles above, ϕ_i, ψ_j and γ_k are doubly systematic: ϕ, ψ, and γ being any formula and $i, j, k \in \mathbb{N}^+$.

We conclude this section with a an important remark concerning the relationship between the principles Q and S. We mentioned earlier that Q can be endorsed plausibly and independently from S. In other words, accepting Q doesn't hing on construing statements of quantification as some 'long' Boolean sentences. If one is on board with us in this, then one can be neutral about, or even against any structured picture of propositions. But even if, for whatever reason, one accepts Q only as an instance of S, we noticed in §2 that it's still not necessary to commit to highly granular propositions that are susceptible to paradoxes of grain, such as the Russell-Myhill result. Either way, this puts those who reject structured propositions based on such inconsistencies (e.g., Dorr 2016; Goodman 2016; Uzquiano 2015) in an awkward position: they are now offered independent reasons to admit a ramified reality, which, if they do, they end up having access to the resources that allow for highly structured propositions, as well.

In the next section, we will expand the scope of our project from the propositional fragment of e-grounding to all that can be said about it. Accordingly, we will expand our linguistic resources by adding, among other

[16] For instance, $\phi_m \rightarrow (\psi_n \rightarrow \phi_m)$ in place of $\phi \rightarrow (\psi \rightarrow \phi)$, where each indexed letter doubly schematically stands for a formula of a certain level.

things, variables and quantificational tools for individuals as well as relational entities of different types and arities. In a similar way to this section, we will also try to find appropriate languages that can capture the talk of e-grounding and its logic.

4 Relational Structure and Ramification

We now extend the scope of our project by aiming at capturing the talk of e-grounding in its entirety. In particular, we will allow for statements of e-grounding that hold between entities of any pair of types: individuals e-grounding properties, propositions e-grounding operators, relations and individuals grounding propositions, etc. We can argue that, as long as properties and relations are properly structured, they should come in infinitary hierarchies of levels. The argument is similar in its spirit to the one from the previous section to the effect that propositions should come in infinite levels. First, we assume in the background that properties and other types of relations can have structure. For instance, it seems plausible to say that the property of being loved by everyone has the relation of loving as a constituent, or the property of being identical to Mike and such that Mike has some property has both Mike and the proposition that Mike has some property as a constituent.

Later in the section, we will propose a rigorous account of relational structure and constituency, but for now, consider the latter property, namely, *being identical to Mike and such that Mike has some property*. We can argue that this property is e-grounded by itself. Here's how: by S, the property is e-grounded by its constituent proposition, *Mike has some property*. On the other hand, by Q, that proposition itself is e-grounded by *all* properties of individuals. So by TR, the property of being identical to Mike and such that Mike has some property is e-grounded by *all* individual properties, including itself, which goes against IR.

Similar to the case of the propositional fragment, and given our strong commitment to the principles of e-grounding, it can be argued that the most viable option to resolve this contradiction while retaining those principles is to posit a new kind of individual properties whose inhabitants are obtained through quantification over all members of the other kind. The rest of the story is also similar to the one from before: we can run analogous arguments to the effect that there has to be at least 3, at least 4, and for any natural number n, at least n kinds of individual properties. As before, and to make these kinds traceable, we assign levels to these relations. Other relational entities can be argued to come in infinitary hierarchies of levels.

Throughout the rest of this section, we will devise formal languages that can rigorously express statements of e-grounding in their full generality and capture the line of argument for segregating relational entities into infinitely many levels. Now, while the spirit of the project here is quite similar to the one from the previous section, the syntactic complexities involved are considerably more complicated than the ones found there. In particular, we will see that given the way many predicates are constructed via lambda abstraction in higher-order languages, and certain complications attached to free and bound variables in λ-terms, finding a notion of constituency that properly and rigorously capture our intuitions of structure and constituency for properties and relations is, by no count, a trivial task and deserves special attention.

In what follows, we add variables and quantifiers of different types into our language. In the previous section, we were only interested in the propositional fragment of the talk of e-grounding, so we only focused on propositional variables and quantifiers. But now we want to capture everything that can be said about e-grounding. So, we add variables and quantifiers for individuals, propositions and relations of different arities.

We start with types. Simple types provide a way of tracking the grammatical categories of terms.[17]

Definition 2 (Simple Types)**.** The set $\mathcal{T}^s$ of *simple types* is recursively defined as follows: $e \in \mathcal{T}^s$, and for any $t_0, ..., t_n \in \mathcal{T}^s$, $\langle t_0, ..., t_n \rangle \in \mathcal{T}^s$.

When $n = 0$, the relational type is shown by $\langle\rangle$, which is the type of propositions. We assume that for any $t \in \mathcal{T}^s$ there's a denumerably infinite set of *variables* Var^t of type t and a (possibly empty) set of typed *constants* CST^t. We will reserve CST^t for the set of all constants of type t. We define the sets of all variables and constants respectively as $\mathrm{Var} := \bigcup_{t \in \mathcal{T}^s} \mathrm{Var}^t$ and $\mathrm{CST} := \bigcup_{t \in \mathcal{T}^s} \mathrm{CST}^t$.

In the previous section, we introduced the logical statements of our languages through clauses—what's sometimes called a 'syncategorematic' representation of logical statements. For instance, in Definition 1 we took it that whenever ϕ is a formula, then so is $\neg\phi$ (similarly for other connectives and quantifiers). Introducing the logical vocabulary via clauses is common in many textbooks and papers on logic, but there's an alternative, categorematic approach that specially dominates the literature on simple type theory (see, e.g., Church 1940; Dorr 2016; Henkin 1950; Mitchell 1996). According to the alternative approach, logical connectives and quantifiers are constants of certain types, and logical statements are formed using a certain operation called *application*. For instance, we take negation to be represented by a constant $\neg$ of type $\langle\langle\rangle\rangle$, and a negative statement like $\neg\phi$ to be a

[17] The type theories presented in this paper will be *relational* (as opposed to functional). Also, for higher readability, the style of typing will by *Church-typing* (as opposed to Curry-typing), where the types of variables are fixed and attached to them as superscripts, instead of depending on 'contexts'. Alternative formulations are possible as well.

shorthand for application of the constant $\neg$ to a term ϕ of the appropriate type $\langle\rangle$, which is shown by $\neg(\phi)$. (A similar attitude can be taken for other connectives and quantifiers.) Below we will discuss some of the advantages of treating the logical vocabulary categorematically, using typed constants.

In any case, here's the list of our primitive, typed logical constants: negation $\neg$ of type $\langle\langle\rangle\rangle$, implication $\rightarrow$, disjunction $\vee$ and conjunction $\wedge$ each of type $\langle\langle\rangle,\langle\rangle\rangle$, and for any type t, there is a constant $=^t$ for identities between t-type entities and two constants for quantification, one for (higher-order) universal quantifier $\forall^t$ and one for (higher-order) existential quantifier $\exists^t$, each being of type $\langle\langle t\rangle\rangle$. Notice that our quantifier constants apply to *predicates* of t-type entities, not those entities themselves. As will become clear through the proof system, however, there won't make any difference in the truth-conditional behavior of the logical statements in the constant-based and the clausal approach.

Definition 3 (Simple Terms). The *terms* of simple type theory (STT) are recursively defined as follows: **(i)** if $x^t \in \mathrm{Var}^t$, then x^t is a term of type t; **(ii)** if $c \in \mathrm{CST}^t$, then c is a term of type t; **(iii)** if ϕ is a terms of type $\langle\rangle$ and for $n \geq 1$, the variables $x_1^{t_1}, ..., x_n^{t_n}$ are pairwise distinct, then $\lambda x_1^{t_1}, ..., x_n^{t_n}.\phi$ is a term of type $\langle t_1, ..., t_n\rangle$; **(iv)** if τ is a term of type $\langle t_1, ..., t_n\rangle$, where $n \geq 1$, and for each $i = 1, ..., n$, σ_i is a term of type t_i, then $\tau(\sigma_1, ..., \sigma_n)$ is a term of type $\langle\rangle$.

The operations at (iii) and (iv) are called, *application* and *abstraction*, respectively. We sometimes drop parentheses when no risk of ambiguity, and write e.g., Fa instead of $F(a)$. We call a term of type $\langle\rangle$ a *formula*, and when it contains no free variables, a *sentence*. We use the letter t with or without subscripts as metavariables for types, lower-case Greek letters $\tau, \sigma, \phi, \psi, ...$ with or without subscripts as metavariables for general terms, and lower-case or capital English letters $x, y, z, p, q, X, Y, Z, P, Q$, with or without subscripts, as metavariables for variables. The notions of *free* and *bound* variables of terms, substitutions of terms for variables, and *being free for a variable* are defined as usual. We show the set of free variables in a term σ by $FV(\sigma)$. Also, the set of all terms of STT is denoted by TERM_s.

From now on, by convention we write things like $\phi \vee \psi$ or $x = y$ to indicate the application instances $\vee(\phi, \psi)$ or $= (x, y)$, respectively. Similarly, quantified statements of the forms $\forall x^t \phi$ and $\exists x^t \phi$ are now construed as shorthands for application instances $\forall^t(\lambda x^t.\phi)$ and $\exists^t(\lambda x^t.\phi)$.

Before attending to the logic of our typed language, let's briefly discuss some of the advantages of our categorematic, constant-based approach to logical statements. Not only this approach is more elegant than the alternative, syncategorematic approach, with fewer axioms or term-formation rules in place and a unified way (i.e., application) to produce logical statements, but it also has the metaphysical advantage of allowing us to intelligibly ask certain questions and theorize about the granularity of the logical connectives and quantifiers—an option that is not available to the rival approach. For instance, one could theorize about whether the operation of disjunction should be treated as a primitive relation or identified with truth-functionally similar properties such as $\lambda pq.\neg(\neg p\wedge\neg q)$ or $\lambda pq.(\neg p\to q)$. To pre-theoretically settle this question is to prejudge matters of grain. But more importantly, to be unable to even ask such questions rigorously would be a loss of expressiveness. It is only the categorematic approach that allows for expressing and defending any of the positions above. This, in effect, constitutes our main reason to choose a categorematic vs. syncategorematic treatment of logical statements in this paper.[18]

We now spell out the proof system for our simple type theory. In what follows, expressions like $\bar{\sigma}_i$ will stand for tuples like $(\sigma_1, ..., \sigma_n)$, and $[\bar{\sigma}_i/\bar{x}_i]$ stands for the simultaneous substitution of σ_i's with x's in τ.[19] Also, in each case, it's been assumed that the substitutants are *free for* the substituent. Intuitively, that guarantees that (i) no bound variable is allowed to be substituted (that is, the notion of substitution only applies to free variables), and (ii) no free variable can get bound after substitution.

Proof System $\mathcal{S}$:

Axioms:

1. Axioms of propositional logic PC
2. $(\lambda x_1^{t_1}, ..., x_n^{t_n}.\phi)(\bar{\sigma}_i) \leftrightarrow [\bar{\sigma}_i/\bar{x}_i]\phi$, where the type of σ_i is t_i $(i = 1, ..., n)$ β_E
3. $\forall^t F \to F\sigma$, where F and σ are, respectively, of types $\langle t\rangle$ and t UI
4. $F\sigma \to \exists^t F$, where F and σ are, respectively, of types $\langle t\rangle$ and t EG
5. $\forall^t(\lambda x^t.\phi\to Fx) \to (\phi \to \forall^t F)$, where F is of type $\langle t\rangle$ and $x \notin FV(\phi)$ UD

[18] With that in mind, one can object to the categorematic approach by saying that, in English, the talk of, e.g., identity, quantification and many other relations and logical operators doesn't seem to be bound to types—we seem to use the same locution of 'is identical to' or 'for all' for most if not all claims regrading individual, properties and relations. So, contra to the popular view, the thought goes, the categorematic might fall short of capturing the talk of properties and quantifiers in English. See footnote 21 for a related discussion regarding the categorematic vs. syncategorematic treatment of e-grounding statements, and a potential, novel reply to these sorts of objections.

[19] For a rigorous definition of substitution, see Mitchell (1996, p. 53). Mitchell's definition is given for functional type theory. Similar definitions can be given for relational type theory.

6. $\forall^t(\lambda x^t.Fx\rightarrow\psi)\rightarrow(\exists^t F\rightarrow\psi)$, where F is of type $\langle t\rangle$ and $x\notin FV(\psi)$ ED

7. $\sigma{=}^t\sigma$, where σ is of type t REF

8. $\sigma{=}^t\tau\rightarrow(F\sigma\rightarrow F\tau)$, where F is of type $\langle t\rangle$ LBZ

Rules of Inference:

9. If $\vdash\phi$ and $\vdash\phi\rightarrow\psi$, then $\vdash\psi$ MP

10. If $\vdash Fx^t$, then $\vdash\forall^t F$, where F is of type $\langle t\rangle$ UG

Now that we have our primary language set up, we need to find a way to rigorously define a suitable notion of syntactic constituency that reflects the sense of constituency that we've seen earlier through examples of structured properties and relations. In other words, we want a syntactic criterion that, whenever applied to any relational term reveals the structure of the thing denoted by the term. For example, we want to find the constituents of the property of being friends with Gary by 'scanning through' the predicate that expresses it in our language, namely $\lambda x^e.F(x,g)$, where F is a constant standing for the relation of friendship, and g is Gary's name. In this specific case, we want to systematically recognize g and the F to be among the constituents of the predicate, because that would reflect the fact that the property in question has Gary and the relation of friendship as a constituent. Such rigorous syntactic specifications matter to us in particular due to the way we have introduced our e-grounding principle S, as relying on the syntax-semantics interplay. Recall that S, in its most general form, says that entities denoted by syntactic expressions are e-grounded by the things picked by the constituents of those expressions.

In general, we take it that many properties and relations that are expressible in our language by the lambda device are structured, and we want to find a general way to specify their constituents through the λ-terms that denote to them. But implementing the ideas of a structuredness and constituency can be perplexing in the presence of λ. For example, even though we may convincingly find the constituents of the property of being friends with Gary from the corresponding λ-term, $\lambda x^e.F(x,g)$, it's not clear how we can pinpoint the constituents (if any) of the property of having every property of individuals from its corresponding λ-term, i.e., the predicate $\lambda x^e.\forall Y^{\langle e\rangle}Y(x)$. An immediate, though naive thought is to take $\forall Y^{\langle e\rangle}Y(x)$ to be a constituent of that predicate. But that sentence doesn't express a unique proposition: depending on what value x takes by an assignment function, it expresses a different proposition.

In general, it's not clear what's the contribution of abstraction or free and bound variables involved in the determination of the constituency of λ-terms. In the rest of the section, we will explore three main options regarding constituency, and choose one of them as the correct definition of constituency. But as will become clear in the end, *all* of them can be used equally well to motivate examples where stratification of relational types is needed. The rest of this section mainly attempts to find the best account of relational constituency among the three options that will be discussed.

We start with the broadest sense of relational constituency, which is the same as being a sub-expression. The idea is to take constituents of terms, in general, just to be their sub-expressions. (Sub-expressions of terms are the terms that contribute to the recursive definition of them, as expressed for all terms in Definition 3.) For instance, by Sub we have it that the predicate $\lambda x^e.F(x,g)$ has F, g and x as constituents. Or that $\lambda x^t.(\forall y^{t'} Fy) \wedge x = a$ has as constituents the sentences $(\forall y^t Fy) \wedge x = a$, $\forall y^t Fy$, $x = a$; plus all of their constituents, i.e., $\wedge$, x, a, $\forall^t, y, Fy$ and F, as well. Similarly, $\lambda x^t.\forall z^t R(z,x)$ has as constituents, $\forall z^t R(z,x)$, $\forall^t$, $z, R(z,x), R$ and x.

Thus here's the first attempt:

- τ is a *constituent* of σ iff τ is a sub-expression of σ. Sub

Sub is the most liberal account of constituency To motivate Sub, remember that in the case of the propositional fragment of e-grounding (see §3), the constituents of a Boolean expression were taken to be the things connected by the relevant connectives: constituents of a conjunctive sentence were taken to be its conjuncts, etc. We can generalize the idea for applicational terms in STT, by taking the constituents of an application term $F(a)$ to be F, a and their respective constituents. One might further expand the notion of constituency of terms, including λ-terms of the general form $\lambda x_1^{t_1}, ..., x_n^{t_n}.\phi$, as well.

But the sub-expressional sense of constituency is too liberal, and in some cases, unmotivated by our metaphysical considerations. While it seems natural to say that the property of being friends with Gary, expressed by $\lambda x^e.F(x,g)$, has Gary and the property of friendship as constituents, it seems implausible to say the same for the variable x, as due to its boundness, none of the values assigned to it seem to have to do anything with the property of being friends with Gary. For example, x could be assigned Mike, but Mike doesn't seem to have anything to do with structure of the property in question—certainly doesn't seem to be a constituent of it.[20]

[20] Of course, one may want to take, e.g., any property to be e-grounded by all the propositions obtained from it. In that case, the property in question will in particular be e-grounded by all propositions obtained from it, and that includes the proposition that Mike is friends with Gary. From S and propositional TR, it will follow that Mike

So we need to impose some restrictions on our initial definition of constituency, for the sake of metaphysical plausibility. The examples above suggest that we need to rule out bound occurrences of variables as constituents. More generally, they suggest that every free occurrence of a variable in the constituent term should also occur free in the term that it's a constituent of. Clearly, limiting constituents to *closed* sub-expressions satisfies this:

- τ is a *constituent* of σ iff τ is a closed sub-expression of σ. Closed Sub

From Closed Sub it follows that the property of being an individual such that Mike is drinking—expressed by $\lambda x^e.Dm$—has the proposition Dm that Mike is drinking, and accordingly, both Mike and the property of drinking as constituents. We will also have it that the property $\lambda x^e.\forall Y^{\langle e \rangle}Y(m)$ of being an individual such that Mike has every individual property is structured and has as a constituent the proposition $\forall Y^{\langle e \rangle}Y(m)$ that Mike has every individual property.

But the closed conception of constituency is a somewhat too restrictive. Consider, for example, $\sigma := \lambda x^t.L(y, x)$, where y is a variable of some type t' and R is a constant of type $\langle t', t \rangle$. With Closed Sub we can say that L is a constituent of σ, but we can't say that about y. But we *would* want the free variable y to be a constituent of σ, because for any value a that y is assigned, a is in fact a constituent of the property picked by $\lambda x^t.L(a, x)$. For instance, if L stands for the relation of loving, the property of being loved by Sarah, $\sigma := \lambda x^t.L(s, x)$, seems to have Sarah ($s$) as a constituent.

So perhaps the best idea is to just hold onto or sharpen the two restrictions that we had ended up with in discussing Sub, as what determines a syntactic notion of constituency that suitably accommodates our favorite sense of constituency that holds between real entities. Let's see some more examples. Suppose $\sigma := \lambda y^{t'}.((\forall x^t Fx) \wedge Gx)$. We would like the universal statement $\forall x^t Fx$ to be a constituent of σ, because it's a closed term. But not Fx, because the free occurrence of x in Fx doesn't occur free in σ. Neither is the occurrence of x in Fx a constituent of σ, for the same reason.

e-grounds the property of being friends with Gary. But this option doesn't seem to sit well with our constructional intuitions of e-grounding. Remember that we took e-grounding to somehow reflect the sense of 'construction' involved in entities; clearly, no such sense can plausibly be given to justify the claim that Mike e-grounds the property of being friends with Gary, or the tentative principle that properties are e-grounded by their propositional values. Similarly, it sounds unmotivated to say that the property denoted by $\lambda x^e.\forall z^e L(z, x)$ is e-grounded by $\forall z^e L(z, x)$: for any assignment of values to variables, this formula returns an entirely different sentence. Suppose L stands for the property of loving. Then for any value a of x, $\forall z^e L(z, a)$ expresses the proposition that everyone loves a. But no such proposition seems to have anything with the 'construction' of the property of being loved by everyone.

On the other hand, Gx, G and the occurrence of x in Gx are all to be construed as constituents of σ because, whatever value they take, that value would seem to be a constituent of the property denoted by σ. Accordingly, for any assignment of values to variables, the entities picked by Gx and x will e-ground the property picked by σ, the latter in virtue of x being a constituent of σ through its free occurrence in Gx.[21]

Below is the general definition of constituency that suitably accommodates all the examples of relational e-grounding that we have been discussing so far:

- An occurrence of a term τ in a term σ is a *constituent occurrence* of τ in σ if τ is a sub-expression of σ and every free occurrence of a variable in τ occurs freely in σ. The term τ is a *constituent* of σ, written $\tau \in \mathrm{c}(\sigma)$, if τ has a constituent occurrence in σ. Cons

Notice that this definition encompasses the sense constituency for sentences as well. That is, a sentence of the form $R(a_1, ..., a_n)$ has as constituents R and all a_i's, simply because they're all sub-expressions of $R(a_1, ..., a_n)$ and every free occurrence of a variable in each of them occurs freely in $R(a_1, ..., a_n)$.

It can be shown that each of Sub, Closed Sub and Cons can motivates the idea of type-stratification for relational types. This means that as soon as we settle on a notion of syntactic constituency for λ-terms from among these three major candidates, we can motivate our desired type stratification. Of course, for the reasons given earlier, our favorite account of constituency will be Cons and we will use examples along those lines. First, we add to the terms language of STT entity grounding statements between any pair of types t_1 and t_2, to obtain $\mathrm{STT}^{\ll}$. The relevant clause is as follows:

- If τ and σ are terms then $\tau \ll \sigma$ is a term of type $\langle\rangle$.

Notice that here we are treating statements of e-grounding syncategorematically. Alternatively, we could treat them categorematically and take statements of e-grounding to be obtained by, e.g., typed constants (standing for relations of e-grounding) that apply to entities of appropriate types. More specifically, for any pair of types t_1 and t_2 we could associate a constant $\ll_{t_1 t_2}$ of type $\langle t_1, t_2\rangle$ and construe statements of e-grounding $a \ll_{t_1 t_2} b$

[21] This is plausible, especially because due to the principle of 'α-conversion', according to which terms with corresponding bound variables of different names are the same, σ is can be identified with $\lambda y^{t'}.((\forall z^t Fz) \wedge Gx)$. In this α-equivalent variant of σ, x, but not z, still plausibly is a constituent of the term. Such α-equivalent representations of terms may well allow for redefining constituency in terms of variables, instead of variable occurrences.

are in fact abbreviations for applications of the form $\ll_{t_1 t_2}(a, b)$, similar to what we did for the logical vocabulary.

Our syncategorematic treatment of e-grounding statements is mainly because our pre-theoretic talk of e-grounding (as introduced in §2) doesn't discriminate against entities of different types; it appeals to a unified notion that runs across reality. It's the same locution all over as if we are talking about the same relation that holds between entities of different types. So a syncategorematic treatment of e-grounding statements seems closer to our pre-theoretic conception and use of the notion.[22]

In any case, we can now express our desired principles of e-grounding in the extended language, to obtain $\vdash^{\mathrm{STT}^{\ll}}$, which is just $\vdash^{\mathrm{STT}}$ plus the following axiom schemata:

- $(\tau \ll \sigma \wedge \sigma \ll \gamma) \to \tau \ll \gamma$ TR
- $\neg(\tau \ll \tau)$ IR
- $\tau \ll \sigma \to \neg\sigma \ll \tau$ AS
- $\tau \ll \sigma$, if $\tau \in \mathrm{c}(\sigma)$ S
- $\tau \ll \forall x^t \phi \wedge \tau \ll \exists x^t \phi$, where τ is of type t and $x \in FV(\phi)$ Q

[22] One might object to the syncategorematic formulation of e-grounding statements by saying that there is no such relation out there in the reality, after all, that, e.g., would contribute to the truth of statements of e-grounding; at best, there are infinitely many such relations that do the job (and that has to be cashed out on the alternative, categorematic approach). In response to this, although one should admit the structure of the type theories in this paper, and in general in the philosophical literature, don't allow for a unique relation that ignores type differences, that's hardly a unique problem for e-grounding, or any other notion of grounding, for that matter. The same issue can be raised regarding categorematic vs. syncategorematic treatments of the logical vocabulary or identity. This has to do with the design of the kind of type systems that most philosophers use, such as simple and ramified type systems, where no cross-type term can be expressed. There are, however, more recent type theories, though so far mostly in the service of mathematicians and computer scientists, that allow for such entities. For instance, $\lambda 2$ or *System F* is among such type systems, otherwise known as *Dependent Type Theories*. These type systems raise above the type restrictions built into the kind of type theories entertained here, and in general by philosophers, and allow for terms and their denotations that aren't sensitive to the choice of simple types. We believe that the notion of e-grounding, as well as various other notions of grounding that call for ramification, and, in fact, many other relations outside the context of metaphysical priority (e.g., identity and existence), are best captured by such general systems. For a recent argument in favor of System F as the right framework to capture the talk of identity, existence, quantification and various other typed relations, see Anonymous (MS[a]). For a general introduction to dependent type theories see Nederpelt and Geuvers (2014). More detailed discussions of System F and their applications can be found at Girard et al. (1989) and Mitchell (1996, Chapter 9).

We can now see exactly why we need relational ramification. Consider, for example, the property P of being Mike such that Mike has some property, expressed by $\lambda x^e.(x = m \wedge \exists Y^{\langle e\rangle} Y(m))$. We argued at the beginning of this section that the structure of this property calls for an infinitary hierarchy of individual properties. Using Cons and the principles of e-grounding above, this can be shown more rigorously. Notice that according to our definition of relational constituency, P has the proposition $\exists Y^{\langle e\rangle} Y(m)$ that Mike has some property as a constituent, so by S they are e-grounded by it. On the other hand by Q the proposition itself is e-grounded by *all* properties of individuals. A contradiction follows from applying UI and TR. Put formally, we have the following:[23]

Theorem 2. $\varnothing \vdash^{\mathrm{STT}^{\ll}} \bot$

Proof.

$$\begin{array}{lll}
(1) & \lambda x^e.(x = m \wedge \exists Y^{\langle e\rangle} Y(m)) \ll \exists Y^{\langle e\rangle} Y(m) & \text{Q} \\
(2) & \exists Y^{\langle e\rangle} Y(m) \ll \lambda x^e.(x = m \wedge \exists Y^{\langle e\rangle} Y(m)) & \text{S} \\
(3) & \lambda x^e.(x = m \wedge \exists Y^{\langle e\rangle} Y(m)) \ll \lambda x^e.(x = m \wedge \exists Y^{\langle e\rangle} Y(m)) & \text{TR 1, 2} \\
(4) & \neg[\lambda x^e.(x = m \wedge \exists Y^{\langle e\rangle} Y(m)) \ll \lambda x^e.(x = m \wedge \exists Y^{\langle e\rangle} Y(m))] & \text{IR} \\
(5) & \bot & \text{PC 3, 4} \quad \square
\end{array}$$

This Theorem essentially formalizes the inconsistency result outlined at the beginning of the present section. The rest of the story is similar to the previous section. We have *assumed* that the language of simple type theory can capture our stipulative talk of e-grounding. We have then run into contradictions when formulating our desired principles in this language. In line with our arguments at the beginning of the section, the most e-ground-friendly resolution to the problem at stake is to segregate the property picked by $\lambda x^e.\exists Y^{\langle e\rangle} Y(m)$ and the ones it quantifies over, so we implement similar revisions in our syntax.

More specifically, we replace the type $\langle e\rangle$ with two types $\langle e\rangle/1$ and $\langle e\rangle/2$, the first one assigned to predicates that pick the 'building-block' individual properties, and the second one for the predicates that pick the 'buildings'. As a result, we will be able to revise our term-formation rules in a way that, e.g., $\lambda x^e.\exists Y^{\langle e\rangle/1} Y(m)$ will be of type $\langle e\rangle/2$, and so on. As expected, the proof system needs to also be calibrated, accordingly.

As expected, this process improving upon languages and running into inconsistencies leads to positing an infinite array of newer and newer leveled types $\langle e\rangle/1, \langle e\rangle/2, \langle e\rangle/3, \ldots$ for predicates that pick different kinds of individual properties, and $\langle\rangle/1, \langle\rangle/2, \langle\rangle/3, \ldots$ for sentences that express different kinds of

[23] One who goes only as far as committing to the restrictive Closed Sub could use the term $P := \lambda x^e.\exists Y^{\langle e\rangle} Y(m)$ and run similar arguments.

propositions. In general, we can run similar arguments for different expressions of different relational types of the form $\langle t_1, ..., t_n\rangle$, for any $n \geq 0$, and end up with an infinite hierarchy of types $\langle t_1, ..., t_n\rangle/1, \langle t_1, ..., t_n\rangle/2, \langle t_1, ..., t_n\rangle/3, ...$ that behave in the way expected. In the next section, we propose a formal language and logic that fully accommodates the syntactic changes glossed here.

5 Ramified Type Theory

We now introduce a system ramified types based on the previous discussions, in its most general form. First, let's introduce ramified types and their levels:

Definition 4 (Ramified Types and Levels). The set $\mathcal{T}^r$ of *ramified types* t and their levels $\mathrm{l}(t)$ are simultaneously defined as follows: $e \in \mathcal{T}^r$ with $\mathrm{l}(e) = 0$, and for $t_0, ..., t_n \in \mathcal{T}^r$ and $m \geq 1$, if $\mathrm{l}(t_i) \leq m$ for each $i = 0, ..., n$, then $\langle t_0, ..., t_n\rangle/m \in \mathcal{T}^r$, with $\mathrm{l}(\langle t_1, ..., t_n\rangle/m) = m$.

In effect, e is the type of individuals, and for any types $t_1, ..., t_n$, where $n \geq 0$, $\langle t_1, ..., t_n\rangle/m$ is the type of n-ary propositional functions of level m—functions that, as the term-formation rules below show, take arguments of types $t_1, ..., t_n$, respectively, and return an level-m proposition. The type of level-m propositions is obtained as the limiting case of the relational types, when $n = 0$, and is represented by $\langle\rangle/m$.

As before, for any ramified type $t \in \mathcal{T}^r$ we assume there's a denumerably infinite set of *variables* Var^t of type t and a (possibly empty) set of typed *constants* CST^t. We reserve CST^t for the set of all constants) of type t. We define the sets of all variables and constants respectively as $\mathrm{Var} := \bigcup_{t \in \mathcal{T}^r} \mathrm{Var}^t$ and $\mathrm{CST} := \bigcup_{t \in \mathcal{T}^r} \mathrm{CST}^t$. We also represent the set of t-type terms with TERM^t.

As before, we choose the constant-based approach to introduce our logical vocabulary. In line with our discussions of levels from before, we choose our typed, logical constants in RTT, as follows: $\neg_m$ is of type $\langle\langle\rangle/m\rangle/m$; $\rightarrow_{m_1,m_2}$, $\leftrightarrow_{m_1,m_2}$, $\vee_{m_1,m_2}$ and $\wedge_{m_1,m_2}$, each of type $\langle\langle\rangle/m_1, \langle\rangle/m_2\rangle/\max\{m_1, m_2\}$; and, to repeat, for any ramified type t, $\forall^t_m$ is of type $\langle\langle t\rangle/m\rangle/\max\{\mathrm{l}(t)+1, m\}$. As for the identity operator in RTT, for any t we reserve a constant $=^t_r$ of type $\langle t, t\rangle/\max\{1, \mathrm{l}(t)\}$. Notice that since identity statements are essentially formulae, the minimum level they can take should be 1. Notice also that the type of the universal quantifier constant $\forall^t_m$ is determined through the convention $\forall x^t \phi := \forall^t_m(\lambda x^t.\phi_m)$ and level conventions of λ-terms (as introduced below).

Definition 5 (Ramified Terms). The *terms* of RTT are recursively defined as follows: **(i)** If $x^t \in \mathrm{Var}^t$, then x^t is a term of type t; **(ii)** if $c \in \mathrm{CST}^t$, then c is a term of type t; **(iii)** if $x_1 \in \mathrm{Var}^{t_1}, ..., x_n \in \mathrm{Var}^{t_n}$ are pairwise distinct, where $n \geq 1$ and $\mathrm{l}(t_i) \leq m$ for each t_i, and ϕ is a term of type $\langle\rangle/m$, then $\lambda x_1^{t_1}, ..., x_n^{t_n}.\phi$ is a term of type $\langle t_1, ..., t_n\rangle/m$; **(iv)** if τ is a term of type $\langle t_1, ..., t_n\rangle/m$, where $n \geq 1$, and for each $i = 1, ..., n$, τ_i is a term of type t_i, then $\tau(\tau_1, ..., \tau_n)$ is a term of type $\langle\rangle/m$.

The notions of *free* and *bound* variables of terms, *substitutions* of terms for variables and *being free for a variable* are defined as usual. We denote the set of free variables in a term σ by $FV(\sigma)$, and the set of all terms of ramified type theory by TERM_r.[24] We also adopt similar conventions about meta-variables for variables, terms and types as before.

We now introduce the proof system for our ramified language, which we named *System $\mathcal{R}$*.

PROOF SYSTEM $\mathcal{R}$:

Axioms:

1. Leveled appropriate axioms of propositional logic PC$_r$
2. $(\lambda x_1^{t_1}, ..., x_n^{t_n}.\phi_m)(\bar{\sigma}_i) \leftrightarrow [\bar{\sigma}_i/\bar{x}_i]\phi_m$, where the type of σ_i is t_i β_{E_r}
3. $\forall_m^t F \to F\sigma$, where F and σ are, respectively, of types $\langle t\rangle/m$ and t UI$_r$
4. $F\sigma \to \exists_m^t F$, where F and σ are, respectively, of types $\langle t\rangle/m$ and t EG$_r$
5. $\forall_{n^*}^t(\lambda x^t.\,\phi_m \to Fx) \to (\phi_m \to \forall_n^t F)$, where F is of type $\langle t\rangle/n$, $n^* = \max\{m, n\}$ and $x \notin FV(\phi_m)$ UD$_r$
6. $\forall_{n^*}^t(\lambda x^t.Fx \to \psi_m) \to (\exists_n^t F \to \psi_m)$, where F is of type $\langle t\rangle/n$, $n^* = \max\{m, n\}$ and $x \notin FV(\psi_m)$ ED$_r$
7. $\sigma =_r^t \sigma$, where σ is of type t Ref$_r$
8. $\sigma =_r^t \tau \to (F\sigma \to F\tau)$, where F is of type $\langle t\rangle/m$ LBZ$_r$

Rules of Inference:

9. If $\vdash \phi_m$ and $\vdash \phi_m \to \psi_n$, then $\vdash \psi_n$ MP$_r$
10. If $\vdash Fx^t$, then $\vdash \forall_m^t F$, where F is of type $\langle t\rangle/m$ UG$_r$

[24] Notice that our ramified types and terms, as introduced here are very similar to Harold Hodes's System $\Rightarrow^{nr}$, as introduced in Hodes (2013). What we consider as level here is called 'order' by Hodes, and that in our system, but not Hodes's vacuous lambda abstraction is possible.

Notice that each of the axioms and rules of inference above are multiply schematic. For example in PC_r, the axioms hold for any sentence of any level, and the relevant instances of $\neg$ and $\rightarrow$ may differ in type and should be typed carefully. In particular, notice that in UI_r, LBZ_r and UG_r are all schematic in multiple ways: in the occurrence of the terms, types t *and* the level m of the relational types $\langle t\rangle/m$.

Finally, we express the principles of e-grounding in RTT. To do this, we first extend RTT to $\mathrm{RTT}^{\ll}$ by adding the following clause:

- If τ and σ are terms, then $\tau \ll \sigma$ if a term of type $\langle\rangle$.

The e-grounding system looks like what we introduced at the beginning of this section, but now with the types being schematic for different types (individuals, propositional and relational) *and* levels (1, 2, 3, ...). We call the augmentation of $\mathcal{R}$ with the following axiom schemata, resulting in $\mathcal{R}^{\ll}$ or what we also call *System* $\mathcal{G}$:

1. $(\tau \ll \sigma \wedge \sigma \ll \gamma) \rightarrow \tau \ll \gamma$ $\qquad$ TR_r
2. $\neg(\tau \ll \tau)$ $\qquad$ IR_r
3. $\tau \ll \sigma \rightarrow \neg\sigma \ll \tau$ $\qquad$ AS_r
4. $\tau \ll \sigma$, if $\tau \in \mathrm{c}(\sigma)$ $\qquad$ S_r
5. $\tau \ll \forall x^t\phi \wedge \tau \ll \exists x^t\phi$, where τ is of type t and $x \in FV(\phi)$ $\qquad$ Q_r

We conclude the paper with some final remarks. First, notice that in Definition 5 we assumed that the level of the arguments $\tau_0, ..., \tau_n$ of a relational type $\langle\tau_0, ...\tau_n\rangle/m$ are no higher than the level of type, i.e., m. We can see now why we had to make this choice. Suppose, on the contrary, that $\langle\langle\rangle/3\rangle/2$ is a legit type and entities of this type could apply to entities of type $\langle\rangle/3$ in order to produce $\langle\rangle/2$-type propositions. Now, let $F := \lambda p^{\langle\rangle/3}.\phi_2$, where ϕ_2 is any sentence of type $\langle\rangle/2$, and let $\phi := \forall p^{\langle\rangle/2}p$. Then $F(\phi)$ will be of type $\langle\rangle/2$. But by S, we have $\phi \ll F(\phi)$, and by Q, all level-2 propositions would e-ground ϕ. A contradiction then follows from IR and TR: by TR all level-2 propositions would e-ground $F(\phi)$, and that includes $F(\phi)$ itself, which goes against IR. Similar examples can be given if types like $\langle\langle\rangle/3\rangle/1$ were allowed, whereas types of the form $\langle\langle\rangle/3\rangle/n$ for any $n \geq 3$ are safe to inhabit.

The second remark concerns our constant-based, categorematic presentation of the logical vocabulary in this section and the previous one. We mentioned that there are advantages in this approach, both concerning presentation (more elegance and convenience in defining terms and specifying axioms) and expressiveness for metaphysical theorizing. That said, however, one might be skeptical if the constant-based approach is the best one when, in particular, it comes to *ramified* type systems principles. For instance, given the abundance of levels, the constant-based approach requires a very big ontology, with infinitely many entities sitting out there to just do the job of, say, negation. Similarly, we need considerably more axioms, compared to the syncategorematic approach, each crafted for certain levels, in laying out the proof system. These considerations might make the constant-based approach in ramified type theory lose attraction to sparser ontologies within certain big-picture considerations. This might also be why most of the works in the literature on ramified type systems (including Russell's original works) choose the syncategorematic approach to the logical vocabulary. Moreover, from a purely e-ground-theoretic perspective, it may sound somewhat mysterious that some but not other constants raise levels of the things they apply to.

One might think that these go against one of the primary motives of adapting a categorematic approach towards ramified type theory, and wonder if it's possible to present our ramified system syncategorematically while somehow retaining the relevant formalizations regardless of e-grounding. This seems possible. In fact, this is the approach we took in §3, though mostly for convenience. Here we can also associate in our term-formation rules (Definition 1) separate clauses for logical terms. But then we will have to make sure that the notion of constituency (Cons) is also extended with enough clauses concerning logical statements and their constituents. It should be noted, however, that this approach will no longer allow for the attractive thought that logical entities can enter into e-grounding relations. For instance, there is no longer a relation of conjunction that can be said to e-ground a conjunctive proposition. In any case, we leave it open which choice is more appropriate here, all things considered.

Finally, in this paper we argued that simple *relational* entities each have to come in certain infinitary hierarchies of levels, for the principles of e-grounding to go through without facing immediate inconsistencies. One might wonder if individuals should also be fragmented into levels, similarly to relations. Even though this is formally possible (for instance, Bacon et al. 2016, do this), such a move seems unmotivated by the metaphysical views that we have been appealing to, for if a propositional function contains a proposition that quantifies over individuals, then whatever it refers to is *not* an individual: it's a proposition, property or relation. That is to say, the type of propositional functions are *already* different from the type e of individuals, whether or not they quantify over individuals in their structure. That said, however, some might have independent reasons to stratify individuals into levels, as well, e.g., by considering certain mereological relations that hold between them as instances of e-grounding. We also leave that possibility open for future investigations.

6 Conclusion

I argued that considerations of e-grounding, as presented in this paper, naturally call for fragmentation of relational entities into certain infinitary hierarchies of levels, in a way that is best captured by ramified type systems. I proposed a natural ramified type system that nicely captures the principles of e-grounding.

Several problems remain open, which we hope to attend to in the future. We haven't yet verified the consistency of the systems $\mathcal{R}$ and $\mathcal{G}$. In particular, while the consistency of the former is proved in (Anonymous MS[c]), the consistency of the latter still awaits proof. Another issue concerns the choice between categorematic vs. syncategorematic representations of logical statements as well as e-grounding statements. Even though throughout the paper, and for a combination of reasons, we made a certain choice in this regard, namely, a categorematic treatment of logical, and a syncategorematic treatment of e-grounding statements, we also mentioned that alternative options are available, without any serious impact on our arguments for the ramification of relational types, though, each choice has its own pros and cons. We, however, leave it open which choice of options is more appropriate under large-scale considerations.

A somewhat larger open problem concerns the general choice between ramified versus simple type theories as the 'correct' framework in pursuing philosophical, and in particular, metaphysical inquiry. We mentioned that the ramified approach is naturally motivated by considerations of e-grounding, and also settles a cluster of contemporary puzzles and paradoxes of ground and grain in a unified way. Some other puzzles in philosophy and logic have also been proposed ramified solutions (see, e.g., Kaplan 1995; Kripke 2011; Prior 1961; Russell 1908; Whitehead and Russell 1910). This speaks to the immense and unified explanatory power of the ramified approach in doing philosophy, and in particular, metaphysics. But simple type theory has been proved more fruitful in certain other areas, such as in mathematics, where, r.g., those systems can be enriched with certain axioms to serve as a foundation for classical mathematics (see, e.g., Church 1940, as an early work along these lines). Also, some major projects, especially in the recent literature on the metaphysics of modality, have been carried out in simple type systems and supposedly seriously rely on their full expressive power (e.g., Bacon 2018; Williamson 2013).

Finally, we noted that, unless we construe quantificational statements as 'long' conjunctions or disjunctions, and accordingly the principle Q as an instance of S, one doesn't need to embrace structured propositions in order to be receptive to the idea that propositions, along with properties and relations, have to come in infinitary levels, as described by ramified type theory. All that's needed is that other, non-propositional types of relational entities are structured in certain plausible ways and that principles of e-grounding are true. But, and perhaps more importantly, we noted that even if propositions are structured, the principles of e-grounding don't require them to be too structured to be susceptible to paradoxes of grain such as the Russell-Myhill theorem. This makes it possible to have coarse-grained views about propositions, and yet admit that they have to come in a ramified hierarchy. It also puts those who reject structured propositions based on paradoxes of grain in an awkward position, as they now have independent motives to endorse a ramified space of propositions, which if they do, they can secure highly structured propositions, as well.

Only future work on both ramified and simple type systems and their large-scale metaphysical implications will determine which one, if any, is to be preferred as the correct framework for pursuing metaphysical inquiry.

References

Adamek, J., & Rosicky, J. (1994). *Locally Presentable and Accessible Categories.* Cambridge University Press.

Anonymous, A. (MS[a]). Categorematicity and Type-Insensitive Relations: Towards Polymorphic Metaphysics and Beyond [Unpublished Paper Manuscript].

Anonymous, A. (MS[b]). Ramification and Ground: Towards a Unified Predicative Solution to Puzzles of Quantificactional Ground [Unpublished Paper Manuscript].

Anonymous, A. (MS[c]). Ramified Types and Structured Propositions [Unpublished Paper Manuscript].

Anonymous, A. (MS[d]). Russellian Presuppositions, Vicious Circle Principles and Ramified Type Theory [Unpublished Paper Manuscript].

Anonymous, A. (MS[e]). Structured Propositions and a Semantics for Unrestricted, Extended Impure Logics of Ground [Unpublished Paper Manuscript].

Audi, P. (2012). Grounding: Toward a Theory of the In-Virtue-Of Relation. *Journal of Philosophy*, *109*(12), 685–711.

Bacon, A. (2018). The Broadest Necessity. *Journal of Philosophical Logic*, *47*(5), 733–783.

Bacon, A., Hawthorne, J., & Uzquiano, G. (2016). Higher-order Free Logic and the Prior-Kaplan Paradox. *Canadian Journal of Philosophy*, *46*(4), 493–541.

Bennett, K. (2011). By Our Bootstraps. *Philosophical Perspectives*, *25*(1), 27–41.

Church, A. (1940). A Formulation of the Simple Theory of Types. *The Journal of Symbolic Logic*, *5*(2), 56–68.

Correia, F. (2017). An Impure Logic of Representational Grounding. *Journal of Philosophical Logic*, *46*(5), 507–538.

deRosset, L. (2013). Grounding Explanations. *Philosophers' Imprint*, *13*.

Donaldson, T. (2017). The (Metaphysical) Foundations of Arithmetic? *Noûs*, *51*(4), 775–801.

Dorr, C. (2016). To be F is to be G. *Philosophical Perspectives*, *30*.

Fine, K. (1970). Propositional Quantifiers in Modal Logic. *Theoria*, *36*(3), 336–346.

Fine, K. (1995). Ontological Dependence. *Proceedings of the Aristotelian Society*, *95*, 269–290.

Fine, K. (2010). Some Puzzles of Ground. *Notre Dame Journal of Formal Logic*, *51*(1), 97–118.

Fine, K. (2012). Guide to Ground. *Metaphysical Grounding: Understanding the Structure of Reality* (pp. 37–80). Cambridge University Press.

Fine, K. (2017). Truthmaker Semantics. In B. Hale, C. Wright, & A. Miller (Eds.), *A Companion to the Philosophy of Language* (pp. 556–577).

Fritz, P. (2021). Ground and Grain. *Philosophy and Phenomenological Research*.

Fritz, P. (MS). Breaking the Domination of the Word over the Human Spirit [Unpublished Manuscript].

Girard, J.-Y., Taylor, P., & Lafont, Y. (1989). *Proofs and Types*. Cambridge University Press.

Gödel, K. (1984 [1944]). Russell's mathematical logic. In H. Putnam & P. Benacerraf (Eds.), *Philosophy of Mathematics: Selected Readings* (2nd ed., pp. 447–469). Cambridge University Press.

Goodman, J. (2016). Reality is Not Structured. *Analysis*, *77*(1), 43–53.

Henkin, L. (1950). Completeness in the Theory of Types. *The Journal of Symbolic Logic*, *15*(2), 81–91.

Hodes, H. T. (2013). A Report on Some Ramified-Type Assignment Systems and Their Model-Theoretic Semantics. *The Palgrave Centenary Companion to Principia Mathematica* (pp. 305–336). Springer.

Hodes, H. T. (2015). Why Ramify? *Notre Dame Journal of Formal Logic*, *56*(2), 379–415.

Hylton, P. (2008). The Vicious Circle Principle. *Propositions, Functions, and Analysis: Selected Essays on Russell's Philosophy*. Oxford University Press.

Jung, D. (1999). Russell, Presupposition, and the Vicious-Circle Principle. *Notre Dame Journal of Formal Logic*, *40*(1), 55–80.

Kaplan, D. (1970). S5 with quantifiable propositional variables. *Journal of Symbolic Logic*, *35*(2), 355.

Kaplan, D. (1995). A Problem in Possible Worlds Semantics. In W. Sinnott-Armstrong, D. Raffman, & N. Asher (Eds.), *Modality, Morality and Belief: Essays in Honor of Ruth Barcan Marcus* (pp. 41–52). Cambridge University Press.

Kaplan, D. (1989[1977]). Demonstratives: An Essay on the Semantics, Logic, Metaphysics and Epistemology of Demonstratives and Other Indexicals. In J. Almog, J. Perry, & H. Wettstein (Eds.), *Themes From Kaplan* (pp. 481–563). Oxford University Press.

King, J. (1996). Structured Propostions and Sentence Structure. *Journal of philosophical logic*, *25*(5), 495–521.

King, J. C. (2009). *The Nature and Structure of Content.* Oxford University Press.

Krämer, S. (2013). A Simpler Puzzle of Ground. *Thought: A Journal of Philosophy*, *2*(2), 85–89.

Krämer, S. (2018). Towards a Theory of Ground-Theoretic Content. *Synthese*, *195*(2), 785–814.

Kripke, S. A. (2011). A Puzzle about Time and Thought. In S. A. Kripke (Ed.), *Philosophical Troubles. Collected Papers.* Oxford University Press.

Mitchell, J. C. (1996). *Foundations for Programming Languages.* MIT Press.

Myhill, J. (1958). Problems Arising in the Formalization of Intensional Logic. *Logique et Analyse*, *1*(2), 74–83.

Nederpelt, R., & Geuvers, H. (2014). *Type Theory and Formal Proof: An Introduction.* Cambridge University Press.

Prior, A. N. (1961). On a Family of Paradoxes. *Notre Dame Journal of Formal Logic*, *2*(1), 16–32.

Quine, W. (1963). *Set Theory and Its Logic: Revised Edition* (II). Harvard University Press.

Rosen, G. (2010). Metaphysical Dependence: Grounding and Reduction. *Modality: Metaphysics, Logic, and Epistemology* (pp. 109–36). Oxford University Press.

Russell, B. (1903). *The Principles of Mathematics.* Cambridge University Press.

Russell, B. (1908). Mathematical Logic as Based on the Theory of Types. *American Journal of Mathematics*, *30*(3), 222–262.

Russell, B. (2009[1918]). *The Philosophy of Logical Atomism* (1st Edition). Routledge.

Schaffer, J. (2009). On What Grounds What. In D. Manley, D. J. Chalmers, & R. Wasserman (Eds.), *Metametaphysics: New Essays on the Foundations of Ontology* (pp. 347–383). Oxford University Press.

Soames, S. (1987). Direct Reference, Propositional Attitudes, and Semantic Content. *Philosophical Topics*, *15*(1), 47–87.

Uzquiano, G. (2015). A Neglected Resolution of Russell's Paradox of Propositions. *The Review of Symbolic Logic*, *8*(2), 328–344.

Whitehead, A. N., & Russell, B. (1910). *Principia Mathematica* (1st ed., Vol. 1). Cambridge University Press.

Wilhelm, I. (2020a). An Argument for Entity Grounding. *Analysis*, *80*(3), 500–507.

Wilhelm, I. (2020b). Grounding and propositional identity. *Analysis*.

Williamson, T. (2013). *Modal Logic as Metaphysics*. Oxford University Press.