THE LOGIC OF QUANTIFICATIONALISM, PART 1: FOUNDATIONS

ANTONIO MARIA CLEANI

Contents

Draft of January 13, 2026.

1. Introduction

Consider the following sentences:

(1) No country has all five of Earth's climate zones.
(2) In 1783, no country had all five of Earth's climate zones.
(3) Had territories west of the Mississippi not been colonized, no country would have had all five of Earth's climate zones.
(4) With the exception of the United States, no country has all five of Earth's climate zones.

In fact, the United States has all five of Earth's climate zones. That makes (1) false. But no other country enjoys this much climate diversity, which makes (4) true. Moreover, dry climates in North America only occur west of the Mississippi River, which the 1783 Treaty of Paris identified as the western boundary of the United States. That makes (2) and (3) also true.

A natural explanation of the truth of (2) despite the falsity of (1) consists in the claim that (1) expresses a *temporary falsehood*: a proposition that is false, but not *eternally* false. This proposition is false now, but was not false in 1783. Likewise, we can explain the truth of (3) despite the falsity of (1) by claiming that (1) expresses a *contingent falsehood*: a proposition that is false, but could have been true. This proposition is actually false, but would have been true had territories west of the Mississippi not been colonized.

The key insight that truth can vary across times and worlds is at the heart of mature research programs in philosophical logic, centered on the study of tense and modal logics. In these traditions, temporal and modal reasoning are formalized by means of sentential operators, regimenting natural language expressions like 'always,' 'in 2001,' 'necessarily,' and 'had territories west of the Mississippi not been colonized.' The logic of modal and tense operators is by now well understood, and tense and modal logic serve as rigorous frameworks for formalizing a number of philosophical debates concerning time and modality.

The superficial similarity of (4) with (2) and (3) suggests a parallel explanation of the variation in truth value between (1) and (4). The thought here is that the falsehood expressed by (1) is not only temporary and contingent, but also *quantificationally relative*: it is false, though not false *at* or *relative to* all domains of quantification. In particular, this proposition is true relative to the domain of countries other than the United States.

Despite the natural parallel, the idea that truth is quantificationally relative has received virtually no attention. This paper is part of a larger project that aims to change this. Here, I focus on the problem of giving the idea solid formal foundations, comparable to what the frameworks of modal and tense logic are to contingent and temporary truth.

Let me unpack the parallel further. Let *Temporalism* be the thesis that there are temporary truths—truths that are not always true—and let *Modalism* be the thesis that there are contingent truths—truths that are not necessarily true. Since the negation of a temporary truth is a temporary falsehood, Temporalism is equivalent to the thesis that there are temporary falsehoods. Likewise, Modalism is equivalent to the thesis that there are contingent falsehoods. Thus (1)—(3), if the explanations given in the previous paragraphs are correct, witness the truth of both Temporalism and Modalism. Accordingly, let us call these explanations the *temporalist* and *modalist explanations* of the truth values of (1)—(3).

Let me be clear about what Temporalism and Modalism say, as I understand them. I take the glosses of Temporalism and Modalism just given to be shorthands for the following higher-order generalizations.[1]

Temporalism: $\exists p(p \wedge S\neg p)$.
Modalism: $\exists p(p \wedge \Diamond\neg p)$.

Here 'S' means 'sometimes' (the dual of 'always') and $\Diamond$ means 'possibly.' Thus (Temporalism) and (Modalism) assert, respectively, the *non-triviality* of tense and modal operators, where an operator is *non-trivial* when it can turn a truth into a falsehood (or vice versa).[2]

It is helpful to contrast these explanations to another sort of account, common among a family of reductionist views about tense and modality. On this alternative account, somewhere in the syntax of (1), there are covert occurrences of variables ranging over times and worlds. Modifiers like 'in 1783' and 'had territories west of the Mississippi not been colonized' work by binding and valuing such variables. When these variables are not bound, they receive their value from the context of utterance. Let us call this explanation the *reductionist* explanation.[3]

Call a country *climate diverse* when it has all five of Earth's climate zones. On the reductionist explanation, since the variables over times and worlds that occur covertly in (1) are not bound, (1) expresses the proposition that no country is climate diverse *in 2026 at the actual world*. This proposition is not only false, but eternally and necessarily so. On the other hand, covert time variables in (2) are bound by 'in 1783' in (2), so that (2) expresses the proposition that no country was climate diverse *in 1783* (at the actual world). Likewise, covert world variables in (1) are bound by 'had territories West of the Mississippi not been colonized' in (3), so that (3) expresses the proposition that no country is climate diverse (in 2026) at the counterfactual situation territories west of the Mississippi have not been colonized, where the latter is rigidly specified. These propositions, too, are eternally and necessarily true.

Consequently, on the reductionist explanation, the falsity of (1) alongside the truth of (2) and (3) does not license the generalizations Temporalism and Modalism.

---

[1]As is customary in the higher-order metaphysics literature, I adopt the convention of pronouncing higher-order quantification using English expressions that sound like first-order quantification over properties and propositions. This pronunciation convention should be taken to indicate a reductionist attitude towards higher-order quantification.

[2]Temporalism and Modalism resemble but are subtly distinct from a number of views falling under the same label in the literature. There are well established views in the literature that superficially resemble Temporalism and Modalism, but are formulated in terms of first-order quantification over propositions instead of quantification in sentence position. See, for example, Kaplan [1989]; Lewis [1980]; Brogaard [2012]; Schaffer [2012]. Prior and Fine [1977] endorse both Temporalism and Modalism, but use these labels to refer to stronger theses that, in addition, entail that the operators S and $\Diamond$ cannot be reduced to quantification over times and possible worlds. To my knowledge, Temporalism as understood here is only discussed in Bacon [2018]; Dorr and Goodman [2020], whereas Modalism is not discussed as a "free-standing" thesis at all.

[3]See, e.g., Mellor [1981]; [Sider, 2001, Ch. 2.1] and Schaffer [2012].

Due to variable binding, the embedded occurrences of (1) in (2) and (3)—if they express a proposition at all—do not express the same proposition as (1) does when it occurs unembedded. The existence of temporary and contingent falsehoods does not thereby follow on the reductionist account.

Return now to the temporalist and modalist explanations. There is a completely analogous story we can tell about why (1) is false and (4) is true. We can say that (1) expresses—for lack of a better term—a *quantificationally relative falsehood*: a proposition that is false, but not false *relative to all domains of quantification.* This proposition in fact false, but it is true relative to the restricted domain of countries other than the United States. As for tense and modality, so for quantifier domain specification.

Let *Quantificationalism* be the thesis that there are quantificationally relative truths—truths that are not true relative to all domains of quantification. Since the negation of a quantificationally relative truth is a quantificationally relative falsehood, Quantificationalism is equivalent to the claim that there are quantificationally relative falsehoods. On the explanation just given—call it the *quantificationalist explanation*—(1) and (4) witness the truth of Quantificationalism.

Continuing the analogy with Temporalism and Modalism, I take the gloss of Quantificationalism just given to be a shorthand for the following higher-order generalization.

Quantificationalism: $\exists p(p \wedge \blacklozenge\neg p)$.

Here $\blacklozenge$ is a (primitive) sentential operator regimenting the notion of *truth at some domain of quantification*. Thus Quantificationalism asserts the non-triviality "quantificational operators," which, intuitively, are to domains of quantification what tense and modal operators are to times and worlds.

As we did for the temporalist and modalist explanations, we can contrast the quantificationalist explanation with a *reductionist* account. On this view, the syntax of (1) contains covert occurrences of variables ranging over domains of quantification. These variables get their values from the context of utterance when they occur free, as in (1), but can be bound and valued by modifiers like 'with the exception of the United States,' as in (4). Thus, (1) expresses the proposition that no country *in* d is climate diverse, where d rigidly picks out the current domain of quantification. This proposition is not only false, but false relative to all domains of quantification. On the other hand, (4) expresses the proposition that no country *in the domain of countries excluding the United States* is climate diverse, which is true relative to all domains.[4] Thus, on the reductionist account, Quantificationalism does not follow from the falsity of (1) and the truth of (4).

Unlike Temporalism and Modalism, Quantificationalism has received no philosophical attention. I think this is undeserved, because it is a philosophically fruitful thesis. As the introductory examples just discussed suggest, Quantificationalism can serve as the background for a novel analysis of exceptive constructions and other domain-restricting modifiers in natural language. There are a number of additional promising applications of Quantificationalism, which I explore in other work. Here are some examples.

[4]Several variations of this picture are explored in Stanley and Szabó [2000].

"To quantify over a domain": The expression "to quantify over a domain" is used in discussions of a variety of philosophical topics, from absolute generality to ontological disagreement and predicativity.[5] While these uses are often fast and loose, there is important theoretical work that a precisification of this expression can do for us. Quantificationalists can introduce one such precisification, on which it makes sense to think of *propositions, properties and relations* (as opposed to sentences and predicates) as quantifying over a domain, without assuming that reality has anything like quasi-syntactic structure.

Metaphysical predicativity: Through the idea of quantification over a domain, Quantificationalists can also formulate a novel notion of *metaphysical predicativity*: a proposition, property or relation is metaphysically predicative when it only quantifies over domains that do not include that proposition.[6] Metaphysical predicativity differs from traditional conceptions of predicativity in that it is a property of non-linguistic entities rather than sentences or definitions. It can be used to articulate a predicativist metaphysics on which only predicative entities exist, that does not require syntactic complexities of ramified type theory while reaping some of the same benefits thereof.[7]

Uniqueness results for free quantifiers: Any two expressions that behave logically like classical universal quantifiers (at the same type) are provably equivalent [Harris, 1982]. This result is philosophically significant. Classical logicians can avoid entering merely verbal ontological disputes simply by agreeing to use "exists" as synonymous with "is identical to something" [Williamson, 1988]. Classical higher-order logicians can argue that primitive higher-order quantifiers are intelligible because their meaning is uniquely pinned down by inferential roles [Bacon, 2023, Ch. 0.3]. Notoriously, standard free logics are not strong enough to prove similar uniqueness results for free quantifiers, so free logicians cannot reap the same benefits. As we shall see, Quantificationalists have strong reasons to theorize in a free logic. They also have independent reason to regard the correct inferential role of free quantifiers as richer than it is normally taken to be. This richer inferential role turns out to be strong enough to single out the meaning of free quantifiers uniquely.

As I noted, the purpose of this paper is not to develop any of these applications in detail. Before that can be done, Quantificationalism needs solid formal foundations, within which its applications can be rigorously developed. This task is trickier than one might have thought: Quantificationalism faces problems that do not arise for Temporalism and Modalism, which threaten to collapse it into inconsistency. This paper has the twofold purpose of *raising* and *understanding* the problems that arise uniquely for Quantificationalism, and of developing a logical framework for Quantificationalism that avoids these problems.

---

[5]Representative samples include Williamson [2003]; Lewis [1990]; Quine [1953]; Uzquiano [2019].

[6]A gloss along these lines is given in Bacon et al. [2016]; Bacon [2021], who note some difficulties involved in getting precise on the notion of quantification over a domain.

[7]This may prove useful in formulating novel solutions to intensional paradoxes such as Prior's paradox [Prior, 1961], as well as some puzzles in the theory of ground Fine [2010]; Krämer [2013].

Here is the plan for the paper. I begin in Section 2 by articulating the problems just mentioned and motivating my preferred solution in broad strokes, covering some preliminaries along the way. The rest of the paper presents a comprehensive formal framework implementing this solution Sections 3 to 6. Section 3 covers the syntactic background. Sections 4 and 5 cover the model theory. Finally, Section 6 presents a full axiomatization of the logic of Quantificationalism and explores some extensions thereof.

2. Paradoxes of Domain Specification

I introduce the notion of a *domain specifier*, a device that regiments and generalizes the behavior of exceptive constructions and other domain-shifting modifiers in natural language, and explain how domain specifiers are connected to the operator $\blacklozenge$. I then present two paradoxes that arise if we attempt to develop the logic of Quantificationalism in a standard higher-order framework. Finally, I sketch and briefly motivate my preferred way out of these paradoxes. It centers on the idea that domain specifiers are *genuinely syncategorematic* expressions, which do not stand for higher-order entities in their own right.

Before all that, I make the preliminary point that Quantificationalists should theorize in a *free* quantificational logic, in which the classical principle of Universal Instantiation is not derivable. This is a central observation that comes up repeatedly when articulating the intended reading of domain specifiers, so it is worth discussing up front.

2.1. **Free logic.** Classical quantificational logic consists of the Universal Instantiation schema and the rule of Universal Generalization:

$$\forall x P \to P[a/x] \tag{UI}$$

$$\text{if } \Gamma \vdash P, \text{ then } \Gamma \vdash \forall x P \tag{UG}$$

where x is not free in any formula in Γ. Standard tense and modal logics feature necessitation rules, which imply that logical truths are eternally and necessarily true. Together, classical quantificational logic and necessitation rules entail *permanentism* and *necessitism*, which say, respectively, that always, everything eternally exists and that necessarily, everything necessarily exists.[8]

$$A\forall x A \exists y(x = y) \tag{Permanentism}$$

$$\Box \forall x \Box \exists y(x = y) \tag{Necessitism}$$

Here A and $\Box$ are the duals of S and $\Diamond$, respectively.

Many philosophers working with tense and modal logics wish to reject (Permanentism) and (Necessitism), convinced that ordinary examples of temporary and contingent existence abound.[9] For example, while the United States exists now, it did not exist in 1775, and would not have existed had the American Revolution failed. To avoid (Permanentism) and (Necessitism), these philosophers embrace a *free logic* in which (UI) is not derivable.

A similar dialectic arises for Quantificationalists. If our background logic contains classical quantificational logic and is closed under a necessitation rule for

[8]Permanentism and Necessitism are discussed at length in Williamson [2013] and the rich literature that followed; see especially Yli-Vakkuri and McCullagh [2017].

[9]See especially Fine [1977]; Stalnaker [2011].

the dual $\blacksquare$ of $\blacklozenge$—to the effect that logical truths are true at all domains of quantification—then *Antirestrictivism* below is derivable:

$$\blacksquare\forall x\blacksquare\exists y(x = y). \qquad \text{(Antirestrictivism)}$$

Antirestrictivism says that at all domains, everything exists at all domains. In other words, it says that domains of quantification cannot shrink.

But Quantificationalists have compelling reasons to reject Antirestrictivism. The very examples I used to initially motivate Quantificationalism essentially involve the idea of evaluating a proposition for truth at a domain that excludes some things that actually exist: since, e.g., the United States exists, the domain consisting of countries other than the United States is strictly smaller than the domain of all things. In other words, at some domain, namely that of countries other than the United States, there is no such thing as the United States.

The issue here is not that there are intuitive counter-examples to Antirestrictivism. Rather, it is that accepting Antirestrictivism changes the subject. Quantificationalism is a theory of domain specification, the way Temporalism is a theory of tense. Temporalists think that tense should be regimented by means of non-trivial tense operators. Tense divides into past, present, and future tense, so temporalists should admit backward-looking, present-looking, and forward-looking tense operators. Temporalists can disagree about whether Permanentism is true while agreeing that they are giving an account of the same phenomenon, namely tense.

Not so for quantificationalists. Quantificationalists think that domain specification should be regimented by means of non-trivial "quantificational operators." Restricting the domain of quantification is a way of specifying a domain of quantification—to restrict the domain of quantification is to specify a domain smaller than the current one as the "salient" domain. This is just a fact: expressions like 'with the exception of the United States' do that, regardless of whether they are ultimately to be analyzed as sentential operators. A quantificationalist who accepts Antirestrictivism, therefore, is failing to give a full account of domain specification; they are missing out on domain restriction. Put another way, a quantificationalist who accepts Antirestrictivism is in some respects like a temporalist who only posits backward-looking tense operators, failing to give an account of future tense.

Quantificationalists should thus reject Antirestrictivism. The simplest way of doing so is to reject (UI) and theorize in a free quantificational logic instead.[10]

2.2. **Domain specifiers.** The logic of tense operators can be fully characterized by the logic of "tense nominals," which are sentential operators formalizing the notion of truth at a time. A language with tense nominals is equipped with a class of variables and constants ranging over or standing for times, as well as an expression at that takes one such term t and returns a sentential operator $\mathsf{at}(t)$. Intuitively, $\mathsf{at}(t)(P)$ is true precisely when P is true at time t. If we understand the logic of tense nominals, we can characterize the logic of tense operators by endorsing:

Tense Leibnizianism: $\forall p(Sp \leftrightarrow \exists t(\mathsf{at}(t)(p)))$.[11]

[10] Another possibility is to reject the necessitation rule for $\blacksquare$: not all logical truths are true at all domains of quantification. Concurring with Bacon [2013], I think the choice between rejecting necessitation and rejecting (UI) ultimately amounts to a merely verbal disagreement about the notion of a logical truth.

[11] This characterization of the logic of tense is not uncontroversial. It is reasonable to think that a proposition can be sometimes true even though there are no such things as times, but Tense

I take a similar approach to the logic of $\blacklozenge$. We can regiment English expressions like 'with the exception of the United States' (or 'among prime numbers,' 'in Australia,' ...) by means of an operation At that takes a predicate F (of any type) and returns a sentential operator. Intuitively, $\mathsf{At}(F)(P)$ is true precisely when P is true relative to the domain consisting precisely of the Fs. We can think of P as the result of "resetting" the domain of quantification within its scope to precisely the Fs. On this approach, we formalize 'with the exception of the United States' as something like

$$\mathsf{At}(\lambda x.x \neq \text{the U.S.}).$$

More generally, we can allow expressions of the form $\mathsf{At}(F)$ to take values of *any* type in their second argument place.[12] We can then use the device At to formalize a broader range of expressions, for example:

(5) 'Is a country other than the U.S.' as $\mathsf{At}(\lambda x.x \neq \text{the U.S.})(\text{is a country})$;
(6) 'Some country' as $\mathsf{At}(\text{is a country})(\exists)$;
(7) 'The tallest American' as $\mathsf{At}(\text{is American})(\text{the tallest person})$.

To be clear, I am not trying to argue that any of these analyses are correct as a matter of natural language semantics; I just want to give a taste of what is possible.

Call an expression of the form $\mathsf{At}(F)$ a *domain specifier*. I take it that the logic of the operator $\blacklozenge$ can be illuminated by studying the logic of domain specifiers. For, at the very least, every instance of the following schema is true on the target reading:

$$\mathsf{At}(F)(P) \to \blacklozenge P \tag{Master}$$

If P is true at the Fs, then P is true at some domain.

The connection between $\blacklozenge$ and domain specifiers is not as straightforward as that between the standard tense operators and tense nominals. This is because of two main reasons. The first is related to the impossibility of quantification over types in standard higher-order frameworks. Intuitively, $\blacklozenge P$ says that we can make P true by shifting the domain of quantification *at some type or other*. This is not something we can express in our object language. The best we can do is to pick a particular type σ and write down the schema

$$\blacklozenge P \to \exists X^{\sigma \to t} \mathsf{At}(X)(P). \tag{1}$$

But this makes $\blacklozenge$ stronger than it ought to be.

The second reason is that, as I argued in the previous section, Quantificationalists should think that the correct logic of quantification is not classical, but free. Given so, even if we *were* able to somehow quantify over types to get around the first expressive limitation mentioned above, writing something like

$$\blacklozenge P \to \forall \sigma \exists X^{\sigma \to t} \mathsf{At}(X)(P), \tag{2}$$

it would still be true that $\blacklozenge$ is stronger than it is intended to be. For it is consistent in free logic that $\exists X(\mathsf{At}(X)(P))$ is false and yet some claim of the form $\mathsf{At}(F)(P)$ is true.

Leibnizianism rules that out. I will ignore this complication: I introduce Tense Leibnizianism merely as a helpful analogy to my own characterization of the logic of $\blacklozenge$, for which a counterpart to the worry just mentioned does not seem to arise.

[12] We can do this, for example, by theorizing in terms of type-indexed families of domain specifiers, where $\mathsf{At}_\sigma(F)$ has type $\sigma \to \sigma$. Alternatively, we could treat domain specifiers as themselves syncategorematic. For the moment, we can overlook the details, though this is an important choice point we will return to.

We will eventually be able to capture the idea that ⧫ is a *master modality* for all domain specifiers, in much the same sense S is for tense nominals, by means of infinitary schematic rules. For now, however, I will rely on an intuitive grasp of the connection between ⧫ and domain specifiers.

Let me say a little more about the intended interpretation of domain specifiers. Recall that a domain specifier $\mathsf{At}(F)$ is supposed to work by resetting the domain of quantification within its scope to precisely the Fs. I sharpen this gloss through the notion of an *existence predicate*, defined by its inferential role. In a free logic, an *existence predicate* for a free quantifier $\forall$ is any predicate $\mathsf{E!}$ for which the following are both derivable:

$$\mathsf{E!}a \to (\forall x P \to P[a/x]) \tag{E!UI}$$

$$\forall x \mathsf{E!}x. \tag{$\forall$E!}$$

Here a must be free for x in P. The schema (E!UI) is a weakening of the classical principle (UI). A universal generalization $\forall x P$ can fail to imply all its instances, but always implies an instance $P[a/x]$ under the assumption that a exists. The schema ($\forall$E!) simply says that everything exists. If our background logic has an identity predicate, an existence predicate can always be defined as $\lambda x.\exists y(x = y)$: to exist just is to be identical to something.

I think of the *current* domain of quantification as given by *the existing things*, that is, by $\mathsf{E!}$. Part of what it means for a domain specifier $\mathsf{At}(F)$ to reset the domain of quantification to the Fs can thus be captured by writing down a schema describing the result of applying a domain specifier to an existence predicate itself:

$$\mathsf{At}(F^{\sigma\to t})(\mathsf{E!}_\sigma) = F. \tag{AtG}$$

In words: *to exist at the Fs is just to be F.*

This illustrates how *resetting* the domain of quantification to the Fs is different from *restricting* it to the Fs. The latter idea could be regimented by the schema

$$\mathsf{At}(F^{\sigma\to t})(\mathsf{E!}_\sigma) = \lambda x.\mathsf{E!}x \wedge Fx. \tag{AtR}$$

In words: *to exist at the Fs is just to be an existing F.* Resetting is more general than restricting, since we can define

$$\mathsf{At}^{\cap}(F^{\sigma\to t})(M) := \mathsf{At}(\lambda x.\mathsf{E!}x \wedge Fx)(M),$$

and the result of substituting $\mathsf{At}^{\cap}$ for At in (AtR) holds whenever (AtG) holds.

Of course, resetting and restricting are the same when the background quantificational logic is classical, so that $\mathsf{E!}$ is equivalent to $\lambda x.\top$. But they come apart in a free logic. If Pegasus the winged horse does not exist, then At(is a winged horse)($\mathsf{E!}$) applies to Pegasus, but $\mathsf{At}^{\cap}$(is a winged horse)($\mathsf{E!}$) does not.

2.3. **Paradox.** I have allowed domain specifiers to take arguments of any type, without specifying how exactly this is achieved. It is time to get more precise about this. In standard higher-order languages, no expression can take arguments of multiple types. Instead, one *simulates* polymorphic expressions by means of type-indexed families of expressions. Applied to domain specifiers, this means that, for each predicate F, instead of having a single domain specifier $\mathsf{At}(F)$ that literally can combine with arguments of any type, we have a family of domain specifiers $\mathsf{At}_\sigma(F)$, each of type $\sigma \to \sigma$.

This straightforward approach, unfortunately, leads to paradox in the presence of a few other eminently plausible assumptions about the logic of domain specifiers.

There are at least two paradoxes that arise, and multiple variations of each. The first has a familiar form, but requires stronger assumptions to kick in. The second is harder to make intuitive sense of, but is a lot more general. I take them in order.

Both paradoxes have to do with the concept of a *stable entity*. Here is the idea. Some entities *involve quantification*, others do not. For example, the property of *loving someone* involves quantification, negation does not. To involve quantification is not a purely syntactic matter. The syntax of the English verb 'eradicate' does not feature any overt or covert quantification, yet its meaning involves quantification: to eradicate a disease is to make it so that *virtually no one* has it.

I call *stable* those entities that do not involve quantification in the sense just gestured at. Non-stable entities may be "moved" by domain specifiers. For example, $\mathsf{At}(F)(\textit{loves someone})$ should be equivalent to *loves an* F. On the other hand, stable entities ought to be fixed by domain specifiers: when M has a stable meaning, $\mathsf{At}(F)(M)$ should be equivalent to M for any F.

The first paradox is, essentially, a version of the infamous Russell-Myhill paradox [Russell, 1937; Myhill, 1958; Goodman, 2017]. This is a paradox often taken to show that extremely fine-grained conceptions of metaphysical structure are inconsistent. If reality were structured the way language is, one would expect every instance of the following schema to hold:

$$Fa = Gb \to (F = G \wedge a = b). \tag{Struc}$$

However, this is classically inconsistent. Given a fixed q, let

$$O := \lambda p.\forall X((p = Xq) \to \neg Xp).$$

Using (UI) and propositional reasoning, we can prove $\neg O(Oq)$, which is to say that there is some Y such that $Oq = Yq$ and $Y(O)$. But then Y and O are not coextensive, since only the former applies to Oq. This contradicts (Struc).

A similar version of this paradox ensues for quantificationalists, suggesting that theorizing in terms of domain specification may impose excessively fine-grained granularity constraints. The simplest version of this paradox arises in the presence of three assumptions about the logic of domain specifiers. First, that domain specifiers commute with application:

$$\mathsf{At}_\tau(F)(MN) = (\mathsf{At}_{\sigma\to\tau}(F)(M))(\mathsf{At}_\sigma(F)(N)). \tag{App}$$

Intuitively, since MN is built up by applying M to N, any quantification involved in MN must come from either M or N. So, to specify the Fs as the domain of quantification in MN is the same as specifying the Fs as the domain of quantification in both M and N, and then applying the results.

The second assumption concerns the behavior of domain specifiers when applied to quantified claims involving stable properties. It is tempting to think that every instance of the following schema should hold

$$\mathsf{At}_t^\cap(F)(\forall G) = \forall x(Fx \to Gx). \tag{3}$$

In words, saying that considering *just* the Fs, everything is G is the same as saying that every F is G. However, this is not in general the case when F is not stable. As an example, consider the property of *loving someone*, which involves quantification. Call it *being a lover*. Assume that Alice loves only Bob, Bob loves only Carol, but Carol loves no one. Then Alice and Bob are the only lovers. However, only Alice is a *lover among lovers*: for among the lovers—i.e., Alice and Bob—Bob does not love anyone. So it is false that among the lovers, everyone is a lover. That being said, when G is stable, it seems reasonable to assume that the schema above holds. Thus, we assume:

$$\mathsf{Stab}_\sigma(G) \to (\mathsf{At}^\cap(F)(\forall G) = \forall x(Fx \to Gx)). \tag{Stab$\forall$}$$

Here Stab_σ is a predicate of type $(\sigma \to t) \to t$, governed by the schema

$$\mathsf{Stab}_\sigma(F) \to (\mathsf{At}(X)(F) = F). \tag{4}$$

Lastly, we need an assumption generating an abundance of stable properties: every property is coextensive with a stable property:[13]

$$\forall X^{\sigma \to t} \exists Y(\mathsf{Stab}(Y) \wedge \forall z(Xz \leftrightarrow Yz)). \tag{StabC}$$

Notice that(StabC) is only as strong as intended if we assume that (UI) holds at least *outside* the scope of domain specifiers. This is an extra assumption that I will spot myself to run the present version of the paradox, noting it can be dispensed with in some variations.

Problem 2.1 (The Quantificationalist Russell-Myhill Paradox)**.** Under the assumptions just stated, we can prove that whenever F and G are not coextensive, the sentential operators $\mathsf{At}_t^\cap(F)$ and $\mathsf{At}_t^\cap(G)$ are distinct. To see this, assume F and G are not coextensive. Then either there is some F that is not G or the other way around—suppose the former. By (StabC) each of F and G is coextensive with a stable property, so suppose without loss of generality that F and G are already stable. Then $\mathsf{At}_t^\cap(G)(\forall G)$ is true by (Stab$\forall$), but $\mathsf{At}_t^\cap(F)(\forall G)$ is false, since it is equivalent to $\forall x(Fx \to Gx)$. So, $\mathsf{At}_t^\cap(F)$ and $\mathsf{At}_t^\cap(G)$ disagree on $\forall F$, and are therefore distinct.

This is paradoxical, by an argument closely paralleling the Russell-Myhill paradox. We can define a property of operators

$$O := \lambda x^{t \to t}.\forall Y^{(t \to t) \to t}((\mathsf{At}_t^\cap(Y) = x) \to \neg Yx).$$

We can then prove $\neg O(\mathsf{At}_t^\cap(O))$, which is to say—helping ourselves to (UI) here—that there is some $Y^{(t \to t) \to t}$ such that $\mathsf{At}_t^\cap(Y) = \mathsf{At}_t^\cap(O)$ and $Y(\mathsf{At}_t^\cap(O))$. The latter shows that Y and O are not coextensive, which by what established above in turn implies that $\mathsf{At}_t^\cap(Y)$ and $\mathsf{At}_t^\cap(O)$ are distinct. This is a contradiction.

A naive reaction is to blame paradox on classical quantificational logic, which I have already argued quantificationalists should reject. This is misguided, for two reasons. First, to deny that every instance of (UI) is a *logical truth*—which is what I argued quantificationalists should do—is consistent with accepting that every instance of (UI) is *true*. Only the latter assumption is required to run the paradox. And second, as I noted in passing, even this assumption can be dispensed with in some variations of the paradox. The idea here is to replace (UI) with a collection of schemas to the effect that $\mathsf{At}_{\sigma \to t}(\lambda x.\top)(\forall)$ behaves like a classical quantifier, even if $\forall$ itself does not.

Another possible reaction is to blame (StabC). Denying comprehension principles to block paradoxes is certainly not without precedent. As far as I can see, this move might well succeed in blocking variations of this specific paradox. But that is beside the point, as it will do nothing to block the second paradox, which I turn to next.

[13]Compare this with the more familiar principle of *rigid comprehension*, which says that every property is coextensive with a rigid property. A *rigid property* is one that does not change its extension across modal space. See, e.g., Bacon and Dorr [2024] for discussion.

The second paradox is harder to make intuitive sense of, but it is a lot more general. The most general version I was able to articulate requires only two assumptions besides (AtG): (App) and the claim that there are at least two distinct stable properties. By "there are" here I do not mean an object language quantified claim. All that is needed is that we can write down two terms of any type $\sigma \to t$ that express distinct stable properties. This is always possible, for example, if all Boolean connectives and all combinators express stable properties: the relevant properties can then be taken to be negation and the identity combinator $\lambda p.p$ at type $t \to t$. But putatively fundamental physical properties like *is positively charged* and *is negatively charged* also, intuitively, fit the bill.[14]

Here is the paradox, presented by omitting type subscripts from At for ease of exposition.[15]

Problem 2.2 (The Stability Collapse Paradox)**.** Assume that F and G are stable. Thus, in particular, $F = \mathsf{At}(G)(F)$. Using (AtG) and Leibniz's Law, we may substitute F for $\mathsf{At}(F)(\mathsf{E!})$, yielding

$$F = \mathsf{At}(G)(\mathsf{At}(F)(\mathsf{E!})). \tag{5}$$

Now, by (App) on the right-hand side, this is equivalent to

$$F = (\mathsf{At}(G)(\mathsf{At}(F)))(\mathsf{At}(G)(\mathsf{E!})). \tag{6}$$

Since $\mathsf{At}(G)(\mathsf{E!}) = G$ by (AtG), by Leibniz's Law again we may substitute G for $\mathsf{At}_\sigma(G)(\mathsf{E!})$, yielding

$$F = (\mathsf{At}(G)(\mathsf{At}(F)))G. \tag{7}$$

But G is stable, so $\mathsf{At}(G)(G) = G$. So, once more by Leibniz's Law, we get

$$F = (\mathsf{At}(G)(\mathsf{At}(F)))(\mathsf{At}(G)(G)), \tag{8}$$

and in turn, by (App),

$$F = \mathsf{At}(G)(\mathsf{At}(F)(G)). \tag{9}$$

Since $\mathsf{At}(F)(G) = G$ and $\mathsf{At}(G)(G) = G$ both hold by stability, using Leibniz's Law twice we finally conclude that $F = G$.

Thus, (AtG) and (App) imply there can be at most one stable property at any given type, in the strong sense that every instance of the schema $\mathsf{Stab}_\sigma(F) \wedge \mathsf{Stab}_\sigma(G) \to F = G$ becomes derivable if (AtG) and (App) are added as axioms to the background logic. A quantificationalist must either embrace this conclusion or deny one of (AtG) and (App). Or, alternatively, they can claim that something is wrong with the background higher-order logic.

---

[14] More generally, there is a family of views according to which no fundamental property involves quantification. The world can be completely described by specifying (i) which fundamental properties are instantiated where, and (ii) which things exist. A view of this sort is one where all fundamental properties are stable, and surely there are at least two distinct fundamental properties. See, in particular, Kaplan [1995].

[15] The paradox is closely related to the problem discussed in [Bacon, 2019, p. 1062]. Bacon uses an analogous argument to derive, from similar assumptions, the claim that there cannot be more than two *fundamental* entities at any type. That being said, these problems should not be conflated: Bacon's notion of fundamentality is different—both conceptually and formally—from my notion of stability, which instead formally resembles his notion of *purity*. It seems that the stability collapse paradox could be generalized to show that, under similar assumptions in Bacon's framework, there can be at most one *pure* entity at any given type.

This paper articulates a solution of the last sort. Before I describe it in more detail, let me say a few words about why I take this route. I think of domain specification as a theoretical concept. Domain specifiers are connected to exceptives and other domain shifting modifiers in natural language ('in the United States,' 'among prime numbers'...) in a tight enough way that the former can be introduced by analogizing them to the latter. But domain specifiers are not supposed to be stipulative equivalents of such modifiers. Rather, the main way we grasp what they mean is by understanding their inferential role. I take it we are in a similar situation with respect to logical vocabulary like Booleans and quantifiers. Truth-functional conjunction is initially introduced by analogy with the English word 'and.' But as we learn enough propositional logic, we come to grasp the meaning of Boolean conjunction independently of that of 'and,' to the point that we can have substantive disagreements about whether the two are synonymous.[16]

I take both (AtG), and to a somewhat lesser extent (App) and the non-triviality of stability, to be integral parts of the inferential role that I am trying to articulate for domain specifiers. To deny any of these would amount to rejecting the viability of domain specifiers as a theoretical device. This could prompt us to look for alternative logical devices to formulate Quantificationalism, or it could be taken as evidence that Quantificationalism is not a viable philosophical position. I am not rejecting these options outright. But properly debating these questions requires a comprehensive lay of the land: whether anyone should or should not theorize in terms of domain specifiers, or whether any alternative devices do a better job than domain specifiers, depends on what the costs and benefits of theorizing in terms of domain specifiers are. I think we can get a good sense of what these benefits and costs are by asking what sort of changes we need to make to a standard higher-order framework in order to accommodate domain specifiers.

2.4. **Genuine Syncategorematicity.** I think both paradoxes rest on one mistake, i.e., treating domain specifiers as *categorematic expressions*: independently meaningful terms in our language that can be assigned types. We should instead treat domain specifiers as *genuinely syncategorematic* expressions, which only have meaning in relation to the expressions they combine with. In Russellian terms, genuinely syncategorematic expressions do not contribute constituents to the propositions expressed by the sentences they occur in; rather, they indicate how the constituents expressed by their arguments are to be combined to form a proposition.

On this approach, domain specifiers are analogized to application. In standard higher-order languages, application is introduced as a primitive syncategorematic operation that forms new terms from old ones: given a term M of type $\sigma \to \tau$ and a term N of type σ, the application MN is a term of type τ. Application is not itself assigned a type, and does not stand for any higher-order entity. The syncategorematicity of application is visually emphasized by the fact that we notate it using concatenation, rather than with an overt symbol, though this is not essential.

In the framework I propose, domain specifiers are treated similarly. Domain specifiers are not treated as terms that can be assigned types. Instead, we equip the language with an additional syncategorematic term formation rule. Given a predicate F of type $\sigma \to t$ and a term M of type τ, the expression $\mathsf{At}(F)(M)$ is a term of type τ. The meaning of $\mathsf{At}(F)(M)$ is not obtained by *applying* the meaning

[16] See Dorr [2025, 2014]; Williamson [2003] on this attitude towards the meaning of logical vocabulary.

of $\mathsf{At}(F)$ to that of M—the former is not independently meaningful. Rather, it is obtained from the meanings of F and M via a primitive combination operation other than application, which I will eventually gloss model-theoretically as a kind of *metaphysical substitution*.

This approach echoes the early logical atomists' view about Boolean logical vocabulary.[17] On their view, expressions like $\wedge$ and $\rightarrow$ are not meaningful on their own, but merely indicate how the meanings of their arguments are to be combined to form the meaning of a compound expression. Again, to put together the meanings of P and Q via conjunction is not to *apply* the meaning of 'and' to those of P and Q, but to combine them in a fundamentally different way.

Let me briefly sketch how this idea helps with the two paradoxes. The Quantificationalist Russell-Myhill Paradox shows that if F and G are not coextensive, then neither are the domain specifiers $\mathsf{At}_t(F)$ and $\mathsf{At}_t(G)$. This only determines an injection from properties of operators to operators if domain specifiers are treated as categorematic terms of type $t \to t$. If they are instead treated as syncategorematic expressions, the non-coextensiveness of F and G need not imply the existence of distinct higher-order entities corresponding to $\mathsf{At}_t(F)$ and $\mathsf{At}_t(G)$.

As for the Stability Collapse Paradox, the categorematicity of domain specifiers predicts the well-formedness of problematic instances of (App). Given a term of the form $\mathsf{At}(F)(\mathsf{At}(G)(M))$, if $\mathsf{At}(G)$ combines with M through *application*, then

$$\mathsf{At}(F)(\mathsf{At}(G)(M)) = (\mathsf{At}(F)(\mathsf{At}(G)))(\mathsf{At}(F)(M)) \tag{10}$$

is an instance of (App). This identity is not only hard to parse, but suggests a misguided way of thinking about how domain specifiers compose that leads to paradox. If we treat domain specifiers syncategorematically, then (10) is no longer well-formed. And we can tell a different, to my eye more intuitive story about how domain specifiers compose. If $\mathsf{At}(G)$ sets the domain to the Gs and $\mathsf{At}(F)$ sets the domain to the Fs, then their composition $\mathsf{At}(F)(\mathsf{At}(G)(\cdot))$ should set the domain to the *Gs among the Fs*.[18] We would thus have the schema:

$$\mathsf{At}(F)(\mathsf{At}(G)(M)) = \mathsf{At}(\mathsf{At}(F)(G))(M). \tag{Comp}$$

There is a caveat. In standard higher-order frameworks, syncategorematicity is *superficial*. Take the case of application. While application is introduced as a syncategorematic operation, we can always *define* a categorematic counterpart of it via λ-abstraction, as $\lambda X \lambda y.Xy$. Since $\beta\eta$-equivalence suffices for identity, any term MN formed by applying M to N can always be equivalently rewritten as $(\lambda X \lambda y.Xy)MN$, which is constructed by applying the categorematic counterpart of application itself to M, and then applying the result to N. Even if syncategorematic application itself does not stand for a higher-order entity, a higher-order entity that behaves exactly like it can always be found.

The same goes for domain specifiers. Even if we take domain specifiers to be syncategorematic expressions, categorematic counterparts thereof can always be reintroduced through λ-abstraction, writing $\lambda x^\sigma.\mathsf{At}(F)(x)$. Since $\beta\eta$-equivalence

[17] See [Russell, 1940, p. 39] and [Wittgenstein, 1921, §5.4].

[18] One way to see the contrast is to assume that $\mathsf{At}(F)(\mathsf{At}(G)) = \mathsf{At}(\mathsf{At}(F)(G))$—which would follow, for example, from the assumption that At is stable and combines with G through application. If both (10) and (Comp) held, we would then have

$$\mathsf{At}(\mathsf{At}(F)(G))(M) = \mathsf{At}(\mathsf{At}(F)(G))(\mathsf{At}(F)(M)), \tag{11}$$

which one would not expect to hold unless $\mathsf{At}(F)(M) = M$.

suffices for identity, any term of the form $\mathsf{At}(F)(M)$ can always be equivalently rewritten as one formed through application:

$$\mathsf{At}(F)(M) = (\lambda x^\sigma.\mathsf{At}(F)(x))M. \tag{12}$$

In other words, even if domain specifiers themselves do not stand for higher-order entities, higher-order entities that behave exactly like them can always be found. This fact is enough to make both paradoxes resurface: we simply run both arguments using the categorematic counterparts of domain specifiers.

Thus, for the proposed solution to work, we need to work in a framework where syncategorematicity is more than merely superficial. We need a framework where no categorematic counterparts of domain specifiers can be reintroduced. The simplest way to achieve this, and the one I shall pursue here, is to ban all λ-abstraction of variables occurring within the scope of domain specifiers (or the operator $\blacksquare$) from outside such contexts. The following terms, for example, will come out as ill-typed:

$$\lambda x.\mathsf{At}(\bar{F})(x) \qquad \lambda \bar{X} y.\mathsf{At}(\bar{X})(y) \qquad \lambda p.\blacksquare p.$$

This language thus makes domains specifiers *genuinely syncategorematic*: they are syncategorematic, and the language provides no mechanism to introduce categorematic counterparts thereof.

I cannot give a direct argument for the genuine syncategorematicity of domain specifiers independent of the specific framework I propose: I doubt that such an argument can even be formulated in a framework that does not allow for genuine syncategorematicity. What I can do is *show* that genuine syncategorematicity is a viable option, by developing a consistent and powerful framework in which genuine syncategorematicity is accommodated and internally explained. That is what I set out to do in the rest of this paper.

3. Languages and Background Higher-order Logics for Quantificationalism

I now move on to describing the alternative higher-order framework I propose for Quantificationalism in detail. This section covers syntactic preliminaries and background higher-order logics, and explains some noteworthy consequences of the restrictions on λ-abstraction I impose.

For ease of exposition, I have so far presented domain specifiers as having the form $\mathsf{At}(F)$, where F is a single predicate. But in the official version of the framework, for greater generality, At is in fact allowed to take *finite sequences* of predicates of different types in its first argument place. Intuitively, a domain specifier of the form $\mathsf{At}(\bar{F})$, where $\bar{F}$ is one such sequence of predicates, specifies the domain of quantification across multiple types at once. For example, we can form a domain specifier

$$\mathsf{At}(\lambda p.p, \text{is a cat})$$

which specifies *the truths* as the domain of quantification at type t, and *the cats* as the domain of quantification at type e.

3.1. General Higher-order Languages.

Throughout, I assume that we are given a typed family of variables $Var := \{Var^\sigma : \sigma \in Types\}$. A *signature* Σ is a typed collection of sets Σ^σ, one for each type σ. Given a signature Σ, we will generate two languages (classes of terms) over Σ: $\mathcal{L}_+(\Sigma)$ and $\mathcal{L}_\dagger(\Sigma)$. The language we will ultimately theorize in is $\mathcal{L}_\dagger(\Sigma)$, but it is convenient to define it as a fragment of the larger and more standard language $\mathcal{L}_+(\Sigma)$ equipped with full λ-abstraction.

More precisely, given a signature Σ, let $\mathcal{L}_+(\Sigma)$ be defined recursively as follows. Let a *nice type sequence* be a finite, non-repeating sequence of types. When $\bar{\sigma}$ is any sequence of types, I abuse notation slightly by writing $\sigma \in \bar{\sigma}$ to mean that σ occurs in $\bar{\sigma}$. I write $M \cdot \sigma$ to mean that M is a term in $\mathcal{L}_+(\Sigma)$ of type σ.

(1) $x \cdot \sigma$ whenever $x \in \mathit{Var}^\sigma$;
(2) $C \cdot \sigma$ whenever $C \in \Sigma^\sigma$;
(3) $MN \cdot \tau$ whenever $M \cdot \sigma \to \tau$ and $N \cdot \sigma$;
(4) $\lambda x.M \cdot \sigma \to \tau$ whenever $M \cdot \tau$ and $x \in \mathit{Var}^\sigma$;
(5) $QxP \cdot t$ whenever $P \cdot t$, $x \in \mathit{Var}^\sigma$ and $Q \in \{\forall, \exists\}$;
(6) $\mathsf{At}(\bar{F})(M) \cdot \sigma$ whenever $\bar{F} := (F_\sigma)_{\sigma \in \bar{\sigma}}$ such that $\bar{\sigma}$ is a nice type sequence and $F_\sigma \cdot \sigma \to t$ for each $\sigma \in \bar{\sigma}$, and $M \cdot \sigma$;
(7) $\blacksquare P \cdot t$ whenever $P \cdot t$,

Notice that in Item 5, quantifiers are also being treated as syncategorematic expressions. I will return to this point shortly. The notions of a variable occurring *free* or *bound* in a $\mathcal{L}_+$-term are defined as usual, as is the notion of a term being *free for* a variable in another term. We write $\mathsf{FV}_\sigma(M)$ to denote the free variables of type σ of M.

It will be convenient to have a name for the term sequences that can occur in the first argument place of At. I call these *nice term sequences*, and say that $\bar{F} = (F_{\sigma_1}, \dots, F_{\sigma_n})$ is *indexed* by a nice type sequence $\bar{\sigma} := (\sigma_1, \dots, \sigma_n)$ when F_{σ_i} has type $\sigma_i \to t$ for each $1 \leq i \leq n$.

We define the language $\mathcal{L}_\dagger(\Sigma)$ as a fragment of $\mathcal{L}_+(\Sigma)$ that obeys appropriate restrictions on λ-abstraction. Given a term $M \in \mathcal{L}_+(\Sigma)$, the set $\mathsf{AV}_\sigma(M)$ (mnemonic for *abstractable variables*) consists of all variables of type σ that do not occur free within the scope of the second argument place of an occurrence of At, or within the scope of an occurrence of $\blacksquare$. For example, x is not abstractable in $\mathsf{At}(\bar{F})(Gx)$, and p is not abstractable in $\blacksquare p$. More formally, we may define $\mathsf{AV}_\sigma(M)$ by induction on the structure of M.

(1) $\mathsf{AV}_\sigma(v) := \mathit{Var}^\sigma$ whenever $x \in \mathit{Var}^\sigma$;
(2) $\mathsf{AV}_\sigma(C) := \mathit{Var}^\sigma$ whenever $C \in \Sigma^\sigma$;
(3) $\mathsf{AV}_\sigma(MN) := \mathsf{AV}_\sigma(M) \cap \mathsf{AV}_\sigma(N)$;
(4) $\mathsf{AV}_\sigma(\lambda x.M) := \mathsf{AV}_\sigma(M)$;
(5) $\mathsf{AV}_\sigma(QxP) := \mathsf{AV}_\sigma(P)$ for $Q \in \{\forall, \exists\}$;
(6) $\mathsf{AV}_\sigma(\mathsf{At}(\bar{F})(M)) := \mathsf{AV}_\sigma(\bar{F}) \setminus \mathsf{FV}_\sigma(M)$;
(7) $\mathsf{AV}_\sigma(\blacksquare P) := \mathit{Var}^\sigma \setminus \mathsf{FV}_\sigma(M)$.

We can now define $\mathcal{L}_\dagger(\Sigma)$. Writing $M : \sigma$ to mean that M is a term in $\mathcal{L}_\dagger(\Sigma)$ of type σ, we may define $\mathcal{L}_\dagger(\Sigma)$ recursively just like we did for $\mathcal{L}_+$-terms, but replacing condition (4) with

(4*) $\lambda x.M : \sigma \to \tau$ whenever $M : \tau$ and $x \in \mathsf{AV}_\sigma(M)$.

I will need two more languages. The first language $\mathcal{L}_\star(\Sigma)$ consists of those terms in $\mathcal{L}_\dagger(\Sigma)$ containing no occurrences of $\blacksquare$. We will use it to study the logic of domain specifiers without worrying about $\blacksquare$. In addition, we will let $\mathcal{L}_0(\Sigma)$ consist of those terms in $\mathcal{L}_+(\Sigma)$ that contain no occurrences of either $\blacksquare$ or At. We shall use $\mathcal{L}_0(\Sigma)$ mainly for model-building purposes. We thus have four languages in total, ordered

$$
\begin{aligned}
\wedge_t &:= \wedge & \wedge_{\sigma\to\tau} &:= \lambda X \lambda Y \lambda z^\sigma . Xz \wedge_\tau Yz \\
\neg_t &:= \neg & \neg_{\sigma\to\tau} &:= \lambda X \lambda y^\sigma . \neg_\tau Xy \\
\bot &:= \wedge \equiv_{t\to t} \wedge & \top &:= \neg\bot \\
\bot_t &:= \bot & \bot_{\sigma\to\tau} &:= \lambda x^\sigma . \bot_\tau \\
\top_t &:= \top & \top_{\sigma\to\tau} &:= \lambda x^\sigma . \top_\tau \\
\to &:= \lambda p \lambda q . \neg p \vee q & \leftrightarrow &:= \lambda p \lambda q . (p \to q) \wedge (q \to p) \\
\hat{\forall}_\sigma &:= \lambda X^{\sigma\to t} . \forall y^\sigma Xy & \hat{\exists}_\sigma &:= \lambda X^{\sigma\to t} . \exists y^\sigma Xy \\
\forall \bar{x} M &:= \forall x_1 \cdots \forall x_n M & \exists \bar{x} M &:= \exists x_1 \cdots \exists x_n M \\
\lambda \bar{x} . M &:= \lambda x_1 \cdots \lambda x_n . M
\end{aligned}
$$

FIGURE 1. Abbreviations. Here $\bar{x}$ is any sequence of variables $(x_1, \ldots, x_n)$.

by inclusion:

$$\mathcal{L}_0(\Sigma) \subseteq \mathcal{L}_\star(\Sigma) \subseteq \mathcal{L}_\dagger(\Sigma) \subseteq \mathcal{L}_+(\Sigma).$$

When $\mathcal{L}_\heartsuit \in \{\mathcal{L}_+, \mathcal{L}_\dagger, \mathcal{L}_\star, \mathcal{L}_0\}$, an $\mathcal{L}_\heartsuit$-*term over* Σ is just an element of $\mathcal{L}_\heartsuit(\Sigma)$.

3.2. **Background Logics.** I now move on to describing the background higher-order logics I shall refer to throughout the paper. Let us fix a standard logical signature Λ containing the Boolean operations $\wedge, \neg$ of the usual types, a constant $\mathsf{E!}_\sigma$ of type $\sigma \to t$ for each type σ, as well as a constant $\equiv_\sigma$ of type $\sigma \to (\sigma \to t)$ for each type σ. The constants $\mathsf{E!}_\sigma$ are intended to be existence predicates, whereas each constant $\equiv_\sigma$ is intended to express a relation I call *applicative indiscernibility* at the relevant type. This is a relation that has a lot in common with identity, but turns out to be weaker than it in our non-standard framework. I will elaborate on this point in the next section.

All signatures considered in this paper are assumed to be extensions of Λ, and the working signature will always be Λ when a signature is not explicitly specified. For $\mathcal{L}_\heartsuit \in \{\mathcal{L}_+, \mathcal{L}_\dagger, \mathcal{L}_\star, \mathcal{L}_0\}$, we let $\mathcal{L}_\heartsuit$ abbreviate $\mathcal{L}_\heartsuit(\Lambda)$. I fix some standard abbreviations I make use of throughout the paper in Figure 1.

I also fix two (classes of) background higher-order logics: the classical systems $\mathtt{H}_\heartsuit$ and the free systems $\mathtt{FH}_\heartsuit$ (short for *Free* $\mathtt{H}_\heartsuit$.) I give an axiomatization of $\mathtt{FH}_\heartsuit$ in Figure 2. The relevant notion of axiomatization is as follows. Figure 2 defines a relation $\vdash$ between sets of $\mathcal{L}_\heartsuit$-formulas and $\mathcal{L}_\heartsuit$-formulas, the least relation satisfying all the listed constraints. Then the logic $\mathtt{FH}_\heartsuit$ is defined as the smallest set of $\mathcal{L}_\heartsuit$-formulas containing P whenever $\vdash P$ holds. An axiomatization, in the same sense, of $\mathtt{H}_\heartsuit$ can be obtained by adding (UI) below to the schemas from Figure 2.

$$\forall x P \to P[a/x] \qquad\qquad a \text{ free for } x \text{ in } P \qquad\qquad \text{(UI)}$$

We ultimately care most about the free systems $\mathtt{FH}_\heartsuit$, but it will be convenient to view them as fragments of classical systems for model-building purposes.

Both axiomatizations are schematic; for each class of terms introduced in the previous subsection, we may consider the system that results by taking instances only from terms in that class. I notate these systems by adding the names of the relevant classes of terms as subscripts. Thus, for example, we have the system

$\vdash P$	P a classical tautology	(Taut)
$\vdash \forall x \mathsf{E!} x$		($\forall\mathsf{E!}$)
$\vdash \mathsf{E!} a \to (\forall y P \to P[y/a])$	a free for y in P	(FrUI)
$\vdash \forall x(P \to Q) \to (\forall x P \to \forall x Q)$		(Norm)
$\vdash \forall x P \leftrightarrow \neg\exists x \neg P$		(Dual)
$\vdash M \equiv M$		(Refl)
$\vdash (M \equiv N) \to (FM \to FN)$		(WLL)
$\vdash M \equiv N$	M, N $\beta\eta$-equivalent	($\beta\eta$)
$P, P \to Q \vdash Q$		(MP)
If $\Gamma \vdash Q$ then $\Gamma \vdash \forall x Q$	x not free in Γ	(UG)

FIGURE 2. The logic $\mathtt{FH}_{\heartsuit}$

$\mathtt{H}_0$, axiomatized by all $\mathcal{L}_0$-*terms* P such that $\vdash P$ holds. And we have the system $\mathtt{FH}_{\dagger}$, axiomatized by all $\mathcal{L}_{\dagger}$-*terms* P such that $\vdash P$. Unless otherwise specified, the relevant classes of terms are assumed to be over the logical signature Λ.

$\mathtt{H}_+$ can be fully and faithfully translated into the standard, consistent system of higher-order logic that Bacon [2023] calls $\mathtt{H}$ by replacing each occurrence of one of the non-standard syncategorematic expressions by a constant of the appropriate type. Consequently, $\mathtt{H}_+$ is consistent. In turn, every other system introduced in this section is consistent, since all are fragments of $\mathtt{H}_+$.

Note that in each of our systems $\mathtt{FH}_{\heartsuit}$, the terms $\mathsf{E!}$ and $\lambda x.\exists y(x \equiv y)$ are provably coextensive, in the sense that $\mathsf{E!}x \leftrightarrow \exists y(x \equiv y)$ is derivable for any variable x of the right type. Thus the constants $\mathsf{E!}$ could be in principle removed from our logical signature. I keep them as primitive mainly to stress the central role existence predicates play in the logic of domain specifiers and its model theory.

3.3. **The Significance of Restricting Abstraction.** The syntax of the languages introduced in this section is non-standard in two ways. First, the presence of non-standard syncategorematic operations. Second, the restrictions imposed on λ-abstraction.

When working in the full language $\mathcal{L}_+$, the first choice amounts to a mostly superficial difference from standard higher-order languages. This remark is essentially repeating the point made earlier that standard higher-order logic does not accommodate genuine syncategorematicity. Here I want to stress how the point applies to quantifiers.

Item 5 in the definition of $\mathcal{L}_+$ introduces quantifiers as syncategorematic expressions rather than as constants with types of the form $(\sigma \to t) \to t$. However, corresponding terms of such types are definable in $\mathcal{L}_+$:

$$\hat{\exists} := \lambda X.\exists y Xy \qquad \hat{\forall} := \lambda X.\forall y Xy.$$

Call these the *categorematic quantifiers*. In $\mathtt{FH}_+$, every term of the form $\forall x P$ is $\beta\eta$-equivalent to $\hat{\forall}\lambda x.P$, and likewise for $\exists x P$. This means that syncategorematic quantification can always be re-expressed using categorematic quantifiers.

The situation is different when we work with $\mathcal{L}_\dagger$ terms only. Notice that categorematic quantifiers can still be defined in $\mathcal{L}_\dagger$: both $\hat{\exists}$ and $\hat{\forall}$ are $\mathcal{L}_\dagger$-terms. However, syncategorematic quantifiers are more expressive than categorematic ones. Since quantifiers can bind in positions where λ cannot, it is possible that $\forall xP$ is a $\mathcal{L}_\dagger$-term while $\hat{\forall}\lambda x.P$ is not. If so, ($\beta\eta$) alone cannot guarantee that syncategorematic quantification can always be re-expressed using categorematic quantifiers. In fact, it turns out that some instances of the following schema are actually not provable in $\mathtt{H}_\dagger$:[19]

$$\exists X(\forall xP \equiv \hat{\forall}X). \tag{$\forall$-$\hat{\forall}$}$$

Thus, when working in $\mathcal{L}_\dagger$, syncategorematic quantifiers are strictly more expressive than categorematic ones, which is why I have chosen to include them in the language.

The restrictions on abstraction imposed on $\mathcal{L}_\dagger$-terms also have consequences for how we theorize about identity. In the previous section, I said that each constant $\equiv_\sigma$ expresses the *applicative indiscernibility* relation at type σ. Let me say more about what that means. Given a constant $\approx\colon \sigma \to \sigma \to t$, it is worth distinguishing between three versions of Leibniz's Law that $\approx$ might satisfy:

$$M \approx N \to (FM \to FN) \tag{ALL}$$

$$M \approx N \to \forall X(XM \to XN) \tag{QLL}$$

$$M \approx N \to (P[M/x] \to P[N/x]) \qquad M, N \text{ free for } x \text{ in } P \tag{SLL}$$

Call these schemas, respectively, *applicative (schematic) Leibniz's Law*, *quantified Leibniz's Law*, and *substitutional Leibniz's Law*. The constant $\equiv$ is only stipulated to obey (ALL). And whatever identity is, it surely must obey (SLL).[20]

All three schemas are equivalent in $\mathtt{H}_+$.[21] In $\mathtt{FH}_+$, (ALL) and (SLL) are still equivalent, but (QLL) is strictly weaker: schematic generalizations are stronger than quantified generalizations in $\mathtt{FH}_+$. For example, the object language *Leibniz equivalence* relation, defined as

$$\approx_L := \lambda xy.\forall Z(Zx \leftrightarrow Zy)$$

satisfies (QLL), but not (SLL). To get a feel for this, consider Max Black's famous case of Castor and Pollux [Black, 1952], and assume it really is a counter-example to the identity of *qualitative* indiscernibles. Suppose further that every property is qualitative. Then, Castor and Pollux are Leibniz equivalent. Yet Castor is identical to Castor, but Pollux is not identical to Castor. This shows that $\approx_L$ does not satisfy (SLL), and in turn (ALL). Examples like these suggest that free logicians should not reduce identity to Leibniz equivalence.

For the same reasons, (QLL) and (SLL) are not equivalent in $\mathtt{FH}_\dagger$. In addition, the equivalence between (ALL) and (SLL) also breaks down. While (SLL) still implies (ALL), the converse does not hold. In order to prove (SLL) from (ALL)

[19] Of course, the negation of *any* instance of ($\forall$-$\hat{\forall}$) can be consistently added to $\mathtt{FH}_\dagger$, but this has more to do with the fact that $\mathtt{FH}_\dagger$ is a free logic than with the fact that it is formulated over $\mathcal{L}_\dagger$-terms. Indeed, the negation of every instance of ($\forall$-$\hat{\forall}$) can be consistently added to $\mathtt{FH}_+$, as well.

[20] This is not entirely uncontroversial, see, e.g., Bacon and Russell [2019].

[21] Indeed, (WLL) follows from (QLL) by an application of (UG). Moreover, $M \approx N \to ((\lambda x.P)M \to (\lambda x.P)N)$, which is $\beta\eta$-equivalent to $M \approx N \to (P[M/x] \to P[N/x])$, is an instance of (WLL), so (WLL) implies (SLL). Lastly, (SLL) implies (QLL) by (UG).

in $\mathtt{FH}_+$, one notes that $P[M/x]$ is $\beta\eta$-equivalent to $(\lambda x.P)M$, which is of the form FM. But $\lambda x.P$ can fail to be a $\mathcal{L}_\dagger$-term when P contains domain specifiers or $\blacksquare$, so that proof breaks down. As we shall see, there is no way around this: there are consistent extensions of $\mathtt{FH}_\dagger$ in which $\equiv$ does not satisfy (SLL). The intuition, here, is that two entities may share all their (existing and non-existing) higher-order properties, while still being distinguishable by means of syncategorematic terms.

Quantificationalists who theorize in $\mathcal{L}_\dagger$, therefore, should not conflate identity with either applicative indiscernibility or Leibniz equivalence. Identity, Leibniz equivalence, and applicative indiscernibility are relations that collapse in $\mathtt{H}_+$ but come apart in weaker settings, much like truth and double negation classically coincide but diverge in intuitionistic logic. All are interesting relations with their own theoretical roles to play.

Applicative indiscernibility plays an important role in my framework, and does not seem to be definable in terms of identity and other logical vocabulary.[22] That is why I take it as primitive. Part of its role lies in illuminating the nature of identity from a quantificationalist perspective. In what I take the correct $\mathcal{L}_\dagger$-logic for quantificationalism, the $\blacksquare$-necessitation of applicative indiscernibility satisfies (SLL). The quantificationalist can thus think of identity as *indistinguishability at all domains of quantification*. I say more on this in Sections 5 and 6.

4. Quantificational Substitution Structures

In this and the following section, I introduce a model-theoretic framework for interpreting the higher-order languages just introduced. This model theory gives a suggestive picture of the intended interpretation of domain specifiers, and can be used as a tool for checking the consistency of object language theories.

My model theory is an attempt at characterizing a metaphysical notion of domain specification by starting from a well-defined notion of domain specification on linguistic entities and then "de-syntactifying" it. Let me elaborate.

Notice, first, that there seems to be a well defined notion of domain specification as an operation on *language*. For simplicity, assume we are working in $\mathtt{H}_0$, so that we can ignore the complications induced by the distinctions between domain *restriction* and domain *specification*. For each predicate $F : \sigma \to t$, we can define a translation mapping α_F over $\mathcal{L}_0$, where $\alpha_F(M)$ is obtained by replacing every occurrence of the existence predicate $\mathsf{E!}_\sigma$ with F and syntactically restricting every quantifier by F. Thus, for example, if $G : \sigma \to t$ is a constant, we have:

$$\alpha_F(\forall x Gx) = \forall x(Fx \to Gx) \qquad \alpha_F(\exists x Gx) = \exists x(Fx \wedge Gx)$$
$$\alpha_F(\mathsf{E!} \wedge_\sigma G) = F \wedge_\sigma G$$

We can then say that the result of specifying the domain of quantification to the Fs in a sentence P is simply the sentence $\alpha_F(P)$.[23]

[22] Applicative indiscernibility is definable from identity and other logical vocabulary if the background theory of granularity is coarse enough to guarantee that provably coextensive properties are identical, as in Classicism [Bacon and Dorr, 2024] and Free Classicism [Bacon, 2024]. While I am attracted to a quantificationalist analogue of these theories, I do not want to bake too much granularity theory into my framework.

[23] This is not a good model of domain specification if the background logic is $\mathtt{FH}_0$ rather than $\mathtt{H}_0$, as the mappings α_F are not congruences with respect to provable equivalence in $\mathtt{FH}_0$. For example, even though $\exists x(x \equiv a)$ and $\mathsf{E!}a$ are provably equivalent, their α_F-images are not. To get a good model of domain specification in $\mathtt{FH}_0$, we need to embed $\mathtt{FH}_0$ into a classical higher-order

One way to obtain a model of *metaphysical* domain specification from the syntactic operation just sketched would be to force quasi-syntactic structure on reality. On this picture, properties and propositions have metaphysical structure that mirrors the syntactic structure of the linguistic entities that express them. Properties and propositions have *atomic constituents*, corresponding to the entities expressed by constants in the language that mirrors reality's structure. There should then be a well defined operation of substituting an atomic constituent in a property or proposition with some other constituent, just like there is a well defined operation of substituting a term for a constant in a term. We could think of the act of specifying the Fs as the domain of quantification in a property or proposition as the result of applying a *metaphysical substitution* of this sort, modeled after the translation function α_F, to that property or proposition.

The quasi-syntactic picture of metaphysical structure is deeply controversial: as we have seen in Section 2, it is widely taken to be refuted by the Russell-Myhill paradox. So, I do not wish to characterize domain specification by assuming it. But theorizing in terms of metaphysical substitutions does not require assuming the quasi-syntactic picture. Following Bacon [2019], we can instead take the notion of a metaphysical substitution as primitive, and impose constraints on metaphysical substitutions that mirror constraints satisfied by our translation functions α_F above. This leads to the notion of a *quantificational substitution structure*, which is the central notion of my model theory and the subject of the present section.

The higher-order languages discussed in the previous section can then be interpreted over models based on quantificational substitution structures, so that each domain specifier expresses a quantificational substitution and the operator $\blacksquare$ is interpreted as a kind of quantifier over quantificational substitutions. It is worth pointing out, before we begin, that not *every* quantificational substitution structure gives rise to a model for our languages. There are some further constraints that we will need to impose; I will discuss these in the next section.

4.1. **Background on Applicative Structures.** I begin with a brief review of *applicative structures*, the basic structures from which we will build quantificational substitution structures. For more details, the reader may consult [Bacon, 2023, Ch. 14].

Definition 4.1 (Applicative structure)**.** An *applicative structure* is a pair $\mathbf{A} = (A, App)$, where A is a typed family of sets A^σ and App is a typed family of functions $App^{\sigma\tau} : (A^{\sigma\to\tau} \times A^\sigma) \to A^\tau$.

---

logic in a language expanded with new "outer" classical quantifiers, so that our original quantifiers can be seen as restrictions of the new classical quantifiers by the original restricted predicates. More precisely, consider a language $\mathcal{L}_0^+$ expanded with additional syncategorematic quantifiers Π, Σ. We stipulate that these new quantifiers obey classical quantificational logic, and that our original quantifiers $\forall, \exists$ are restrictions of these by the existence predicates:

$$\forall x P \leftrightarrow \Pi x(\mathsf{E!}x \to P) \qquad \exists x P \leftrightarrow \Sigma x(\mathsf{E!}x \wedge P).$$

Note that in this logic, $\mathsf{E!}$ is an existence predicate for $\forall$ and $\exists$, but not for Π and Σ. For each predicate $F : \sigma \to t$, we can then define a translation mapping β_F from $\mathcal{L}_0$ to $\mathcal{L}_0^+$, where $\beta_F(M)$ is obtained by replacing each occurrence quantifier $\forall$ with the restriction of Π by $\mathsf{E!}$ (and likewise for $\exists$ and Σ) and subsequently substituting each occurrence of $\mathsf{E!}_\sigma$ with F in the result. Thus, for example,

$$\beta_F(\forall x G x) = \Pi x(Fx \to Gx) \qquad \beta_F(\exists x G x) = \exists x(Fx \wedge Gx).$$

We can then say that the result of *specifying* the domain of quantification to the Fs in a sentence P is simply the sentence $\beta_F(P)$.

The objects in A^σ represent the entities of type σ and App allows us to apply entities of function type to their arguments.

Given an applicative structure $\mathbf{A}$, an *applicative behavior* is a mapping $f : A^\sigma \to A^\tau$. An applicative behavior f is said to be *realized* in $\mathbf{A}$ when there is $\mathbf{f} \in A^{\sigma\to\tau}$ such that, for each $\mathbf{a} \in A^\sigma$, we have $f(\mathbf{a}) = App(\mathbf{f}, \mathbf{a})$.

We say that $\mathbf{A}$ is *full* when every applicative behavior on $\mathbf{A}$ is realized, and *functional* when no two elements of A^σ realize the same applicative behavior, for all types σ. Lastly, when $\mathbf{A}$ is functional, we say that $\mathbf{A}$ *has combinators* if for all types σ, τ, ρ there are elements $\mathbf{k} \in A^{\sigma\to\tau\to\sigma}$ and $\mathbf{s} \in A^{(\sigma\to\tau\to\rho)\to(\sigma\to\tau)\to\sigma\to\rho}$ such that

- $App((App(\mathbf{k}, \mathbf{a}), \mathbf{b})) = \mathbf{a}$ whenever $\mathbf{a} \in A^\sigma$ and $\mathbf{b} \in A^\tau$;
- $App(App(App(\mathbf{s}, \mathbf{f}), \mathbf{g}), \mathbf{a}) = App(App(\mathbf{f}, \mathbf{a}), App(\mathbf{g}, \mathbf{a}))$ whenever $\mathbf{f} \in A^{\sigma\to\tau\to\rho}$, $\mathbf{g} \in A^{\sigma\to\tau}$ and $\mathbf{a} \in A^\sigma$.

Definition 4.2 (Applicative Congruence)**.** An *applicative congruence* on an applicative structure $\mathbf{A}$ is a typed family of equivalence relations $\sim_\sigma \subseteq A^\sigma \times A^\sigma$ such that $\mathbf{a} \sim_\sigma \mathbf{a}'$ and $\mathbf{f} \sim_{\sigma\to\tau} \mathbf{f}'$ implies $App(\mathbf{f}, \mathbf{a}) \sim_\tau App(\mathbf{f}', \mathbf{a}')$, for all $\mathbf{a}, \mathbf{a}' \in A^\sigma$ and $\mathbf{f}, \mathbf{f}' \in A^{\sigma\to\tau}$.

Given an applicative congruence $\sim$ on $\mathbf{A}$, write $[\mathbf{a}]$ for the equivalence class of $\mathbf{a}$ under $\sim$. The *quotient* of $\mathbf{A}$ *through* $\sim$ is the applicative structure $\mathbf{A}_{/\sim} := ([A], [App])$, where

$$[A]^\sigma := \{[\mathbf{a}] : \mathbf{a} \in A^\sigma\} \qquad [App]^{\sigma\tau}([\mathbf{f}], [\mathbf{a}]) := [App^{\sigma\tau}(\mathbf{f}, \mathbf{a})].$$

Definition 4.3 (Direct product)**.** Let $\{\mathbf{A}_i = (A_i, App_i) : i \in I\}$ be an indexed family of applicative structures. The *direct product* of the $\mathbf{A}_i$s is the applicative structure $\mathbf{B} = (B, App)$, where B^σ consists of all mappings that assign an element of $\mathbf{A}_i^\sigma$ to each index $i \in I$, and where

$$App(\mathbf{f}, \mathbf{a})(i) := App_i(\mathbf{f}(i), \mathbf{a}(i)).$$

As a special case, a *direct power* of an applicative structure $\mathbf{A}$ over an index set I is the direct product of the constant family $\{\mathbf{A}_i : i \in I\}$ where $\mathbf{A}_i = \mathbf{A}$ for each $i \in I$.

It is straightforward to verify that a direct product of any family of functional applicative structures is functional, and that a direct product of any family of functional applicative structures that have combinators itself has combinators. Indeed, the combinators of such a direct products are the constant functions that map each index i to the appropriate combinator in the applicative structure $\mathbf{A}_i$. However, a direct product of a family of full applicative structures need not be full.

4.2. **Background on Substitution Structures.** I now introduce substitution structures, which regiment the idea of *metaphysical substitutions*. The material here is adapted from Bacon [2019], and can be skipped by readers familiar with that paper.

Definition 4.4 (Substitution Structure)**.** A *substitution structure* is a tuple $\mathfrak{A} = (\mathbf{I}, \mathbf{A}, Sub)$ where $\mathbf{A}$ is an applicative structure, $\mathbf{I} = (I, \circ)$ is a monoid, and Sub is a typed family of mappings $Sub^\sigma : I \times A^\sigma \to A^\sigma$ satisfying the following conditions:

(1) $Sub^\sigma(1, \mathbf{a}) = \mathbf{a}$, where 1 is the identity element of $\mathbf{I}$;
(2) $Sub^\sigma(i, Sub^\sigma(j, \mathbf{a})) = Sub^\sigma(i \circ j, \mathbf{a})$;

(3) $Sub^{\tau}(i, App^{\sigma\tau}(\mathbf{f}, \mathbf{a})) = App^{\sigma\tau}(Sub^{\sigma\to\tau}(i, \mathbf{f}), Sub^{\sigma}(i, \mathbf{a}))$.

The elements of $\mathbf{I}$ represent the metaphysical substitutions, while $\circ$ represents the operation of composing substitutions. The mapping *Sub* tells us how each substitution acts on each entity in the applicative structure $\mathbf{A}$.

I will adopt the convention of abbreviating $Sub^{\sigma}(i, \mathbf{a})$ as $i\mathbf{a}$ provided no ambiguities arise. With this notational convention in place, we can rewrite Items 1 to 3 more transparently as follows:

(1') $1\mathbf{a} = \mathbf{a}$, where 1 is the identity element of $\mathbf{I}$;
(2') $ij\mathbf{a} = i \circ j(\mathbf{a})$;
(3') $iApp(\mathbf{f}, \mathbf{a}) = App(i\mathbf{f}, i\mathbf{a})$.

Thus Item 1 says that the identity substitution leaves each entity unchanged; Item 2 says that applying two substitutions in succession is equivalent to applying their composition; and Item 3 says that substitutions distribute over application. I will also generally notate a substitution structure $(\mathbf{I}, \mathbf{A}, Sub)$ simply as $(\mathbf{I}, \mathbf{A})$, when the mapping *Sub* is clear from context.

A substitution structure $\mathfrak{A} = (\mathbf{I}, \mathbf{A})$ is *functional* (resp. *full*) when its underlying applicative structure is. Furthermore, we say that $\mathfrak{A}$ is *quasi-functional* when for any $\mathbf{f}, \mathbf{g} \in A^{\sigma\to t}$, we have $\mathbf{f} = \mathbf{g}$ precisely when $App(i\mathbf{f}, \mathbf{a}) = App(i\mathbf{g}, \mathbf{a})$ for each $\mathbf{a} \in A^{\sigma}$ and any $i \in I$. In other words, an applicative structure is quasi-functional when no two distinct elements realize the same applicative behavior *under all substitutions.*

When $\mathfrak{A} = (\mathbf{I}, \mathbf{A})$ is a quasi-functional substitution structure, we say that $\mathfrak{A}$ *has combinators* when for all types σ, τ, ρ there are elements $\mathbf{k} \in A^{\sigma\to\tau\to\sigma}$ and $\mathbf{s} \in A^{(\sigma\to\tau\to\rho)\to(\sigma\to\tau)\to\sigma\to\rho}$ which satisfy the conditions from the definition of having combinators for applicative structures *under all substitutions.* That is to say, for each $i \in I$, we have:

- $App((App(i\mathbf{k}, \mathbf{a}), \mathbf{b})) = \mathbf{a}$ whenever $\mathbf{a} \in A^{\sigma}$ and $\mathbf{b} \in A^{\tau}$;
- $App(App(App(i\mathbf{s}, \mathbf{f}), \mathbf{g}), \mathbf{a}) = App(App(\mathbf{f}, \mathbf{a}), App(\mathbf{g}, \mathbf{a}))$ whenever $\mathbf{f} \in A^{\sigma\to\tau\to\rho}$, $\mathbf{g} \in A^{\sigma\to\tau}$ and $\mathbf{a} \in A^{\sigma}$.

Definition 4.5 (Substitutional Congruence)**.** Given a substitution structure $\mathfrak{A} = (\mathbf{I}, \mathbf{A})$, a *substitutional congruence* on $\mathfrak{A}$ is a typed family of equivalence relations $\sim_{\sigma} \subseteq A^{\sigma} \times A^{\sigma}$ such that $\mathbf{a} \sim_{\sigma} \mathbf{b}$ implies $i\mathbf{a} \sim i\mathbf{b}$, for each $\mathbf{a}, \mathbf{b} \in A^{\sigma}$ and every $i \in \mathbf{I}$.

It is important to keep in mind that a substitutional congruence need not be an applicative congruence, nor the other way around.

4.3. **Quantificational Substitution Structures.** We now have the necessary tools to introduce the notion of a *quantificational* substitution structure. Basically, a quantificational substitution structure is a substitution structure where every substitution is determined by some *domain* and the way domains *determine* substitutions is well behaved with respect to the operations of the substitution structure.

I think of *domains* as ways of settling what it is to exist at a given type. A domain says, for finitely many types σ, that for an entity of that type to exist is for it to have a certain property. We can model this idea as follows.

Definition 4.6 (Domain)**.** Let $\mathbf{A}$ be an applicative structure. A *domain* on $\mathbf{A}$ is a partial mapping $\overline{\mathbf{f}} := Types \to \bigcup_{\sigma} A^{\sigma\to t}$ defined *on finitely many types*, such that $\overline{\mathbf{f}}(\sigma) \in A^{\sigma\to t}$ for each $\sigma \in Types$ on which $\overline{\mathbf{f}}$ is defined.

Basically, we should think of a domain $\bar{\mathbf{f}}$ as saying that for an entity of type σ to exist is for it to have the property $\bar{\mathbf{f}}(\sigma)$, when the latter is defined. Given the requirement that domains be defined on only finitely many types, any domain will be silent about what exists at types on which it is undefined.

Given a nice type sequence $\bar{\sigma}$, we say $\bar{\mathbf{f}}$ is a $\bar{\sigma}$-*domain* when $\bar{\mathbf{f}}$ is defined on σ precisely when $\sigma \in \bar{\sigma}$. I notate the set of all domains on $\mathbf{A}$ as $Dom(\mathbf{A})$ and just Dom when $\mathbf{A}$ is clear from context. Likewise, I write $Dom_{\bar{\sigma}}(\mathbf{A})$, or simply $Dom_{\bar{\sigma}}$, for the set of all $\bar{\sigma}$-domains on $\mathbf{A}$.

Convention 4.7. I will often use sequential notation for domains, writing $(\mathbf{f}_\sigma)_{\sigma\in\bar{\sigma}}$ for a domain $\bar{\mathbf{f}}$, where $\mathbf{f}_\sigma := \bar{\mathbf{f}}(\sigma)$ for each $\sigma \in \bar{\sigma}$, and speak of domains as if they were type-indexed sequences of properties. I also abuse notation and treat domains defined on a single type σ as elements of $A^{\sigma\to t}$ rather than functions from $\{\sigma\}$ to $A^{\sigma\to t}$. These domains are called *unit domains*.

Each domain *determines* a quantificational substitution, whose function is to specify that domain as the new domain of quantification within its argument. We can model this by means of *substitution assignment functions*.

Definition 4.8 (Substitution Assignment Function)**.** Let $\mathfrak{A} = (\mathbf{I}, \mathbf{A})$ be a substitution structure. A *substitution assignment function* is a surjective mapping $i_{(\cdot)} : Dom \to I$.

We think of $i_{\bar{\mathbf{f}}}$ as the quantificational substitution determined by $\bar{\mathbf{f}}$. In models based on QSSs, we will semantically associate a domain specifier $\mathsf{At}(\bar{F})$ with the substitution $i_{\bar{\mathbf{f}}}$, where $\bar{\mathbf{f}}$ is the unique domain such that $\mathbf{f}_\sigma$ is the interpretation of F_σ whenever the latter is defined and undefined otherwise.

Some domains are *current*, in the sense that for each type σ on which they are defined, what it is for an entity of type σ to exist according to that domain is simply for it to exist. We model this idea through the notion of an *identity domain*.

Definition 4.9 (Identity Domain)**.** Let $\mathfrak{A} = (\mathbf{I}, \mathbf{A}, i_{(\cdot)})$ be a substitution structure equipped with a substitution assignment function. A *identity domain* is any domain $\bar{\mathbf{e}} \in Dom$ such that $i_{\bar{\mathbf{e}}}$ is the identity substitution in $\mathbf{I}$.

I write $\mathbf{e}_\sigma$ for an arbitrary element of $\mathbf{A}^{\sigma\to t}$, if there is one, such that $i_{\mathbf{e}_\sigma} = 1$.

When defining models based on quantificational substitution structures, I will require that an existence predicate $\mathsf{E!}_\sigma$ express $\mathbf{e}_\sigma$. Thus we may think of identity domains as consisting entirely of interpretations of existence predicates. The identity substitution can be thought of as a quantificational substitution that specifies *the existing things* as the new domain of quantification within its argument.

Through substitution assignment functions, we can also define an operation of *domain composition*. Given two nice type sequences $\bar{\sigma}$ and $\bar{\tau}$, let $\bar{\sigma} + \bar{\tau}$ be an arbitrary nice type sequence such that the types occurring in it are exactly the types that occur in either $\bar{\sigma}$ or $\bar{\tau}$. If $\bar{\mathbf{f}}, \bar{\mathbf{g}} \in Dom$ are respectively a $\bar{\sigma}$- and a $\bar{\tau}$-domain, the *composition* $\bar{\mathbf{f}} \bullet \bar{\mathbf{g}}$ is a $\bar{\sigma} + \bar{\tau}$-domain defined by putting

$$(\bar{\mathbf{f}} \bullet \bar{\mathbf{g}})(\sigma) := \begin{cases} i_{\bar{\mathbf{f}}}\mathbf{g}_\sigma & \text{if } \sigma \in \bar{\tau} \\ \mathbf{f}_\sigma & \text{if } \sigma \in \bar{\sigma} \text{ but } \sigma \notin \bar{\tau} \\ \text{undefined} & \text{otherwise.} \end{cases} \tag{$\bullet$}$$

Intuitively, the domain $\bar{\mathbf{f}} \bullet \bar{\mathbf{g}}$ is the domain $\bar{\mathbf{g}}$ seen from the "perspective" of the domain $\bar{\mathbf{f}}$. We can think of domains as characterizing not only what exists, but also what exists according to other domains—including themselves. So, what it is for an entity of the same type to exist according to $\bar{\mathbf{f}} \bullet \bar{\mathbf{g}}$ is the same as what it is for it to *exist according to* $\bar{\mathbf{g}}$, *according to* $\bar{\mathbf{f}}$. Continuing the example from Section 2, the domain of *lovers* specifies the lovers as what exists, and also specifies the *lovers among lovers* as what exists *according to the domain of lovers*. As both the name and the gloss just given suggest, we will use domain composition to characterize the composition of quantificational substitutions.

A *quantificational substitution structure* is defined as a substitution structure equipped with a substitution assignment function that is "well behaved" with respect to domain composition.

Definition 4.10 (Quantificational Substitution Structure)**.** A *quantificational substitution structure*, henceforth a *QSS*, is a tuple $\mathfrak{A} = (\mathbf{I}, \mathbf{A}, \mathit{Sub}, i_{(\cdot)})$, where $(\mathbf{I}, \mathbf{A}, \mathit{Sub})$ is a substitution structure, $i. : \mathit{Dom} \to I$ is a substitution assignment function, such that an identity unit σ-domain $\mathbf{e}_\sigma$ exists for every type σ, and the conditions

$$i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{g}}} = i_{\bar{\mathbf{f}} \bullet \bar{\mathbf{g}}} \qquad \text{(Composition)}$$
$$\bar{\mathbf{e}} \bullet \bar{\mathbf{f}} = \bar{\mathbf{f}} \bullet \bar{\mathbf{e}} \qquad \text{(Identity)}$$

hold for all domains $\bar{\mathbf{f}}, \bar{\mathbf{g}}$ and any identity domain $\bar{\mathbf{e}}$.

Some comments are in order. Composition says that domain composition commutes with the substitution assignment function. Since $i_{(\cdot)}$ is surjective, for any two domains $\bar{\mathbf{f}}, \bar{\mathbf{g}}$ there must be a unique domain $\bar{\mathbf{h}}$ such that $i_{\bar{\mathbf{h}}} = i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{g}}}$. Composition identifies this domain as $\bar{\mathbf{f}} \bullet \bar{\mathbf{g}}$. So, Composition can be seen as a model-theoretic regimentation of the way of thinking about the composition of domain specifiers discussed at the end of Section 2: to specify the domain as the Gs and then as the Fs is the same as specifying the domain as *the Fs at the Gs*.

The existence assumption about identity unit domains is motivated by the use I intend to make of identity domains as domains consisting entirely of interpretations of existence predicates. In fact, the existence of all identity unit domains guarantees that an identity $\bar{\sigma}$-domain exists for every nice type sequence $\bar{\sigma}$. Indeed, when $\bar{\sigma} = (\sigma_1, \ldots, \sigma_n)$, an identity $\bar{\sigma}$-domain $\bar{\mathbf{e}}$ can be defined as

$$\bar{\mathbf{e}} := \mathbf{e}_{\sigma_1} \bullet \mathbf{e}_{\sigma_2} \bullet \cdots \bullet \mathbf{e}_{\sigma_n}.$$

It follows immediately that $\bar{\mathbf{e}}(\sigma) = \mathbf{e}_\sigma$ for each $\sigma \in \bar{\sigma}$. That $\bar{\mathbf{e}}$ is actually an identity domain follows from Composition.

Identity is a bit more complicated. We could obtain an equivalent definition if we replaced it with the claim that both the following identities hold for any $\bar{\sigma}$-domain

$\bar{\mathbf{f}}$:[24]

$$i_{\bar{\mathbf{f}}}\mathbf{e}_\sigma = \mathbf{f}_\sigma \qquad \text{whenever } \sigma \in \bar{\sigma} \qquad \text{(Generality)}$$

$$i_{\bar{\mathbf{f}}}\mathbf{e}_\tau = \mathbf{e}_\tau \qquad \text{whenever } \tau \notin \bar{\sigma} \qquad \text{(Modularity)}$$

Both conditions concern what quantificational substitutions do to identity unit domains. Generality requires that applying $i_{\bar{\mathbf{f}}}$ to $\mathbf{e}_\sigma$ retrieve the σ-th projection of the domain $\bar{\mathbf{f}}$. It is this condition that will eventually ensure the schema (AtG) is valid. Note, also, that Generality explains the second case in the definition of domain composition ($\bullet$): when $\mathbf{f}_\sigma$ is defined but $\mathbf{g}_\sigma$ is not, then $(\bar{\mathbf{f}} \bullet \bar{\mathbf{g}})(\sigma) = \mathbf{f}_\sigma$. In view of Generality, this is exactly what would happen if we were to expand $\bar{\mathbf{g}}$ to a domain $\bar{\mathbf{g}}'$ defined at σ as well, with $\bar{\mathbf{g}}'(\sigma) = \mathbf{e}_\sigma$.

On the other hand, Modularity requires that $\mathbf{e}_\tau$ be a fixpoint of $i_{\bar{\mathbf{f}}}$ whenever $\tau \notin \bar{\sigma}$. This condition corresponds to the object-language claim *to exist*$_\tau$ *at the* $F^{\sigma\to t}$*s is to be an* $F^{\sigma\to t}$ when $\tau \neq \sigma$, to be properly formalized. This claim is inspired by the behavior of the translations α_F discussed in the introduction to the present section, specifically by the fact that $\alpha_{\mathsf{E!}_\sigma}(\mathsf{E!}_\tau) = \mathsf{E!}_\tau$ whenever $\sigma \neq \tau$.

Together, Generality and Modularity paint a picture of higher-order existence as *freely recombinable*. What exists at one type is completely independent on what exists at another type. I take this to be a core aspect of the intended interpretation of domain specification. But it is certainly a philosophically controversial picture: given reasonable background assumptions, a theory of domain specifiers based on QSSs is incompatible with theses about higher-order existence familiar from the literature on higher-order contingentism. I discuss this point in more detail in other work, though it is worth pointing out here that alternative views of higher-order existence can be accommodated in a variation of the present framework by replacing Generality and Modularity with appropriate conditions.

4.4. **The Algebraic Structure of Domains.** Composition and Identity are also well motivated from a technical perspective. Together, they ensure that in any QSS $\mathfrak{A}$, the family of all domains can be seen as a monoid having essentially the same structure as the monoid of substitutions $\mathbf{I}$.

Let me elaborate. First, as a near immediate consequence of Composition it follows that the domain composition $\bullet$ is associative in any QSS. Thus, its restriction to any set $Dom_{\bar{\sigma}}$ is associative as well. Moreover, we have:

Lemma 4.11. *Let $\mathfrak{A}$ be a QSS. For any nice type sequence $\bar{\sigma}$, any identity $\bar{\sigma}$-domain is an identity element with respect to $\bullet$, in the sense that the identities*

$$\bar{\mathbf{e}} \bullet \bar{\mathbf{f}} = \bar{\mathbf{f}} = \bar{\mathbf{f}} \bullet \bar{\mathbf{e}}$$

hold for each $\bar{\sigma}$-domain $\bar{\mathbf{f}}$.

[24] Note $i_{\bar{\mathbf{f}}}\mathbf{e}_\sigma = (\bar{\mathbf{f}} \bullet \mathbf{e}_\sigma)(\sigma)$. By Identity the right-hand side equals $(\mathbf{e}_\sigma \bullet \bar{\mathbf{f}})(\sigma) = \mathbf{f}_\sigma$. This gives us Generality. For Modularity, note $i_{\bar{\mathbf{f}}}\mathbf{e}_\tau = (\bar{\mathbf{f}} \bullet \mathbf{e}_\tau)(\tau)$. By Identity, the right-hand side equals $(\mathbf{e}_\tau \bullet \bar{\mathbf{f}})(\tau)$, which in turn equals $\mathbf{e}_\tau$ by the definition of $\bullet$. Conversely, assume a tuple $\mathfrak{A} = (\mathbf{I}, \mathbf{A}, \mathit{Sub}, i_{(\cdot)})$ satisfies the first three conditions of Definition 4.10, plus Generality and Modularity. Let $\bar{\mathbf{e}}$ be an identity $\bar{\sigma}$-domain and $\bar{\mathbf{f}}$ be a $\bar{\tau}$-domain. We show that $(\bar{\mathbf{e}}\bullet\bar{\mathbf{f}})(\rho) = (\bar{\mathbf{f}}\bullet\bar{\mathbf{e}})(\rho)$ holds for each type $\rho \in \bar{\sigma} + \bar{\tau}$. If ρ occurs in both $\bar{\sigma}$ and $\bar{\tau}$, the claim follows from Generality and the fact that $\bar{\mathbf{e}}$ is an identity domain. If ρ occurs in $\bar{\sigma}$ but not in $\bar{\tau}$, then $(\bar{\mathbf{e}} \bullet \bar{\mathbf{f}})(\rho) = \mathbf{e}_\rho$. The right-hand side equals $i_{\bar{\mathbf{f}}}\mathbf{e}_\rho$ by Modularity, which is just $(\bar{\mathbf{f}} \bullet \bar{\mathbf{e}})(\rho)$. Finally, suppose ρ occurs $\bar{\tau}$ but not in $\bar{\sigma}$. Then $(\bar{\mathbf{e}} \bullet \bar{\mathbf{f}})(\rho) = i_{\bar{\mathbf{e}}}\mathbf{f}_\rho = \mathbf{f}_\rho$. By Generality, the last item equals $i_{\bar{\mathbf{f}}}\mathbf{e}_\rho$, which is just $(\bar{\mathbf{f}} \bullet \bar{\mathbf{e}})(\rho)$.

Proof. By Identity, $\bar{\mathbf{e}} \bullet \bar{\mathbf{f}} = \bar{\mathbf{f}} \bullet \bar{\mathbf{e}}$. Further, since $i_{\bar{\mathbf{e}}}$ is the identity substitution, by the definition of $\bullet$ the left-hand side equals $\bar{\mathbf{f}}$. □

Consequently any QSS contains a *unique* identity $\bar{\sigma}$-domain, for each nice type sequence $\bar{\sigma}$.[25]

We have now established:

Lemma 4.12. *Let $\mathfrak{A}$ be a QSS. For any nice type sequence $\bar{\sigma}$, the set of all $\bar{\sigma}$-domains $Dom_{\bar{\sigma}}$ is a monoid under the restriction of $\bullet$ to $\bar{\sigma}$-domains. The identity element is given by the unique identity $\bar{\sigma}$-domain.*

We then unify the class of monoids $(Dom_{\bar{\sigma}}, \bullet)$ from a QSS $\mathfrak{A}$ into a single monoid via a direct limit construction. When $\bar{\sigma}$ and $\bar{\tau}$ are nice type sequences such that every type that occurs in $\bar{\sigma}$ occurs in $\bar{\tau}$, there is a mapping from $pad^{\bar{\tau}}_{\bar{\sigma}} : Dom_{\bar{\sigma}} \to Dom_{\bar{\tau}}$ given by

$$pad^{\bar{\tau}}_{\bar{\sigma}}(\bar{\mathbf{f}}) = \bar{\mathbf{e}} \bullet \bar{\mathbf{f}} \qquad \bar{\mathbf{e}} \text{ an identity } \bar{\tau}\text{-domain.} \tag{Padding}$$

We should think of $\bar{\mathbf{e}} \bullet \bar{\mathbf{f}}$ as the "representative" of $\bar{\mathbf{f}}$ among $\bar{\tau}$-domains. For note that $i_{\bar{\mathbf{f}}}\mathbf{a} = i_{\bar{\mathbf{e}}\bullet\bar{\mathbf{f}}}\mathbf{a}$ holds for any $\mathbf{a} \in A^{\rho}$, for any type ρ. Indeed, since $\bar{\mathbf{e}}$ is an identity domain we have $i_{\bar{\mathbf{f}}}\mathbf{a} = (i_{\bar{\mathbf{e}}} \circ i_{\bar{\mathbf{f}}})\mathbf{a}$ and by Composition the right-hand side equals $i_{\bar{\mathbf{e}}\bullet\bar{\mathbf{f}}}\mathbf{a}$.

Note that each mapping $pad^{\bar{\tau}}_{\bar{\sigma}}$ is a monoid homomorphism from $(Dom_{\bar{\sigma}}, \bullet)$ to $(Dom_{\bar{\tau}}, \bullet)$. For since the identity $\bar{\sigma}$-domain is always uniquely decomposable into unit identity domains, $pad^{\bar{\tau}}_{\bar{\sigma}}$ maps identity domains to identity domains. Moreover, $pad^{\bar{\tau}}_{\bar{\sigma}}$ commutes with domain composition, which is to say that

$$\bar{\mathbf{e}} \bullet (\bar{\mathbf{f}} \bullet \bar{\mathbf{g}}) = (\bar{\mathbf{e}} \bullet \bar{\mathbf{f}}) \bullet (\bar{\mathbf{e}} \bullet \bar{\mathbf{g}}) \tag{13}$$

whenever $\bar{\mathbf{e}}$ is the identity $\bar{\tau}$ domain and $\bar{\mathbf{f}}, \bar{\mathbf{g}}$ are $\bar{\sigma}$-domains. This follows from the associativity of $\bullet$ and the fact that $\bar{\mathbf{e}} \bullet \bar{\mathbf{e}} = \bar{\mathbf{e}}$.

Note the class of monoids $(Dom_{\bar{\sigma}}, \bullet)$ is partially order by the relation $(Dom_{\bar{\sigma}}, \bullet) \leq (Dom_{\bar{\tau}}, \bullet)$ which holds iff any $\bar{\tau}$-domain is defined on at least as many types as any $\bar{\sigma}$-domain. We can abbreviate this as $\bar{\sigma} \leq \bar{\tau}$. Further, the padding map $pad^{\bar{\sigma}}_{\bar{\sigma}}$ is the identity mapping on $Dom_{\bar{\sigma}}$, and composing $pad^{\bar{\tau}}_{\bar{\rho}}$ with $pad^{\bar{\rho}}_{\bar{\sigma}}$ yields $pad^{\bar{\tau}}_{\bar{\sigma}}$ whenever $\bar{\sigma} \leq \bar{\rho} \leq \bar{\tau}$. We have thus shown that the class of monoids $(Dom_{\bar{\sigma}}, \bullet)$ is a directed system.

Now, define

$$\bar{\mathbf{f}} \approx \bar{\mathbf{g}} :\Longleftrightarrow \bar{\mathbf{e}} \bullet \bar{\mathbf{f}} = \bar{\mathbf{e}} \bullet \bar{\mathbf{g}} \text{ for some identity domain } \bar{\mathbf{e}}.$$

Basically, $\bar{\mathbf{f}} \approx \bar{\mathbf{g}}$ holds when $\bar{\mathbf{f}}$ and $\bar{\mathbf{g}}$ have the same "representative" among domains living higher-up in the directed system. The *direct limit* of our directed system of monoids is the structure $([Dom], [\bullet])$ consisting of the set $[Dom]$ of equivalence classes under $\approx$, together with the operation $[\bullet]$ defined as the lifting of $\bullet$ to equivalence classes under $\approx$. This operation is well defined, since $\approx$ is a congruence

[25]This fact has significant philosophical consequences. It allows us to prove that, in what I take to be the correct logic for Quantificationalism, existence predicates—and consequently free quantifiers—are uniquely determined by their inferential role. Similar uniqueness results have been used by classical logicians to argue for the substantivity of ontological disputes [Williamson, 1988], as well as for the intelligibility of primitive higher-order quantifiers [Bacon, 2023]. But these strategies seem not to generalize to a free setting, since standard systems of free logic are not strong enough to prove uniqueness results of this sort. I discuss this point in more detail other work.

with respect to $\bullet$ in the sense that $\bar{\mathbf{f}} \approx \bar{\mathbf{f}}'$ and $\bar{\mathbf{g}} \approx \bar{\mathbf{g}}'$ implies $\bar{\mathbf{f}} \bullet \bar{\mathbf{g}} \approx \bar{\mathbf{f}}' \bullet \bar{\mathbf{g}}'$. Now, let $[i] : [Dom] \to I$ be the mapping given by $[\bar{\mathbf{f}}] \mapsto i_{\bar{\mathbf{f}}}$. This mapping, too, is well defined, for $\bar{\mathbf{f}} \approx \bar{\mathbf{g}}$ implies that $i_{\bar{\mathbf{f}}} = i_{\bar{\mathbf{g}}}$. We can now establish our desired result.

Theorem 4.13. *Let $\mathfrak{A}$ be any QSS. $([Dom], [\bullet])$ is a monoid. The identity element is given by $[\bar{\mathbf{e}}]$, where $\bar{\mathbf{e}}$ is any identity domain. Moreover, the mapping $[i]$ is a monoid homomorphism from $([Dom], [\bullet])$ to $(I, \circ)$.*

Proof. The direct limit of a directed system of monoids is always a monoid, where the identity element is given by the equivalence class of all identity elements of the monoids in the directed system. For the second part of the theorem, by definition of identity domains, $[i][\bar{\mathbf{e}}] = [i_{\bar{\mathbf{e}}}]$ is always the identity element of I. Moreover,

$$\begin{aligned} [i]([\bar{\mathbf{f}}][\bullet][\bar{\mathbf{g}}]) &= i_{\bar{\mathbf{f}} \bullet \bar{\mathbf{g}}} \\ &= i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{g}}} && \text{by Composition} \\ &= i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{g}}} \\ &= [i][\bar{\mathbf{f}}] \circ [i][\bar{\mathbf{g}}] \end{aligned}$$

holds for all domains $\bar{\mathbf{f}}, \bar{\mathbf{g}}$. □

4.5. **Congruences and Quotients.** Let me now turn to the topic of congruences. To the notions of *applicative* and *substitutional* congruences, we add the notion of a *quantificational congruence*: an equivalence relation that is "well behaved" with respect to the mapping $i_{(\cdot)}$. More precisely, let $\sim$ be a typed family of equivalence relations on an applicative structure $\mathbf{A}$ and let $\bar{\mathbf{f}}, \bar{\mathbf{g}}$ be respectively a $\bar{\sigma}$- and a $\bar{\tau}$-domain on $\mathbf{A}$. Write $\bar{\mathbf{f}} \sim \bar{\mathbf{g}}$ to mean that

$$(pad_{\bar{\sigma}}^{\bar{\sigma}+\bar{\tau}}(\bar{\mathbf{f}}))(\rho) \sim (pad_{\bar{\tau}}^{\bar{\sigma}+\bar{\tau}}(\bar{\mathbf{g}}))(\rho) \text{ for all } \rho \in \bar{\sigma} + \bar{\tau}.$$

In other words, $\bar{\mathbf{f}} \sim \bar{\mathbf{g}}$ holds when the "representatives" of $\bar{\mathbf{f}}$ and $\bar{\mathbf{g}}$ among $\bar{\sigma} + \bar{\tau}$-domains have $\sim$-equivalent projections.

Definition 4.14 (Quantificational congruence)**.** A *quantificational congruence* on a QSS $\mathfrak{A}$ is a typed family $\sim$ of equivalence relations on $\mathfrak{A}$, such that $\bar{\mathbf{f}} \sim \bar{\mathbf{g}}$ implies $i_{\bar{\mathbf{f}}}\mathbf{a} \sim i_{\bar{\mathbf{g}}}\mathbf{a}$ for every $\mathbf{a} \in A^{\sigma}$, whenever $\bar{\mathbf{f}}, \bar{\mathbf{g}}$ are domains over $\mathbf{A}$.

By putting together this notion with those of an applicative congruence and of a substitutional congruence, we reach the general notion of a *congruence*.

Definition 4.15 (Congruence)**.** A *congruence* on a QSS $\mathfrak{A}$ is a quantificational congruence on $\mathfrak{A}$ that is also a substitutional congruence for the underlying substitution structure and an applicative congruence for the underlying applicative structure.

It is worth noting for later reference that congruences as just defined are automatically congruences for domain composition:

Proposition 4.16. *Let $\sim$ be a congruence on a QSS $\mathfrak{A}$ and let $\bar{\mathbf{f}}, \bar{\mathbf{f}}', \bar{\mathbf{g}}, \bar{\mathbf{g}}'$ be domains on $\mathfrak{A}$ with $\bar{\mathbf{f}} \sim \bar{\mathbf{f}}'$ and $\bar{\mathbf{g}} \sim \bar{\mathbf{g}}'$. Then $\bar{\mathbf{f}} \bullet \bar{\mathbf{g}} \sim \bar{\mathbf{f}}' \bullet \bar{\mathbf{g}}'$.*

Proof. Wlog, assume $\bar{\mathbf{f}}, \bar{\mathbf{f}}'$ are both $\bar{\sigma}$-domains and $\bar{\mathbf{g}}, \bar{\mathbf{g}}'$ are both $\bar{\tau}$-domains.[26] Let ρ be any type. We show $(\bar{\mathbf{f}} \bullet \bar{\mathbf{g}})(\rho) \sim (\bar{\mathbf{f}}' \bullet \bar{\mathbf{g}}')(\rho)$ whenever $\rho \in \bar{\sigma} + \bar{\tau}$. If $\rho \in \bar{\tau}$, then $(\bar{\mathbf{f}} \bullet \bar{\mathbf{g}})(\rho) = i_{\bar{\mathbf{f}}}\mathbf{g}_\rho \sim i_{\bar{\mathbf{f}}'}\mathbf{g}'_\sigma$. Otherwise, $(\bar{\mathbf{f}} \bullet \bar{\mathbf{g}})(\rho) = \mathbf{f}_\rho \sim \mathbf{f}'_\rho$. □

Congruences can be used to define an operation of *quotienting* for QSSs. When $\sim$ is a congruence on a QSS $\mathfrak{A}$, write $i_{\bar{\mathbf{f}}} \sim i_{\bar{\mathbf{g}}}$ to mean that $i_{\bar{\mathbf{f}}}\mathbf{a} \sim i_{\bar{\mathbf{g}}}\mathbf{a}$ holds for every $\mathbf{a} \in A^\sigma$ and write $[i_{\bar{\mathbf{f}}}]$ for the equivalence class of $i_{\bar{\mathbf{f}}}$ under this lifting of $\sim$.

Definition 4.17 (Quotient)**.** Let $\mathfrak{A}$ be a QSS and $\sim$ a congruence on $\mathfrak{A}$. When $\bar{\mathbf{f}}$ is a $\bar{\sigma}$-domain, write $[\bar{\mathbf{f}}]$ for the partial function with $[\bar{\mathbf{f}}](\sigma) := [\mathbf{f}_\sigma]$ when $\sigma \in \bar{\sigma}$ and undefined otherwise. The *quotient* of $\mathfrak{A}$ under $\sim$ is the QSS $\mathfrak{A}_{/\sim} := ([\mathbf{I}], [\mathbf{A}], [Sub], [i_{(\cdot)}], [\xi])$, where

1. $[\mathbf{A}]$ is the quotient of $\mathbf{A}$ under $\sim$;
2. $[\mathbf{I}] = ([I], [\circ])$ is the quotient of the monoid $(I, \circ)$ under the lifting of $\sim$ to I;
3. $[Sub]([i_{\bar{\mathbf{f}}}], [\mathbf{a}]) = [Sub(i_{\bar{\mathbf{f}}}, \mathbf{a})]$;
4. $[i]_\cdot$ is the mapping $[\bar{\mathbf{f}}] \mapsto [i_{\bar{\mathbf{f}}}]$.

It is straightforward to verify that this is, indeed, a QSS.

4.6. **Stability.** Finally, I introduce the model-theoretic notion of *stability* and some of its weakenings. Stability is essentially the restriction of the concept of *purity* from Bacon [2019] to QSSs.

Definition 4.18 (Stability)**.** Let $\mathfrak{A}$ be a QSS. An entity $\mathbf{a} \in A^\sigma$ is called

- $\bar{\sigma}$*-stable* in $\mathfrak{A}$ when $i_{\bar{\mathbf{f}}}\mathbf{a} = \mathbf{a}$ for every $\bar{\sigma}$-domain $\bar{\mathbf{f}}$;
- *Stable* in $\mathfrak{A}$ when it is $\bar{\sigma}$-stable for all nice type sequences $\bar{\sigma}$.

If we think of quantificational substitutions as only moving entities that *involve* quantification, then stable entities are naturally thought of as entities that *do not* involve quantification. Weakenings of the notions of stability characterize entities that do not involve quantification ranging over entities of certain types only.

A *stable domain* is any domain $\bar{\mathbf{f}}$ such that $\mathbf{f}_\sigma$ is stable whenever it is defined. A *stabilizing substitution* is any substitution $i_{\bar{\mathbf{f}}}$ such that $\bar{\mathbf{f}}$ is a stable domain. Likewise, the notions of a $\bar{\sigma}$*-stable domain* and of a $\bar{\sigma}$*-stabilizing substitution* are defined.

It turns out that given minimal assumptions about the existence of stable domains, stable (resp. $\bar{\sigma}$-stable) entities can be equivalently characterized as those entities that lie in the range of some stabilizing (resp. $\bar{\sigma}$-stabilizing) substitution. This result will come in handy in Section 6.

Proposition 4.19. *Let $\mathfrak{A}$ be a QSS containing at least one stable (resp. $\bar{\sigma}$-stable) $\bar{\tau}$-domain for each nice type sequence $\bar{\tau}$, and let $\mathbf{a} \in A^\sigma$. Then $\mathbf{a}$ is stable (resp. $\bar{\sigma}$-stable) in $\mathfrak{A}$ iff for all nice type sequences $\bar{\tau}$ there a stable (resp. $\bar{\sigma}$-stable) $\bar{\tau}$-domain $\bar{\mathbf{f}}$ such that $i_{\bar{\mathbf{f}}}\mathbf{a} = \mathbf{a}$.*

Proof. The left-to-right direction is obvious given the existence assumptions about stable domains. Conversely, take any stabilizing substitution $i_{\bar{\mathbf{g}}}$. When $\bar{\mathbf{g}}$ is a

[26] If they were not, we could find appropriate paddings of these sequences which satisfy the assumption just stated and run the argument with the padded sequence. The desired claim about the original sequences would follow from this argument.

$\bar{\tau}$-domain, we know there is a stable $\bar{\tau}$-domain $\bar{\mathbf{f}}$ such that $i_{\bar{\mathbf{f}}}\mathbf{a} = \mathbf{a}$. But by (Composition) it follows that

$$i_{\bar{\mathbf{g}}}\mathbf{a} = i_{\bar{\mathbf{g}}}(i_{\bar{\mathbf{f}}}\mathbf{a}) = i_{\bar{\mathbf{g}}\circ\bar{\mathbf{f}}}(\mathbf{a}) = i_{\bar{\mathbf{f}}}(\mathbf{a}) = \mathbf{a}.$$

So, $\mathbf{a}$ is stable in $\mathfrak{A}$. The same argument works for $\bar{\sigma}$-stability. □

4.7. **Existence results.** Now that the main definitions have been set up, it is time to show that this was not all for nothing: QSSs actually exist. It is straightforward to come up with degenerate cases of QSSs. For example, let $\mathfrak{A}$ be any applicative structure containing a single entity at each type. We can expand $\mathfrak{A}$ to a substitution structure by equipping it with the trivial monoid of substitution, containing only the identity substitution. We can then define $i_{(\cdot)}$ by having every domain determine the identity substitution. The resulting structure is a QSS, albeit a rather uninteresting one.

I give two more interesting examples of QSSs. In my first example, I show how to construct a more realistic QSS by starting from the syntactic model of domain specification sketched in the introduction to the present section. The construction is similar to that used to build term models of higher-order logics. We start from terms and the translations α_F sketched out earlier, form equivalence classes of terms under an appropriate equivalence relation, then lift the translation mappings to substitutions defined on equivalence classes of terms. This is still not a perfect example, however: the resulting QSS cannot be turned into a $\mathcal{L}_\dagger$-model (in the sense of Section 5), because it is not rich enough to interpret syncategorematic quantification.

In the second example, I show how to construct a QSS as a direct power of any applicative structure. This example shows that while the intended understanding of domain specification is inspired by the syntactic model, it does not requires us to think of reality as somehow built out of entities with quasi-syntactic structure. Moreover, as I shall prove in the next section, this construction yields QSSs that can be turned into $\mathcal{L}_\dagger$-models with interesting properties, so long as the original applicative structure we take the power of is nice enough.

Example 4.20 (Term Structure Construction)**.** We work in $\mathcal{L}_0$. For any nice term sequence $\bar{F}$, define the mapping $\alpha_{\bar{F}}$ recursively as follows.

(1) $\alpha_{\bar{F}}(C) := C$ for each constant $C \in \Sigma^\sigma$ other than $\mathsf{E!}_\sigma$;
(2) $\alpha_{\bar{F}}(\mathsf{E!}_\sigma) := F_\sigma$ if F_σ is defined, $\alpha_{\bar{F}}(\mathsf{E!}_\sigma) := \mathsf{E!}_\sigma$ otherwise;
(3) $\alpha_{\bar{F}}(x) := x$ for each variable $x \in \mathit{Var}^\sigma$;
(4) $\alpha_{\bar{F}}(MN) := \alpha_{\bar{F}}(M)\alpha_{\bar{F}}(N)$ whenever $M : \sigma \to \tau$ and $N : \sigma$;
(5) $\alpha_{\bar{F}}(\lambda x.M) := \lambda x.\alpha_{\bar{F}}(M)$;
(6) $\alpha_{\bar{F}}(QxP) := Qx\alpha^t_{\bar{F}}(P)$ whenever $x \in \mathit{Var}^\sigma$ and $Q \in \{\forall, \exists\}$.

Notice a slight difference between the definition $\alpha_{\bar{F}}$ just given and that sketched at the beginning of the present section. In the current version, $\alpha_{\bar{F}}$ need not "move" quantified claims where the syncategorematic quantifiers occur unguarded—where a quantifier occurrence is guarded when it is syntactically restricted by some predicate. Thus, e.g., we have $\alpha_F(\exists xYx) = \exists xYx$ rather than $\alpha_F(\exists xYx) = \exists xFx \wedge Yx$, whenever Y is a variable. The quantifier occurrences whose domain can be specified in this model are quantifiers that occur guarded by some predicate containing occurrences of an appropriate existence predicate. Unguarded quantifiers are taken, for modelling purposes, to have a fixed, all inclusive domain that cannot be re-specified.

It is well known that $\mathcal{L}_0$ carries an applicative structure $\boldsymbol{\mathcal{L}_0} := (\mathcal{L}_0, App)$, where $App^{\sigma\tau}(M, N) := MN$. Thus a *domain* over $\boldsymbol{\mathcal{L}_0}$ is essentially a nice term sequence. More exactly, for each nice term sequence $\bar{F}$ there is a domain that assigns F_σ to σ whenever F_σ is defined and is undefined otherwise. In view of this correspondence, I will notate and speak of domains over $\boldsymbol{\mathcal{L}_0}$ as if they were nice term sequences. Of course, nice term sequences that are mutual permutations of one another correspond to the same domain.

We define a typed family of equivalence relations $\sim_\sigma \subseteq \mathcal{L}_0^\sigma \times \mathcal{L}_0^\sigma$ on $\mathcal{L}_0$ by setting

$$M \sim_\sigma N \iff \mathtt{H}_0 \vdash \alpha^\sigma_{\bar{F}}(M)\bar{v} \leftrightarrow \alpha^\sigma_{\bar{F}}(N)\bar{v} \text{ for every nice term sequence } \bar{F}.$$

Intuitively, then, $M \sim N$ means that M and N are provably equivalent, and remain so under arbitrary translations. It is easy to see that $\sim$ is an applicative congruence of our term applicative structure $(\mathcal{L}_0, App)$. We then let $\mathbf{A} := (A, App)$ be the quotient of $\boldsymbol{\mathcal{L}_0}$ through $\sim$.

To obtain a substitution structure, we can lift the mappings $\alpha^\sigma_{\bar{F}}$ to equivalence classes of terms. Write $[M]$ for the equivalence class of M under $\sim$. A domain over $\mathbf{A}$ is then the result of lifting a domain over $\mathcal{L}_0$ to equivalence classes under $\sim$. This justifies notating domains in $\mathbf{A}$ as $[\bar{F}] := ([F_\sigma])_{\sigma \in \bar{\sigma}}$, once more slightly abusing notation.

To each such domain $[\bar{F}]$ we assign a substitution $i_{[\bar{F}]}$ by setting

$$i_{[\bar{F}]}[M] := [\alpha_{\bar{F}}(M)].$$

It is not difficult to verify that $i_{[\bar{F}]}$ and $i_{[\bar{G}]}$ are identical whenever $[\bar{F}] = [\bar{G}]$.

Let I be the set of all such $i_{[\bar{F}]}$s. It is then straightforward to check that $\mathbf{I} := (I, \circ)$ is a monoid, where $\circ$ is function composition. The identity substitution is given by $i_{[\bar{\mathsf{E!}}_\sigma]}$, for any type σ, as $\alpha_{\mathsf{E!}_\sigma}$ fixes every term by construction. Closure under composition is ensured by defining:

$$\bar{G} \bullet \bar{F} := \begin{cases} \alpha_{\bar{G}}(F_\sigma) & \text{if } F_\sigma \text{ is defined} \\ G_\sigma & \text{if } F_\sigma \text{ is not defined but } G_\sigma \text{ is} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Notice the definition just given mirrors that of domain composition given in ($\bullet$). This is no accident, as we can show

$$i_{[\bar{G}]} \circ i_{[\bar{F}]} = i_{[\bar{G} \bullet \bar{F}]} = i_{[\bar{G}] \bullet [\bar{F}]}, \tag{14}$$

where $\bullet$ is defined as in ($\bullet$). The proof is a routine induction on the structure of terms.

Moreover, $(\mathbf{A}, \mathbf{I}, Sub)$ is a substitution structure, where Sub is function application. Indeed, the identity substitution $i_{\bar{\mathsf{E!}}}$ fixes every element of $\mathbf{A}$, the condition on substitution composition is automatic from the choice of $\circ$ as function composition, and the requirement that substitutions commute with application follows because the mappings $\alpha_{\bar{F}}$ do. Finally, $\mathfrak{A} := (\mathbf{A}, \mathbf{I}, Sub, i_{(\cdot)})$ is a QSS. For $i_{(\cdot)}$ is clearly a domain assignment function. That identity unit domains exist has already been shown, as have Identity and Composition.

For simplicity, in this example we started from from a term model of $\mathtt{H}_0$, where $\mathsf{E!}_\sigma$ is provably equivalent to $\top_\sigma$. This is not really necessary. All that is required for this construction to go through is that that there be some term that behaves like a classical quantifier according to the model, though that term need not be connected to E! the way (∀E!) and (FrUI) require (see Section 4).

Example 4.21 (Direct Power Construction)**.** Let $\mathbf{A}$ be any applicative structure. An *infinite domain* over $\mathbf{A}$ is any sequence $(\mathbf{f}_\sigma)_{\sigma\in \mathit{Types}}$ with $\mathbf{f}_\sigma \in A^{\sigma\to t}$. We will show that the direct power of $\mathbf{A}$ over an index set of infinite domains over $\mathbf{A}$ can be turned into a QSS.

Let $\bar{\mathbf{d}}$ be a fixed arbitrary infinite domain over $\mathbf{A}$. An infinite domain $\bar{\mathbf{f}}$ over $\mathbf{A}$ is said to be *accessible* when $\mathbf{d}_\sigma = \mathbf{f}_\sigma$ holds for cofinitely many types σ. Let $\mathbf{B}$ be the direct power of $\mathbf{A}$ over the index set containing all accessible infinite domains over $\mathbf{A}$. Thus, the elements of $\mathbf{B}^\sigma$ are functions from the set of all accessible infinite domains over $\mathbf{A}$ to A^σ. I call an element of some B^σ an *upper entity (property, proposition)*, and an element of some A^σ a *lower entity (property, proposition)*. I extend this terminology to domains: I shall call a domain over $\mathbf{B}$ an *upper domain*, and an (infinite) domain over $\mathbf{A}$ a *lower (infinite) domain*.

In this construction, we should think of infinite lower domains as akin to possible worlds. Then when $\mathbf{b}$ is an upper entity and $\bar{\mathbf{f}}$ is a lower infinite domain, we can think of $\mathbf{b}(\bar{\mathbf{f}})$ as what the entity $\mathbf{b}$ "looks like" from the perspective of the world corresponding to $\bar{\mathbf{f}}$. For brevity, by a *point* I henceforth mean a lower accessible infinite domain.

When $\bar{\mathbf{h}}$ is an upper $\bar{\sigma}$-domain and $\bar{\mathbf{f}}$ is a point, let us define a (full) function $\bar{\mathbf{h}}[\bar{\mathbf{f}}]$ on types by putting

$$\bar{\mathbf{h}}[\bar{\mathbf{f}}](\sigma) := \begin{cases} \mathbf{h}_\sigma(\bar{\mathbf{f}}) & \text{when } \sigma \in \bar{\sigma} \\ \mathbf{f}_\sigma & \text{otherwise.} \end{cases}$$

Since each $\mathbf{h}_\sigma$ is a function from points to elements of $A^{\sigma\to t}$ and $\bar{\mathbf{h}}[\bar{\mathbf{f}}]$ can only disagree with $\bar{\mathbf{f}}$ on finitely many types, we have that $\bar{\mathbf{h}}[\bar{\mathbf{f}}]$ is a point as well.

Thus each upper domain induces a mapping from the set of points to itself. The key idea in the proof is to use this perspective to match each upper domain to a substitution. An element $\mathbf{b} \in \mathbf{B}^\sigma$ is a mapping from points to entities in A^σ. Given an upper domain $\bar{\mathbf{h}}$, applying $i_{\bar{\mathbf{h}}}$ to $\mathbf{b}$ will result in an element that maps the point $\bar{\mathbf{f}}$ to the value of $\mathbf{b}$ on the argument $\bar{\mathbf{h}}[\bar{\mathbf{f}}]$. Formally:

$$(i_{\bar{\mathbf{h}}}\mathbf{b})\bar{\mathbf{f}} := \mathbf{b}(\bar{\mathbf{h}}[\bar{\mathbf{f}}]).$$

Thus, if $\mathbf{b}(\bar{\mathbf{f}})$ is what $\mathbf{b}$ "looks like" from the perspective of the point $\bar{\mathbf{f}}$, then $(i_{\bar{\mathbf{h}}}\mathbf{b})\bar{\mathbf{f}}$ is what $\mathbf{b}$ "looks like" from the perspective of the point $\bar{\mathbf{h}}[\bar{\mathbf{f}}]$.

The set of all mappings $i_{\bar{\mathbf{h}}}$ as above, equipped with function composition $\circ$, is a monoid. Clearly, $\circ$ is associative. The identity element is given by $i_{\bar{\mathbf{e}}}$, where $\bar{\mathbf{e}}$ is any upper $\bar{\sigma}$-domain with $\bar{\mathbf{e}}(\sigma) = \mathbf{e}_\sigma$ and $\mathbf{e}_\sigma$ is the unique mapping in $B^{\sigma\to t}$ such that

$$\mathbf{e}_\sigma(\bar{\mathbf{f}}) = \mathbf{f}_\sigma.$$

That is, each $\mathbf{e}_\sigma$ is the projection function at the σ-th coordinate. It should be clear that each such $\bar{\mathbf{e}}$ determines the constant mapping from points to points, which implies that $i_{\bar{\mathbf{e}}}$ is indeed an identity element relative to $\circ$. To check closure under composition, we first show that whenever $\bar{\mathbf{h}}, \bar{\mathbf{k}}$ are respectively $\bar{\sigma}$ and $\bar{\tau}$ upper domains, for all points $\bar{\mathbf{f}}$ and for all types σ we have

$$(\bar{\mathbf{k}}[\bar{\mathbf{h}}[\bar{\mathbf{f}}]])(\sigma) = (\bar{\mathbf{h}} \bullet \bar{\mathbf{k}}[\bar{\mathbf{f}}])(\sigma), \tag{15}$$

where $\bullet$ is domain composition as defined in ($\bullet$). If $\sigma \notin \bar{\sigma} + \bar{\tau}$, then both sides equal $\bar{\mathbf{f}}_\sigma$. If $\sigma \in \bar{\sigma} + \bar{\tau}$, then depending on whether $\sigma \in \bar{\tau}$ (left) or $\sigma \notin \bar{\tau}$ (right) we have

$$\begin{aligned} (\bar{\mathbf{h}} \bullet \bar{\mathbf{k}}[\bar{\mathbf{f}}])(\sigma) &= i_{\bar{\mathbf{h}}}\mathbf{k}_\sigma(\bar{\mathbf{f}}) \\ &= \mathbf{k}_\sigma[\bar{\mathbf{h}}[\bar{\mathbf{f}}]] \\ &= (\bar{\mathbf{k}}[\bar{\mathbf{h}}[\bar{\mathbf{f}}]])(\sigma) \end{aligned} \qquad \begin{aligned} (\bar{\mathbf{h}} \bullet \bar{\mathbf{k}}[\bar{\mathbf{f}}])(\sigma) &= \mathbf{h}_\sigma(\bar{\mathbf{f}}) \\ &= [\bar{\mathbf{h}}[\bar{\mathbf{f}}]](\sigma) \\ &= (\bar{\mathbf{k}}[\bar{\mathbf{h}}[\bar{\mathbf{f}}]])(\sigma) \end{aligned}$$

Thus, indeed,

$$\begin{aligned} (i_{\bar{\mathbf{h}}}(i_{\bar{\mathbf{k}}}\mathbf{b}))\bar{\mathbf{f}} &= (i_{\bar{\mathbf{k}}}\mathbf{b})(\bar{\mathbf{h}}[\bar{\mathbf{f}}]) \\ &= \mathbf{b}(\bar{\mathbf{k}}[\bar{\mathbf{h}}[\bar{\mathbf{f}}]]) \\ &= \mathbf{b}(\bar{\mathbf{h}} \bullet \bar{\mathbf{k}}[\bar{\mathbf{f}}]) \\ &= (i_{\bar{\mathbf{h}} \bullet \bar{\mathbf{k}}}\mathbf{b})\bar{\mathbf{f}}. \end{aligned}$$

Call the structure just shown to be a monoid $\mathbf{I}$. Then $(\mathbf{I}, \mathbf{B}, \mathit{Sub})$ is a substitution structure, where Sub is function application. Indeed, the condition that $i_{\bar{\mathbf{e}}}\mathbf{b} = \mathbf{b}$ is obvious from the choice of $i_{\bar{\mathbf{e}}}$, and the remaining conditions from the definition of a substitution structure are straightforward to verify.

To conclude, I show that $\mathfrak{B} := (\mathbf{I}, \mathbf{B}, \mathit{Sub}, i_{(\cdot)})$ is in fact a QSS. Obviously, $i_{(\cdot)}$ is a substitution assignment function. We have already shown that an identity unit upper domain $\mathbf{e}_\sigma$ exists for each type σ. Furthermore, we have shown that Composition holds when checking that $\mathbf{I}$ is a monoid. The only remaining condition to verify is Identity. So let $\bar{\mathbf{e}}$ be any identity upper $\bar{\sigma}$-domain and $\bar{\mathbf{h}}$ be any upper $\bar{\tau}$ domain. Let $\bar{\mathbf{f}}$ be any point and σ be any type. In view of (15), it suffices to show that

$$(\bar{\mathbf{e}}[\bar{\mathbf{h}}[\bar{\mathbf{f}}]])(\sigma) = (\bar{\mathbf{h}}[\bar{\mathbf{e}}[\bar{\mathbf{f}}]])(\sigma). \tag{16}$$

If $\sigma \notin \bar{\sigma} + \bar{\tau}$, then both sides equal $\mathbf{f}_\sigma$. Otherwise, depending on whether $\sigma \in \bar{\tau}$ (left) or not (right), we have

$$\begin{aligned} (\bar{\mathbf{e}}[\bar{\mathbf{h}}[\bar{\mathbf{f}}]])(\sigma) &= (\bar{\mathbf{h}}[\bar{\mathbf{f}}])(\sigma) \\ &= \mathbf{h}_\sigma(\bar{\mathbf{f}}) \\ &= \mathbf{h}_\sigma(\bar{\mathbf{e}}[\bar{\mathbf{f}}]) \\ &= \bar{\mathbf{h}}[\bar{\mathbf{e}}[\bar{\mathbf{f}}]](\sigma) \end{aligned} \qquad \begin{aligned} (\bar{\mathbf{e}}[\bar{\mathbf{h}}[\bar{\mathbf{f}}]])(\sigma) &= (\bar{\mathbf{h}}[\bar{\mathbf{f}}])(\sigma) \\ &= \mathbf{f}_\sigma \\ &= \bar{\mathbf{e}}[\bar{\mathbf{f}}](\sigma) \\ &= \bar{\mathbf{h}}[\bar{\mathbf{e}}[\bar{\mathbf{f}}]](\sigma). \end{aligned}$$

Thus, $\mathfrak{B}$ is indeed a QSS.

4.8. **A Limitative Result.** To conclude this section, I prove a limitative result that, as we shall see in Section 5.4, is connected to Stability Collapse Paradox outlined in Problem 2.2.

In substitution structures, substitutions are not identified with any element of the underlying applicative structure. But we can think of various ways in which the monoid of substitutions might be connected with the associated applicative structure. QSS impose a connection of this sort. Domains are determined by finite sequences of properties drawn from the applicative structure, and each substitution is determined by a domain through $i_{(\cdot)}$.

We can also imagine connections along the opposite direction, from substitutions to elements of the applicative structure. A very natural connection of this sort is the *realization* relation. While substitutions are not literally identified with elements of the underlying applicative structure, some substitution structures are rich enough to contain *realizers* of substitutions: higher-order entities whose applicative behavior matches that of substitutions restricted to a particular type.

Definition 4.22 (Realization)**.** Let $\mathfrak{A}$ be a substitution structure and i be a substitution. An element $\mathbf{f} \in A^{\sigma \to \sigma}$ is said to *realize i at type σ*—or that $\mathbf{f}$ *is a realizer of i at type σ*—if the identity

$$i\mathbf{a} = App(\mathbf{f}, \mathbf{a})$$

holds for every $\mathbf{a} \in A^{\sigma}$. We say that i is *realized in* $\mathfrak{A}$ when i has a realizer at every type, and that $\mathfrak{A}$ is *fully realized* when every substitution is realized.

The next two results show that fully realized QSSs are severely impoverished of stable properties. The argument closely parallels that sketched in Problem 2.2—a parallel we will make precise in Section 5.4.

Lemma 4.23. *Let $\mathfrak{A}$ be a fully realized QSS. Then for any substitutions $i_{\bar{\mathbf{f}}}, i_{\bar{\mathbf{g}}}$ and any $\mathbf{a}, \mathbf{b} \in A^{\sigma}$, if $i_{\bar{\mathbf{f}}}\mathbf{a} = i_{\bar{\mathbf{f}}}\mathbf{b}$, then $(i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{g}}})\mathbf{a} = (i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{g}}})\mathbf{b}$.*

Proof. Suppose $i_{\bar{\mathbf{f}}}\mathbf{a} = i_{\bar{\mathbf{g}}}\mathbf{b}$. By full realization, there are $\mathbf{r}_{\bar{\mathbf{f}}}, \mathbf{r}_{\bar{\mathbf{g}}} \in A^{\sigma \to \sigma}$ that, respectively, realize $i_{\bar{\mathbf{f}}}$ and $i_{\bar{\mathbf{g}}}$ at type σ, where $\mathbf{a}, \mathbf{b} \in A^{\sigma}$. Then $i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{g}}}\mathbf{a} = i_{\bar{\mathbf{f}}}App(\mathbf{r}_{\bar{\mathbf{g}}}, \mathbf{a})$, which in turn equals $App(i_{\bar{\mathbf{f}}}\mathbf{r}_{\bar{\mathbf{g}}}, i_{\bar{\mathbf{f}}}\mathbf{a})$ because substitutions distribute over application. Likewise, $i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{g}}}\mathbf{a} = App(i_{\bar{\mathbf{f}}}\mathbf{r}_{\bar{\mathbf{g}}}, i_{\bar{\mathbf{f}}}\mathbf{b})$. But since $i_{\bar{\mathbf{f}}}\mathbf{a} = i_{\bar{\mathbf{g}}}\mathbf{b}$, we must have $i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{g}}}\mathbf{a} = i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{g}}}\mathbf{b}$. ☐

Theorem 4.24. *In any fully realized QSS $\mathfrak{A}$, any $A^{\sigma \to t}$ contains at most one stable property.*

Proof. Let $\mathfrak{A}$ be a fully realized. Suppose $\mathbf{f}, \mathbf{g} \in A^{\sigma \to t}$ are stable. We reason as follows:

(1) $i_{\mathbf{f}}\mathbf{e}_{\sigma} = \mathbf{f}$ by Generality;
(2) $i_{\mathbf{f}}\mathbf{f} = \mathbf{f}$ by stability;
(3) $i_{\mathbf{f}}i_{\mathbf{g}}\mathbf{e}_{\sigma} = i_{\mathbf{f}}i_{\mathbf{g}}\mathbf{f}$ by the above and Lemma 4.23;
(4) $i_{\mathbf{g}}\mathbf{e}_{\sigma} = \mathbf{g}$ by Generality;
(5) $i_{\mathbf{g}}\mathbf{f} = \mathbf{f}$ by stability;
(6) $i_{\mathbf{f}}\mathbf{g} = i_{\mathbf{f}}\mathbf{f}$ by Items 3 to 5;
(7) $\mathbf{g} = \mathbf{f}$ by Item 6 and stability.

☐

5. Models Based on Quantificational Substitution Structures

I now discuss models based on QSSs. These models represent the intended interpretation of domain specification in terms of quantificational substitutions. After defining the notion of a model, I discuss a few properties of models of philosophical interest. I close this section by proving two important results concerning the existence of models.

5.1. **Models.** As usual, we let a *variable assignment* on an applicative structure $\mathbf{A}$ be a function mapping each variable $x \in Var^{\sigma}$ to an element of A^{σ}.

Definition 5.1 (Model)**.** Let $\mathcal{L}_{\heartsuit} \in \{\mathcal{L}_+, \mathcal{L}_{\dagger}, \mathcal{L}_{\star}, \mathcal{L}_0\}$. An $\mathcal{L}_{\heartsuit}$-*model* consists of a QSS $\mathfrak{A}$ together with a stable $\mathcal{L}_{\heartsuit}$-interpretation $[\![\cdot]\!]^{(\cdot)}$ on $\mathfrak{A}$ and an $\mathcal{L}_{\heartsuit}$-satisfaction relation $\models$ on $\mathfrak{A}$.

By a *model* I mean a $\mathcal{L}_{\heartsuit}$-model for some $\mathcal{L}_{\heartsuit} \in \{\mathcal{L}_+, \mathcal{L}_{\dagger}, \mathcal{L}_{\star}, \mathcal{L}_0\}$, without specifying which.

Let me now elaborate on what (stable) interpretations and satisfaction relations are.[27]

Definition 5.2 (Interpretation)**.** Let $\mathcal{L}_{\heartsuit} \in \{\mathcal{L}_+, \mathcal{L}_{\dagger}, \mathcal{L}_{\star}, \mathcal{L}_0\}$ and let $\mathfrak{A} = (\mathbf{I}, \mathbf{A}, i_{(\cdot)})$ be a QSS. An $\mathcal{L}_{\heartsuit}$-*interpretation* on $\mathfrak{A}$ is a function $[\![\cdot]\!]^{(\cdot)}$ that maps every $\mathcal{L}_{\heartsuit}$-term $M \cdot \sigma$ and any variable assignment g on $\mathbf{A}$ to an element of A^{σ}. When $\bar{F}$ is a nice term sequence indexed by $\bar{\sigma}$, write $[\![\bar{F}]\!]^g$ for the sequence $([\![F_{\sigma}]\!]^g)_{\sigma \in \bar{\sigma}}$ We require that $[\![\cdot]\!]^g$ satisfy the following conditions:

1. $[\![x]\!]^g = g(x)$ for every variable $x \in Var^{\sigma}$;
2. $[\![MN]\!]^g = App^{\sigma\tau}([\![M]\!]^g, [\![N]\!]^g)$ whenever $M : \sigma \to \tau$ and $N : \sigma$;
3. $[\![M]\!]^g = [\![N]\!]^h$ whenever M and N are $\beta\eta$-equivalent and g, h agree on $\mathsf{FV}(M) \cap \mathsf{FV}(N)$;
4. $[\![\mathsf{E!}_{\sigma}]\!]^g$ is an identity unit domain;
5. $[\![\mathsf{At}(\bar{F})(M)]\!]^g = i_{[\![\bar{F}]\!]^g}[\![M]\!]^g$.

Here and throughout, $[\![\bar{F}]\!]^g$ denotes the sequence $([\![F_{\sigma}]\!]^g)_{\sigma \in \bar{\sigma}}$, where $\bar{\sigma}$ is the nice type sequence indexing $\bar{F}$. Item 5 is only required when $\mathcal{L}_{\heartsuit} \neq \mathcal{L}_0$.

Up to Item 4, the definition just given is completely standard. Item 4 fixes the intended interpretation of existence predicates, which was anticipated in Section 4.3. Item 5 specifies the role of the map $i_{(\cdot)}$ in determining the interpretation of domain specifiers, which I have sketched earlier. Both conditions are written in conformity to Convention 4.7: while $[\![\mathsf{E!}_{\sigma}]\!]^g$ is not strictly speaking a unit domain, it corresponds uniquely to the domain that maps σ to $[\![\mathsf{E!}_{\sigma}]\!]^g$ and is undefined otherwise. Likewise, in $i_{[\![\bar{F}]\!]^g}[\![M]\!]^g$, by $i_{[\![\bar{F}]\!]^g}$ I denote the unique domain defined only on types that occur in $\bar{\sigma}$ that maps σ to $[\![F_{\sigma}]\!]^g$ whenever the latter is defined, rather than the type-indexed sequence of properties $([\![F_{\sigma}]\!]^g)_{\sigma \in \bar{\sigma}}$. Thus, $[\![\mathsf{At}(\bar{F})(M)]\!]^g$ equals the result of applying the quantificational substitution determined by the domain $[\![\bar{F}]\!]^g$ to $[\![M]\!]^g$. I apply Convention 4.7 in this manner throughout the paper.

In the definition of a model, I work with a more demanding notion of an interpretation, which requires that the interpretations of certain terms we intuitively take to have nothing to do with quantification to be stable.

Definition 5.3 (Stable interpretation)**.** Let $\mathcal{L}_{\heartsuit} \in \{\mathcal{L}_+, \mathcal{L}_{\dagger}, \mathcal{L}_{\star}, \mathcal{L}_0\}$ and let $\mathfrak{A}$ be a QSS. A *stable* $\mathcal{L}_{\heartsuit}$-*interpretation* on $\mathfrak{A}$ is a $\mathcal{L}_{\heartsuit}$-interpretation such that

1. $[\![\wedge]\!]^g$ and $[\![\neg]\!]^g$ are stable;
2. $[\![M]\!]^g$ is stable whenever M is a combinator;
3. $[\![\equiv]\!]^g$ is stable.

[27]Note that Definition 5.2 is not a *recursive* definition of an interpretation function. It simply spells out some constraints that a mapping must satisfy in order to count as an interpretation. The same goes for Definition 5.4. See [Bacon, 2023, Sec. 4.3] for discussion of why a non-recursive approach is desirable.

It is very plausible that Boolean connectives and combinators do not, in the target sense, *involve* quantification.[28] And recall that applicative indiscernibility does not coincide with the obviously quantified relation of Leibniz equivalence. Applicative indiscernibility is only subject to the schema (ALL), which generalizes over properties schematically, not quantificationally.[29] There is thus no obstacle to equipping it with a stable interpretation.

Definition 5.4 (Satisfaction Relation)**.** Let $\mathcal{L}_\heartsuit \in \{\mathcal{L}_+, \mathcal{L}_\dagger, \mathcal{L}_\star, \mathcal{L}_0\}$, let $\mathfrak{A}$ be a QSS and let $[\![\cdot]\!]^{(\cdot)}$ be an $\mathcal{L}_\heartsuit$-interpretation on $\mathfrak{A}$. A $\mathcal{L}_\heartsuit$*-satisfaction relation* is a relation $\models$ between $\mathfrak{A}$ and elements of A^t that satisfies the following conditions, for every variable assignment g and every domain $\bar{\mathbf{f}} \in \mathit{Dom}$:

(1) $\mathfrak{A} \models \mathit{App}(\mathit{App}([\![\wedge]\!]^g, [\![P]\!]^g), [\![Q]\!]^g)$ iff $\mathfrak{A} \models [\![P]\!]^g$ and $\mathfrak{A} \models [\![Q]\!]^g$;
(2) $\mathfrak{A} \models \mathit{App}([\![\neg]\!]^g, [\![P]\!]^g)$ iff $\mathfrak{A} \not\models [\![P]\!]^g$;
(3) $\mathfrak{A} \models \mathit{App}(\mathit{App}([\![\equiv]\!]^g, [\![M]\!]^g), [\![N]\!]^g)$ iff for every $\mathbf{f} \in A^{\sigma\to t}$ we have $\mathfrak{A} \models \mathit{App}(\mathbf{f}, [\![M]\!]^g)$ precisely when $\mathfrak{A} \models \mathit{App}(\mathbf{f}, [\![N]\!]^g)$;
(4) $\mathfrak{A} \models i_{\bar{\mathbf{f}}}[\![\forall x P]\!]^g$ iff $\mathfrak{A} \models i_{\bar{\mathbf{f}}}[\![P]\!]^{g[x\mapsto \mathbf{a}]}$ for every $\mathbf{a} \in A^\sigma$ such that $\mathfrak{A} \models \mathit{App}(i_{\bar{\mathbf{f}}}[\![\mathsf{E!}]\!]^g, i_{\bar{\mathbf{f}}}\mathbf{a})$;
(5) $\mathfrak{A} \models i_{\bar{\mathbf{f}}}[\![\exists x P]\!]^g$ iff $\mathfrak{A} \models i_{\bar{\mathbf{f}}}[\![P]\!]^{g[x\mapsto \mathbf{a}]}$ for some $\mathbf{a} \in A^\sigma$ such that $\mathfrak{A} \models \mathit{App}(i_{\bar{\mathbf{f}}}[\![\mathsf{E!}]\!]^g, i_{\bar{\mathbf{f}}}\mathbf{a})$;
(6) $\mathfrak{A} \models i_{\bar{\mathbf{f}}}[\![\blacksquare P]\!]^g$ iff $\mathfrak{A} \models (i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{g}}})[\![P]\!]^g$ holds for all $\bar{\mathbf{g}} \in \mathit{Dom}$.

Of course, the last condition is not required when $\mathcal{L}_\heartsuit \in \{\mathcal{L}_\star, \mathcal{L}_0\}$.

Some comments are needed. Items 1 and 2 are standard and fix the truth-functional behavior of $[\![\wedge]\!]^g$, $[\![\neg]\!]^g$. Item 3 is motivated by the intended interpretation of the $\equiv_\sigma$ constants as expressing relations of applicative indiscernibility, as explained in Section 3. Thus $[\![M \equiv N]\!]^g$ is true in a model when, according to the model, $[\![M]\!]^g$ and $[\![N]\!]^g$ share *all* their higher-order properties—not just those falling within the interpretation of the relevant existence predicate.

It will be convenient to have some compact notation for this model-theoretic analogue of applicative indiscernibility. Given a model $\mathfrak{A}$, let us write

$$\mathbf{a} \equiv_{\mathfrak{A}} \mathbf{b} :\Longleftrightarrow \mathfrak{A} \models \mathit{App}(\mathbf{f}, \mathbf{a}) \text{ iff } \mathfrak{A} \models \mathit{App}(\mathbf{f}, \mathbf{b}).$$

When $\mathbf{a} \equiv_{\mathfrak{A}} \mathbf{b}$ holds, we say that $\mathbf{a}$ and $\mathbf{b}$ are *applicatively indiscernible in* $\mathfrak{A}$. Thus condition (3) becomes:

(3') $\mathfrak{A} \models \mathit{App}(\mathit{App}([\![\equiv]\!]^g, [\![M]\!]^g), [\![N]\!]^g)$ iff $[\![M]\!]^g \equiv_{\mathfrak{A}} [\![N]\!]^g$.

---

[28]To be sure, there are reasons to deny the stability of Boolean connectives. For example, one might think there is a reading of conjunction on which $P \wedge Q$ is synonymous with *both* P *and* Q *are true*, where *both* is intended to be read as a quantifier. Likewise, there might be a reading of negation as synonymous with *materially implies every proposition.* Now (skipping ahead a bit) modifying the appropriate conditions in the definition of a satisfaction relation to capture these readings is straightforward:

(1') $\mathfrak{A} \models \mathit{App}(\mathit{App}([\![\wedge]\!]^g, [\![P]\!]^g), [\![Q]\!]^g)$ iff $\mathfrak{A} \models [\![\mathsf{E!}P]\!]^g$ and $\mathfrak{A} \models [\![\mathsf{E!}Q]\!]^g$ and $\mathfrak{A} \models [\![P]\!]^g$ and $\mathfrak{A} \models [\![Q]\!]^g$;
(2') $\mathfrak{A} \models \mathit{App}([\![\neg]\!]^g, [\![P]\!]^g)$ iff $\mathfrak{A} \models [\![P]\!]^g$ implies $\mathfrak{A} \models [\![Q]\!]^g$ whenever $\mathfrak{A} \models [\![\mathsf{E!}Q]\!]^g$;

But the resulting propositional logic would not be classical. These readings of Boolean operators seem to fit much better in the context of a *negative* semantics, on which existence is always required for truth. That is not the framework I am working in here, so I will set this picture aside.

[29]For more on the idea that there are non-quantificational ways to generalize, see Russo [Unpublished].

I occasionally suppress the subscript $\mathfrak{A}$ (and the qualifiers 'in $\mathfrak{A}$' or 'in a model') when it can be inferred from context.

Notice how the remaining conditions all involve substitutions: they constrain what it takes to satisfy the interpretation of a formula of a given form *after* a substitution has been applied to it. This is due to syncategorematicity. In the case of, say, (2), we can figure out whether $\mathfrak{A} \models i_{\bar{\mathbf{f}}}(App([\![\neg]\!]^g, [\![P]\!]^g))$ simply by checking whether $\mathfrak{A} \models App(i_{\bar{\mathbf{f}}}[\![\neg]\!]^g, i_{\bar{\mathbf{f}}}[\![P]\!]^g)$ holds. So, (2) suffices to fix the satisfaction conditions for interpretations of negated formulas under all substitutions. But it does not make sense to write something like $[\![\forall x]\!]^g$ or $[\![\blacksquare]\!]^g$, because syncategorematic operators are not assigned independent interpretations, so we need to explicitly build in substitutions in the satisfaction conditions.

Items 4 and 5 constrain, simultaneously, the meanings of quantifiers and existence predicates, as well as the behavior of substitutions. In the special case where $i_{\bar{\mathbf{f}}}$ is the identity substitution Items 4 and 5 yield the more familiar-looking

(4a) $\mathfrak{A} \models [\![\forall x P]\!]^g$ iff $\mathfrak{A} \models [\![P]\!]^{g[x \mapsto \mathbf{a}]}$ for every $\mathbf{a} \in A^\sigma$ such that $\mathfrak{A} \models App([\![\mathsf{E!}]\!]^g, \mathbf{a})$;
(5a) $\mathfrak{A} \models [\![\exists x P]\!]^g$ iff $\mathfrak{A} \models [\![P]\!]^{g[x \mapsto \mathbf{a}]}$ for some $\mathbf{a} \in A^\sigma$ such that $\mathfrak{A} \models App([\![\mathsf{E!}]\!]^g, \mathbf{a})$.

These are just the standard satisfaction clauses for quantifiers in free logic, mirroring the key quantificational axioms of $\mathtt{FH}_\heartsuit$. The idea behind Items 4 and 5 is that these standard clauses should hold true in a model *and* remain true after applying any quantificational substitution—just like (FrUI) should hold true within the scope of arbitrary domain specifiers. Indeed, given Item 5 in the definition of an interpretation, Items 4 and 5 imply the following satisfaction clauses for quantified statements within the scope of domain specifiers (provided x does not occur free in $\bar{F}$):

(4b) $\mathfrak{A} \models [\![\mathsf{At}(\bar{F})(\forall x P)]\!]^g$ iff $\mathfrak{A} \models [\![\mathsf{At}(\bar{F})(P)]\!]^{g[x \mapsto \mathbf{a}]}$ for every $\mathbf{a} \in A^\sigma$ such that $\mathfrak{A} \models [\![\mathsf{At}(\bar{F})(\mathsf{E!}x)]\!]^{g[x \mapsto \mathbf{a}]}$;
(5b) $\mathfrak{A} \models [\![\mathsf{At}(\bar{F})(\exists x P)]\!]^g$ iff $\mathfrak{A} \models [\![\mathsf{At}(\bar{F})(P)]\!]^{g[x \mapsto \mathbf{a}]}$ for some $\mathbf{a} \in A^\sigma$ such that $\mathfrak{A} \models [\![\mathsf{At}(\bar{F})(\mathsf{E!}x)]\!]^{g[x \mapsto \mathbf{a}]}$.

Thus, intuitively, quantifiers within the scope of domain specifiers range over the things which *exist among the* F*s*.

Another way to understand the satisfaction clauses for quantifiers is by thinking of domains as akin to possible worlds. Given any model with underlying QSS $\mathfrak{A}$, stable interpretation $[\![\cdot]\!]^{(\cdot)}$ and satisfaction relation $\models$, it is always possible to construct a new model with the same stable interpretation by taking a domain $\bar{\mathbf{f}}$ and defining a new satisfaction relation

$$\mathfrak{A} \models_{\bar{\mathbf{f}}} [\![P]\!]^g :\Longleftrightarrow \mathfrak{A} \models i_{\bar{\mathbf{f}}}[\![P]\!]^g.$$

I call this construction the *re-pointing* of a model by $i_{\bar{\mathbf{f}}}$. We can notate satisfaction in the re-pointing of a model using notation familiar from Kripke semantics, writing $\mathfrak{A}, \bar{\mathbf{f}} \models [\![P]\!]^g$ to mean $\mathfrak{A} \models_{\bar{\mathbf{f}}} [\![P]\!]^g$. In this notation, Items 4 and 5 are equivalent to the following conditions:

(4c) $\mathfrak{A}, \bar{\mathbf{f}} \models [\![\forall x P]\!]^g$ iff $\mathfrak{A}, \bar{\mathbf{f}} \models [\![P]\!]^{g[x \mapsto \mathbf{a}]}$ for every $\mathbf{a} \in A^\sigma$ such that $\mathfrak{A}, \bar{\mathbf{f}} \models App([\![\mathsf{E!}]\!]^g, \mathbf{a})$;
(5c) $\mathfrak{A}, \bar{\mathbf{f}} \models [\![\exists x P]\!]^g$ iff $\mathfrak{A}, \bar{\mathbf{f}} \models [\![P]\!]^{g[x \mapsto \mathbf{a}]}$ for some $\mathbf{a} \in A^\sigma$ such that $\mathfrak{A}, \bar{\mathbf{f}} \models App([\![\mathsf{E!}]\!]^g, \mathbf{a})$.

These essentially say that the standard satisfaction conditions for free quantifiers must hold at all domains.

Lastly, Item 6 regiments the intended interpretation of ■ as generalizing over domains. Here, too, it is important to recall from Section 2.2 that ■ generalizes schematically, not quantificationally: the key axiom fixing is behavior is (Master), not

$$\mathsf{At}(\bar{F})(P) \to \left(\bigwedge_{\sigma \in \bar{\sigma}} \mathsf{E!}F_\sigma \to \blacklozenge P \right). \tag{17}$$

This is why quantification over domains in the right-hand side of Item 6 is not restricted to domains consisting entirely of *existing* properties.

Given a model $\mathfrak{A}$ and some term $P \cdot t$ in the language of the model, we say that P is *valid* in $\mathfrak{A}$ when $\mathfrak{A} \models [\![P]\!]^g$ holds for every variable assignment g. A term $P \cdot t$ is said to be *valid* in a class of models when it is valid in every model in that class.

5.2. **Regularity and Quasi-Leibnizianness.** I now move on to discussing some model-theoretic properties of philosophical interest. In this subsection, I cover some properties related to the question of what a quantificationalist who theorizes in $\mathcal{L}_\dagger$ should say about the nature of identity.

For any model $\mathfrak{A}$ and $\mathbf{a}, \mathbf{b} \in A^\sigma$, I write

$$\mathbf{a} \cong_{\mathfrak{A}} \mathbf{b} :\iff i_{\bar{\mathbf{f}}}\mathbf{a} \equiv_{\mathfrak{A}} i_{\bar{\mathbf{f}}}\mathbf{b} \text{ for every } \mathbf{f} \in \mathit{Dom}.$$

Again, I suppress the subscript when it is clear from context. I call this relation *substitutional indiscernibility* (in a model). Intuitively, two entities are substitutionally indiscernible (in a model) when they are applicatively indiscernible (in that model) under any substitution.

Substitutional indiscernibility need not coincide with applicative indiscernibility. This is the semantic counterpart of the fact—noted in Section 3.3—that (SLL) does not follow from (WLL) in $\mathtt{H}_\dagger$ (and so in $\mathtt{FH}_\dagger$ either). If a model contains applicatively indiscernible entities $\mathbf{a}, \mathbf{b}$ that are not substitutionally indiscernible, then $\mathbf{a}$ and $\mathbf{b}$ share all their higher-order properties, but they are distinguished by some substitution. Such models, then, are counter-examples to (SLL) but not to (WLL), both formulated in terms of the object language constants $\equiv$.

The claim that applicative indiscernibility and substitutional indiscernibility do not coincide in a model is equivalent to the claim that said model is not *regular*, in the following sense.

Definition 5.5 (Regularity)**.** A model $\mathfrak{A}$ is *regular* when $\equiv$ is a substitutional congruence on the underlying QSS.

Clearly, there cannot be irregular models based on fully realized QSSs. For in such models, each substitution corresponds to a higher-order property whose applicative behavior matches the "substitutional behavior" of the substitution. Thus any two entities distinguished by some substitution can also be distinguished by some higher-order property. However, regularity fails in some models based on applicative structures that are not fully realized. In such models, there can be entities that share all their order properties, but are distinguished by some substitution.

Applicative indiscernibility is the most natural applicative congruence for standard models of higher-order logic, based on applicative structures only. It is guaranteed to be an applicative congruence in any such model. This is part of the reason why it makes sense to reduce identity to applicative indiscernibility in such models. *Any* standard model of higher-order logic can be turned into a *Leibnizian* model—one that identifies all applicatively indiscernible entities—by quotienting the underlying applicative structure under $\equiv$, then defining an interpretation function and a satisfaction relation that commute with the quotient map in the right way.

The existence of irregular models—which I prove later—shows that applicative indiscernibility cannot play the same role for models in our sense. It would be good to have some other relation playing this role. Substitutional indiscernibility is intended to do just that. Indeed, we have the following result. Recall from Section 4.5 that a *congruence* on a QSS is an applicative, substitutional, and quantificational congruence on the relevant underlying structures.

Proposition 5.6. *Substitutional indiscernibility is a congruence on the QSS underlying any $\mathcal{L}_\dagger$-model (and so in any $\mathcal{L}_\star$-model).*

Proof. Let $\mathfrak{A}$ be a $\mathcal{L}_\dagger$-model. Since $\cong$ implies $\equiv$ and substitutions commute with applications, $\cong$ is an applicative congruence on the applicative structure underlying $\mathfrak{A}$. Since substitutions compose, $\cong$ is also a substitutional congruence on the underlying substitution structure. To see that $\cong$ is also a quantificational congruence, let $\bar{\mathbf{f}}, \bar{\mathbf{g}}$ be respectively a $\bar{\sigma}$- and a $\bar{\tau}$-domain such that $\bar{\mathbf{f}} \cong \bar{\mathbf{g}}$. Recall this means that

$$(pad_{\bar{\sigma}}^{\bar{\sigma}+\bar{\tau}}(\bar{\mathbf{f}}))(\rho) \cong (pad_{\bar{\tau}}^{\bar{\sigma}+\bar{\tau}}(\bar{\mathbf{g}}))(\rho) \text{ for all } \rho \in \bar{\sigma} + \bar{\tau}.$$

We now show that $i_{\bar{\mathbf{f}}}\mathbf{a} \cong i_{\bar{\mathbf{g}}}\mathbf{a}$ holds for each $\mathbf{a}$. Let $i_{\bar{\mathbf{h}}}$ be any substitution. By Proposition 4.16 we have $\bar{\mathbf{h}} \bullet \bar{\mathbf{f}} \cong \bar{\mathbf{h}} \bullet \bar{\mathbf{g}}$, which implies $\bar{\mathbf{h}} \bullet \bar{\mathbf{f}} \equiv \bar{\mathbf{h}} \bullet \bar{\mathbf{g}}$. Now consider the term $\lambda\bar{X}.\mathsf{At}(\bar{X})(y)$, where $\bar{X}$ is a nice term sequence of variables indexed by $\bar{\sigma} + \bar{\tau}$. Let g be a variable assignment with $g(y) = \mathbf{a}$. By $\bar{\mathbf{h}} \bullet \bar{\mathbf{f}} \equiv \bar{\mathbf{h}} \bullet \bar{\mathbf{g}}$ we have

$$App([\![\lambda\bar{X}.\mathsf{At}(\bar{X})(y)]\!]^g, \bar{\mathbf{h}} \bullet \bar{\mathbf{f}}) \equiv App([\![\lambda\bar{X}.\mathsf{At}(\bar{X})(y)]\!]^g, \bar{\mathbf{h}} \bullet \bar{\mathbf{g}}).$$

This is equivalent to $(i_{\bar{\mathbf{h}}} \circ i_{\bar{\mathbf{f}}})\mathbf{a} \equiv (i_{\bar{\mathbf{h}}} \circ i_{\bar{\mathbf{g}}})\mathbf{a}$. Since $\bar{\mathbf{h}}$ was arbitrary, this shows $i_{\bar{\mathbf{f}}}\mathbf{a} \cong i_{\bar{\mathbf{g}}}\mathbf{a}$, as desired. □

Thus, in our models, substitutional indiscernibility plays a role similar to that played by applicative indiscernibility in standard models of higher-order logic. In particular, just like every standard model of higher-order logic can be turned into a Leibnizian model satisfying the same formulas, every model in our sense can be turned into a *quasi-Leibnizian* model satisfying the same formulas.

Definition 5.7 (Quasi-Leibnizianness)**.** A model is *quasi-Leibnizian* when substitutionally indiscernible entities in the model are identical.

Let the *quotient* of a model $\mathfrak{A}$ under a congruence $\sim$ be the unique model $\mathfrak{A}_{/\sim}$, if it exists, obtained by taking the quotient of the underlying QSS of $\mathfrak{A}$ under $\sim$, and defining the interpretation function and satisfaction relation as follows:

$$[\![M]\!]_{/\sim}^{[g]} := [[\![M]\!]^g] \qquad\qquad \mathfrak{A}_{/\sim} \models [\![M]\!]_{/\sim}^{[g]} :\iff \mathfrak{A} \models [\![M]\!]^g.$$

Clearly, $\mathfrak{A}_{/\sim}$ satisfies the same formulas as $\mathfrak{A}$ whenever it exists, for the satisfaction relation on $\mathfrak{A}_{/\sim}$ is well defined precisely when $\sim$ never relates a truth with a falsehood.

Proposition 5.8. *Every model has a quotient under $\cong$.*

Proof. In any such model, $\cong$ is a congruence. Since $\cong$ implies $\equiv$, it follows that $\cong$ never relates a truth with a falsehood. □

This makes $\cong$ an extremely natural relation to serve as the interpretation of identity in models. It is this fact which ensures the quantificationalist who theorizes in $\mathcal{L}_\dagger$ can prove that an account of identity as *applicative indiscernibility at all domains of quantification* is consistent. In other words, *identity is applicative indiscernibility under all quantificational substitutions.*

5.3. **Stabilization and Quantifier Generality.** Other interesting properties of models are formulated in terms of the notion of stability. We might be interested in models where entities of a particular kind are stable. For example:

Definition 5.9 (Individual stability)**.** A model $\mathfrak{A}$ is *individual stable* when $\mathbf{a}$ is stable whenever $\mathbf{a} \in A^e$.

Individual stability captures the idea that individuals have nothing to do with quantification. This seems intuitive on a structured picture of reality in which individuals do not have properties and relations as constituents, and in which to involve quantification is to have a constituent corresponding to the interpretation of an existence predicate. But it is worth noting there are seemingly coherent philosophical pictures in which one would expect individuals not to be stable. For example, one might hold that for the number of planets to be 8 *just is* for there to be (exactly) 8 planet, where the right-hand side is formalized in the usual purely logical fashion using quantifiers and identities. On this view, 'the number of planets is 8' would express a non-stable proposition. This might be explained by insisting that 8 is a non-stable individual. Perhaps this is because the nature of numbers is completely determined by an appropriate abstraction principle that involves quantification, so that numbers involve quantification as well. Alternatively, one might propose that the definite description 'the number of planets' expresses a non-stable individual that, relative to a given domain, is applicatively indiscernible from the cardinality of planets in that domain.

A second class of stability-theoretic conditions consists of *comprehension* conditions for the class of stable entities. As I argue in other work, some applications of Quantificationalism—most importantly the notion of metaphysical predicativity—rely on the existence of *enough* stable properties. Comprehension principles for stability are a natural way of ensuring just that.

Let $\mathfrak{A}$ be a model and $\mathbf{a} \in \mathbf{A}^\sigma$. A *stabilizer* for $\mathbf{a}$ is a stable entity $\mathbf{s}$ such that $\mathbf{a} \equiv \mathbf{s}$. The notion of a *$\bar{\sigma}$-stabilizer* is defined the same way, but substituting '$\bar{\sigma}$-stable' for 'stable.' We can lift the notion of a stabilizer to domains. A *stabilizer* for a $\bar{\sigma}$-domain $\bar{\mathbf{f}}$ is a $\bar{\sigma}$-domain $\bar{\mathbf{g}}$ such that $\mathbf{g}_\sigma$ stabilizes $\mathbf{f}_\sigma$ whenever $\mathbf{f}_\sigma$ is defined. Likewise for the notion of a $\bar{\sigma}$-stabilizer.

Definition 5.10 (Stabilization)**.** A model $\mathfrak{A}$ is called *stabilized* (resp. *$\bar{\sigma}$-stabilized*) when every entity has a stabilizer (resp. a $\bar{\sigma}$-stabilizer)

All these notions admit a *weak* variant (*weak stabilizer, weakly stabilized,* etc.), formulated by replacing $\equiv$ with coextensiveness in the corresponding definition.

Clearly, stabilization does not make sense for regular, Leibnizian models. For a stabilized, regular, and Leibnizian model would just be a model where every entity is stable, and there are no such models. But stabilization can be had by irregular, quasi-Leibnizian models in which substitutions have non-trivial behavior.

Note that if a domain $\bar{\mathbf{f}}$ has a $\bar{\sigma}$-stabilizer $\bar{\mathbf{s}}$, then $i_{\bar{\mathbf{f}}}$ and $i_{\bar{\mathbf{s}}}$ have the same "substitutional behavior" up to applicative indiscernibility.

Proposition 5.11. *Let $\mathfrak{A}$ be a model and $\bar{\mathbf{f}}$ a $\bar{\sigma}$-domain. If $\bar{\mathbf{f}}$ has a stabilizer $\bar{\mathbf{s}}$, then $i_{\bar{\mathbf{f}}}\mathbf{a} \equiv i_{\bar{\mathbf{s}}}\mathbf{a}$ for every $\mathbf{a} \in A^\sigma$.*

Proof. Since each $\mathbf{f}_\sigma$ is applicatively indiscernible from $\mathbf{s}_\sigma$, we must have

$$\mathfrak{A} \models App([\![\lambda\bar{X}.\mathsf{At}(\bar{X})(y)]\!]^{g[y\mapsto\mathbf{a}]}, \bar{\mathbf{f}}) \equiv App([\![\lambda\bar{X}.\mathsf{At}(\bar{X})(y)]\!]^{g[y\mapsto\mathbf{a}]}, \bar{\mathbf{s}}),$$

where $\bar{X}$ is a nice term sequence indexed by $\bar{\sigma}$. This is to say that $i_{\bar{\mathbf{f}}}\mathbf{a} \equiv i_{\bar{\mathbf{s}}}\mathbf{a}$. □

We thus have a handy method for constructing $\bar{\sigma}$-stabilizers in models where every *identity $\bar{\sigma}$-domain* has a $\bar{\sigma}$-stabilizer, for all $\bar{\sigma}$. To construct a $\bar{\sigma}$-stabilizer for an entity $\mathbf{a}$, we simply take a $\bar{\sigma}$-stabilizer of the identity $\bar{\sigma}$-domain and apply the corresponding substitution to $\mathbf{a}$. By proposition 4.19, the result must be $\bar{\sigma}$-stable.

Proposition 5.12. *Let $\mathfrak{A}$ be a model where every domain has a $\bar{\sigma}$-stabilizer, for each $\bar{\sigma}$. Then every entity has a $\bar{\sigma}$-stabilizer.*

Proof. Take any $\mathbf{a} \in A^\sigma$. Let $\bar{\mathbf{s}}$ be a $\bar{\sigma}$-stabilizer for the identity $\bar{\sigma}$-domain $\bar{\mathbf{e}}$. By Proposition 4.19, $i_{\bar{\mathbf{s}}}\mathbf{a}$ is stable. Furthermore, by Proposition 5.11, $i_{\bar{\mathbf{e}}}\mathbf{a} \equiv i_{\bar{\mathbf{s}}}\mathbf{a}$ and the left-hand side is just $\mathbf{a}$. □

Stabilization is connected with a property concerning the interpretation of *categorematic* quantifiers in models. Recall their definition:

$$\hat{\exists} := \lambda X.\exists y Xy \qquad\qquad \hat{\forall} := \lambda X.\forall y Xy.$$

We might expect models to uniquely fix how the interpretations of categorematic quantifiers are moved by substitutions. While I have not confirmed this, I conjecture that they do not: the current definition of models is consistent with more than one way in which the interpretations of categorematic quantifiers behaves under substitutions.

Thus, it makes sense to consider properties of models that fix one such behavior. The most natural choice seems to be the following:

Definition 5.13 (Quantifier Generality)**.** A model $\mathfrak{A}$ is *quantifier general* when the following conditions obtain:

(1) $\mathfrak{A} \models App(i_{\bar{\mathbf{f}}}[\![\hat{\forall}]\!]^g, \mathbf{g})$ iff $\mathfrak{A} \models App(\mathbf{g}, \mathbf{a})$ holds for every $\mathbf{a} \in A^\sigma$ with $\mathfrak{A} \models App(\mathbf{f}_\sigma, \mathbf{a})$;
(2) $\mathfrak{A} \models App(i_{\bar{\mathbf{f}}}[\![\hat{\exists}]\!]^g, \mathbf{g})$ iff $\mathfrak{A} \models App(\mathbf{g}, \mathbf{a})$ holds for some $\mathbf{a} \in A^\sigma$ with $\mathfrak{A} \models App(\mathbf{f}_\sigma, \mathbf{a})$;

Quantifier generality guarantees that $i_{\bar{\mathbf{f}}}[\![\hat{\forall}]\!]^g$ is coextensive with the interpretation of the restriction of a *classical* quantifier quantifier by $\mathbf{f}_\sigma$, if the model contains such a classical quantifier. A stronger version of quantifier generality could require that $i_{\bar{\mathbf{f}}}[\![\hat{\forall}]\!]^g$ be *identical* with the restriction of a classical quantifier by $\mathbf{f}_\sigma$, again assuming that a classical quantifier exists. As it turns out, this assumption is automatically satisfied: every quantifier general model contains a classical quantifier, definable as $i_{[\![\bar{\top}]\!]^g}[\![\hat{\forall}_\sigma]\!]^g$.

It is important to understand exactly how quantifier generality differs from the satisfaction condition for quantifiers stated in Definition 5.4. While the latter is formulated in terms of syncategorematic quantifiers, it implies the following clause for the categorematic quantifiers:

(1′) $\mathfrak{A} \models App(i_{\bar{\mathbf{f}}}[\![\hat{\forall}]\!]^g, i_{\bar{\mathbf{f}}}(\mathbf{g}))$ iff $\mathfrak{A} \models App(i_{\bar{\mathbf{f}}}(\mathbf{g}), i_{\bar{\mathbf{f}}}(\mathbf{a}))$ holds for every $\mathbf{a} \in A^\sigma$ with $\mathfrak{A} \models App(\mathbf{f}, i_{\bar{\mathbf{f}}}(\mathbf{a}))$;

Unlike in quantifier generality, here the substitution $i_{\bar{\mathbf{f}}}$ targets both $\mathbf{g}$ and $\mathbf{a}$, in addition to the quantifier.

This difference between (1′) and quantifier generality, however, collapses in stabilized models.

Proposition 5.14. *Every stabilized model is quantifier general.*

Proof. Let $\mathfrak{A}$ be stabilized. I show only the clause for the universal quantifier. Given any $\mathbf{a} \in A^\sigma$, let $\mathbf{s_a}$ be an arbitrarily chosen stabilizer of $\mathbf{a}$. Suppose $\mathfrak{A} \not\models App(i_{\bar{\mathbf{f}}}[\![\hat{\forall}]\!]^g, \mathbf{g})$. Then $\mathfrak{A} \not\models i_{\bar{\mathbf{f}}}App([\![\hat{\forall}]\!]^g, \mathbf{s_g})$. By generality and the satisfaction condition on quantifiers, there must be some $\mathbf{a} \in A^\sigma$ with $\mathfrak{A} \models App(\mathbf{f}, i_{\bar{\mathbf{f}}}\mathbf{a})$, such that $\mathfrak{A} \not\models App(\mathbf{s_g}, i_{\bar{\mathbf{f}}}\mathbf{a})$. Since applicatively indiscernible entities are also coextensive, it follows that $\mathfrak{A} \not\models App(\mathbf{g}, i_{\bar{\mathbf{f}}}\mathbf{a})$. Thus $i_{\bar{\mathbf{f}}}\mathbf{a}$ is the desired witness mentioned in the definition of quantifier generality.

Conversely, suppose there is some $\mathbf{a} \in A^\sigma$ such that $\mathfrak{A} \models App(\mathbf{f}, \mathbf{a})$ and $\mathfrak{A} \not\models App(\mathbf{g}, \mathbf{a})$. Then $\mathfrak{A} \models App(\mathbf{f}, i_{\bar{\mathbf{f}}}\mathbf{s_a})$ and $\mathfrak{A} \not\models App(i_{\bar{\mathbf{f}}}\mathbf{s_g}, i_{\bar{\mathbf{f}}}\mathbf{s_a})$ both hold by stabilization. By generality and the satisfaction condition on quantifiers, this implies $\mathfrak{A} \not\models i_{\bar{\mathbf{f}}}App([\![\hat{\forall}]\!]^g, \mathbf{s_g})$. By stabilization again, this is equivalent to $\mathfrak{A} \not\models App(i_{\bar{\mathbf{f}}}[\![\hat{\forall}]\!]^g, \mathbf{g})$. □

5.4. **Model Existence Results.** I introduced models to be used as tools for developing object language theories for quantificationalism, so the question of whether there are any models with interesting properties is of prime importance.

I present two results concerning model existence: the first negative and the second positive. The first is a model-theoretic counterpart to Problem 2.2: it shows that there are no $\mathcal{L}_+$-models whatsoever. The positive result shows that an interesting class of $\mathcal{L}_\dagger$-models, on the other hand, is non-empty. Together, I take it, these results motivate theorizing in $\mathcal{L}_\dagger$.

The reason why there are no $\mathcal{L}_+$ models is that any $\mathcal{L}_+$-model must be based on a fully realized QSS, in the sense of Definition 4.22, and we have already proved that any such QSS contains at most one stable property. However, any model contains more than one stable property at each type.

Theorem 5.15. *There is no $\mathcal{L}_+$-model.*

Proof. Suppose otherwise that $\mathfrak{A}$ is a $\mathcal{L}_+$-model. Take any $\bar{\sigma}$-domain $\bar{\mathbf{f}}$ and for each type σ consider the term

$$\lambda y^\sigma.\mathsf{At}(\bar{X})(y),$$

where $\bar{X}$ is a nice term sequence indexed by $\bar{\sigma}$. Letting g be any variable assignment with $g(\bar{X}) = \bar{\mathbf{f}}$, it is straightforward to verify that, because $\beta\eta$-equivalent terms have the same interpretations, $[\![\lambda y^\sigma.\mathsf{At}(\bar{X})(y)]\!]^g$ realizes $i_{\bar{\mathbf{f}}}$ at type σ.

Generalizing, the QSS underlying $\mathfrak{A}$ is fully realized, so by Theorem 4.24 it contains at most one stable property at each type. But since $[\![\cdot]\!]^{(\cdot)}$ is a stable interpretation, this cannot be: for example, both $[\![\bot_\sigma]\!]^g$ and $[\![\top_\sigma]\!]^g$ are stable and obviously distinct. □

The next result, on the other hand, shows that there are $\mathcal{L}_\dagger$-models. In fact, there is a very general construction that can be used to establish the existence of $\mathcal{L}_\dagger$-models with several philosophically interesting properties. It is an extension of the direct power construction spelled out in Example 4.21.

Theorem 5.16. *There are stabilized, functional, quasi-Leibnizian $\mathcal{L}_\dagger$-models.*

Proof. We begin by taking a full and functional applicative structure $\mathbf{A}$, where $\mathbf{A}^t$ carries a complete atomic Boolean algebra. We choose $\mathbf{A}$ to be full to ensure we can interpret $\equiv$, atomic to easily define a satisfaction relation, and complete to ensure we can interpret quantifiers and $\blacksquare$ as infinite conjunctions. Since it is full, $\mathbf{A}$ also has combinators.[30]

Using the same method described in Example 4.21, we construct a QSS $\mathbf{B}$ by taking a direct power of $\mathbf{A}$ over an index set of infinite domains over $\mathbf{A}$. Using the same terminology as in Example 4.21, I call entities in $\mathbf{A}$ *lower entities* and entities in $\mathbf{B}$ *upper entities*, and extend this terminology to domains. Recall that the relevant index set was chosen as follows: given a fixed arbitrary lower infinite domain $\bar{\mathbf{d}}$, we call an infinite lower domain $\bar{\mathbf{f}}$ *accessible* when it disagrees with $\bar{\mathbf{d}}$ on only finitely many types. Our index set is the set of all accessible lower infinite domains, or the *points* for short.

Recall that each projection mapping $\mathbf{e}_\sigma$ from points to $\mathbf{A}^{\sigma\to t}$ with $\bar{\mathbf{f}} \mapsto \mathbf{f}_\sigma$ corresponds to an identity upper domain, as does any upper domain $\bar{\mathbf{e}}$ consisting entirely of projection mappings. Furthermore, observe that $\mathbf{B}$ is functional and has combinators, for both of these properties are preserved by direct powers. For the same reason, $\mathbf{B}^t$ carries a complete Boolean algebra.

We now need to construct a model over $\mathfrak{B}$. To define an interpretation, the strategy will be to assign interpretations to constants first, then show that this assignment can be recursively extended to a full interpretation—thanks to functionality. We let $[\![\neg]\!]^g$ and $[\![\wedge]\!]^g$ be the constant functions mapping every point to the Boolean complement and meet operations of $\mathbf{A}^t$ respectively. For existence predicates, we put $[\![\mathsf{E!}_\sigma]\!]^g = \mathbf{e}_\sigma$. To define $[\![\equiv_\sigma]\!]^g$ we proceed as follows. Let U be any principal ultrafilter on $\mathbf{A}^t$ (which must exist by atomicity), and for any $\mathbf{p} \in A^t$ write $\mathbf{A} \Vdash \mathbf{p}$ iff $\mathbf{p} \in U$. For any $\mathbf{a}, \mathbf{a}' \in A^\sigma$, write $\mathbf{a} \rightleftharpoons \mathbf{a}'$ to mean that, for any $\mathbf{f} \in A^{\sigma\to t}$, we have that $\mathbf{A} \Vdash App(\mathbf{f}, \mathbf{a})$ iff $\mathbf{A} \Vdash App(\mathbf{f}, \mathbf{a}')$. We may assume, wlog, that $\mathbf{a} \rightleftharpoons \mathbf{a}'$ implies $\mathbf{a} = \mathbf{a}'$.[31] We then choose one element $\mathbf{r} \in A^{\sigma\to\sigma\to t}$ such that $\mathbf{A} \Vdash App(App(\mathbf{r}, \mathbf{a}), \mathbf{a}')$ implies $\mathbf{a} \rightleftharpoons \mathbf{a}'$, and let $[\![\equiv_\sigma]\!]^g$ be the constant function mapping every point to $\mathbf{r}$. At least one such $\mathbf{r}$ must exist by the fullness of $\mathbf{A}$.

We extend $[\![\cdot]\!]^g$ to a full stable interpretation by recursion on the structure of terms. For applications, we put $[\![MN]\!]^g = App([\![M]\!]^g, [\![N]\!]^g)$. For λ-abstracts, we let $[\![\lambda x.M]\!]^g$ be the unique element, if it exists, realizing the applicative behavior $\mathbf{b} \mapsto [\![M]\!]^{g[x\mapsto\mathbf{b}]}$. Both these conditions are standard. For At terms, we put

$$[\![\mathsf{At}(\bar{H})(M)]\!]^g := i_{[\![\bar{H}]\!]^g}[\![M]\!]^g.$$

[30]The assumption of fullness is stronger than necessary. We could instead just assume that $\mathbf{A}$ is rich enough to support a model of $\mathtt{H}_0$, in the standard sense.

[31]If this were not true, we could have quotiented $\mathbf{A}$ by $\rightleftharpoons$ at the beginning of our construction

For quantifiers, we put

$$[\![\forall xP]\!]^g(\bar{\mathbf{f}}) := \bigwedge\{[\![P]\!]^{g[x\mapsto\mathbf{b}]}(\bar{\mathbf{f}}) : \mathbf{b} \in \mathbf{B}^\sigma \text{ with } \mathbf{A} \models \mathit{App}(\mathbf{f}_\sigma, \mathbf{b}(\bar{\mathbf{f}}))\},$$
$$[\![\exists xP]\!]^g(\bar{\mathbf{f}}) := \bigvee\{[\![P]\!]^{g[x\mapsto\mathbf{b}]}(\bar{\mathbf{f}}) : \mathbf{b} \in \mathbf{B}^\sigma \text{ with } \mathbf{A} \models \mathit{App}(\mathbf{f}_\sigma, \mathbf{b}(\bar{\mathbf{f}}))\}.$$

Both are well defined by the completeness of A^t. Finally, we set

$$[\![\blacksquare P]\!]^g(\bar{\mathbf{f}}) := \bigwedge\{[\![P]\!]^g(\bar{\mathbf{h}}(\bar{\mathbf{f}})) : \bar{\mathbf{h}} \text{ an upper domain}\}.$$

This yields a full interpretation if we can show that interpretations of λ-abstracts always exist. Because we are working in a non-standard setting involving syncategorematic expressions, we cannot simply infer this from functionality and the existence of combinators. Still, it can be done: it is a consequence of the following claim. For any $\mathbf{a} \in \mathbf{A}^\sigma$, let $\mathbf{c_a}$ be the constant mapping on points $\bar{\mathbf{f}} \mapsto \mathbf{a}$.

Claim. *Let $\bar{x} \subseteq \mathsf{AV}(M)$. Then for each point $\bar{\mathbf{f}}$, we have*

$$[\![M]\!]^{g[\bar{x}\mapsto\bar{\mathbf{b}}]}(\bar{\mathbf{f}}) = [\![M]\!]^{g[\bar{x}\mapsto\mathbf{c}_{\bar{\mathbf{b}}(\bar{\mathbf{f}})}]}(\bar{\mathbf{f}}).$$

The proof is a tedious but straightforward induction on the structure of terms. I should point out that here the restrictions on abstraction imposed on $\mathcal{L}_\dagger$-terms play a crucial role: the above claim would be false had we attempted to define $[\![\cdot]\!]^g$ the same way on $\mathcal{L}_+$-terms.

In fact, $[\![\cdot]\!]^g$ is a stable interpretation. We have effectively stipulated that $[\![\neg]\!]^g$, $[\![\wedge]\!]^g$ and $[\![\equiv_\sigma]\!]^g$ are always stable: every constant element of $\mathbf{B}$ is stable in $\mathbf{B}$. Furthermore, $[\![M]\!]^g$ is also a constant mapping over whenever M is a combinator. For a closed term can only express a non-stable entity if it contains occurrences of quantifiers, existence predicates, At or $\blacksquare$; combinators do not contain occurrences of any such expressions.[32]

Our last task is to equip $\mathfrak{B}$ with a satisfaction relation. Here, we set

$$\mathfrak{B} \models [\![P]\!]^g :\iff \mathbf{A} \Vdash [\![P]\!]^g(\bar{\mathbf{d}}),$$

where $\bar{\mathbf{d}}$ is the distinguished point chosen at the beginning of the construction, when defining the notion of accessibility. This is indeed a satisfaction relation given how we defined $[\![\cdot]\!]^g$. So, $\mathfrak{B}$ is a model.

Finally, let us check that our model $\mathfrak{B}$ belongs to the desired class. We have already noted that $\mathfrak{B}$ is functional. Moreover, $\mathfrak{B}$ is stabilized. For note that for each $\mathbf{b} \in B^\sigma$, a stabilizer is given by the constant $\mathbf{c}_{\mathbf{b}(\bar{\mathbf{d}})}$, which maps every point to $\mathbf{b}(\bar{\mathbf{d}})$. As already observed, constant mappings of this sort are not moved by any substitution, so $\mathbf{c}_{\mathbf{b}(\bar{\mathbf{d}})}$ is stable. Now, take any $\mathbf{h} \in B^{\sigma\to t}$. If $\mathfrak{B} \models \mathit{App}(\mathbf{h}, \mathbf{b})$, this is to say $\mathbf{A} \Vdash \mathit{App}(\mathbf{h}(\bar{\mathbf{d}}), \mathbf{b}(\bar{\mathbf{d}}))$. But $\mathbf{b}(\bar{\mathbf{d}}) = \mathbf{c}_{\mathbf{b}(\bar{\mathbf{d}})}(\bar{\mathbf{d}})$, so this is equivalent to $\mathbf{A} \Vdash \mathit{App}(\mathbf{h}(\bar{\mathbf{d}}), \mathbf{c}_{\mathbf{b}(\bar{\mathbf{d}})}(\bar{\mathbf{d}}))$. And that, in turn, is equivalent to $\mathfrak{B} \models \mathit{App}(\mathbf{h}, \mathbf{c}_{\mathbf{b}(\bar{\mathbf{d}})})$. Thus, $\mathfrak{B}$ is indeed stabilized.

Finally, $\mathfrak{B}$ is also quasi-Leibnizian. For $\mathbf{b} \cong \mathbf{b}'$ implies that $\mathbf{b}(\bar{\mathbf{f}}) = \mathbf{b}(\bar{\mathbf{f}})$ for every point $\bar{\mathbf{f}}$, since every point corresponds to an upper domain consisting entirely of constant functions from points to lower properties. But then $\mathbf{b}$ and $\mathbf{b}'$ are the same function from points to lower entities, and so $\mathbf{b} = \mathbf{b}'$. □

[32]When doing this construction in signatures richer than Λ, we might want to interpret additional non-logical constants as expressing non-stable entities. Even so, combinators will still have stable interpretations, since they cannot contain occurrences of constants.

Note that the resulting model must be irregular. For the model evidently contains non-stable entities (for example, any $\mathbf{e}_\sigma$), yet it is stabilized. Thus, there are non-stable entities that are applicatively indiscernible but distinct from stable entities. A non-stable entity and its stabilizer will be moved differently by some substitution, hence the model is irregular.

The construction used in the the proof just given can be generalized to show the non-emptiness of other interesting classes of models. For example, individual stability can be ensured by restricting the product so that only constant functions over points are included in B^e. Further, while the construction was carried out in the logical signature Λ, it is clear that nothing prevents us to generalize it to richer signatures containing additional non-logical constants.

6. Object Language Theories

I have developed a model-theoretic framework in which our higher-order language for domain specifiers and the operator $\blacksquare$ can be interpreted. In this section, I apply these model-theoretic results to formulate object language theories of these expressions. I present a logic, $\mathtt{Q}_\star$, which is sound and complete for validity over $\mathcal{L}_\star$-models. I then expand this logic to an $\mathcal{L}_\dagger$-theory $\mathtt{Q}_\dagger$, which is sound and complete for validity over $\mathcal{L}_\dagger$-models. Finally, I discuss various object-language principles tracking some of the model-theoretic properties discussed in the previous section.

6.1. **Domain Specifiers.** The system $\mathtt{Q}_\star$ is obtained by extending our background free logic $\mathtt{FH}_\star$ with the axioms and rules in Figure 3. Recall the operative notion of axiomatization: together, Figures 2 and 3 define the least relation $\vdash$ satisfying all the listed condition, and $\mathtt{Q}_\star$ is the least set of formulas containing P whenever $\vdash P$. $\mathtt{Q}_\star$ is put forward as an axiomatization of the theory of all $\mathcal{L}_\star$-models.

$\vdash \mathsf{At}(\bar{\mathsf{E}}!)(M) \equiv M$		(Id)
$\vdash \mathsf{At}(\bar{F})(MN) \equiv (\mathsf{At}(\bar{F})(M))(\mathsf{At}(\bar{F})(N))$		(App)
$\vdash \mathsf{At}(\bar{F})(\mathsf{At}(\bar{G})(M)) \equiv \mathsf{At}(\bar{F} \bullet \bar{G})(M)$		(CompG)
$\vdash \mathsf{At}(\bar{F})(M) \equiv \mathsf{At}(\bar{G})(M)$	$\bar{F}, \bar{G}$ permutations	(Perm)
$\vdash \mathsf{At}(\bar{F})(\heartsuit) \equiv \heartsuit$	$\heartsuit \in \{\wedge, \neg, \equiv\}$	(Stab$\heartsuit$)
$\vdash \mathsf{At}(\bar{F})(M) \equiv M$	M a combinator	(Stabλ)
$\vdash \mathsf{At}(\bar{F})(\mathsf{E}!_\sigma) \equiv F_\sigma$	when F_σ defined	(Gen)
$\vdash \mathsf{At}(\bar{F})(\mathsf{E}!_\sigma) \equiv \mathsf{E}!_\sigma$	when F_σ undefined	(Mod)
If $\vdash P$, then $\vdash \mathsf{At}(\bar{F})(P)$		(At-Nec)
If $\Gamma \vdash \mathsf{At}(\bar{F})(Q)$, then $\Gamma \vdash \mathsf{At}(\bar{F})(\forall x Q)$	$x \notin \mathsf{FV}(\Gamma, \bar{F})$	(At-UG)

FIGURE 3. The logic $\mathtt{Q}_\star$

Let me comment on the axioms and rules of $\mathtt{Q}_\star$ in turn. The first three axiom schemas pertain to the substitutional interpretation of At terms. They correspond to the existence of an identity substitution (Id), the condition that substitutions commute with application (App), and that the condition that substitutions be closed under composition (CompG). But they do more than just that. In particular, (Id) also characterizes the identity substitution as the substitution induced by any domain consisting entirely of interpretations of existence predicates. Likewise, (CompG) spells out how to compute the composition of two substitutions. As the notation suggests, $\bar{F} \bullet \bar{G}$ is defined in a way that mirrors the model-theoretic notion of domain composition.

$$(\bar{F} \bullet \bar{G})_\sigma := \begin{cases} \mathsf{At}(\bar{F})(G_\sigma) & \text{if } G_\sigma \text{ is defined} \\ F_\sigma & \text{if } G_\sigma \text{ is undefined but } F_\sigma \text{ is} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Thus, (CompG) corresponds to the Composition condition in the definition of a QSS.

(Perm) is just an auxiliary axiom guaranteeing that the order in which the arguments of a nice term sequence appear does not matter. (Stab♡) and (Stabλ) should be self-explanatory, and correspond to the three requirements imposed in the definition of a stable interpretation.

(Gen) and (Mod) correspond, respectively, to the conditions Generality and Modularity mentioned in Section 4.3. Recall that these were shown to be jointly equivalent to the Identity condition in the definition of a QSS.

The remaining two schemas are inference rules. (At-Nec) is a necessitation rule for domain specifiers. It ensures that logical truths remain true under every possible way of specifying the domain of quantification. The last rule, (At-UG), is a strengthening of (UG) required to prove completeness. We could avoid adding it if we could derive, in its absence, a version of the Barcan formula for domain specifiers:

$$\forall x \mathsf{At}(\bar{F})(P) \to \mathsf{At}(\bar{F})(\forall x P) \qquad x \text{ not free in } \bar{F}. \qquad (\mathsf{At}\text{-Barcan})$$

But clearly, (At-Barcan) should *not* be derivable on the intended interpretation on domain specifiers, on which the latter, so to speak, are capable of *expanding* the domain of quantification. And it is readily verified that (At-Barcan) is not valid in the class of all $\mathcal{L}_\star$-models. In the absence of (At-Barcan), (At-UG) is strictly stronger than (UG).

The consistency of $\mathtt{Q}_\star$ follows from its soundness over validity over all $\mathcal{L}_\star$-models and the model existence result established in Section 5.4. In fact, we can prove that $\mathtt{Q}_\star$ is precisely the theory of validity over all $\mathcal{L}_\star$-models.

Theorem 6.1. *$\mathtt{Q}_\star$ is sound and complete with respect to validity over all $\mathcal{L}_\star$-models. Consequently, it is consistent.*

The proof of Theorem 6.1 is by a variation of standard canonical model constructions. I leave it for Appendix A.

6.2. **Adding ■.** I now explore expansions of $\mathtt{Q}_\star$ over $\mathcal{L}_\dagger$-terms. The base system of this sort is $\mathtt{Q}_\dagger$. It is axiomatized by all the axioms and rules from **??** and fig. 3 plus those in Figure 4, this time taking instances from $\mathcal{L}_\dagger$ rather than $\mathcal{L}_\star$. Let me comment on the additional schemas from Figure 4.

We have informally characterized ■ as the least modality broader than any domain specifiers. The schema (Master) and the rule (Surj) formally regiment

$$\vdash \blacksquare P \to \mathsf{At}(\bar{F})(P) \qquad \text{(Master)}$$

$$\vdash \blacksquare(M \equiv N) \to (P[M/x] \to P[N/x]) \qquad (\text{SLL}_{\blacksquare})$$

$$\begin{array}{l} \text{If } \Gamma \vdash \mathsf{At}(\bar{F})(\mathsf{At}(\bar{X})(P)) \text{ for all } \bar{\sigma},\ \bar{X} : \bar{\sigma} \text{ with } \bar{X} \notin \mathsf{FV}(\Gamma, P, \bar{F}), \\ \text{then } \Gamma \vdash \mathsf{At}(\bar{F})(\blacksquare P). \end{array} \qquad \text{(Surj)}$$

FIGURE 4. Additional schemas axiomatizing $\mathtt{Q}_\dagger$.

this characterization.[33] (Master), whose dual we have already encountered in Section 2.2, ensures that $\blacksquare$ is at least as broad as any domain specifier. On the other hand, (Surj) ensures $\blacksquare$ is the *least* modality with this property. If we can prove from assumptions Γ that within the scope of a domain specifier $\mathsf{At}(\bar{F})$, P holds under arbitrary domain specifiers, then we can conclude, from the same assumption, that $\blacksquare P$ holds within the scope of $\mathsf{At}(\bar{F})$. The case where $\mathsf{At}(\bar{F})$ is just $\mathsf{At}(\bar{\mathsf{E}!})$ reduces to the simpler rule:[34]

$$\begin{array}{l} \text{if } \Gamma \vdash \mathsf{At}(\bar{X})(P) \text{ for all } \bar{\sigma},\ \bar{X} : \bar{\sigma} \text{ with } \bar{X} \notin \mathsf{FV}(\Gamma, P), \\ \text{then } \Gamma \vdash \blacksquare P. \end{array} \qquad \text{(CSurj)}$$

Thus (Surj) generalizes (CSurj) in much the same way that (At-UG) generalizes (UG).

I should point out that (Surj) is an *infinitary* rule, though in a fairly harmless way. The rule is infinitary because it (schematically) universally generalizes over nice type sequences, which cannot be done in the object language. It is an open question whether (Surj) can be expressed as a finitary rule in the present language.

The remaining schema ($\text{SLL}_{\blacksquare}$) was anticipated in Section 3.3. It captures the Quantificationalist analysis of identity as *applicative indiscernibility at all domains of quantification*. Thus, when working in systems containing this schema, I will abbreviate

$$M = N := \blacksquare(M \equiv N).$$

Again using an adaptation of standard canonical model constructions, we can prove the following completeness result.

Theorem 6.2. $\mathtt{Q}_\dagger$ *is sound and complete for validity over all* $\mathcal{L}_\dagger$*-models. Consequently, it is consistent.*

The proof is presented in in Appendix A.

As anticipated, $\mathtt{Q}_\dagger$ is consistent with Quantificationalism. This is a simple consequence of the completeness result just mentioned, but let me illustrate with a concrete instance. An easy consequence of (Gen), the stability axioms, (App) and (At-Nec) is the following schema

$$F_\sigma(\mathsf{At}(\bar{F})(M)) \leftrightarrow \mathsf{At}(F_\sigma)(\mathsf{E}!M). \qquad (18)$$

[33]The label (Surj) stands for *Surjectivity*. Intuitively, (Surj) says that domain specifiers pick out all quantificational substitutions, a condition that was baked into our model theory by requiring that every quantificational substitution be determined by some domain.

[34]The label (CSurj) stands for *contingent surjectivity*. Intuitively (CSurj) says that domain specifiers pick out all metaphysical substitutions, perhaps contingently in the sense that they may fail to do so at some other domain.

An instance of this is

$$\vdash \bot_{t\to t}(\mathsf{At}(\bot_{t\to t})(\top)) \leftrightarrow \mathsf{At}(\bot_{t\to t})(\mathsf{E!}\top). \tag{19}$$

Since, obviously, $\mathtt{Q}_\star \vdash \neg\bot_{t\to t}(\mathsf{At}(\bot_{t\to t})(\top))$, using the stability axioms and (App) it follows that

$$\vdash \mathsf{At}(\bot_{t\to t})(\neg\mathsf{E!}\top). \tag{20}$$

Consequently,

$$\vdash \mathsf{E!}\top \to (\mathsf{E!}\top \wedge \blacklozenge\neg\mathsf{E!}\top). \tag{21}$$

Using the dual of (FrUI), this implies

$$\vdash \mathsf{E!}\top \to \exists p(p \wedge \blacklozenge\neg p). \tag{22}$$

Since $\mathsf{E!}\top$ is consistent in $\mathtt{Q}_\dagger$ (this can be verified using the soundness result above) it follows that Quantificationalism is consistent in $\mathtt{Q}_\dagger$.

For the remainder of the paper I shall treat $\mathtt{Q}_\dagger$ as the base system over $\mathcal{L}_\dagger$-terms. I review some notable theorems of $\mathtt{Q}_\dagger$ and consider various philosophically interesting extensions thereof. If Γ is a set of $\mathcal{L}_\dagger$-formulas, I write $\mathtt{Q}_\dagger \oplus \Gamma$ for the least set of $\mathcal{L}_\dagger$-formulas extending both $\mathtt{Q}_\dagger$ and Γ that is closed under the rules of $\mathtt{Q}_\dagger$.

6.3. **The Modal Logic of $\blacksquare$.** Let me start by saying a little more about the logic of $\blacksquare$ in $\mathtt{Q}_\dagger$. First, note that $\blacksquare$ obeys all axioms of the normal modal logic $\mathtt{S4.D}$:

$$\blacksquare(P \to Q) \to (\blacksquare P \to \blacksquare Q) \tag{K}$$

$$\blacksquare P \to P \tag{T}$$

$$\blacksquare P \to \blacksquare\blacksquare P \tag{4}$$

$$\blacksquare P \to \blacklozenge P \tag{D}$$

That (T) is a theorem can be shown (Master), (Id) and (WLL). Moreover, (D) follows by an instance of (Master) and its dual. For (K), from $\blacksquare(P \to Q) \wedge \blacksquare P$ we may infer $\mathsf{At}(\bar{X})(P \to Q) \wedge \mathsf{At}(\bar{X})(P)$ for an arbitrary nice term sequence $\bar{X}$, using (Master) and propositional logic. This implies $\mathsf{At}(\bar{X})(Q)$ using the fact that $\mathsf{At}(\bar{X})$ is a normal modality. For (4), note that for arbitrary term sequences of variables $\bar{X}, \bar{Y}$ we can derive $\blacksquare P \to \mathsf{At}(\bar{X})(\mathsf{At}(\bar{Y})(P))$, by first using (Master) to derive $\blacksquare P \to \mathsf{At}(\mathsf{At}(\bar{X})(\bar{Y}))(P)$ and then applying (CompG). (4) then follows by applying (Surj) first, then (CSurj).

Second, as a straightforward consequence of (At-Nec) and (Surj), we can show that $\mathtt{Q}_\dagger$ is closed under a necessitation rule for $\blacksquare$:

$$\text{if } \vdash P \text{ then } \vdash \blacksquare P. \tag{$\blacksquare$Nec}$$

Thus, while all axioms of $\mathtt{Q}_\star$ are stated in terms of applicative indiscernibility, $\mathtt{Q}_\dagger$ derives their $\blacksquare$-necessitations by ($\blacksquare$Nec). So, $\mathtt{Q}_\dagger$ can be equivalently axiomatized by the schemas in Figure 4 and the result of substituting $=$ for $\equiv$ in the schemas from (3).

We might wonder what happens if we require that the logic of $\blacksquare$ be at least $\mathtt{S5}$, by adding

$$P \to \blacksquare\blacklozenge P. \tag{$\mathrm{B}_\blacksquare$}$$

A sufficient condition for a model to validate ($\mathrm{B}_\blacksquare$) is that substitutions in the underlying QSS form a group: that is, for any $i_{\bar{\mathbf{f}}}$ there should be some $i_{\bar{\mathbf{g}}}$ such that $i_{\bar{\mathbf{g}}} \circ i_{\bar{\mathbf{f}}} = i_{[\![\bar{\mathsf{E!}}]\!]^g}$. I have not confirmed whether there are any models satisfying this condition. But it turns out this is not a necessary condition: the model constructed in the proof of Theorem 5.16 does not satisfy it, and yet it validates ($\text{B}_{\blacksquare}$). Substitutions fail to form a group because whenever $\bar{\mathbf{h}}$ consists entirely of constant mappings over points, $i_{\bar{\mathbf{h}}}\mathbf{a}$ is always a constant mapping over points as well. But no substitution can turn a constant mapping over points into a non-constant one.

To see that our model, nonetheless, validates ($\text{B}_{\blacksquare}$), suppose $\mathfrak{A} \Vdash [\![P]\!]^g$. That means $\mathfrak{A} \models [\![P]\!]^g(\bar{\mathbf{d}})$. Let $\bar{\mathbf{k}}$ be any upper domain. It suffices to find an upper domain $\bar{\mathbf{c}}$ such that $\bar{\mathbf{c}}(\bar{\mathbf{k}}(\bar{\mathbf{d}})) = \bar{\mathbf{d}}$. This can be easily done: we simply let $\bar{\mathbf{c}}$ consist of the constant functions $\mathbf{c}_{\mathbf{d}_\sigma}$, each mapping every point to $\mathbf{d}_\sigma$. Thus $\mathfrak{A} \models (i_{\bar{\mathbf{k}}} \circ i_{\bar{\mathbf{c}}}[\![P]\!]^g)(\bar{\mathbf{d}})$ holds for every upper domain $\bar{\mathbf{k}}$, which implies $\mathfrak{A} \models [\![\blacksquare\blacklozenge P]\!]^g$.[35] Consequently, $\mathtt{Q}_\dagger \oplus \text{B}_{\blacksquare}$ is consistent.

6.4. **Granularity.** Theorizing in terms of domain specifiers forces one to take a stand on questions regarding how fine-grained reality is. The balance seems to be towards finer distinctions: $\mathtt{Q}_\dagger$ negatively settles some identity questions that $\mathtt{FH}$ leaves open. A simple example is the question whether $\exists p.p$ and $\top$ are identical. Either answer to this question is consistent in $\mathtt{FH}$, whereas $\mathtt{Q}_\dagger$ implies a negative answer: indeed, $\mathtt{Q}_\dagger$ proves both $\mathsf{At}(\bot_t)(\top)$ and $\neg\mathsf{At}(\bot_t)(\exists p.p)$, which by ($\text{SLL}_{\blacksquare}$) implies $(\exists p.p) \neq \top$.

Surprisingly, however, $\mathtt{Q}_\dagger$ turns out to be consistent with a fairly coarse-grained identity criterion. Bacon [2024] calls *free classicism* (something equivalent to) the system obtained by adding the following rule to $\mathtt{FH}_0$:

$$\text{if } \vdash P \leftrightarrow Q \text{ then } \vdash (\lambda\bar{x}.P) = (\lambda\bar{x}.Q). \qquad (\text{Equiv}^+)$$

Intuitively, free classicism identifies all properties that are provably coextensive in $\mathtt{FH}_0$. In our setting, it is natural to consider the system $\mathtt{QFC}_\dagger$—for *Quantificationalist Free Classicism*—which results from closing $\mathtt{Q}_\dagger$ under (Equiv^+). This system identifies all properties that are provably coextensive in $\mathtt{Q}_\dagger$.

To establish the consistency of $\mathtt{QFC}_\dagger$, let me introduce the notion of a *quasi-extensional* model.[36]

Definition 6.3. A model $\mathfrak{A}$ is *quasi-extensional* when the following conditions hold:

(1) Its underlying QSS is quasi-functional;
(2) For any $\mathbf{p}, \mathbf{q} \in A^t$, if $\mathfrak{A} \models i_{\bar{\mathbf{f}}}\mathbf{p}$ holds precisely when $\mathfrak{A} \models i_{\bar{\mathbf{f}}}\mathbf{q}$ does for all substitutions $i_{\bar{\mathbf{f}}}$, then $\mathbf{p} = \mathbf{q}$ (*quasi-Fregeanness*).

Intuitively, the property of quasi-Fregeanness captures the idea that propositions having the same truth value at all domains of quantification are identical.

Theorem 6.4. $\mathtt{QFC}_\dagger$ *is sound with respect to validity in quasi-extensional, stabilized models.*

Proof. Let C be the class of quasi-extensional, stabilized models. Note it is closed under re-pointing. To see that quasi-Fregeanness is closed under repointing, take

[35] In the presence of ($\text{B}_{\blacksquare}$), as one would expect, we can prove that distinctness is $\blacksquare$-necessary. Since identity is *defined* in terms of $\blacksquare$ itself, the argument is even simpler than Prior's famous derivation of the necessity of distinctness in $\mathtt{S5}$ and relies only on ($\text{B}_{\blacksquare}$) and the duality between $\blacksquare$ and $\blacklozenge$. If $M \neq N$, that means $\neg\blacksquare(M \equiv N)$. This implies $\blacklozenge\neg(M \equiv N)$, which by in turn implies $\blacksquare\blacklozenge\neg(M \equiv N)$. This is equivalent to $\blacksquare(M = N)$.

[36] Bacon and Dorr [2024] introduce a notion of *intensionality* for categories of models based on ordinary applicative structures, using a structurally similar definition.

any $\mathfrak{A} \in \mathsf{C}$ and let $\bar{\mathbf{f}}$ be any $\bar{\sigma}$-domain. Assume that for every substitution $i_{\bar{\mathbf{g}}}$, we have $\mathfrak{A} \models_{\bar{\mathbf{f}}} i_{\bar{\mathbf{g}}}\mathbf{p}$ iff $\mathfrak{A} \models_{\bar{\mathbf{f}}} i_{\bar{\mathbf{g}}}\mathbf{q}$. This is to say that for all $i_{\bar{\mathbf{g}}}$ we have $\mathfrak{A} \models (i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{g}}})\mathbf{p}$ iff $\mathfrak{A} \models (i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{g}}})\mathbf{q}$. I claim this is equivalent to saying that for all substitutions $i_{\bar{\mathbf{h}}}$ we have $\mathfrak{A} \models i_{\bar{\mathbf{h}}}\mathbf{p}$ iff $\mathfrak{A} \models i_{\bar{\mathbf{h}}}\mathbf{q}$. The right-to-left direction is obvious. Conversely, suppose that for some $i_{\bar{\mathbf{h}}}$, the second equivalence fails. Let $\bar{\mathbf{s}}$ be a $\bar{\sigma}+\bar{\tau}$-stabilizer of $pad_{\bar{\sigma}+\bar{\tau}}\bar{\mathbf{h}}$. By Proposition 5.12 and the fact that $i_{\bar{\mathbf{h}}}\mathbf{a} = i_{pad_{\bar{\sigma}+\bar{\tau}}\bar{\mathbf{h}}}\mathbf{p}$, it follows that $i_{\bar{\mathbf{s}}}\mathbf{p}$ and $i_{\bar{\mathbf{s}}}\mathbf{q}$ are $\bar{\sigma}$-stabilizers for $i_{\bar{\mathbf{h}}}\mathbf{p}$ and $i_{\bar{\mathbf{h}}}\mathbf{q}$ respectively. But then $(i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{s}}})\mathbf{p} \equiv i_{\bar{\mathbf{s}}}\mathbf{p} \equiv i_{\bar{\mathbf{h}}}\mathbf{p}$ and likewise for $\mathbf{q}$. Since the equivalence $\mathfrak{A} \models i_{\bar{\mathbf{h}}}\mathbf{p}$ iff $\mathfrak{A} \models i_{\bar{\mathbf{h}}}\mathbf{q}$ fails, so must the equivalence $\mathfrak{A} \models (i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{s}}})\mathbf{p}$ iff $\mathfrak{A} \models (i_{\bar{\mathbf{f}}} \circ i_{\bar{\mathbf{s}}})\mathbf{q}$, as desired.

We now check that (Equiv$^+$) is sound with respect to C. So suppose $\mathsf{C} \models P \leftrightarrow Q$. Take any model $\mathfrak{A} \in \mathsf{C}$. Take any model $\mathfrak{A} \in \mathsf{C}$. We want to show that for every domain $\bar{\mathbf{f}}$ we have $i_{\bar{\mathbf{f}}}[\![\lambda\bar{x}.P]\!]^g \equiv i_{\bar{\mathbf{f}}}[\![\lambda\bar{x}.Q]\!]^g$. For simplicity, assume $\bar{x}$ is a single variable $x : \sigma$; the general case is analogous.

Since $\mathfrak{A}$ is quasi-functional, it suffices to show that for each domain $i_{\bar{\mathbf{f}}}$ and every $\mathbf{a} \in A^\sigma$ we have $App(i_{\bar{\mathbf{f}}}[\![\lambda x.P]\!]^g, \mathbf{a}) = App(i_{\bar{\mathbf{f}}}[\![\lambda x.Q]\!]^g, \mathbf{a})$. Take any $\mathbf{a} \in A^\sigma$ and suppose $\mathfrak{A} \models App(i_{\bar{\mathbf{f}}}[\![\lambda x.P]\!]^g, \mathbf{a})$. Since $\mathfrak{A}$ is stabilized we can find a stabilizer $\mathbf{s} \in A^\sigma$ of $\mathbf{a}$. It follows that $\mathfrak{A} \models i_{\bar{\mathbf{f}}}(App([\![\lambda x.P]\!]^g, \mathbf{s}))$. This is equivalent to saying $\mathfrak{A} \models_{\bar{\mathbf{f}}} App([\![\lambda x.P]\!]^g, \mathbf{s})$, so since C is closed under repointing we infer $\mathfrak{A} \models_{\bar{\mathbf{f}}} App([\![\lambda x.Q]\!]^g, \mathbf{s})$ and in turn $\mathfrak{A} \models App(i_{\bar{\mathbf{f}}}[\![\lambda x.Q]\!]^g, \mathbf{a})$. We have thus shown that for all substitutions $i_{\bar{\mathbf{f}}}$, if $\mathfrak{A} \models App(i_{\bar{\mathbf{f}}}[\![\lambda x.P]\!]^g, \mathbf{a})$, then $\mathfrak{A} \models App(i_{\bar{\mathbf{f}}}[\![\lambda x.Q]\!]^g, \mathbf{a})$; the converse follows analogously. But $\mathfrak{A}$ is quasi-extensional, so this implies $i_{\bar{\mathbf{f}}}[\![\lambda x.P]\!]^g = i_{\bar{\mathbf{f}}}[\![\lambda x.Q]\!]^g$, as desired. ☐

Corollary 6.5. $\mathtt{QFC}_\dagger$ *is consistent.*

Proof. We use the construction form the proof of Theorem 5.16, but require that the applicative structure we start from be *Fregean*: A^t must consist of exactly two elements. If so, then the final model is quasi-Fregean. Moreover, we have already noted it is functional, and functionality clearly implies quasi-functionality. Thus, the final model belongs to C, showing C is non-empty. By Theorem 6.4, $\mathtt{QFC}_\dagger$ must be consistent. ☐

6.5. **Quantifier Principles.** So far, I have not presented any schemas that constrain how domain specifiers interact with quantifiers. Some such constraints can already be derived from the other axioms. For example, by using (At-Nec) on (FrUI) and distributing the domain specifier in accordance to (Stab♡) we can show that $\mathtt{Q}_\dagger$ derives

$$\mathsf{At}(\bar{F})(\mathsf{E!}_\sigma a) \to \mathsf{At}(\bar{F})(\forall x P \to P[a/x]) \tag{AtUI}$$

When F_σ is defined, using (Gen) we can further derive

$$F_\sigma \mathsf{At}(\bar{F})(a) \to \mathsf{At}(\bar{F})(\forall x P \to P[a/x]) \tag{AtGUI}$$

We can impose additional constraints corresponding to the model-theoretic property of quantifier generality. One way of doing so is to add schemas ensuring that each term $\mathsf{At}(\bar{F})(\hat{\forall}_\sigma)$, when F_σ is defined, behaves like the restriction of a *classical* quantifier by the predicate F_σ. In $\mathtt{Q}_\dagger$, this can be done by adding the following schemas:[37]

$$\mathsf{At}(\bar{F})(\mathsf{E!}_\sigma)a \to (\mathsf{At}(\bar{F})(\hat{\forall}_\sigma)G \to Ga) \quad (\widehat{\text{AtFrUI}})$$

$$\mathsf{At}(\bar{F})(\hat{\forall}_\sigma)(\lambda x.Gx \to Hx) \to (\mathsf{At}(\bar{F})(\hat{\forall}_\sigma)G \to \mathsf{At}(\bar{F})(\hat{\forall}_\sigma)H) \quad (\widehat{\text{AtNorm}})$$

$$\mathsf{At}(\bar{F})(\hat{\forall}_\sigma)G \leftrightarrow \neg\mathsf{At}(\bar{F})(\hat{\exists}_\sigma)(\lambda x.\neg Gx) \quad (\widehat{\text{AtDual}})$$

I write $\mathtt{QG}_\dagger$ for the theory $\mathtt{Q}_\dagger \oplus (\widehat{\text{AtFrUI}})$–$(\widehat{\text{AtDual}})$.

Alternatively, we could add to $\mathtt{Q}_\dagger$ only instances of the above where $\bar{F} = \bar{\top}\sigma = (\top_{\sigma_1}, \dots, \top_{\sigma_n})$—which ensure that $\mathsf{At}(\bar{\top})(\hat{\forall})$ behaves like a classical quantifier whenever $\bar{\top}$ is one such sequence—then add

$$\mathsf{At}(\bar{F})(\hat{\forall}_\sigma) \equiv \lambda X.\mathsf{At}(\bar{\top})(\hat{\forall})(\lambda y.F_\sigma y \to Xy) \quad (23)$$

$$\mathsf{At}(\bar{F})(\hat{\exists}_\sigma) \equiv \lambda X.\mathsf{At}(\bar{\top})(\hat{\exists})(\lambda y.F_\sigma y \wedge Xy). \quad (24)$$

This would ensure that the remaining instances of $(\widehat{\mathsf{At}\forall\mathsf{E!}})$–$(\widehat{\text{AtDual}})$ are derivable. These two approaches are equivalent over $\mathtt{QFC}_\dagger$, in the sense that the least $\mathtt{Q}_\dagger$-theory containing $\mathtt{QFC}_\dagger$ and the first set of axioms is the same as the least $\mathtt{Q}_\dagger$-theory containing $\mathtt{QFC}_\dagger$ and the second set of axioms. However, I conjecture that the second approach is stronger in $\mathtt{Q}_\dagger$, in the sense that $\mathtt{QG}_\dagger$ does not derive (23) and (24). All the theories just mentioned are consistent, as all the schemas above are valid in the model used for the proof of Theorem 5.16 and any re-pointing thereof.

It is important to keep in mind that all of the above schemas only constrain the behavior of the definable quantifiers $\hat{\forall}$ and $\hat{\exists}$, but not the behavior of the more expressive syncategorematic quantifiers. In fact, none of the schemas above can even be restated as a constraint on the syncategorematic quantifiers, since $\mathsf{At}(\bar{F})(\forall)$ and $\mathsf{At}(\bar{F})(\exists)$ are not even well formed. Let me illustrate using the duality of quantifiers as an example. There are two legitimate questions to ask regarding how the duality between $\hat{\forall}$ and $\hat{\exists}$ interacts with domain specifiers. One is whether $(\widehat{\text{AtDual}})$ is true, the other is whether the following is true:

$$\mathsf{At}(\bar{F})(\hat{\forall}G) \leftrightarrow \mathsf{At}(\bar{F})(\neg\hat{\exists}\lambda x.\neg Gx) \quad (25)$$

The above schema differs from $(\widehat{\text{AtDual}})$ in that G is also within the scope of the domain specifier $\mathsf{At}(\bar{F})$. $\mathtt{Q}_\dagger$ leaves open the first question, but answers the second question affirmatively. However, there is only one legitimate question to ask regarding how the duality between the *syncategorematic quantifiers* interacts with domain specifiers, namely whether the following is true:

$$\mathsf{At}(\bar{F})(\forall xP) \leftrightarrow \mathsf{At}(\bar{F})(\neg\exists x\neg P) \quad (26)$$

This question is analogous to the second question about the definable quantifiers and, like the latter, it is answered affirmatively in $\mathtt{Q}_\dagger$.

Another interesting principle to consider is the following:

$$F_\sigma a \to \mathsf{At}(\bar{F})(\forall xP \to P[a/x]). \quad (27)$$

[37]Note the schema

$$\mathsf{At}(\bar{F})(\hat{\forall}_\sigma)F_\sigma \quad (\widehat{\mathsf{At}\forall\mathsf{E!}})$$

is already a theorem of $\mathtt{Q}_\dagger$, derived using (At-Nec), ($\beta\eta$) and (Gen).

(27) differs from (AtGUI) in that $\mathsf{At}(\bar{F})(a)$ has been replaced with a. It is equivalent to

$$F_\sigma a \to \mathsf{At}(\bar{F})(\mathsf{E!}a). \tag{28}$$

These principles are tempting and seems to fit the intuitive gloss of the meaning of domain specifiers. They license natural language inferences that strike us as valid. For example:

(2) Among those who attended the dinner, nobody enjoyed the food.
(3) Alice attended the dinner.
(4) So, Alice did not enjoy the food.

Under an analysis under which 'Among those who attended the dinner' semantically associates with a domain specifier (something like $\mathsf{At}(\lambda x.x$ attended the dinner$))$ taking 'nobody enjoyed the food' as its argument, (27) would seem to license this inference.

But (27) is inconsistent in $\mathsf{Q}_\dagger$. It has provably false instances when $F : t \to t$. Note that in $\mathsf{Q}_\dagger$, (28) implies

$$Fa \to F\mathsf{At}(F)(a). \tag{29}$$

An instance of the above is

$$(\lambda p.p)\mathsf{E!}\bot \to (\lambda p.p)\mathsf{At}(\lambda p.p)(\mathsf{E!}\bot), \tag{30}$$

which is $\beta\eta$-equivalent to

$$\mathsf{E!}\bot \to \mathsf{At}(\lambda p.p)(\mathsf{E!}\bot), \tag{31}$$

In words: if $\bot$ exists, then *at the truths*, $\bot$ exists. This certainly sounds false, and indeed it is inconsistent in $\mathsf{Q}_\dagger$: the antecedent is consistent in $\mathsf{Q}_\dagger$, but the consequent is provably false. Obviously, $\mathsf{Q}_\dagger \vdash \neg\bot$. Using (Gen) and (Stab♡), it follows that $\mathsf{Q}_\dagger \vdash \neg\mathsf{At}(\lambda p.p)(\mathsf{E!}\bot)$.

That being said, the restriction of (27) obtained by requiring $F : e \to t$ is consistent in $\mathsf{Q}_\dagger$. It is a theorem of any extension of $\mathsf{Q}_\dagger$ which proves a schema to the effect that individuals are finitely stable. Given such a schema, the relevant restriction of (27) is a simple consequence of (AtGUI). We can thus view (27) as a sort of limit case of (AtGUI), restrictions of which become provable in the presence of stability principles of the right sort.

To conclude this subsection, I want to return to the issues of Barcan (and converse Barcan) formulas for domain specifiers. We can explicitly define syncategorematic expressions whose semantic functions are respectively to "restrict" or "expand"—rather than "reset"—the domain of quantification to a particular value. Given a nice term sequence $\bar{F}$ indexed by $\bar{\sigma}$, write

$$\bar{F}^\cap := (F_\sigma \wedge_\sigma \mathsf{E!}_\sigma)_{\sigma\in\bar{\sigma}} \qquad \bar{F}^\cup := (F_\sigma \vee_\sigma \mathsf{E!}_\sigma)_{\sigma\in\bar{\sigma}}.$$

We then put

$$\mathsf{At}^\cap(\bar{F})(M) := \mathsf{At}(\bar{F}^\cap)(M) \qquad \mathsf{At}^\cup(\bar{F})(M) := \mathsf{At}(\bar{F}^\cup)(M).$$

I have noted that (At-Barcan) should not be a theorem of the correct logic of domain specification, given the intended reading of domain specifiers. However, in view of the gloss just given of the expression $\mathsf{At}^\cap(\bar{F})$, one might have thought that (At-Barcan) should be derivable in when we replace $\mathsf{At}^\cap$ for At:

$$\forall x\mathsf{At}^\cap(\bar{F})(P) \to \mathsf{At}^\cap(\bar{F})(\forall xP) \qquad x \text{ not free in } \bar{F}. \tag{$\mathsf{At}^\cap$-Barcan}$$

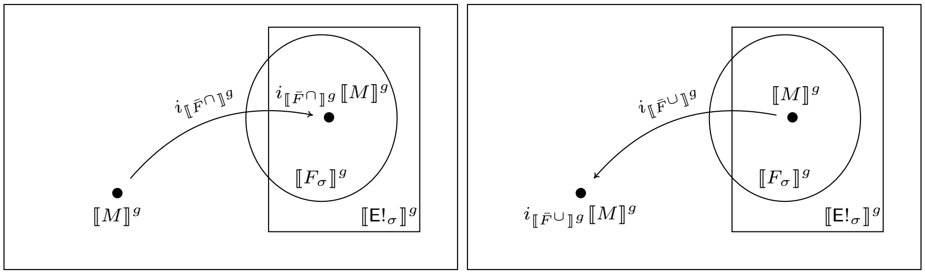

FIGURE 5. On the left, a counter-example to ($\mathsf{At}^\cap$-Barcan). On the right, a counter-example to ($\mathsf{At}^\cup$-CBarcan).

Likewise, in view of the gloss just given of the expression $\mathsf{At}^\cup(\bar{F})$, one might have thought that the following *converse Barcan* formula for $\mathsf{At}^\cup(\bar{F})$ should be derivable.

$$\mathsf{At}^\cup(\bar{F})(\forall xP) \to \forall x\mathsf{At}^\cup(\bar{F})(P) \qquad x \text{ not free in } \bar{F}. \qquad (\mathsf{At}^\cup\text{-CBarcan})$$

But neither is the case. Here is a counter-example to ($\mathsf{At}^\cap$-Barcan). Consider the proposition that something is identical to the proposition that there are only truths.

$$\exists p(p = \forall q.q). \tag{32}$$

It is provable in $\mathsf{Q}_\dagger$ that $\mathsf{At}^\cap(\lambda p.p)(\exists p(p = \forall q.q))$. For both $\mathsf{At}^\cap(\lambda p.p)(\forall q.q)$ and $\mathsf{At}^\cap(\lambda p.p)(\forall q.q = \forall q.q)$ are provable and $\mathsf{At}^\cap(\lambda p.p)(\exists p(p = \forall q.q))$ can be inferred from the latter by applying (3) and the $\mathsf{At}^\cap(\lambda p.p)$-necessitation of (FrUI). However, $\exists p.\mathsf{At}^\cap(\lambda p.p)(p = \forall q.q)$ is false in some models. For example, it can be falsified in a model generated by the direct power construction used in the proof of Theorem 5.16. We simply need to choose the distinguished domain $\bar{\mathbf{d}}$ so that $\mathbf{d}_t$ contains some truths and some falsehoods, but none of the falsehoods in the domain are applicatively indiscernible from the interpretation of $\forall q.q$. In any such model, $\mathsf{At}^\cap(\lambda p.p)(p = \forall q.q)$ is equivalent to $p = \forall q.q$, so $\exists p.\mathsf{At}^\cap(\lambda p.p)(p = \forall q.q)$ is equivalent to $\exists p(p = \exists q.q)$, which is false. This means that the substitution $i_{[\![\lambda p.\mathsf{E!}p \wedge p]\!]^g}$ maps $[\![\exists p(p = \exists q.q)]\!]^g$, a falsehood, to a truth. Not only that: the unique *stable* falsehood that is applicatively indiscernible from $[\![\exists p(p = \exists q.q)]\!]^g$ is *not* applicatively indiscernible from the unique *stable* truth that is applicatively indiscernible from $i_{[\![\lambda p.\mathsf{E!}p \wedge p]\!]^g}[\![\exists p(p = \exists q.q)]\!]^g$.

More generally, counter-examples to ($\mathsf{At}^\cap$-Barcan) of this sort can always be generated in direct product models where there is some $[\![M]\!]^g \in A^\sigma$ and some substitution $i_{[\![\bar{F}^\cap]\!]^g}$ such that $App([\![\mathsf{E!}_\sigma]\!]^g, [\![M]\!]^g)$ is false but

$$App([\![\lambda x.\mathsf{E!}x \wedge F_\sigma x]\!]^g, [\![\mathsf{At}^\cap(\bar{F})(M)]\!]^g) \tag{33}$$

is true. In such models, the instance of ($\mathsf{At}^\cap$-Barcan) obtained by letting $P :=$ $\exists y.M = y$ is always false. Likewise, counter-examples to ($\mathsf{At}^\cup$-CBarcan) can always be generated in direct product models where there is some $[\![M]\!]^g \in A^\sigma$ and some substitution $i_{[\![\bar{F}^\cup]\!]^g}$ such that $App([\![\mathsf{E!}_\sigma]\!]^g, [\![M]\!]^g)$ is true but

$$App([\![\lambda x.\mathsf{E!}x \vee F_\sigma x]\!]^g, [\![\mathsf{At}^\cap(\bar{F})(M)]\!]^g) \tag{34}$$

is false. I give graphical representations of both sorts of counter-examples in Figure 5.

The failures of ($\mathsf{At}^{\cap}$-Barcan) and ($\mathsf{At}^{\cup}$-CBarcan) are important issues relating to the intended interpretation of domain specifiers, so let me unpack things further. In the presence of axioms ensuring that the quantifiers in ($\mathsf{At}^{\cap}$-Barcan) are somehow restricted to *stable* entities only, then ($\mathsf{At}^{\cap}$-Barcan) is indeed derivable. Likewise for ($\mathsf{At}^{\cup}$-CBarcan). For example, as I shall discuss in Section 6.6, the intuitive idea that individuals involve no quantification at all motivates extending $\mathtt{Q}_{\dagger}$ with a schema ensuring that all entities at type e are stable. In this extension of $\mathtt{Q}_{\dagger}$, all instances of ($\mathsf{At}^{\cap}$-Barcan) and ($\mathsf{At}^{\cup}$-CBarcan) where the quantifiers bind variables of type e are derivable. Counter-examples of the above sort crucially involve the use of non-stable entities, and it turns out this is a general feature of *all* possible counter-examples to ($\mathsf{At}^{\cap}$-Barcan) and ($\mathsf{At}^{\cup}$-CBarcan). To put it in a slogan, while not every entity *exists at the existing Fs*, every *stable* entity does.

Further, instances of ($\mathsf{At}^{\cap}$-Barcan) and ($\mathsf{At}^{\cup}$-CBarcan) where the variable x is abstractable in P are in fact derivable in the stronger system $\mathtt{QG}_{\dagger}$ and thus are valid in all direct product models. So, all possible counter-examples to these principles, in $\mathtt{QG}_{\dagger}$, must involve instances where P contains occurrences of domain specifiers or $\blacksquare$. Without these expressions, it is impossible to "harness" the lack of stability of the desired entities to generate counter-examples.

6.6. **Object-Language Stability Principles.** I now turn to stability principles. There are two main ways we can extend $\mathtt{Q}_{\dagger}$ with further stability constraints. One is to add more schemas similar to (Stab$\heartsuit$) and (Stabλ). The other is to introduce object-language terms expressing various notions of stability, then add axioms restricting the behavior of these terms. The first works fine for requiring that entities of a certain sort be themselves stables, while the second is needed to formulate comprehension principles on the notion of stability. I discuss each approach in turn.

One idea that I have just mentioned in the previous section is to extend $\mathtt{Q}_{\dagger}$ with a requirement that individuals be stable. This can be done by adding the following schema:

$$a \equiv \mathsf{At}(\bar{F})(a) \qquad a : e. \tag{StabInd}$$

$\mathtt{Q}_{\dagger} \oplus$ StabInd also derives the $\blacksquare$-necessitation of (StabInd), so it only has models where all the entities of type e are stable.

Of course, $\mathtt{Q}_{\dagger} \oplus$ StabInd derives

$$F_e\mathsf{At}(\bar{F})(a) \leftrightarrow F_e a \qquad a : e. \tag{35}$$

Since (AtGUI) is also derivable, it follows that $\mathtt{Q}_{\dagger} \oplus$ StabInd proves every instance of (27) where $a : e$.

It is worth noting that while (StabInd) is consistent in $\mathtt{Q}_{\dagger}$, it need not be consistent in systems formulated in richer signatures. In particular, it may not be consistent in systems with signatures containing non-logical constants whose type features e as a terminal type, if those constants are forced to have non-stable interpretations.[38]

[38] I have already sketched a view on which definite descriptions stand for special sorts of non-stable individuals: 'the tallest F' stands for a non-stable individual applicatively indiscernible from whomever *in the current domain* is taller than any F. We may want to regiment this by adding a constant $T : (e \to t) \to e$ with a non-stable interpretation that captures the desired meaning. Then, by (App), TF is guaranteed to be non-stable whenever F is stable, and can be non-stable even when F is non-stable as well.

Another interesting schema to consider is one that requires all *constants* in the signature to express finitely stable entities:

$$M \equiv \mathsf{At}(\bar{F})(M) \qquad\qquad M \text{ a non-logical constant.} \qquad\qquad \text{(StabC)}$$

The plausibility of (StabC) varies depending on what we take the intended interpretations of our constants to be. I am intrigued by the view that (StabC) should hold if we take ourselves to be theorizing in a *fundamental language*, where every constant denotes a fundamental entity and no two constants co-refer. Fundamental entities are naturally taken to involve no quantification at all. Grounding theorists, for example, seem to be committed to this idea at least for entities of type t: most grounding theorists define a fundamental proposition as one that is not grounded by anything and accept that "quantified propositions" are grounded in their instances.[39]

Let us now turn to the question of expressing stability-theoretic notions in the object-language. The most naive attempt at expressing $\bar{\sigma}$-stability involves quantifying into the first argument place of domain specifiers:

$$\mathsf{Stab}^N_{\bar{\sigma}}(M) := \forall \bar{X} \blacksquare (M \equiv \mathsf{At}(\bar{X})(M)).$$

But this does not have the intended reading in $\mathtt{Q}_\dagger$, since universal generalizations fail to imply all their instances. Theorizing in $\mathtt{QG}_\dagger$ to use the definable classical quantifiers $\mathsf{At}(\lambda x.\top_\sigma)(\hat{\forall})$ instead of the syncategorematic quantifiers does not help either, since one would need to abstract the variables $\bar{X}$, which occur within the scope of $\blacksquare$, from outside that occurrence of $\blacksquare$.

The best way to overcome these expressive limitations is to theorize in $\mathtt{QFC}_\dagger$ and express generality not by means of quantification, but by using higher-order identities. We can define the abbreviation

$$\mathsf{Stab}_{\bar{\sigma}}(M) := \blacksquare((\lambda \bar{X}.M) \equiv (\lambda \bar{X}.\mathsf{At}(\bar{X})(M))).$$

Here $\bar{X}$ must be indexed by $\bar{\sigma}$ and none of the variables occurring in it can be free in M. By ($\mathrm{SLL}_\blacksquare$), $\mathsf{Stab}_{\bar{\sigma}}(M)$ implies $M = \mathsf{At}(\bar{F})(M)$ whenever $\bar{F}$ is indexed by $\bar{F}$. Moreover, if $M : \sigma_1 \to \cdots \to \sigma_n \to t$ and $\mathtt{QFC}_\dagger \vdash M\bar{Y} \leftrightarrow \mathsf{At}(\bar{X})(M)\bar{Y}$ for an arbitrary sequence of variables $\bar{Y} := (Y_1, \ldots, Y_n)$ with $Y_i : \sigma_i$, none of which are free in $\mathsf{At}(\bar{X})(M)$, we may conclude using (Equiv^+) that $\mathtt{QG}_\dagger \vdash (\lambda \bar{X}\bar{Y}.M\bar{Y}) = (\lambda \bar{X}\bar{Y}.\mathsf{At}(\bar{X})(M)\bar{Y})$ and thus $\mathtt{QFC}_\dagger \vdash \mathsf{Stab}_{\bar{\sigma}}(M)$ using ($\beta\eta$). This shows that $\mathsf{Stab}_{\bar{\sigma}}(M)$ has the expected inferential behavior. It is then straightforward to verify that in every model of $\mathtt{QFC}_\dagger$, if $[\![\mathsf{Stab}_{\bar{\sigma}}(M)]\!]^g$ is true, then $[\![M]\!]^g$ is $\bar{\sigma}$-stable.

The notion of stability proper is likely not expressible in the object-language. Expressing stability would require somehow generalizing over nice type sequences, which cannot be done in the present framework. One could expand the language with additional syncategorematic expressions standing for these notions. But this is not really necessary: the notions of $\bar{\sigma}$-stability turn out to be enough for most purposes.

A case in point are stability-theoretic comprehension principles. It seems difficult to characterize stabilized models in the object language, at least in the purely logical signature Λ. For syncategorematic quantifiers do not behave classically. Moreover, we cannot use the definable classical quantifier in $\mathtt{QFC}_\dagger \oplus \mathtt{QG}_\dagger$ either, since we cannot abstract variables within the scope of Stab from outside its scope.

[39] Kaplan [1995] also suggests a view that fits with this picture.

We can, however, express a weaker condition:

$$\forall x \exists y (\mathsf{Stab}_{\bar{\sigma}}(y) \wedge x \equiv y). \qquad (\mathsf{StabComp})$$

Even though our language cannot express finite stability, a model that validates every instance of (Stabλ) must be such that any *existing* entity is applicatively indiscernible from some finitely-stable entity, which, by applicative indiscernibility, also exists.

7. Conclusion

In this paper, I introduced Quantificationalism, a quantificational analog of Temporalism and Modalism. While one may have expected the logic of Quantificationalism to be straightforward to develop using tools from tense and modal logics, I have shown there are distinctive problems begetting Quantificationalism, which seem to have no analogue in the temporal and modal case.

I have then developed a comprehensive higher-order framework in which a logic for quantificationalism can be articulated and shown to be consistent, overcoming these problems. The key idea behind the framework is the treatment of domain specifiers as genuinely syncategorematic operations, which do not stand for higher-order entities but rather describe how to combine higher-order entities via the execution of metaphysical substitutions.

The resulting picture is an admittedly exotic one—where distinct entities can share all their higher-order properties and higher-order existence is freely recombinable. But the space of philosophically promising applications of quantificationalism is rich, and I hope the present framework will prove its worth by serving as foundations for such endeavors.

References

Bacon, A. [2013]. Quantificational Logic and Empty Names. *Philosophers' Imprint*, 13.

——— [2018]. Tense and Relativity. *Noûs*, 52(3):667–696.

——— [2019]. Substitution Structures. *Journal of Philosophical Logic*, 48(6):1017–1075.

——— [2021]. Opacity and Paradox. Routledge.

——— [2023]. *A Philosophical Introduction to Higher-order Logics*. Oxford University Press.

——— [2024]. Mathematical Modality: An Investigation in Higher-order Logic. *Journal of Philosophical Logic*, 53(1):131–179.

Bacon, A. and Dorr, C. [2024]. Classicism. In P. Fritz and N. K. Jones, eds., *Higher-order Metaphysics*. Oxford University Press.

Bacon, A., Hawthorne, J., and Uzquiano, G. [2016]. Higher-Order Free Logic and the Prior-Kaplan Paradox. *Canadian Journal of Philosophy*, 46(4-5):493–541.

Bacon, A. and Russell, J. S. [2019]. The Logic of Opacity. *Philosophy and Phenomenological Research*, 99(1):81–114.

Black, M. [1952]. The Identity of Indiscernibles. *Mind*, 61(242):153–164.

Brogaard, B. [2012]. *Transient Truths: An Essay in the Metaphysics of Propositions*. Oxford University Press.

Dorr, C. [2014]. Quantifier Variance and the Collapse Theorems. *The Monist*, 97(4):503–570.

——— [2025]. Higher-Order Quantification and the Elimination of Abstract Objects. *Disputatio*.

Dorr, C. and Goodman, J. [2020]. Diamonds Are Forever. *Noûs*, 54(3):632–665.

Fine, K. [1977]. Properties, Propositions and Sets. *The Journal of Philosophy*, 79(11):565–589.

——— [2010]. Some Puzzles of Ground. *Notre Dame Journal of Formal Logic*, 51(1):97–118.

Goodman, J. [2017]. Reality is Not Structured. *Analysis*, 77(1):43–53.

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

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

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

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

Lewis, D. [1990]. Noneism or Allism? *Mind*, XCIX(393):24–31.

Lewis, D. K. [1980]. A Subjectivist's Guide to Objective Chance. In R. C. Jeffrey, ed., *Studies in Inductive Logic and Probability, Volume II*, pp. 263–293. Berkeley: University of California Press.

Mellor, D. H. [1981]. *Real Time*. Cambridge University Press.

Myhill, J. [1958]. Problems Arising in the Formalization of Intensional Logic. *Logique Et Analyse*, 1(1):78–83.

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

Prior, A. W. and Fine, K. [1977]. *Worlds, Times and Selves*. Duckworth.

Quine, W. V. [1953]. On What There Is. In W. V. Quine, ed., *From a Logical Point of View*, pp. 1–19. Cambridge, Mass.: Harvard University Press.

Russell, B. [1937]. *Principles of Mathematics*. Routledge.

Russell, B., ed. [1940]. *The Philosophy of Logical Atomism*. Routledge.

Russo, E. [Unpublished]. Absolute Generality as Higher-Order Identity.

Schaffer, J. [2012]. Necessitarian Propositions. *Synthese*, 189(1):119–162.

Sider, T. [2001]. *Four Dimensionalism: An Ontology of Persistence and Time*. Oxford University Press.

Stalnaker, R. C. [2011]. *Mere Possibilities: Metaphysical Foundations of Modal Semantics*. Princeton University Press.

Stanley, J. and Szabó, Z. G. [2000]. On Quantifier Domain Restriction. *Mind and Language*, 15(2-3):219–61.

Uzquiano, G. [2019]. Impredicativity and Paradox. *Thought: A Journal of Philosophy*, 8(3):209–221.

Williamson, T. [1988]. Vii*–Equivocation and Existence. *Proceedings of the Aristotelian Society*, 88(1):109–128.

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

——— [2013]. *Modal Logic as Metaphysics*. Oxford University Press.

Wittgenstein, L. [1921]. *Tractatus Logico-Philosophicus (Trans. Pears and Mcguinness).* New York,: Routledge.

Yli-Vakkuri, J. and McCullagh, M., eds. [2017]. *Williamson on Modality*. Routledge.

Appendix A. Soundness and Completeness Results

In this section, I prove the main completeness results of this paper, repeated here for convenience.

Theorem A.1. $\mathtt{Q}_\star$ *is strongly sound and complete with respect to validity over all* $\mathcal{L}_\star$*-models.*

Theorem A.2. $\mathtt{Q}_\dagger$ *is strongly sound and complete with respect to validity over all* $\mathcal{L}_\dagger$*-models.*

I spell out the (somewhat more complicated) proof of Theorem A.2 in full, then sketch the necessary adaptations to obtain a proof of Theorem A.1.

A.1. **Syntactic preliminaries.** In this section I will show that consequence in $\mathtt{Q}_\dagger$ can be characterized by a Gentzen-style deductive system where derivations are *trees*, namely, rooted posets in which every chain is well founded. $\mathtt{Q}_\dagger$ is an infinitary system in the sense that these trees can be *infinitely branching*, though each branch will only have finite height. I use this deductive system to better regiment my soundness proof and to establish a key lemma in my completeness proof, which one would not in general expect to hold in the presence of infinitary inference rules.

A *decorated tree* in $\mathtt{Q}_\dagger$ is a tree each node of which is labeled by a $\mathcal{L}_\dagger$-formula. The formulas labeling the leaves of the tree are called *assumptions*, whereas the formula labeling the root is the *conclusion*. We write

$$\begin{array}{c}\Gamma\\ \vdots\\ P\end{array}$$

to denote a decorated tree with conclusion P and whose assumptions belong to Γ.

A *derivation* in $\mathtt{Q}_\dagger$ is a decorated tree that can be generated through the instructions collected in Figure 6. These instructions should be read as saying: if all the

$$\frac{}{P}\ P \text{ axiom instance} \qquad \frac{\begin{array}{c}\Gamma\\ \vdots\\ P\end{array} \quad \begin{array}{c}\Delta\\ \vdots\\ P \to Q\end{array}}{Q} \qquad \frac{\begin{array}{c}\Gamma\\ \vdots\\ \mathsf{At}(\bar{F})(P)\end{array}}{\mathsf{At}(\bar{F})(\forall x P)}\ x \notin \mathsf{FV}(\bar{F}, P) \qquad \frac{\begin{array}{c}\vdots\\ P\end{array}}{\mathsf{At}(\bar{F})(P)}$$

$$\frac{\begin{array}{c}\Gamma_1\\ \vdots\\ \mathsf{At}(\bar{F})(\mathsf{At}(\bar{X}^{\bar{\sigma}_1})(P))\end{array} \quad \begin{array}{c}\Gamma_2\\ \vdots\\ \mathsf{At}(\bar{F})(\mathsf{At}(\bar{X}^{\bar{\sigma}_2})(P))\end{array} \quad \cdots}{\mathsf{At}(\bar{F})(\blacksquare P)}\ \begin{array}{l}\{\bar{\sigma}_i : i \in \omega\} = \mathit{Nice}(\mathit{Types})\\ \bar{X}^{\bar{\sigma}_i} \notin \mathsf{FV}(\Gamma_i, \bar{F}, P)\end{array}$$

Figure 6. Derivations

decorated trees above the displayed line are derivations, then the result of adding a root below all such derivations, labeled as indicated, is also a derivation.

A derivation with assumptions in Γ and conclusion P is called a derivation *of* P *from* Γ. Write $\Gamma \rhd P$ to mean that there is a derivation of P from Γ.

A $\mathtt{Q}_\dagger$-*theory*, or just *theory* for short, is a set T of terms of type t containing $\mathtt{Q}_\dagger$ that is closed under the rules included in the axiomatization of $\mathtt{Q}_\dagger$. Note that $\mathtt{Q}_\dagger$ is the least theory. It is straightforward to verify that when Γ is a set of $\mathcal{L}_\dagger$-formulas, the least theory containing Γ equals the set of all $\mathcal{L}_\dagger$-formulas P such that $\Gamma \rhd P$. In particular, P is a theorem of $\mathtt{Q}_\dagger$ iff $\varnothing \rhd P$. Intuitively, this is so because the instructions in Figure 6 mirrors the inference rules of $\mathtt{Q}_\dagger$. Notice that (UG) has no counterpart instruction, but as we noted already it can be seen as a special case of (At-UG), which does correspond to an instruction. Thus, we henceforth identify $\rhd$ with $\vdash$ as defined by Figures 2 to 4.

A.2. **Soundness of $\mathtt{Q}_\dagger$.** Let us begin with the soundness component of Theorem A.2.

Lemma A.3. *Whenever $\Gamma \cup \{P\}$ is a set of $\mathcal{L}_\dagger(\Sigma)$-formulas, if $\Gamma \vdash P$, then $\Gamma \models P$.*

Proof. We need to check that every instance of any axiom schema of $\mathtt{Q}_\dagger$ is valid over all $\mathcal{L}_\dagger$-models and that the rules of $\mathtt{Q}_\dagger$ preserve validity over all models. The cases of axiom schemas and rules of $\mathtt{FH}_\dagger$ are routine generalizations of soundness proofs of $\mathtt{FH}_+$ with respect to standard models of higher-order free logic. That every instance of any axiom schema from Figures 3 and 4 is valid in any $\mathcal{L}_\dagger$-model follows from routine arguments based on observations already made throughout the paper. I spell out the arguments to the effect that the rules from Figures 3 and 4 preserve validity over all $\mathcal{L}_\dagger$-models. These are also routine, but perhaps not as immediately apparent.

Let us begin with (At-Nec). Assume P is valid in all $\mathcal{L}_\dagger$ models. Let $\mathfrak{A}$ be any $\mathcal{L}_\dagger$-model. Then P is valid in every repointing of $\mathfrak{A}$, which is to say that $\mathfrak{A} \models i_{\bar{\mathbf{f}}}[\![P]\!]^g$ holds for every domain $\bar{\mathbf{f}}$ and variable assignment g. That implies $\mathfrak{A} \models [\![\blacksquare P]\!]^g$ holds for every variable assignment g.

Now consider (At-UG). Assume $P \to \mathsf{At}(\bar{F})(Q)$ is valid in every $\mathcal{L}_\dagger$-model and let $x \notin \mathsf{FV}(P, \bar{F})$. Let $\mathfrak{A}$ be a $\mathcal{L}_\dagger$ model and assume $\mathfrak{A} \models [\![P]\!]^g$ for arbitrary g. Since $x \notin \mathsf{FV}(P, \bar{F})$, also $\mathfrak{A} \models [\![P]\!]^{g[x \mapsto a]}$ for every $\mathbf{a}$ of the right type. Consequently, $\mathfrak{A} \models [\![\mathsf{At}(\bar{F})(P)]\!]^{g[x \mapsto a]}$ for each $\mathbf{a}$. Let $\bar{\mathbf{f}} := [\![\bar{F}]\!]^g$ and observe that $\bar{\mathbf{f}} := [\![\bar{F}]\!]^{g[x \mapsto a]}$ for each $\mathbf{a}$, as $x \notin \mathsf{FV}(\bar{F})$. So, $\mathfrak{A} \models_{\bar{\mathbf{f}}} [\![Q]\!]^{g[x \mapsto a]}$ for each $\mathbf{a}$, in particular for each $\mathbf{a}$ with $\mathfrak{A} \models_{\bar{\mathbf{f}}} App([\![\mathsf{E!}]\!]^{g[x \mapsto a]}, \mathbf{a})$. This implies $\mathfrak{A} \models_{\bar{\mathbf{f}}} [\![\forall x Q]\!]^{g[x \mapsto a]}$ and in turn $\mathfrak{A} \models_{\bar{\mathbf{f}}} [\![\forall x Q]\!]^g$ because $x \notin \mathsf{FV}(\forall x Q)$. Thus, indeed, $\mathfrak{A} \models [\![\mathsf{At}(\bar{F})(\forall x Q)]\!]^g$.

Finally, consider (Surj). Assume that $P \to \mathsf{At}(\bar{F})(\mathsf{At}(\bar{X})(Q))$ is valid in every $\mathcal{L}_\dagger$-model any nice term sequence consisting of variables $\bar{X} : \bar{\sigma}$ with $\bar{X} \notin \mathsf{FV}(P, Q, \bar{F})$, for all nice type sequences $\bar{\sigma}$. Let $\mathfrak{A}$ be a $\mathcal{L}_\dagger$-model and assume $\mathfrak{A} \models [\![P]\!]^g$ for arbitrary g. Let $\bar{X} : \bar{\sigma}$ be arbitrary as above and let $\bar{\mathbf{f}} := [\![\bar{F}]\!]^g$. Then also $\mathfrak{A} \models [\![P]\!]^h$ for every h that differs from g at most on $\bar{X}$. So, $\mathfrak{A} \models_{\bar{\mathbf{f}}} [\![\mathsf{At}(\bar{X})(Q)]\!]^h$ for each such h. This implies that $\mathfrak{A} \models_{\bar{\mathbf{f}}} i_{\bar{\mathbf{g}}}[\![Q]\!]^g$ holds for every $\bar{\sigma}$-domain $\bar{\mathbf{g}}$: simply choose the variable assignments h so that $[\![\bar{X}]\!]^h = \bar{\mathbf{g}}$ and observe that $[\![Q]\!]^g = [\![Q]\!]^h$ by the conditions on $\bar{X}$. We can repeat this reasoning for every nice type sequence $\bar{\sigma}$. So, we may infer $\mathfrak{A} \models_{\bar{\mathbf{f}}} [\![\blacksquare Q]\!]^g$, and in turn $\mathfrak{A} \models [\![\mathsf{At}(\bar{F})(\blacksquare Q)]\!]^g$. $\square$

A.3. **Completeness of $\mathtt{Q}_\dagger$.** That was the easy part; let us turn to completeness. I will adopt a standard term model-based strategy to the present setting. The main obstacle we need to overcome is that strategies of this sort typically appeal to compactness properties: if P follows from a set of assumptions Γ, then it follows from a finite subset thereof. However, consequence in $\mathsf{Q}_\dagger$ is not obviously compact, in view of the infinitary rule (Surj).

I will show that the construction can nonetheless be run without appealing to compactness. This has to do with the particular shape of (Surj): while the rule does require infinitely many assumptions, the form of its assumptions is uniform. Using this observation, we can prove that the union of a countable chain of theories each of which conservatively extends its predecessors is itself a theory (Lemma A.6). This claim, in general, can fail in the presence of infinitary rules. It is usually derived as a consequence of compactness, but one can of course just apply it directly insted.[40]

Besides this epicycle, the other novelty of the present proof is its use of a strengthened notion of witness completeness. Normally, in using a construction of this sort to prove completeness (for free logics), one seeks to construct a term model from a maximal consistent theory that is *witnessed* in the sense that for every existentially quantified statement we can find a close term that witnesses the existential. In our setting, we will also want to witness existentials within the scope of domain specifiers, so that for every formula $\mathsf{At}(\bar{F})(\exists x P)$ can be witnessed by some closed term within the scope of that very domain specifier. In addition, we will want to "witness" claim made with $\blacklozenge$: for every term of the form $\mathsf{At}(\bar{F})(\blacklozenge P)$ we must be able to find a nice term sequence $\bar{C}$ made up of closed terms that witnesses the possibility claim within the scope of $\mathsf{At}(\bar{F})$.

Starting with a maximal consistent theory with these properties, we construct a term structure by identifying terms up to substitutional indiscernibility. In this structure, we can identify substitutions with maps that result from lifting functions over terms of the form $M \mapsto \mathsf{At}(\bar{F})(M)$ to equivalence classes.

A theory T is called *consistent* when $T \nvdash \bot$, and *maximal consistent* when it is consistent and has no consistent proper extensions.

Definition A.4 (Witnessing concepts)**.** A theory T is called *strongly witnessed* when the following conditions hold:

(1) *Quantifier witnessing*: for each $\mathsf{At}(\bar{F})(\exists x P) : t$, there is a closed term c such

$$\mathsf{At}(\bar{F})(\exists x P \to (\mathsf{E!}c \wedge P[c/x]))$$

is a theorem of T;

(2) *Domain witnessing*: for each $\mathsf{At}(\bar{F})(\blacklozenge P) : t$, there is a nice term sequence of closed terms $\bar{C}$ such that

$$\mathsf{At}(\bar{F})(\blacklozenge P \to \mathsf{At}(\bar{C})(P))$$

is a theorem of T.

Henceforth we call a $\bar{C}$ as above a *nice closed sequence*.

Lemma A.5. *Let $\bar{a}$ be non-logical constants and $\bar{x}$ be variables whose types match those of $\bar{a}$. If $\Gamma \vdash P$, then the following hold:*

(1) $\Gamma[\bar{x}/\bar{a}] \vdash P[\bar{x}/\bar{a}]$, so long as the substitution can be done without variable capture.

(2) $\Gamma[\bar{a}/\bar{x}] \vdash P[\bar{a}/\bar{x}]$, so long as no constant in $\bar{a}$ occurs in Γ.

[40] I do not know whether compactness actually fails for Q. If (Surj) admits a finitary reformulation, then it does not, in which case this epicycle would not be necessary. I have not settled this question, so the epicycle is necessary.

Proof. Routine induction on derivation length. □

Lemma A.6. *Let $(T_i)_{i\in\omega}$ be a chain of theories with respect to inclusion such that each T_i conservatively extends T_j whenever $j<i$. Then the union T_ω of the T_i's is a theory.*

Proof. We need to show that T_ω is closed under derivability. Assume $T_\omega \vdash P$ and let $\mathtt{D}$ be a derivation that witnesses this. We show that $P \in T_\omega$ by induction on $\mathtt{D}$.

If $\mathtt{D}$ has a single node, this is obvious. Assume inductively that the claim holds for all subderivations of $\mathtt{D}$. The induction steps for finitary rules are all proved analogously; I illustrate with the case of (MP). Suppose $\mathtt{D}$ has the form

$$\dfrac{\begin{matrix}\Gamma\\ \vdots\\ Q\end{matrix} \quad \begin{matrix}\Delta\\ \vdots\\ Q\to P\end{matrix}}{P}\ .$$

By the induction hypothesis, $Q, Q\to P \in T_\omega$. Since only finitely many constants can occur in a finite set of terms, the conservativity assumptions imply there must be a least i such that $Q, Q\to P \in T_i$. Then $Q \in T_i$ because $Q, Q\to P \vdash P$.

Now assume $\mathtt{D}$ has the form

$$\dfrac{\begin{matrix}\Gamma_1\\ \vdots\\ \mathsf{At}(\bar{F})(\mathsf{At}(\bar{X}^{\bar{\sigma}_1})(P))\end{matrix} \quad \begin{matrix}\Gamma_2\\ \vdots\\ \mathsf{At}(\bar{F})(\mathsf{At}(\bar{X}^{\bar{\sigma}_2})(P))\end{matrix} \quad \cdots}{\mathsf{At}(\bar{F})(\blacksquare P)}\ {\scriptstyle \{\bar{\sigma}_i : i\in\omega\} = \mathit{Nice}(\mathit{Types}) \atop \bar{X}^{\bar{\sigma}_i} \notin \mathsf{FV}(\Gamma_i, \bar{F}, P)}$$

By the induction hypothesis, $\mathsf{At}(\bar{F})(\mathsf{At}(\bar{X}^{\bar{\sigma}_i})(P)) \in T_\omega$ for each $i \in \omega$. All these terms differ from one another only in the variable sequences $\bar{X}^{\bar{\sigma}_i}$. In particular, they all contain the same constants. Since a term only contains finitely many constants, there must be a least j such that $\mathsf{At}(\bar{F})(\mathsf{At}(\bar{X}^{\bar{\sigma}_i})(P)) \in T_j$ for all $i \in \omega$. But then $\mathsf{At}(\bar{F})(\blacksquare P) \in T_j$ by (Surj), so $\mathsf{At}(\bar{F})(\blacksquare P) \in T_\omega$. □

Notice that in logics axiomatized via infinitary rules, the union of a countable chain of theories conservatively extending one another need not be a theory. This can considerably complicate completeness proofs. In our case, luckily, Lemma A.6 goes through through because of the special shape of (Surj).

Lemma A.7. *Every consistent theory can be extended to a domain witnessed, consistent theory in a richer signature.*

Proof. Let T be a consistent theory in signature Σ and let Σ^+ expand Σ with countably many constants at each type of the form $\sigma \to t$. Fix an enumeration of the nice closed sequences constructed entirely out of new constants from $\Sigma^+ \setminus \Sigma$. Let $(\bar{F}, P)_1, (\bar{F}, P)_2, \ldots$ enumerate the pairs $(\bar{F}, P)$ such that $\bar{F} \in \mathit{Nice}(\Sigma_+)$ and $P \in \mathcal{L}^t_\dagger(\Sigma^+)$. Write $\bar{F}_i$ and P_i, respectively, for the first and second projection of $(\bar{F}, P)_i$.[41]

[41] Notice that $\bar{F}_1, \bar{F}_2, \ldots$ is not an enumeration of $\mathit{Nice}(\Sigma_+)$, since we will sometimes have $\bar{F}_i = \bar{F}_j$ for $i \neq j$. Likewise for the other projection.

We construct countable chains of signatures $\Sigma_0, \Sigma_1, \dots$ and theories $T_0, T_1, \dots$. Let $\Sigma_0 := \Sigma$ and $T_0 := T$. Assume Σ_n and T_n have been defined in such a way that T_n is a Σ_n-theory that conservatively extends its predecessors. To construct Σ_{n+1} and T_{n+1}, look at $(\bar{F}, P)_{n+1}$. Say that a nice type sequence $\bar{\sigma}$ is *available* when there is no derivation witnessing $\Gamma \vdash \mathsf{At}(\bar{F}_{n+1})(\mathsf{At}(\bar{X})(\neg P_{n+1}))$ for $\Gamma \subseteq T_n$ and $\bar{X} \notin \mathsf{FV}(\Gamma, \bar{F}_{n+1}, P_{n+1})$. For each available $\bar{\sigma}$, choose the least nice closed sequence of fresh constants (with respect to the previously fixed enumeration) indexed by $\bar{\sigma}$ that does not occur in $\bar{F}_{n+1}, P_{n+1}$ and has not been used previously in the construction. Define Σ_{n+1} by adding all constants occurring in some chosen nice closed sequence or other to Σ_n. Then, define T_{n+1} as the least Σ_{n+1} theory expanding T_n with a witnessing axiom

$$\mathsf{DWit}^{\bar{C}}_{n+1} := \mathsf{At}(\bar{F}_{n+1})(\blacklozenge P_{n+1} \to \mathsf{At}(\bar{C})(P_{n+1}))$$

for each chosen $\bar{C}$.

I claim that T_{n+1} conservatively extends T_n. For let $Q \in \mathcal{L}^t_\dagger(\Sigma_n)$ and assume $T_{n+1} \vdash Q$. That is to say

$$T_n, \{\mathsf{DWit}^{\bar{C}}_{n+1} : \bar{C} \text{ chosen } \} \vdash Q. \tag{36}$$

Because no constant in any chosen $\bar{C}$ occurs in T_n nor Q, we can apply Lemma A.5 to infer that

$$T_n, \{\mathsf{DWit}^{\bar{C}}_{n+1}[\bar{X}/\bar{C}] : \bar{C} \text{ chosen } \} \vdash Q, \tag{37}$$

where each $\bar{X}$ is chosen to consist only of fresh variable (relabeling variables if necessary).

Thus, for each available $\bar{\sigma}$ we have

$$T_n, \neg Q \vdash \mathsf{At}(\bar{F}_{n+1})(\mathsf{At}(\bar{X})(\neg P_{n+1})), \tag{38}$$

for some fresh $\bar{X}$ indexed by $\bar{\sigma}$. Further, note that, by the definition of availability and the fact that derivability is monotonic, the above also holds true for each $\bar{\sigma}$ that is not available. This allows us to apply (Surj) to infer

$$T_n, \neg Q \vdash \mathsf{At}(\bar{F}_{n+1})(\blacksquare \neg P_{n+1}). \tag{39}$$

Because $T_n, \neg Q \vdash \mathsf{At}(\bar{F}_{n+1})(\blacklozenge P_{n+1})$ also holds, we conclude

$$T_n, \mathsf{At}(\bar{F}_{n+1})(\blacklozenge P_{n+1} \to \blacklozenge P_{n+1}) \vdash Q. \tag{40}$$

But obviously $\mathsf{At}(\bar{F}_{n+1})(\blacklozenge P_{n+1} \to \blacklozenge P_{n+1})$ is a theorem of T_n, whence $T_n \vdash Q$, as desired.

We have proved that each T_{n+1} conservatively extends T_n. Thus, we may apply Lemma A.6 to infer that the union T_ω of the T_n's is a theory. By construction, T_ω is domain witnessed. Moreover, the conservative extension of a consistent theory is consistent, and T_ω is clearly a conservative extension of all the T_n's. □

Lemma A.8. *Every consistent theory can be extended to a quantifier witnessed, consistent theory in a richer signature.*

Proof. Let T be a consistent theory in signature Σ. Let Σ^+ expand Σ with countably many constants at each type. Fix an enumeration of the new constants in $\Sigma^+ \setminus \Sigma$. Let $(\bar{F}, x, P)_1, (\bar{F}, x, P)_2, \dots$ be an enumeration of the pairs $(\bar{F}, x, P)$ such that $\bar{F} \in Nice(\Sigma_n)$, $x \in Var$ and $P \in \mathcal{L}^t_\dagger(\Sigma_n)$. Write $\bar{F}_i$, x_i, and P_i for the first, second, and third projection of $(\bar{F}, \exists x P)_i$.[42]

As before, we define a chain of signatures $\Sigma_0, \Sigma_1, \ldots$ and a chain of theories $T_0, T_1, \ldots$ with respect to inclusion. Let $T_0 := T$ and $\Sigma_0 := \Sigma$. Assume T_n and Σ_n have been defined and that T_n is a Σ_n theory that conservatively extends its predecessors. Define Σ_{n+1} by adding to Σ_n the least constant c, relative to the previously fixed enumeration, that has the same type as x_{n+1}, does not occur in either $\bar{F}_{n+1}$ or P_{n+1}, and has not been used previously in the construction. Then, define T_{n+1} as the least Σ_{n+1}-theory expanding T_n with a witnessing axiom

$$\mathsf{QWit}_{n+1} := \mathsf{At}(\bar{F}_{n+1})(\exists x_{n+1} P_{n+1} \to (\mathsf{E!}c \wedge P_{n+1}[c/x_{n+1}])).$$

Again, I claim that T_{n+1} conservatively extends T_n. The argument is analogous to that used to prove conservativity in Lemma A.7, using (At-UG) instead of (Surj). Thus, applying (A.5), we infer that the union T_ω of the T_n's is a theory. For the same reasons as before, T_ω is consistent, and by construction it is quantifier witnessed. □

Lemma A.9. *Every consistent theory can be extended to a strongly witnessed theory in a richer language.*

Proof. Let T be a consistent theory. Apply Lemma A.7 and Lemma A.8 at alternate steps to construct a chain of consistent theories $T_0, T_1, \ldots$ such that, for each n even, T_{n+1} is a domain witnessed conservative extension of T_n and T_{n+2} is a quantifier witnessed conservative extension of T_{n+1}. Then the union $T_\omega := \bigcup_{n\in\omega} T_n$ is a consistent theory by Lemma A.6, and both domain and quantifier witnessed by construction. □

Lemma A.10. *Every consistent theory T can be extended to a maximal consistent theory in the same language. Moreover, if T is domain (quantifier) witnessed, then so is any maximal consistent extension thereof in the same language.*

Proof. Given Lemma A.6, the first part of the lemma can be proved by a standard Zorn's lemma argument. The second part is straightforward to verify. □

We are now ready to prove completeness. There is nothing very surprising in the remainder of the argument; most of the non-standard moves were made to prove the lemmas above.

Lemma A.11. *Whenever $\Gamma \cup \{P\}$ is a set of $\mathcal{L}_\dagger(\Sigma)$-formulas, if $\Gamma \models P$, then $\Gamma \vdash P$.*

Proof. Assume $\Gamma \nvdash P$. Let $\bar{x}$ enumerate $\mathsf{FV}(\Gamma \cup \{P\})$. Expand the language with fresh non-logical constants $\bar{a}$ whose types match those of $\bar{x}$. Then $\Gamma[\bar{a}/\bar{x}] \nvdash P[\bar{a}/\bar{x}]$. For if otherwise $\Gamma[\bar{a}/\bar{x}] \vdash P[\bar{a}/\bar{x}]$, then Lemma A.5 would imply $\Gamma[\bar{a}/\bar{x}][\bar{x}/\bar{a}] \vdash P[\bar{a}/\bar{x}][\bar{x}/\bar{a}]$, which is to say $\Gamma \vdash P$.

Thus $\Gamma[\bar{a}/\bar{x}]$ is consistent. Let T be a maximal consistent, strongly witnessed extension of the least theory generated by $\Gamma[\bar{a}/\bar{x}]$. This must exist by Lemmas A.8 and A.10. Clearly, $P[\bar{a}/\bar{x}] \notin T$. Let $\mathcal{L}_\dagger(\Sigma)$ be the language of T.

[42]The same comment made in Section A.3 applies here.

Define $M \sim N$ iff $T \vdash M = N$. This is clearly a congruence with respect to application, in view of (SLL$_\blacksquare$). Let $[M]$ be the equivalence class of M under $\sim$. We define an applicative structure $\mathbf{Tm} := ([\mathfrak{Cl}_\dagger(\Sigma)], [App])$ by letting $[\mathfrak{Cl}_\dagger(\Sigma)]$ be the quotient, through $\sim$, of the set $\mathfrak{Cl}_\dagger(\Sigma)$ of closed terms in $\mathcal{L}_\dagger(\Sigma)$, and

$$[App]([M],[N]) := [MN].$$

Since $\sim$ is a congruence with respect to application, $[App]$ is well defined.

For each $\bar{F} \in \mathit{Nice}_\dagger(\Sigma)$ consisting entirely of closed terms, define a partial function on types $[\bar{F}]$ with $[\bar{F}](\sigma) = [F_\sigma]$ whenever the latter is defined, and undefined otherwise. Let $[Dom]$ be the set of all such partial functions. For each $[\bar{F}] \in [Dom]$, define a substitution $[i]_{[\bar{F}]}$ by putting, for each $[M] \in [\mathfrak{Cl}_\dagger(\Sigma)]$,

$$[i]_{[\bar{F}]}([M]) := [\mathsf{At}(\bar{F})(M)].$$

Let I be the set of such substitutions. It forms a monoid $\mathbf{I}$ under the operation

$$[i]_{[\bar{F}]}[\circ][i]_{[\bar{G}]} := [i]_{[\bar{F}\circ\bar{G}]}.$$

This is easily seen by applying (Id) and (CompG). Thus, $(\mathbf{I}, \mathbf{Tm}, \mathit{Sub})$ is a substitution structure, with Sub identified with function application.

Moreover, $\mathfrak{Tm} := (\mathbf{I}, \mathbf{Tm}, \mathit{Sub}, [i])$ is a quantificational substitution structure. Using (Id), we can see that the identity unit σ-domain exists and coincides with $[\mathsf{E!}_\sigma]$. Further, (Composition) and (Identity) hold because T contains every instance of (Id) and (CompG). Since, in $\mathtt{Q}_\dagger$, the claim $M \equiv N$ is weaker than $M = N$, it follows that $\mathfrak{Tm}$ is not regular. However, it is clearly quasi-Leibnizian, as we identified elements $[M], [N]$ precisely when $T \vdash (\blacksquare M \equiv N)$.

Let us now turn $\mathfrak{Tm}$ into a model. When g is a variable assignment on $\mathfrak{Tm}$, define

$$[\![M]\!]^g := [M[\bar{c}/\bar{x}]],$$

where $\bar{x} := (x_1, \dots, x_n)$ are all the free variables in M and $\bar{c} := (c_1, \dots, c_n)$ is such that $g(x_i) = [c_i]$ for each x_i. It is routine to verify that $[\![\cdot]\!]^g$ is a stable interpretation. Finally, for $[P] \in [\mathfrak{Cl}_\dagger(\Sigma)]^t$, define

$$\mathfrak{Tm} \models [P] :\Longleftrightarrow T \vdash P.$$

The result is a model. That (1) and (2) from Definition 5.1 hold follows from the fact that T contains every instance of (Taut). That (3) holds follows because T contains every instance of (WLL).

For (4), assume $\mathfrak{Tm} \models [i]_{[\bar{F}]}[\![\forall x P]\!]^g$. Then $\mathfrak{Tm} \models [i]_{[\bar{F}]}[\forall x P[\bar{c}/\bar{y}]]$, with $\bar{c}$ as in the definition of $[\![\cdot]\!]^g$ above. This is equivalent to $T \vdash \mathsf{At}(\bar{F})(\forall x P[\bar{c}/\bar{y}])$. Take $[a]$ with $\mathfrak{Tm} \models [i]_{[\bar{F}]}([App]([\mathsf{E!}],[a]))$, i.e., $T \vdash \mathsf{At}(\bar{F})(\mathsf{E!}a)$. Using (FrUI), At-Nec, (App) (Stab$\heartsuit$) and (WLL), this implies $T \vdash \mathsf{At}(\bar{F})(P[a\bar{c}/x\bar{y}])$ which is equivalent to $\mathfrak{Tm} \models [i]_{[\bar{F}]}[\![P]\!]^{g[x\mapsto[a]]}$. Conversely, assume $\mathfrak{Tm} \not\models [i]_{[\bar{F}]}[\![\forall x P]\!]^g$. Since T is maximal consistent it follows that $T \vdash \mathsf{At}(\bar{F})(\exists x \neg P)$. Since T is strongly witnessed and consistent it follows that $T \vdash \mathsf{At}(\bar{F})(\mathsf{E!}a \wedge \neg P[\bar{c}a/\bar{y}x])$ for some closed term a. Thus, we have found $[a]$ with $\mathfrak{Tm} \models [i]_{[\bar{F}]}(App([\mathsf{E!}],[a]))$ and $\mathfrak{Tm} \not\models [i]_{[\bar{F}]}[\![P]\!]^{g[x\mapsto[a]]}$. (5) is proved analogously.

For (6), assume $\mathfrak{Tm} \models [i]_{[\bar{F}]}[\![\blacksquare P]\!]^g$. Then $T \vdash \mathsf{At}(\bar{F})(\blacksquare P[\bar{c}/\bar{y}])$. If $[\bar{G}] \in [Dom]$, then $T \vdash \mathsf{At}(\bar{F})(\mathsf{At}(\bar{G})(P[\bar{c}/\bar{y}]))$ can be easily derived using (Master), (At-Nec), (App) and (Stab$\heartsuit$). This shows $\mathfrak{Tm} \models ([i]_{[\bar{F}]} \circ [i]_{[\bar{G}]})[\![P]\!]^g$. Conversely, assume $\mathfrak{Tm} \not\models [i]_{[\bar{F}]}[\![\blacksquare P]\!]^g$. That means $T \nvdash \mathsf{At}(\bar{F})(\blacksquare P[\bar{c}/\bar{y}])$, so $T \vdash \mathsf{At}(\bar{F})(\blacklozenge\neg P[\bar{c}/\bar{y}])$ because T is maximal consistent. Since T is domain witnessed, there must be some nice closed sequence $\bar{G}$ such that $T \vdash \mathsf{At}(\bar{F})(\mathsf{At}(\bar{G})(\neg P))$. This shows that $\mathfrak{Tm} \not\models ([i]_{[\bar{F}]} \circ [i]_{[\bar{G}]})[\![P]\!]^g$, as desired.

We have shown that $\mathfrak{Tm}$ with interpretation and satisfaction as just defined, is a model. Now, consider any variable assignment g with $g(\bar{x}) = \bar{a}$, where $\bar{x}$ and $\bar{a}$ are chosen as in the beginning of the proof. Then $\mathfrak{Tm} \models [\![Q]\!]^g$ for each $Q \in \Gamma$ and $\mathfrak{Tm} \not\models [\![P]\!]^g$, as desired. □

This completes our proof of Theorem A.2.

A.4. **Soundness and completeness of $\mathtt{Q}_\star$.** The proof of Theorem A.1 follows the same blueprint. Write $\vdash_*$ for derivability in $\mathtt{Q}_\star$—the relation defined by Figures 2 and 3 taking only instances from $\mathcal{L}_\star$. It is straightforward to verify that $\Gamma \vdash_* P$ iff there is a derivation of P from Γ that uses only $\mathcal{L}_\star$-formulas.

Thus, the soundness proof for $\mathtt{Q}_\dagger$ already contains a soundness proof for $\mathtt{Q}_\star$: no derivation containing only $\mathcal{L}_\star$-formulas can use the rule (Surj), so we can just skip that case in the induction of derivations and work with $\mathcal{L}_\star$-models rather than $\mathcal{L}_\dagger$-models.

As for completeness, the proof can be streamlined somewhat. Since derivations that do not use (Surj) are always finite, it follows that consequence in $\mathtt{Q}_\star$ is compact: $\Gamma \vdash_* P$ implies that $\Gamma' \vdash_* P$ for some finite subset $\Gamma' \subseteq \Gamma$. This has an analog of Lemma A.6 as a straightforward consequence. The rest of the proof proceeds the same way, though we do not need to go through domain witnessed theories since $\blacksquare$ is no longer in the language.