Logic and Logical Philosophy
Volume 30 (2021), 227–280
DOI: 10.12775/LLP.2020.021

Oliver Tatton-Brown

Semantics and Completeness for Schematic Logic

Abstract. This paper gives a semantics for schematic logic, proving soundness and completeness. The argument for soundness is carried out in ontologically innocent fashion, relying only on the existence of formulae which are actually written down in the course of a derivation in the logic. This makes the logic available to a nominalist, even a nominalist who does not wish to rely on modal notions, and who accepts the possibility that the universe may in fact be finite.

Keywords: logic; nominalist; schematic logic; semantics; completeness

1. Introduction

Physics still gives us no guarantee that the universe is infinite. We do not know whether the universe will last indefinitely, and the assumption that spacetime is continuous is a useful one but not one we can be sure is true (see Maddy, 1997, pp. 146–152, for a careful discussion). This presents a problem for a nominalist — one who does not believe in abstract entities. As discussed in (Burgess and Rosen, 1997, p. 83) there are two ways nominalists normally respond to this: to either resort to talk of what is possible, or to bank on the assumption that the universe is in fact infinite in some respect (such as due to space being continuous).

Each of these options has its downsides. Regarding the first option, there are similar controversies around the status of what isn't but could be as around abstract objects, in particular concerning how we can know about what could be if it is never to be found in our universe. Regarding the second, to base one's philosophy on a physical hypothesis which

Received January 14, 2020. *Revised* October 28, 2020. *Published online* November 3, 2020

may (not implausibly) turn out to be false would be better avoided if possible.[1]

This paper avoids the dilemma for the nominalist by investigating how far one can go without taking either option: by accepting that our concrete world may be all there is, and that it may be finite. In this we follow in the footsteps of the original (modern) nominalists Quine and Goodman. As Quine says:

> [...] long ago, Goodman and I got what we could in the way of mathematics, or more directly metamathematics, on the basis of a nominalist ontology and without assuming an infinite universe. We could not get enough to satisfy us. But we would not for a moment have considered enlisting the aid of modalities. The cure would in our view have been much worse than the disease. (Quine, 1999, p. 397)

We will see that even on this restrictive possibility, we can justify a certain fragment of second order logic known as schematic logic. This is a fragment of predicative second order logic which in effect allows one to make predicative Π^1_1 statements (but not Σ^1_1 statements or Π^1_n statements for $n > 1$). We give an ontologically innocent semantics for the logic, and prove a completeness theorem — the first time a completeness theorem has been proved for it, to the author's knowledge.

It would not be difficult to give a substitutional semantics for the logic — it is a fragment of predicative second order logic. However that would require belief in an infinity of linguistic entities to serve as substitutions. Here we show that we can do one better, and use the logic while only relying on the existence of formulae that we actually write down (in the course of a derivation).

The semantics is inspired by remarks of Lavine's (inspired in turn by remarks of Hilbert: see Lavine, 1998, pp. 189–203) about the use of variables as placeholders, which can later be taken to mean other expressions. As he says for the case of schematic arithmetic,

> If 'c' stands for a numeral, then, given a proof that $1 + c = c + 1$, one does not infer $1 + 3 = 3 + 1$ by substituting '3' for 'c'. The letter 'c' is not a variable. What one does instead is specify that c is the

[1] This is not at odds with the strategy of for instance (Field, 1980). When one is trying to nominalise a scientific theory that involves infinitely many objects, it is reasonable to give a nominalist alternative that also involves infinitely many objects. But it is preferable not to rest one's philosophy on the assumption that some such theory (involving infinitely many objects) is correct.

> numeral '3'. That amounts to substituting '3' for 'c' in the entire proof of $1 + c = c + 1$, which results in a proof of $1 + 3 = 3 + 1$ in which 'c' does not occur.

However Lavine never went into the details of how this connects to and justifies the formalism, for instance why one would be able to deduce multiple different substitution instances of a statement such as $1 + c = c+1$. Additionally his approach does not quite work for the second order case, where one substitutes a formula $\psi(x_1, \ldots, x_n)$ for occurrences of a placeholder P: there may be collisions between quantifiers and variables in ψ and in any formula containing P. Instead, we work out a semantics based on taking a placeholder P to mean a given open formula. We prove a soundness and completeness theorem for the logic in terms of this semantics. Though we address the second order case, one could give a similar (and much simpler) account of the use of first order schematic variables along the lines given here.

This schematic logic is not the only logic available to a nominalist. They may accept plural logic, as using the quantifier "there are" can be argued not to rely on any abstract entities, or indeed any entities beyond the objects the first order variables range over. This allows an interpretation of monadic second order logic, but does not allow the use of second order variables of arity 2 or higher, unless a pairing function is available; and if the universe is finite then no such pairing function can exist. To use predicative second order logic, based on a substitutional interpretation, would require there to exist an infinity of linguistic entities, which is the kind of assumption we wish to avoid.

Thus the availability of schematic logic is an advance for a nominalist, allowing them an otherwise unavailable portion of second order reasoning. If they also accept plural logic, then they can combine the two, allowing them to make statements which in effect have arbitrary complexity in terms of first order and monadic second order quantifiers, and a string of leading universal second order quantifiers for variables of arity at least 2.

The semantics and completeness theorem also has implications beyond nominalism. One is that since schematic logic is ontologically innocent it becomes available as an arena for formalizing mathematics at minimal cost, and does a better job of formalizing some aspects of mathematical practice than first order logic does. When working in first order logic one has to regard certain statements as axiom schemes or theorem schemes; so in fact one is not working in first order logic at all, but in a metatheory for it (where one can make statements about these infinite lists of axioms or theorems). In schematic logic these statements are handled natively as axioms and theorems, as they should be. Furthermore since the logic is ontologically innocent and has a complete semantics, it avoids the drawbacks of second order logic as a candidate setting for mathematics. Its credentials in this regard are discussed in Section 9.

Also, a number of authors have taken schematic logic to be useful when discussing issues of determinacy: Lavine (1998, pp. 224–240) advocates a schematic second order set theory as a foundation for mathematics, which (he argues) avoids the problem of scepticism about non standard models; McGee (1997, pp. 56–62) gives another argument using schematic logic which aims to rule out non standard models of arithmetic, and of set theory; and Parsons (2007, pp. 290–293) discusses unpublished work of Lavine along similar lines for the case of arithmetic. In all these cases the arguments would be aided by an interpretation of schematic logic which does not rely on substitution instances in some language, since the point of the schematic variables is to capture a kind of open-endedness, not limited to any particular language. The interpretation of schematic logic given here naturally fits with these ideas of open-endedness, and the entailment relation for the logic (with respect to which it is complete) has this open-endedness built in. These points are discussed in Section 10.

2. Motivating idea

Consider the following argument.

> Suppose that everything which is austerulous is boscaresque. Suppose also that everything which is boscaresque is caprizant. Then if something is austerulous, it is boscaresque; and since it is boscaresque, it is caprizant. Thus everything which is austerulous is caprizant. In conclusion, if everything which is austerulous is boscaresque, and everything which is boscaresque is caprizant, then everything which is austerulous is caprizant.

If the words involved are meaningful, this is a valid argument, and to recognise it as valid one does not need to know what "austerulous", "boscaresque" and "caprizant" mean. If we made the argument to someone, and they were confused and asked for the meanings of these terms, we could reply "It doesn't matter. They are adjectives" (and be justified in doing so).

With the great majority of words we use, there will be subtleties to their meaning that we do not grasp. Even an apparently straightforward word like "blue" will have a much more nuanced meaning to someone with a well developed understanding of optics and the workings of vision in the brain than to most people. That does not make the rest of us any less able to use the word as we do. To use it in any particular case, we need only know enough about its meaning to use it correctly in that case. In some cases, like in the above argument, the knowledge required reduces to a bare minimum: their grammatical category. Additional information might be interesting, and might make us better able to appreciate and use the conclusion, but is not required to follow the argument itself.

Thus we can say the following.

> Suppose that everything which is A is B. Suppose also that everything which is B is C. Then if something is A, it is B; and since it is B, it is C. Thus everything which is A is C. In conclusion, if everything which is A is B, and everything which is B is C, then everything which is A is C. Oh, and by the way, by A I mean "scary", by B I mean "in Australia", and by C I mean "far away".

This is a valid argument. Nothing is lost by using A, B and C as adjectives meaning certain longer adjectives. One can follow the argument as it is given, understanding that A, B and C are adjectives from the context, and not needing to know their meaning beyond that: then, once their meaning is given, we can understand the conclusion more fully.

We could even write down the main body of the argument

> Suppose that everything which is A is B. Suppose also that everything which is B is C. Then if something is A, it is B; and since it is B, it is C. Thus everything which is A is C. In conclusion, if everything which is A is B, and everything which is B is C, then everything which is A is C.

one day on a piece of paper, and leave it there; then on a later day, even on a different piece of paper, write down

> Oh, and by the way, by A I mean "scary", by B I mean "in Australia", and by C I mean "far away".

Thus we would be able to conclude that

> If everything which is scary is in Australia, and everything which is in Australia is far away, then everything which is scary is far away.

We might subsequently want to reach a similar conclusion but with different A, B and C. We could at this point find our original piece of paper with the main body of the argument on it and photocopy it; then write

> Oh, and by the way, by A I mean "gold plated", by B I mean "overpriced", and by C I mean "desirable".

Thus we could conclude that

> If everything which is gold plated is overpriced, and everything which is overpriced is desirable, then everything which is gold plated is desirable.

Of course the photocopying of the original piece of paper is of no epistemic significance. We could with equal validity, instead of photocopying it, simply say

> Earlier I wrote down an argument using A, B and C, and later decided that by A I meant "scary", by B I meant "in Australia" and by C I meant "far away". However that is no longer the case: now by A I mean "gold plated", by B I mean "overpriced" and by C I mean "desirable".

Thus we could again conclude that

> If everything which is gold plated is overpriced, and everything which is overpriced is desirable, then everything which is gold plated is desirable.

Having thus drawn a second conclusion from the main argument, by reinterpreting A, B, and C, we need not see the original conclusion (concerning scary things and Australia) as being in any way undermined. Who is to say that we cannot make two different arguments at two different times, the first meaning one thing by A, B and C and the second meaning another, with these two arguments just happening to employ a common sequence of symbols (the main body of the argument) previously written down on a piece of paper?

Note that in carrying out an argument like this there is no mention of thinking in advance of some particular totality of things for A, B and C to range over; we simply write down an argument in which they appear, and later decide to regard them as meaning certain things.

The ability to make an argument employing certain placeholders — like A, B and C in the above — and later decide to mean certain things by them, is the basic idea motivating schematic logic here. Schematic logic is first order logic supplemented by placeholder variables P, Q, ..., each of fixed arity:[2] the arity is the number of terms which go together with a placeholder variable to give an atomic formula, and we can signify that P has arity n by writing P^n if the context does not make this clear. The distinctive feature of the logic is the substitution rule for the placeholder variables. This allows us to deduce $\phi[\psi(x_1, \dots, x_n)|P^n]$ from ϕ, where $\phi[\psi(x_1, \dots, x_n)|P]$ is ϕ but with $P(x_1, \dots, x_n)$ "replaced by the formula $\psi(x_1, \dots, x_n)$". In other words, each occurrence of $P(t_1, \dots, t_n)$ in ϕ (with $t_1, \dots, t_n$ terms) is replaced by $\psi[t_1|x_1, \dots, t_n|x_n]$.

The idea is that when we carry out a suitable derivation in schematic logic, we can take ourselves to be implicitly arguing in the fashion described above. We use placeholder letters P, Q and so on which are initially uninterpreted, and are then free at a later point in the derivation to take the letter P to have a particular meaning — for instance that $P(x_1, \dots, x_n)$ says that $\psi(x_1, \dots, x_n)$, which justifies us in deducing $\phi[\psi(x_1, \dots, x_n)|P]$ from ϕ. In line with the above discussion we are still free to subsequently take P to mean something else; so further down in the derivation we could take $P(y_1, \dots, y_n)$ to say that $\theta(y_1, \dots, y_n)$, and deduce $\phi[\theta(y_1, \dots, y_n)|P]$ as another consequence.

As we go we will formalize this and see how it leads to a natural justification for the use of schematic logic.

Before proceeding we will quickly address one worry about the idea of arguing using placeholders with unspecified meanings, and then stipulating meanings for them. The worry is that some meanings that the placeholders could be given might lead to invalid conclusions. For instance if we give an argument involving repeated *modus ponens* using placeholders, and then take the placeholders to mean something involving a vague predicate such as "small" or "heap", we may run into dubious conclusions. This is a reasonable point, but is not really an objection to the use of placeholders as such. Once we become wary of carrying out re-

[2] They are placeholder *variables* in that we can vary what we take them to mean.

peated *modus ponens* involving vague predicates, then for any argument involving repeated *modus ponens* to be valid we should really phrase it more carefully: maybe prefacing it with a premise that the notions involved are sharp (do not have vagueness about what they apply to).[3] We can do exactly the same for an argument involving placeholders; for instance we could say

> All predicates in A, B, C, D, E are sharp. We have that A, and that if A then B, and that if B then C, and that if C then D, and that if D then E. Thus by *modus ponens*, B; again by *modus ponens*, C; again by *modus ponens*, D; and finally by *modus ponens*, E.

Having given this placeholder argument with the sharpness premise, we can only take each of $A, \ldots, E$ to mean something involving sharp predicates; otherwise the sharpness premise will not be satisfied. In this way our response to problems of vagueness for placeholder arguments can be exactly the same as the response for arguments not involving placeholders. The same goes for other troublesome semantic issues, such as self reference.

3. Meaning stipulations

The basic idea behind schematic logic here is that we can make a statement ϕ which contains placeholder variables $P_1, \ldots, P_n$ and later take these placeholder variables to have certain meanings (which we stipulate at that point). To prove a soundness theorem in these terms for the logic we need to formally represent this situation, of taking P_i in ϕ to have some given meaning. We do that in this section, defining the notion of *unparametrized meaning stipulation*, and then the more general notion of *parametrized meaning stipulation*. We will show how to adjust the definition of satisfaction to use these notions. We will use λ, μ, $\ldots$ to denote meaning stipulations.

First, more about the syntax of schematic logic. The signature of a language in schematic logic is the same as in the first order case (we can allow languages with or without equality), and terms are formed as in the first order case. For every arity $n \geqslant 0$, we have a countably infinite stock of placeholder variables of that arity. We usually signify these

[3] If one is an epistemicist then this is not needed.

variables using capital letters P, Q, ... Applying a placeholder variable P of arity n to terms t_1, ..., t_n gives an atomic formula $P(t_1, \dots, t_n)$. Atomic formulae are also given by applying relation symbols to terms in the usual way. Further formulae are built out of atomic formulae as usual by conjunction, disjunction, implication, negation and (first order) quantification. We use $\mathrm{FV}(\phi)$ to denote the set of free first order variables of ϕ, and $\mathrm{Plc}(\phi)$ to denote the set of its placeholder variables.

To simplify the argument, we will only formalize the notion of "meaning stipulation" for a situation in which a structure for the language is given: the idea being that we only need to provide for a situation in which our language actually has some particular interpretation (and more generality than this is superfluous). One could take a different approach if desired. A structure for the language is the same as in the first order case (a set equipped with constants, functions and relations corresponding to the symbols in the signature of the language). We will use variables a, b, ... to range over the structure's base set.

Now for the formalization of the notion of "meaning stipulation". If A is a structure for the language, an **unparametrized meaning stipulation over** A is a function μ such that:

- $\mathrm{Domain}(\mu)$ is a finite set S of placeholder variables
- For each Q in S, if n is the arity of Q then $\mu(Q)$ is a subset of A^n (an n-arity relation on A).

We will often leave out the reference to A when its identity is unimportant or understood.

We say that μ covers a placeholder P if $P \in \mathrm{Domain}(\mu)$, covers a set X of placeholders if $X \subseteq \mathrm{Domain}(\mu)$, and covers a formula ϕ if $\mathrm{Plc}(\phi) \subseteq \mathrm{Domain}(\mu)$. We also phrase this by saying that μ is an unparametrized meaning stipulation for P, X or ϕ respectively.

It is important to be clear that the fact that we define these μ as certain functions, which take subsets or relations on A as their values, does not require us to be Platonists or to believe in any infinite totalities. When one works in a set theoretic metalanguage, every object one defines is technically a set — even the symbols occurring in the signature of an object language are really just sets. This does not require one to actually believe in sets, since the fact that one is using sets to play these roles is inessential: any other kind of object that allowed one to carry out equivalent reasoning could be used instead. Set theory is just a well established and familiar metatheory. Here μ is representing an act, or series of actions, that we take, determining the meanings of the placeholders Q. If we had a theory that allowed direct reasoning about such actions then we could use that in place of set theory.

The values $\mu(Q)$ that μ takes represent the extensions of the properties or relations that are stipulated. It will be clear that the only extensions needed for soundness are those corresponding to meaning stipulations actually made (or that can be taken to be implicitly made) in the course of a derivation in the logic. There is no need to quantify over an infinite totality of meanings, formulae or sets to serve as the value of such a μ.

There will likely be many subsets of A, and relations on A, that could not be obtained in this way (so that not all "unparametrized meaning stipulations for ϕ over A", as defined above, represent meaning stipulations we could actually make). We will see later in Section 7 a narrower but also more complicated definition of *definable meaning stipulation* that could be used throughout instead, making clear that only properties and relations that are explicitly definable are needed.[4] Even if one only worked with definable meaning stipulations, it would still be convenient to use their extensions rather than syntactic representatives. Thinking in terms of extensions helps when setting up ways that the meaning of one formula can depend on meanings given to placeholders in another formula, as we will see in the next section.

To define satisfaction we use first order variable assignments v over a structure A as in the first order case. If t is a term then the interpretation of t in A under a variable assignment v is the same as in the first order case, and is denoted $v(t)$. If $a \in A$ we write $v(a|x)$ for the variable assignment which agrees with v everywhere except maybe at x, where it takes value a.

We now define satisfaction for formulae of schematic logic in terms of unparametrized meaning stipulations in the obvious way. Satisfaction is a relation between a structure A, a variable assignment v over A, a formula ϕ and an unparametrized meaning stipulation μ for ϕ, denoted: $A, v, \mu \vDash \phi$. It is defined by recursion on ϕ in the usual manner, with an extra clause for atomic statements containing placeholder variables. This clause is the obvious one:

$$A, v, \mu \vDash P(t_1, \ldots, t_n) \text{ if and only if } (v(t_1), \ldots, v(t_n)) \in \mu(P).$$

[4] The definition in Section 7 is actually a narrower version of the notion of *parametrized* meaning stipulation, which we will see below.

The clause for other atomic statements is the usual one — μ is irrelevant. In the recursive clauses one just passes μ down to the level below, for instance

$$A, v, \mu \vDash \forall x\, \phi \text{ if and only if for every } a \in A, \text{ we have } A, v(a|x), \mu \vDash \phi.$$

Note that if ϕ' is a subformula of ϕ and μ is an unparametrized meaning stipulation for ϕ then it is also an unparametrized meaning stipulation for ϕ', so the recursion is well defined.

Note that, as in the first order case, the values the variable assignment takes only matters for variables which are free in the formula. If v and u are variable assignments that agree on the free variables of ϕ, then for all μ we have $A, v, \mu \vDash \phi$ if and only if $A, u, \mu \vDash \phi$. If the free variables of ϕ are amongst $x_1, \dots, x_n$ then for each $a_1, \dots, a_n \in A$ we will use $A, x_i \mapsto a_i, \mu \vDash \phi$ to denote the situation that for all variable assignments v with $v(x_i) = a_i$, we have $A, v, \mu \vDash \phi$; or equivalently that there exists a variable assignment v with $v(x_i) = a_i$, such that $A, v, \mu \vDash \phi$.

Similarly, satisfaction only depends on the values a meaning stipulation gives to placeholder variables that actually occur in the formula: if μ and λ are unparametrized meaning stipulations for ϕ which agree on $\mathrm{Plc}(\phi)$ then $A, v, \mu \vDash \phi$ if and only if $A, v, \lambda \vDash \phi$.

Also as in the first order case if the terms t_i are free for x_i in ϕ (no x_i appearing in the scope of a quantifier which binds a variable of t_i) then A, v, $\mu \vDash \phi[t_1|x_1, \dots, t_n|x_n]$ if and only if A, $v(v(t_1)|x_1, \dots, v(t_n)|x_n)$, $\mu \vDash \phi$.

Finally note that if ϕ is actually a placeholder free statement, then the above is equivalent to the normal notion of satisfaction: if we use $\vDash_1$ to denote normal first order satisfaction, then for any A, v and μ we have $A, v, \mu \vDash \phi$ if and only if $A, v \vDash_1 \phi$.

The employment of structures and variable assignments here is similar to in the case of first order logic, so in as much as the use of first order logic is ontologically innocent we have (as far as the use of this notion of satisfaction goes) no reason to think any different of this semantics for schematic logic.

As well as the notion of unparametrized meaning stipulation, we also need the notion of *parametrized meaning stipulation*. This represents a situation in which we have stated what a placeholder means in terms of values that certain parameters can take. For instance we might state that $P(x)$ means that x is a descendant of a particular man, without specifying who this particular man is. We can later finish determining what P means by giving the identity of the man in question. There is no problem with determining the meaning of P in stages like this, and using it while its meaning is only partially determined (as there is no problem using it when it is just a placeholder with completely unspecified meaning). Similarly we can state the meaning of P in terms of a statement ϕ with free variables, in which case the meaning of P depends on the values those variables take, as parameters.

Formally, if A is a structure for the language, a **parametrized meaning stipulation over** A is a function λ with domain A^m for some $m \geqslant 0$, such that for all $a_1, \ldots, a_m$ in A, $\lambda(a_1, \ldots, a_m)$ is an unparametrized meaning stipulation (over A). We also require that for all $a, b \in A^m$, $\lambda(a)$ and $\lambda(b)$ have the same domain — the same finite set of placeholders for which they determine meanings. We say that λ covers a given placeholder if it is in this set. We say that λ covers a set of placeholders if it covers every placeholder in that set, and that it covers a formula if it covers every placeholder in that formula. Again we also phrase these by saying that λ is a parametrized meaning stipulation for the placeholder, set of placeholders or formula respectively.

Thus what meanings λ assigns to placeholders depends on m objects from A, the values taken by parameters for the meanings of the placeholders. This is similar to the way that the meaning (and truth) of a statement with free variables depends on the values those variables take.

Note that parametrized meaning stipulations are in effect a special case of parametrized meaning stipulations, since we have an obvious correspondence between parametrized meaning stipulations with domain A^0 and unparametrized meaning stipulations: if λ has domain A^0 then λ applied to the empty list is an unparametrized meaning stipulation (and we identify these).

Since unparametrized meaning stipulations are essentially a special case of parametrized meaning stipulation, we will often just use the term **meaning stipulation** for parametrized meaning stipulations (which includes unparametrized meaning stipulations).

We extend the notion of satisfaction to parametrized meaning stipulations in a straightforward manner. Let A be a structure, v a variable assignment, ϕ a formula and λ a parametrized meaning stipulation for ϕ over A. Let the domain of λ be A^m. Then $A, v, \lambda \vDash \phi$ if and only if for all $a_1, \ldots, a_m$ in A, $A, v, \lambda(a_1, \ldots, a_m) \vDash \phi$.

Thus in essence we take asserting a statement in which placeholders have parametrized meaning stipulations to be equivalent to asserting the universal closure over the parameters of these placeholders. This is much like how asserting a statement $\phi(x)$, with x free, plays the same role as asserting $\forall x\,\phi(x)$ (as long as x is not free in any premise).

Again for placeholder free ϕ this is equivalent to the normal notion of satisfaction: for any A, v and λ we have $A, v, \lambda \vDash \phi$ if and only if $A, v \vDash_1 \phi$.

4. The substitution rule

The difference between the deductive system of schematic logic and that of first order logic is the presence of a substitution rule for placeholder variables. The idea is that if we have obtained a formula ϕ with a placeholder P, then we can take $P(x_1, \ldots, x_n)$ to mean ψ and substitute ψ for P in ϕ appropriately. We will discuss the exact form this rule takes, seeing two slightly different versions of it. We will then explain how it arises naturally from the motivating idea, and prove a soundness lemma for it in terms of the semantics of the previous section.

4.1. Versions of the rule

It will be convenient to denote a formula ψ with distinguished distinct variables $x_1, \ldots, x_n$ as $\psi(x_1, \ldots, x_n)$. This could be coded for instance as a pair $(\psi, (x_1, \ldots, x_n))$. The x_i are not actually required to occur in ψ. We will also denote a list of free variables $x_1, \ldots, x_n$ by $\vec{x}$, so $\psi(\vec{x})$ denotes a formula ψ with distinguished free variables $x_1, \ldots, x_n$ for some n.

We present two slightly different versions of the substitution rule, which differ in the conditions on when a substitution is valid. The need for conditions here is similar to in the case of $\forall$-elimination: to deduce $\phi[t|x]$ from $\forall x\,\phi$, we need that t be free for x in ϕ, or in other words that no free occurrence of x in ϕ can lie in the scope of a quantifier that binds a variable of t.

The first version of the substitution rule we present is what we call the *narrow* version. This is the version given by Shapiro (2000, pp. 68–69). Suppose ϕ is a formula, P^n a placeholder variable, and $\psi(x_1, \ldots, x_n)$ another formula (possibly containing placeholders) with distinguished variables $x_1, \ldots, x_n$, where n is the arity of P. We say that $\psi(\vec{x})$ is **substitutable for P in ϕ in the narrow sense** if wherever $P(t_1, \ldots, t_n)$

occurs in ϕ, with the t_i any terms, we have that each t_i is free for x_i in ψ (see the previous paragraph), and if also $\psi(\vec{x})$ is **free for P in ϕ in the narrow sense** — by which we mean that if u is a free variable of ψ, then no occurrence of P in ϕ lies in the scope of a quantifier that binds u.

If these conditions are satisfied, then the substitution of $\psi(\vec{x})$ for P in ϕ, written $\phi[\psi(\vec{x})|P]$, is defined (in the narrow sense). We obtain it by replacing every occurrence of $P(t_1, \ldots, t_n)$ in ϕ, for any terms $t_1, \ldots, t_n$, by $\psi[t_1|x_1, \ldots, t_n|x_n]$.

There is an issue with this formulation of the substitution rule however. Consider the placeholder formulation of the axiom of separation,

$$\phi_{\text{sep}} = \forall z\, \exists y\, \forall x\, (x \in y \leftrightarrow (x \in z \wedge P(x))).$$

Suppose we wanted to draw a consequence from this, for instance to form the set of nonempty members of z. We would most naturally do this by taking $P(x)$ to mean that x is nonempty, and thus substituting $\psi(x)$ for P where ψ is "$x \neq \varnothing$". The problem is that by the above formulation of the substitution rule the substitution $\phi_{\text{sep}}[\psi(x)|P]$ is not defined — $\psi(x)$ is not substitutable for P in the narrow sense. This is because $\psi(x)$ is not free for P in ϕ in the narrow sense, since P occurs in the scope of the quantifier $\forall x$, and x is free in ψ.

The above substitution rule is too dogmatic in this respect: nothing "goes wrong" when we substitute $\psi(x)$ for P in ϕ_{sep}. The result of carrying out the substitution (if it was allowed) would be

$$\theta = \forall z\, \exists y\, \forall x\, (x \in y \leftrightarrow (x \in z \wedge (x \neq \varnothing)))$$

which is exactly what is intended. One could obtain θ by a substitution in the narrow sense by taking ψ' to be "$x' \neq \varnothing$" and substituting $\psi'(x')$ for P in ϕ_{sep}, which is a valid substitution in the narrow sense. However having to relabel x in this way is an irritation.

Thus we define a second version of the substitution rule, the *wide* version. The definition of substitutability is the same except that we do not take variables in $\vec{x}$ into account when deciding whether $\psi(\vec{x})$ is free for P in ϕ. Thus we define $\psi(\vec{x})$ to be **substitutable for P in ϕ in the wide sense** if wherever $P(t_1, \ldots, t_n)$ occurs in ϕ, with the t_i any terms, we have that each t_i is free for x_i in ψ, and if also $\psi(\vec{x})$ is **free for P in ϕ in the wide sense**, i.e. if for every free variable u of ψ which is not one of the x_i we have that no occurrence of P in ϕ lies in the scope of a quantifier which binds u. If this is satisfied then the substitution of $\psi(\vec{x})$ for P in ϕ is defined as before by replacing every occurrence of $P(t_1, \ldots, t_n)$ in ϕ, for any terms $t_1, \ldots, t_n$, by $\psi[t_1|x_1, \ldots, t_n|x_n]$. We again denote it by $\phi[\psi(\vec{x})|P]$.

In a sense the difference between these is not important: we can always relabel, as in the above example with ϕ_{sep}.

PROPOSITION 1. *If the substitution of $\psi(\vec{x})$ for P in ϕ is defined in the wide sense, then there is a formula $\psi'(\vec{x'})$ such that the substitution of $\psi'(\vec{x'})$ for P in ϕ is defined in the narrow sense, and such that $\phi[\psi'(\vec{x'})|P] = \phi[\psi(\vec{x})|P]$ (the former defined in either sense, the latter in the wide sense).*

PROOF. Let $\psi(x_1, \ldots, x_n)$ be substitutable for P in ϕ in the wide sense. Let $x'_1, \ldots, x'_n$ be variables not occurring in $\vec{x}$ or ψ or ϕ, and let ψ' be the formula $\psi[x'_1|x_1, \ldots, x'_n|x_n]$. Then for every occurrence of $P(t_1, \ldots, t_n)$ in ϕ, each t_i is free for x_i in ψ, so each t_i is free for x'_i in ψ'. If u is a free variable of ψ' which isn't one of the x'_i then there is no occurrence of P in ϕ in the scope of a quantifier that binds u (by the definition of substitutability in the wide sense); and if u is one of the x'_i, then there is no quantifier binding u in ϕ, so there is certainly no occurrence of P in ϕ in the scope of a quantifier that binds u. Thus $\psi'(\vec{x'})$ is substitutable for P in ϕ in the narrow sense. Finally, $\phi[\psi'(\vec{x'})|P]$ is obtained by replacing every occurrence of $P(t_1, \ldots, t_n)$ in ϕ by $\psi'[t_1|x'_1, \ldots, t_n|x'_n]$, which is just $\psi[t_1|x_1, \ldots, t_n|x_n]$. Thus indeed $\phi[\psi'(\vec{x'})|P] = \phi[\psi(\vec{x})|P]$. $\square$

The converse to this is obviously true since substitutability in the narrow sense implies substitutability in the wide sense. Thus we have that one can obtain the same results by using either version of the substitution rule. However the wide version can definitely be more convenient — it can avoid the need to relabel variables before using the rule, as in the ϕ_{sep} example. This extra flexibility will be helpful at points, for instance when proving the completeness theorem in Section 8 (though of course completeness does not rely on using the wide version of the substitution rule, as it cannot by this lemma).

Thus we will work with the wide version of substitutability and the substitution rule. Soundness results still of course apply to uses of the narrow version of the substitution rule, and if one dislikes the slight extra complication in the definition of the wide version then one can use the narrow version when working in the logic if one wishes. We will sometimes refer to the wide version of the substitution rule as the

single substitution rule to distinguish it from the multiple substitution rule defined below.

We make a few notes about this substitution rule and our use of it. The first is just that when it is not important we will not always mention the list of x_i, and will sometimes just write the substitution as $\phi[\psi|P]$ for brevity.

Secondly, if there are actually no occurrences of P in ϕ, then any ψ is substitutable for P in ϕ and $\phi[\psi(\vec{x})|P]$ is always defined, just being ϕ.

Thirdly, if ψ does not contain Q, and $\psi(\vec{x})$ is substitutable for P in ϕ, then for any $\theta(\vec{y})$ we have that $\theta(\vec{y})$ is substitutable for Q in ϕ if and only if it is substitutable for Q in $\phi[\psi(\vec{x})|P]$. This is clear from the definition of substitutability, since Q is in the scope of the same first order quantifiers in $\phi[\psi(\vec{x})|P]$ as it is in ϕ, and is applied to the same terms in $\phi[\psi(\vec{x})|P]$ as it is in ϕ.

Now we generalize the substitution rule to one for multiple substitutions, as follows. Suppose ϕ is a formula, $P_1, \ldots, P_m$ are distinct placeholder variables, and $\psi_1(x_1^1, \ldots, x_{n_1}^1), \ldots, \psi_n(x_1^m, \ldots, x_{n_m}^m)$ are formulae with distinguished variables. Then the multiple substitution

$$\phi[\psi_1(x_1^1, \ldots, x_{n_1}^1)|P_1, \ldots, \psi_m(x_1^m, \ldots, x_{n_m}^m)|P_m],$$

is defined iff $\phi[\psi_i(x_1^i, \ldots, x_{n_i}^i)|P_i]$ is defined for each i, in which case

$$\phi[\psi_1(x_1^1, \ldots, x_{n_1}^1)|P_1, \ldots, \psi_m(x_1^m, \ldots, x_{n_m}^m)|P_m],$$

is obtained from ϕ by replacing each occurrence of $P_i(t_1, \ldots, t_{n_i})$ in ϕ by $\psi_i[t_1|x_1^i, ..., t_{n_i}|x_{n_i}^i]$, for each i and each $t_1, \ldots, t_{n_i}$. One does not then go on and for instance replace any occurrences of P_j in $\psi_i[t_1|x_1^i, \ldots, t_{n_i}|x_{n_i}^i]$.

We will sometimes write $\vec{\psi}|\vec{P}$ for the set $\{(\psi_1(\vec{x^1}), P_1), \ldots, (\psi_m(\vec{x^m}), P_m)\}$, and use $\phi[\vec{\psi}|\vec{P}]$ as shorthand for $\phi[\psi_1(\vec{x^1})|P_1, \ldots, \psi_m(\vec{x^m})|P_m]$.

Note that if each ψ_i contains none of the P_i, then if $\phi[\psi_1|P_1, \ldots, \psi_m|P_m]$ is defined then so is $\phi[\psi_1|P_1] \ldots [\psi_m|P_m]$, by the note just before this definition of multiple substitutions. In this case the two are equal.

Then we can argue that we do not need to add the multiple substitution rule as a rule in our deductive system, as we can use iteration of the single substitution rule instead.

PROPOSITION 2. *If $\phi[\psi_1(x_1^1, \ldots, x_{n_1}^1)|P_1, \ldots, \psi_m(x_1^m, \ldots, x_{n_m}^m)|P_m]$ can be obtained from ϕ by a multiple substitution then it can be obtained by repeated uses of the single substitution rule.*

Proof. Let $Q_1, \ldots, Q_m$ be placeholders that are distinct from each other and the P_i, that do not occur in ϕ or any of the ψ_i, and such that Q_i has the same arity as P_i. Let ψ'_i be ψ_i with every occurrence of P_i replaced by Q_i. Then for each i, $\psi'_i(\vec{x^i})$ is substitutable for P_i in ϕ since $\psi_i(\vec{x^i})$ is. Thus $\phi[\psi'_1(\vec{x^1})|P_1, \ldots, \psi'_m(\vec{x^m})|P_m]$ is defined, and since each ψ'_i contains none of the P_i we have

$$\phi[\psi'_1(\vec{x^1})|P_1, \ldots, \psi'_m(\vec{x^m})|P_m] = \phi[\psi'_1(\vec{x^1})|P_1] \ldots [\psi'_m(\vec{x^m})|P_m]$$

as noted before the proposition. Thus $\phi[\psi'_1(\vec{x^1})|P_1, \ldots, \psi'_m(\vec{x^m})|P_m]$ can be reached from ϕ by repeated uses of the single substitution rule. Now for each i let $\theta_i = P_i(y^i_1, \ldots y^i_{n_i})$ with the y^i_j any distinct variables. Then there are no quantifiers in θ_i and no free variables beyond the y^i_j so from the definition of the wide version of the substitution rule we have that $\theta_i(\vec{y^i})$ is substitutable for Q_i in any formula (this is one example of the usefulness of the wider version of the substitution rule). Further, for any χ we have that $\chi[\theta_i(\vec{y^i})|Q_i]$ is just $\chi[P_i|Q_i]$, the result of replacing P_i by Q_i everywhere it appears. Substituting each of these $\theta_i(\vec{x^i})$ for Q_i one by one into $\phi[\psi'_1(\vec{x^1})|P_1, \ldots, \psi'_m(\vec{x^m})|P_m]$ gives us $\phi[\psi'_1(\vec{x^1})|P_1, \ldots, \psi'_m(\vec{x^m})|P_m][P_1|Q_1 \ldots P_m|Q_m]$, which is easily seen to be $\phi[\psi_1(\vec{x^1})|P_1, \ldots, \psi_m(\vec{x^m})|P_m]$ as required. □

4.2. Semantics of substitution

We have already seen the idea behind the use of the substitution rule: we make an argument involving a placeholder P with unspecified meaning, and are later free to stipulate P to mean what we like. In the formal setting if P has arity n then we stipulate its meaning in terms of a formula with n free variables, saying for instance that $P(x_1, \ldots, x_n)$ means $\psi(x_1, \ldots, x_n)$. If ϕ is a statement we have derived containing P, this allows us to deduce $\phi[\psi(\vec{x})|P]$.

Importantly we allow that ψ itself — the formula being used to specify P's meaning — can contain placeholder variables. In the same way that we can give an argument in which P's meaning is unspecified, we can give a meaning stipulation for P which uses a placeholder variable Q with currently unspecified meaning. We can then later specify Q's meaning, and thus finish specifying P's meaning. There is nothing to prevent us from giving the meaning of P in stages like this, and nothing to prevent us from using it while its meaning is only partially determined (in the same way there is nothing to prevent us using it when meaning is completely unspecified).

This is much like the situation with parametrized meaning stipulations discussed in the previous section. However the two cases play very different roles when arguing for the soundness of schematic logic, so we distinguish them. When we specify that a placeholder P in ϕ means $\psi(x_1, \ldots, x_n)$, and ψ itself contains placeholder variables, we say that we have given a *meaning dependence of ϕ on ψ*: we have stipulated that the meaning of ϕ depends on the meaning of ψ, so that when we specify the meaning of ψ we will then have also specified the meaning of ϕ.

Formally, if A is a structure, by a **meaning dependence** over A we just mean a function g whose domain and range are sets of meaning stipulations over A. If a is a formula, a placeholder or a set of placeholders, and the same is true of b, then a meaning dependence g is a **meaning dependence of a on b** if g maps every meaning stipulation for b over A to a meaning stipulation for a over A. g embodies how once we have given a (parametrized) meaning stipulation for b, we will have done the same for a.

We represent how the meaning of placeholders in x depends on the meaning of placeholders in y by showing how the extensions of the former meanings depend on the extensions of the latter meanings — i.e. how the values $g(\mu)$ gives to a placeholder depend on the values μ gives to placeholders.

As with the definitions of meaning stipulations, the use of functions here does not require us to believe in Platonism — they just formally represent acts we take. Also, not all such functions actually represent meaning dependencies that we could bring about. We are really only interested in certain examples of meaning dependencies that arise from the use of the substitution rule. We will now work towards defining these.

To start, let P be a placeholder with arity n, and $\psi(x_1, \ldots, x_n)$ a formula (with distinguished variables $x_1, \ldots, x_n$). We will show we can obtain a meaning stipulation for P from a meaning stipulation for ψ, which corresponds to the idea of taking $P(x_1, \ldots, x_n)$ to mean $\psi(x_1, \ldots, x_n)$. Let $y_1, \ldots, y_p$ be variables distinct from each other and the x_i, such that $(\mathrm{FV}(\psi) \setminus \{x_1, \ldots, x_n\}) \subseteq \{y_1, \ldots, y_p\}$. Let λ be a meaning stipulation for ψ, with domain A^q. Then we define a meaning stipulation $\delta^{\vec{y},\lambda}_{\psi(\vec{x})|P}$ for P only with domain A^{p+q} as follows. We define

$\delta^{\vec{y},\lambda}_{\psi(\vec{x})|P}(b_1 \ldots b_p, c_1, \ldots, c_q)(P)$ to be

$$\{(a_1, \ldots, a_n) \in A^n \mid A,\, x_i \mapsto a_i, y_j \mapsto b_j,\, \lambda(c_1, \ldots, c_q) \vDash \psi\}.$$

Note that the value of $\delta^{\vec{y},\lambda}_{\psi(\vec{x})|P}(b_1 \ldots b_p, c_1, \ldots, c_q)(P)$ only depends on those of the b_i that correspond to y_i that actually occur free in ψ. Indeed letting J be the set of j such that y_j occurs free in ψ, if we have that $b_j = b'_j$ for $j \in J$ then $\delta^{\vec{y},\lambda}_{\psi(\vec{x})|P}(b_1 \ldots b_p, c_1, \ldots, c_q)(P) = \delta^{\vec{y},\lambda}_{\psi(\vec{x})|P}(b'_1 \ldots b'_p, c_1, \ldots, c_q)(P)$ for all $c_1, \ldots, c_q$.

This definition exactly matches the informal idea: we have that P applies to $a_1, \ldots, a_n$ if and only if $\psi(x_1, \ldots, x_n)$ holds of them. The resulting meaning stipulation for P is parametrized, with parameters corresponding to the parameters of the meaning stipulation for ψ, and the free variables of ψ. If ϕ is any formula only containing the placeholder P, then $\delta^{\vec{y},\lambda}_{\psi(\vec{x})|P}$ is a parametrized meaning stipulation for ϕ.

We will use these $\delta^{\vec{y},\lambda}_{\psi(\vec{x})|P}$ to obtain meaning dependencies corresponding to the substitution rule. We actually treat the case of multiple substitutions, which is not needed for the soundness theorem, Theorem 10, but is useful for the conservativeness and completeness results in Sections 7 and 8.

So suppose that we have distinct placeholders $P_1^{n_1}, \ldots, P_m^{n_m}$, and formulae $\psi_1(x^1_1, \ldots, x^1_{n_1}), \ldots, \psi_m(x^m_1, \ldots, x^m_{n_m})$. If we have a meaning stipulation λ that covers every ψ_i, then we can take each P_i in some formula ϕ to mean $\psi_i(\vec{x^i})$, and obtain the meaning stipulations $\delta^{\vec{y},\lambda}_{\psi_i(\vec{x^i})|P_i}$ for P_i for each i. We then want to put these together to obtain a meaning stipulation for ϕ. There may be leftover placeholders in ϕ too—placeholders beyond $P_1, \ldots, P_m$. We can potentially use λ to give these meanings too if it covers them.

Thus suppose we have $q \in \mathbb{N}$, $q \geqslant 1$, with variables $y_1, \ldots, y_p$ distinct from each other and all the x^i_j, such that for each i, $(\mathrm{FV}(\psi_i) \setminus \{x^i_1, \ldots x^i_{n_i}\}) \subseteq \{y_1, \ldots, y_p\}$. Then if λ is a meaning stipulation for $\bigcup_{i=1}^m \mathrm{Plc}(\psi_i)$, with domain A^q, with S the set of placeholders it covers, we define $g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)$ to be the meaning stipulation with domain A^{p+q} which covers the set $S \cup \{P_1, \ldots, P_m\}$ of placeholders with

- $g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(b_1 \ldots b_p, c_1, \ldots, c_q)(P_i) = \delta^{\vec{y},\lambda}_{\psi_i(\vec{x^i})|P_i}(b_1 \ldots b_p, c_1, \ldots, c_q)(P_i)$
- $g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(b_1 \ldots b_p, c_1, \ldots, c_q)(Q) = \lambda(c_1, \ldots, c_p)(Q)$ for $Q \in S \setminus \{P_1, \ldots, P_m\}$

Thus $g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)$ is a meaning stipulation for $\{P_1, \ldots, P_m\} \cup \bigcup_{i=1}^m \mathrm{Plc}(\psi_i)$ for each meaning stipulation λ for $\bigcup_{i=1}^m \mathrm{Plc}(\psi_i)$, so $g^{\vec{y}}_{\vec{\psi}|\vec{P}}$ is a meaning dependence of $\{P_1, \ldots, P_m\} \cup \bigcup_{i=1}^m \mathrm{Plc}(\psi_i)$ on $\bigcup_{i=1}^m \mathrm{Plc}(\psi_i)$. In fact these are the only meaning dependencies that we are interested in (so one could give a narrower definition of meaning dependence if one wished).

As with $\delta^{\vec{y},\lambda}_{\psi_i(\vec{x^i})|P_i}(b_1 \ldots b_p, c_1, \ldots, c_q)(P_i)$, the value of

$$g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(b_1 \ldots b_p, c_1, \ldots, c_q)(P_i)$$

only depends on those b_j that correspond to y_j that are actually free in ψ_i. Indeed letting J be the set of j such that y_j occurs free in ψ_i, if we have that $b_j = b'_j$ for $j \in J$ then

$$g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(b_1 \ldots b_p, c_1, \ldots, c_q)(P_i) = g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(b'_1 \ldots b'_p, c_1, \ldots, c_q)(P_i)$$

for all $c_1 \ldots c_q$. We also have that if Q is a placeholder variable that isn't one of the P_i and is covered by $g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)$ then $g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(b_1 \ldots b_p, c_1, \ldots, c_q)(Q)$ only depends on $c_1 \ldots c_q$: for any $b_1 \ldots b_p, b'_1 \ldots b'_p, c_1 \ldots c_q$ we have

$$g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(b_1 \ldots b_p, c_1, \ldots, c_q)(Q) = g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(b'_1 \ldots b'_p, c_1, \ldots, c_q)(Q).$$

We now turn back to the case of a single rather than multiple substitution, which is what's needed for the soundness theorem. When $m = 1$ and $\vec{\psi}|\vec{P}$ is just $\{\psi(\vec{x})|P\}$, we write $g^{\vec{y}}_{\psi(\vec{x})|P}$ for $g^{\vec{y}}_{\vec{\psi}|\vec{P}}$. Now if we use the substitution rule to deduce $\phi[\psi(\vec{x})|P]$ from ϕ, then every placeholder in ϕ is either P or appears in $\phi[\psi(\vec{x})|P]$, and so $g^{\vec{y}}_{\psi(\vec{x})|P}$ is a meaning dependence of ϕ on $\phi[\psi(\vec{x})|P]$. We can take any use of the substitution rule to be implicitly setting up a meaning dependence in this way. So that the meaning dependence is determined by the use of the substitution rule, we will assume some canonical ordering of the first order variables, and let $g_{\psi(\vec{x})|P}$ be $g^{\vec{y}}_{\psi(\vec{x})|P}$ where $y_1, \ldots, y_m$ is the ordering of $\mathrm{FV}(\psi) \setminus \{x_1, \ldots, x_n\}$ according to this canonical order; this $g_{\psi(\vec{x})|P}$ is what we take to be implicit in uses of the substitution rule.

One point about the above that should be made explicit is that when we substitute $\psi(x_1, \ldots, x_n)$ for P in ϕ, we allow that ψ itself can contain the placeholder P. We can then stipulate some other meaning for occurrences of P in ψ, or $\phi[\psi(\vec{x})|P]$, for instance that P in $\phi[\psi(\vec{x})|P]$ means

$\chi(z_1, \ldots, z_n)$. This is not inconsistent. The whole point of placeholders is that we are free to stipulate our own meanings for them, and there is nothing to stop us from giving different meanings to occurrences of a placeholder in different statements (even if one of the statements is used to give a meaning to occurrences of the placeholder in the other). This is not completely unlike how a word can have multiple senses, and in some situations it could even be possible to explain one sense of a word by employing another: "well in the biblical sense, 'know' means, you know, to get to know someone sexually". One could if one wanted avoid occurrences of a placeholder variable in meaning stipulations for other occurrences of the same variable, but it would make the deductive system and the soundness theorem more complicated, and be an unnecessary inconvenience.

We now prove a soundness lemma for the substitution rule in terms of these meaning dependencies $g^{\vec{y}}_{\vec{\psi}|\vec{P}}$.

LEMMA 3. *Suppose that ϕ is a formula, and that $P_1^{n_1}, \ldots, P_m^{n_m}$ are distinct placeholders and $\psi_1(\vec{x^1}), \ldots, \psi_m(\vec{x^m})$ formulae such that $\psi_i(\vec{x^i})$ is substitutable for P_i in ϕ for each i. Let $y_1, \ldots, y_p$ be distinct from each other and all the x^i_j such that for each i, $(FV(\psi_i) \setminus \{x^i_1, \ldots x^i_{n_i}\}) \subseteq \{y_1, \ldots, y_p\}$. Let λ be a meaning stipulation for $(Plc(\phi) \setminus \{P_1, \ldots, P_m\}) \cup \bigcup_i Plc(\psi_i)$ over A. Then for any variable assignment v, we have*

$$A,\, v,\, \lambda(\vec{c}) \vDash \phi[\vec{\psi}|\vec{P}] \text{ iff } A,\, v,\, g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(v(y_1), \ldots, v(y_p), \vec{c}) \vDash \phi.$$

In particular if $A,\, v,\, g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda) \vDash \phi$ then $A,\, v,\, \lambda \vDash \phi[\vec{\psi}|\vec{P}]$.

PROOF. Fix structure A, distinct placeholder variables $P_1^{n_1}, \ldots, P_m^{n_m}$, formulae $\psi_1(\vec{x^1}) \ldots \psi_m(\vec{x^m})$, and meaning stipulation λ for $\psi_1(\vec{x^1}) \ldots \psi_m(\vec{x^m})$. Let the domain of λ be A^q, and fix $c_1, \ldots, c_q \in A$. Let $y_1, \ldots, y_p$ be distinct from each other and all the x^i_j such that for each i, $(\mathrm{FV}(\psi_i) \setminus \{x^i_1, \ldots x^i_{n_i}\}) \subseteq \{y_1, \ldots, y_p\}$. We prove first that for all ϕ in which each $\psi_i(\vec{x^i})$ is substitutable for P_i, and such that λ is a meaning stipulation for $(\mathrm{Plc}(\phi) \setminus \{P_1, \ldots, P_m\}) \cup \bigcup_i \mathrm{Plc}(\psi_i)$, we have

$$A,\, v,\, \lambda(\vec{c}) \vDash \phi[\vec{\psi}|\vec{P}] \text{ iff } A,\, v,\, g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(v(y_1), \ldots, v(y_p), \vec{c}) \vDash \phi.$$

for all variable assignments v over A. The proof is by induction on ϕ.

If ϕ is an atomic formula not containing a placeholder, then this is trivial: $\phi[\psi_1(\vec{x^1})|P_1, \ldots, \psi_m(\vec{x^m})|P_m]$ is just ϕ, and the meaning stipulations in both clauses are irrelevant to its satisfaction.

Suppose that ϕ is an atomic formula $Q(s_1, \ldots, s_k)$ with Q distinct from the P_i. Then $\phi[\psi_1(\vec{x^1})|P_1, \ldots, \psi_m(\vec{x^m})|P_m]$ is just ϕ, so the conclusion follows from the fact that $\lambda(\vec{c})$ and $g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(v(y_1), \ldots, v(y_p), \vec{c})$ agree on all placeholders in ϕ (as noted after the definition of satisfaction in Section 3, if λ and λ' agree on the placeholders in ϕ then $A, v, \lambda \vDash \phi$ if and only if $A, v, \lambda' \vDash \phi$).

Suppose that ϕ is an atomic formula $P_i(t_1, \ldots, t_{n_i})$. Then

$$\phi[\vec{\psi}|\vec{P}] = \psi_i[t_1|x^i_1, \ldots, t_{n_i}|x^i_{n_i}]$$

so A, v, $\lambda(\vec{c}) \vDash \phi[\vec{\psi}|\vec{P}]$ iff A, v, $\lambda(\vec{c}) \vDash \psi_i[t_1|x^i_1, \ldots, t_{n_i}|x^i_{n_i}]$ iff $A, v(v(t_1)|x^i_1, \ldots, v(t_{n_i})|x^i_{n_i})$, $\lambda(\vec{c}) \vDash \psi_i$ iff $(v(t_1), \ldots, v(t_{n_i})) \in \{(a_1, \ldots, a_{n_i}) \in A^{n_i} \mid A,\, x_i \mapsto a_i, y_i \mapsto v(y_i),\, \lambda(\vec{c}) \vDash \psi\}$ iff $(v(t_1), \ldots, v(t_{n_i})) \in g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(v(y_1), \ldots, v(y_p), \vec{c})(P_i)$ iff A, v, $g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(v(y_1), \ldots, v(y_p), \vec{c}) \vDash P(t_1, \ldots, t_{n_i})$ iff A, v, $g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(v(y_1), \ldots, v(y_p), \vec{c}) \vDash \phi$ as desired.

The cases of the propositional connectives are easy.

Suppose that ϕ is of the form $\forall z\, \theta$. Let i be such that P_i does occur in ϕ. Then it is in the scope of a quantifier that binds z, so by the definition of $\psi_i(\vec{x^i})$ being substitutable for P_i in ϕ, if z is free in ψ_i then it is one of the x^i_j. In particular if z is free in ψ_i then it is not one of the y_i. Thus for any $a \in A$, for any of the y_j which are free in ψ_i, we have $v(a|z)(y_j) = v(y_j)$. Thus we have that

$$g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(v(a|z)(y_1), \ldots, v(a|z)(y_p), \vec{c})(P_i)$$

is equal to

$$g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(v(y_1), \ldots, v(y_p), \vec{c})(P_i)$$

by the remark after the definition of $g^{\vec{y}}_{\vec{\psi}|\vec{P}}$. For any Q which occurs in ϕ but is none of the P_i, we have that $g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(v(a|z)(y_1), \ldots, v(a|z)(y_p), \vec{c})$ and $g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(v(y_1), \ldots, v(y_p), \vec{c})$ also agree on Q. Thus they agree on every placeholder in ϕ (and in θ).

Thus

$$\begin{aligned}
&\quad A,\, v,\, \lambda(\vec{c}) \vDash \forall z\, \theta[\vec{\psi}|\vec{P}]\\
&\text{iff for all } a \in A,\ A,\, v(a|z),\, \lambda(\vec{c}) \vDash \theta[\vec{\psi}|\vec{P}]\\
&\text{iff for all } a \in A,\ A,\, v(a|z),\, g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(v(a|z)(y_1), \ldots, v(a|z)(y_p), \vec{c}) \vDash \theta\\
&\quad \text{(by the induction hypothesis for } \theta)\\
&\text{iff for all } a \in A,\ A,\, v(a|z),\, g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(v(y_1), \ldots, v(y_p), \vec{c}) \vDash \theta\\
&\text{iff } A,\, v,\, g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(v(y_1), \ldots, v(y_p), \vec{c}) \vDash \forall z\, \theta
\end{aligned}$$

as desired.

The case of existential quantifiers is similar.

That finishes the induction, giving the first conclusion. The second conclusion now follows immediately. Suppose that $A,\, v,\, g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda) \vDash \phi$. Then for any $\vec{c}$, we have $A,\, v,\, g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda)(v(y_1), \ldots, v(y_p), \vec{c}) \vDash \phi$, so that we obtain that $A,\, v,\, \lambda(\vec{c}) \vDash \phi[\vec{\psi}|\vec{P}]$. Thus $A,\, v,\, \lambda \vDash \phi[\vec{\psi}|\vec{P}]$, as required. □

5. The deductive system

The deductive system in schematic logic is a normal first order deductive system supplemented by the substitution rule for placeholder variables. It is formulated here as a natural deduction system. The syntax of formulae of schematic logic is as described at the start of Section 3.

We use a natural deduction system rather than a Hilbert style system like that of (Shapiro, 2000, pp. 68–69) since we want a deductive system which one could plausibly work in. To write out an actual derivation in a standard Hilbert style deductive system is a tremendous chore; instead one standardly works in a metatheory, using the deduction theorem to demonstrate that a derivation exists. The idea behind schematic logic here is that we can see ourselves as licensed to make substitutions based on meaning stipulations that are implicit in uses of the substitution rule. This would be undermined if we did not work in the logic itself and instead had to appeal to a metatheorem to demonstrate the existence of a derivation. Also, if one was using a metatheorem to demonstrate that a derivation existed, rather than constructing an actual derivation oneself, then one would have to have some way of understanding this kind of existence claim about derivations. This is potentially awkward for a nonmodal nominalist. One could try to use a nominalist approach to syntax, but this would require limiting one's talk to derivations of at most a certain finite length (if the universe was finite). It is better to avoid these complications and use a natural deduction system. There is still of course a question about how the formal derivations relate to the kind of informal derivations one would actually produce in practice, but this is a less threatening question for natural deduction than for Hilbert style deductive systems.

The relation between our natural deduction system and the Hilbert style system of (Shapiro, 2000, pp. 68–69) is discussed later in the section.

The main subtlety about formulating schematic logic in natural deduction is that there are two roles premises can play. One can treat a premise containing schematic variables in a similar way to how one would treat a premise in second order logic with free second order variables. Otherwise one can treat a premise containing schematic variables in a similar way to how one would treat its universal closure in second order logic. We call the former a specific premise, and the latter a general premise. The fact that there are these two different roles premises can play in natural deduction for schematic logic was pointed out by Heck (2011, pp. 274–282). He did not go into the details of how the deductive system would work however. We spell the details out here so we can prove our ontologically innocent soundness theorem, and later the completeness theorem.

We saw examples of specific premises earlier in Section 2, where we said:

> Suppose that everything which is A is B. Suppose also that everything which is B is C. Then if something is A, it is B; and since it is B, it is C. Thus everything which is A is C. In conclusion, if everything which is A is B, and everything which is B is C, then everything which is A is C.

In this case A, B and C have unspecified meaning throughout the argument. As a result we can take them to have meanings of our choice afterwards, and deduce as a consequence what we obtain by substituting these meanings for A, B and C in the conclusion of the argument (as we did back in Section 2). Note that in this case $\to$-introduction is valid as usual — going from $\Gamma, \phi \vdash \psi$ to $\Gamma \vdash \phi \to \psi$. However we cannot use the substitution rule on the placeholders in the course of the argument, as the placeholders are required to have some constant meaning throughout it (though we can use the substitution rule on the conclusion of the argument afterwards, having applied $\to$-introduction as many times as we needed). Premises of this kind are analogous to premises in second order logic with free second order variables. They are called specific premises to distinguish them from general premises, which act as though they have the universal closure interpretation.

We now turn to these general premises. Suppose we had established some statement with placeholders, without specifying any meaning for the placeholders. This need not be via a deductive argument. For instance we could argue for induction over the natural numbers

$$\phi_{\text{ind}} = (P(0) \wedge \forall x\,(P(x) \to P(S(x)))) \to \forall x P(x)$$

by saying "if P holds for 0, and it holds for $S(x)$ whenever it holds for x, then it holds for 0, and thus also for $S(0)$, and thus also for $S(S(0))$ and so on on, and thus for all natural numbers". Having established this, we can go on to take P to have a meaning of our choice (setting aside worries like the Sorites paradox), and obtain substitution instances of ϕ_{ind} by appropriately substituting that meaning for P. This is called a general premise, since its role is analogous to a universally closed premise in second order logic.

Importantly, we do not actually have to have established this statement of induction to reason like this: we can reason hypothetically *as though* it had been established with no meaning specified for its placeholders. We can then draw hypothetical consequences of it, reasoning using the substitution rule on it. If at some later time we do manage to establish the statement without specifying a meaning for its placeholders, then we can drop it as an assumption and regard all these consequences as true.

When working from a general premise, we cannot use $\to$-introduction to discharge the assumption. For instance if P is an arity 1 placeholder then for any ϕ we can deduce $\phi(0)$ from $P(0)$ if we make the latter a general premise. Even $\phi(0) = \neg P(0)$ can be deduced. But if we discharge the premise $P(0)$ via $\to$-introduction in this case we obtain $P(0) \to \neg P(0)$, which we would then have established with no premises (since the premise was discharged). Thus we would have established $P(0) \to \neg P(0)$ without specifying a meaning for P, so we could deduce any statement $\phi(0) \to \neg\phi(0)$ as a consequence, which is absurd and quickly leads to a contradiction.

Thus we have two importantly different kinds of premises. There are the general premises Γ which we suppose we have established without specifying a meaning for their placeholders; this allows us to use the substitution rule on these statements, but we cannot use $\to$-introduction on them. Then there are specific premises Δ which will have some constant unspecified meaning throughout the argument; we cannot use the substitution rule on their placeholders but can use $\to$-introduction on them. We will denote a formula ϕ with a set Γ of general premises and a set Δ of specific premises associated with it by $(\Gamma;\Delta) \Rightarrow \phi$.

Now for the notion of derivation in schematic logic. This is intended to be something that would be convenient to actually write out, so a derivation is a finite sequence of formulae rather than a tree. Each line is either a premise or is deduced from previous lines by one of the below rules of inference. We take each formula occurring in the derivation to be labelled with its line number, and with the kind of inference used to infer it: what the rule of inference was, what the previous lines used as premises of this rule were, and which line served as which premise of the rule. We equivocate between the line i and the formula at that line ϕ_i where this will not cause confusion.

Each line ϕ also has associated with it a set Γ of lines which are general premises, and a set Δ of lines which are specific premises — the sets of premises used to infer that line, as discussed above. We can write the line as $(\Gamma;\Delta) \Rightarrow \phi$ to signify this. The inference rules show how the premises of the conclusion depend on the premises of the lines it is inferred from, so these sets Γ and Δ are determined for each line by the derivation (with its labels); otherwise one could take the line numbers of premises in Γ and Δ to also be labels on ϕ.

The rules are standard, except that we have these two different sets of premises, of which only the specific premises can be discharged. We also have the substitution rule for placeholder variables which do not appear in any specific premise.

For premises, we can infer any formula ϕ from ϕ as a premise of either type: we have the two rules

$$(\{\gamma\};\varnothing) \Rightarrow \gamma \qquad\qquad (\varnothing;\{\gamma\}) \Rightarrow \gamma$$

These can be used for any line of the proof, to introduce a premise. Then we have the rule of weakening, where if $\Delta' \subseteq \Delta$ and $\Gamma' \subseteq \Gamma$ then

$$\frac{(\Gamma';\Delta') \Rightarrow \phi}{(\Gamma;\Delta) \Rightarrow \phi}$$

We then have introduction and elimination rules for the various bits of logical vocabulary. We write for instance Δ, ϕ for $\Delta \cup \{\phi\}$.

$$\frac{(\Gamma;\Delta) \Rightarrow \phi \qquad (\Gamma;\Delta) \Rightarrow \psi}{(\Gamma;\Delta) \Rightarrow \phi \wedge \psi} \wedge\text{-I}$$

$$\frac{(\Gamma;\Delta) \Rightarrow \phi \wedge \psi}{(\Gamma;\Delta) \Rightarrow \phi} \wedge\text{-E} \qquad \frac{(\Gamma;\Delta) \Rightarrow \phi \wedge \psi}{(\Gamma;\Delta) \Rightarrow \psi} \wedge\text{-E}$$

$$\frac{(\Gamma;\Delta) \Rightarrow \phi}{(\Gamma;\Delta) \Rightarrow \phi \vee \psi} \vee\text{-I} \qquad \frac{(\Gamma;\Delta) \Rightarrow \psi}{(\Gamma;\Delta) \Rightarrow \phi \vee \psi} \vee\text{-I}$$

$$\frac{(\Gamma;\Delta) \Rightarrow \phi \vee \psi \qquad (\Gamma;\Delta,\phi) \Rightarrow \chi \qquad (\Gamma;\Delta,\psi) \Rightarrow \chi}{(\Gamma;\Delta) \Rightarrow \chi} \vee\text{-E}$$

$$\frac{(\Gamma;\Delta,\phi) \Rightarrow \psi}{(\Gamma;\Delta) \Rightarrow \phi \to \psi} \to\text{-I}$$

$$\frac{(\Gamma;\Delta) \Rightarrow \phi \to \psi \qquad (\Gamma;\Delta) \Rightarrow \phi}{(\Gamma;\Delta) \Rightarrow \psi} \to\text{-E}$$

$$\frac{(\Gamma;\Delta,\phi) \Rightarrow \psi \qquad (\Gamma;\Delta,\phi) \Rightarrow \neg\psi}{(\Gamma;\Delta) \Rightarrow \neg\phi} \neg\text{-I,E}$$

For the quantifier rules, recall that a term t is free for a variable x in formula ϕ if there is no free occurrence of x in ϕ in the scope of a quantifier that binds a variable of t.

$$\frac{(\Gamma;\Delta) \Rightarrow \phi}{(\Gamma;\Delta) \Rightarrow \forall x\, \phi} \forall\text{-I} \qquad \text{if } x \text{ is not free in any member of } \Gamma \text{ or } \Delta$$

$$\frac{(\Gamma;\Delta) \Rightarrow \forall x\, \phi}{(\Gamma;\Delta) \Rightarrow \phi[t|x]} \forall\text{-E} \qquad \text{if } t \text{ free for } x \text{ in } \phi$$

$$\frac{(\Gamma;\Delta) \Rightarrow \phi[t|x]}{(\Gamma;\Delta) \Rightarrow \exists x\, \phi} \exists\text{-I} \qquad \text{if } t \text{ free for } x \text{ in } \phi$$

$$\frac{(\Gamma;\Delta) \Rightarrow \exists x\, \phi \qquad (\Gamma;\Delta,\phi) \Rightarrow \psi}{(\Gamma;\Delta) \Rightarrow \psi} \exists\text{-E} \qquad \begin{array}{l}\text{if } x \text{ is not free in } \psi \text{ or}\\ \text{any member of } \Gamma \text{ or } \Delta\end{array}$$

To make the logic classical we have double negation elimination:

$$\frac{(\Gamma;\Delta) \Rightarrow \neg\neg\phi}{(\Gamma;\Delta) \Rightarrow \phi} \neg\neg_{\mathrm{C}}$$

Finally we have the substitution rule:

$$\frac{(\Gamma;\Delta)\Rightarrow\phi}{(\Gamma;\Delta)\Rightarrow\phi[\psi(\vec{x})|P]}\,\text{Subst}$$

if $\psi(\vec{x})$ is substitutable (in the wide sense) for P in ϕ, and P does not appear in any member of Δ

In the rules where multiple previous lines are used as premises, we number these premises left to right; so $(\Gamma;\Delta)\Rightarrow\phi\vee\psi$ is the first premise in $\vee$-elimination, $(\Gamma;\Delta)\Rightarrow\chi$ the second and so on.

A derivation is a finite sequence of formulae each of which is either a premise, or is deduced from previous line(s) by one of the above rules of inference. As discussed before the description of the rules, we take each formula in the derivation to be labelled with information such as its line number and how it was inferred. Each formula also has a set of assumptions of each type associated with it by the above rules.

If there is a derivation with $(\Gamma;\Delta)\Rightarrow\phi$ as its last line, then we say that $(\Gamma;\Delta)\Rightarrow\phi$ is derivable (equivalently if there is a derivation with $(\Gamma;\Delta)\Rightarrow\phi$ as any line). We write this by $(\Gamma;\Delta)\vdash\phi$. We write just $\Gamma\vdash\phi$ if $(\Gamma;\varnothing)\vdash\phi$.

Note that it is obvious from Proposition 1 that we would obtain the same notion of derivability if we had used the narrow definition of substitutability instead.

Shapiro's system for schematic logic (Shapiro, 2000, pp. 68–69) differs from the one here in that it is formulated as a Hilbert style deductive system. It also employs function variables, and has a special rule relating function variables to relation variables. If one removes the schematic function variables and the special rule for them from $D2-$ (a minor change) then the result has an equivalent derivability relation to the relation $\Gamma\vdash\phi$.

It is straightforward to show for our deductive system that whenever a specific premise is used, a general premise could have been used instead.

Proposition 4. *Suppose that Δ and Λ are disjoint sets of formulae, and that $(\Gamma;\Delta\cup\Lambda)\vdash\phi$. Then $(\Gamma\cup\Lambda;\Delta)\vdash\phi$.*

Proof. This is an easy induction on derivation length, that for all derivations, if the last line is of the form $(\Gamma;\Delta\cup\Lambda)\Rightarrow\phi$ then there is a derivation (of the same length, using the same inference rules but with different premises) of $(\Gamma\cup\Lambda;\Delta)\Rightarrow\phi$. □

Thus by the above proposition whenever $\Gamma;\Delta\vdash\phi$ then $\Gamma\cup\Delta\vdash\phi$.

Before proving a version of the cut rule for this deductive system we prove some preliminary results.

If D is a derivation by the **significant length** of D we mean the number of lines of D which are not inferred by the weakening rule. By an easy induction on D we obtain:

PROPOSITION 5. *If D is a derivation then is a derivation D' of the same significant length as D, where every line of D' is of the form $(\Gamma'; \Delta') \Rightarrow \phi$ with $(\Gamma; \Delta) \Rightarrow \phi$ a line of D and Γ', Δ' finite subsets of Γ, Δ respectively, and where if $(\Gamma; \Delta) \Rightarrow \phi$ is a line of D then there is a line $(\Gamma'; \Delta') \Rightarrow \phi$ of D' with Γ', Δ' finite subsets of Γ, Δ.*

Furthermore, we have:

PROPOSITION 6. *Let D be a derivation of $(\Gamma; \Delta) \Rightarrow \phi$, let x be a variable, and let y be a variable not appearing in D. Then there is a derivation D' of*

$$(\{\gamma[y|x] \mid \gamma \in \Gamma\}; \{\delta[y|x] \mid \delta \in \Delta\}) \Rightarrow \phi[y|x].$$

the same length as D, whose every line is inferred by the same rule as in D, and with the same premises and conclusions but with y substituted for x in them.

PROOF. This is an easy induction on D. The most subtle case is that where the last line of D is $(\Gamma; \Delta) \Rightarrow \phi[\psi(\vec{x})|P]$, inferred by the substitution rule from $(\Gamma; \Delta) \Rightarrow \phi$. In this case by the induction hypothesis replacing every x with y in every previous line of D gives a derivation of

$$(\{\gamma[y|x] \mid \gamma \in \Gamma\}; \{\delta[y|x] \mid \delta \in \Delta\}) \Rightarrow \phi[y|x].$$

Then letting $\vec{x} = (x_1, \ldots, x_n)$, if x is one of the x_i then we reach final line

$$(\{\gamma[y|x] \mid \gamma \in \Gamma\}; \{\delta[y|x] \mid \delta \in \Delta\}) \Rightarrow \phi[\psi(\vec{x})|P][y|x].$$

by substituting $\psi(\vec{x})$ for P in $\phi[y|x]$; and if x is not one of the x_i, we reach final line

$$(\{\gamma[y|x] \mid \gamma \in \Gamma\}; \{\delta[y|x] \mid \delta \in \Delta\}) \Rightarrow \phi[\psi(\vec{x})|P][y|x].$$

by substituting $\psi[y|x](\vec{x})$ for P in $\phi[y|x]$. $\square$

If ϕ is a formula and P and Q are placeholder variables of the same arity, by $\phi[Q|P]$ we mean the result of replacing P by Q everywhere it appears in ϕ. Moreover, by an easy induction on D we obtain:

PROPOSITION 7. *Let D be a derivation of $(\Gamma; \Delta) \Rightarrow \phi$, let P be a placeholder variable, and let Q be a placeholder variable of the same arity as P not appearing in D. Then there is a derivation D' of*

$$(\{\gamma[Q|P] \mid \gamma \in \Gamma\}; \{\delta[Q|P] \mid \delta \in \Delta\}) \Rightarrow \phi[Q|P].$$

which is the same as D but with P replaced by Q in every formula (in premises and conclusions of lines and uses of inference rules).

PROPOSITION 8. *Let P be a placeholder variable, and let Q be a placeholder variable of the same arity n as P. Let $x_1, \ldots, x_n$ be any distinct variables, and let ψ be the formula $Q(x_1, \ldots, x_n)$. Then for any ϕ, $\psi(\vec{x})$ is substitutable (in the wide sense) for P in ϕ, and $\phi[\psi(\vec{x})|P] = \phi[Q|P]$.*

PROOF. For any ϕ, $\psi(\vec{x})$ is substitutable (in the wide sense) for P in ϕ since ψ contains no quantifiers and variables beyond the x_i (this is an example of the usefulness of the wide version of the substitution rule). Carrying out the substitution $\phi[\psi(\vec{x})|P]$ results in replacing every occurrence of $P(t_1, \ldots, t_n)$ in ϕ by $\psi[t_i|x_i]$, i.e. replacing every occurrence of $P(t_1, \ldots, t_n)$ in ϕ by $Q(t_1, \ldots, t_n)$, i.e. results in $\phi[Q|P]$. $\square$

Now we prove our version of the cut rule. First, if Γ, Δ, X and Y are sets of formulae, we write $(\Gamma; \Delta) \vdash (X; Y)$ if $(\Gamma; \varnothing) \vdash \theta$ for all $\theta \in X$, and $(\Gamma; \Delta) \vdash \theta$ for all $\theta \in Y$.

PROPOSITION 9. *Suppose Γ, Δ, X, Y are sets of formulae and ϕ a formula. Then if $(X; Y) \vdash \phi$ and $(\Gamma; \Delta) \vdash (X; Y)$, then $(\Gamma; \Delta) \vdash \phi$.*

PROOF. We prove by induction on the significant length of derivations that if D is a derivation of $(X; Y \cup \Lambda) \Rightarrow \phi$, for some X, Y, Λ and ϕ, and we have $(\Gamma; \Delta) \vdash (X; Y)$ for some Γ, Δ, then $(\Gamma; \Delta \cup \Lambda) \vdash \phi$. It is clear that if this property holds for D it also holds for a derivation consisting of D followed by an application of the weakening rule. In the induction the cases of premises, and the rules for propositional connectives are easy, as are the cases of $\forall$-elimination and $\exists$-introduction.

For $\forall$-introduction, we suppose the induction hypothesis holds for derivations of significant length n, and we have a derivation D of significant length $n + 1$, whose final line is $(X; Y \cup \Lambda) \Rightarrow \forall x\, \phi$, deduced from previous line $(X; Y \cup \Lambda) \Rightarrow \phi$ using $\forall$-introduction, with x not free in any member of X or Y or Λ, and suppose also we have Γ, Δ such that $(\Gamma; \Delta) \vdash (X; Y)$. By Proposition 5 (and weakening) there is a derivation D' of $(X'; Y' \cup \Lambda) \Rightarrow \phi$ of significant length at most n with X', Y', finite subsets of X and Y respectively. Then by Proposition 5 again we can find finite subsets Γ' and Δ' of Γ and Δ such that $(\Gamma'; \Delta') \vdash (X'; Y')$. Then we can find a variable y which does not appear in D' or any element of Γ' or Δ', and then by Proposition 6 we obtain a derivation of $(X'; Y' \cup \Lambda) \Rightarrow \phi[y|x]$ of significant length at most n. Thus by the induction hypothesis we have $(\Gamma'; \Delta' \cup \Lambda) \vdash \phi[y|x]$, and then by $\forall$-introduction we obtain $(\Gamma'; \Delta' \cup \Lambda) \vdash \forall y\, \phi[y|x]$. But then using $\rightarrow$-elimination on this together with the fact that

$$(\varnothing; \varnothing) \vdash (\forall y\, \phi[y|x]) \ \rightarrow\ (\forall x \phi)$$

(which is easily verified) gives that $(\Gamma'; \Delta' \cup \Lambda) \vdash \forall x\, \phi$. Thus by weakening we obtain $(\Gamma; \Delta \cup \Lambda) \vdash \forall x\, \phi$, as required.

For $\exists$-elimination, we suppose the induction hypothesis holds for derivations of significant length n, and we have a derivation D of significant length $n + 1$, whose final line is $(X; Y \cup \Lambda) \Rightarrow \psi$, deduced from previous lines $(X; Y \cup \Lambda) \Rightarrow \exists x\, \phi$ and $(X; Y \cup \Lambda \cup \{\phi\}) \Rightarrow \psi$ using $\exists$-elimination, with x not free in any member of X or Y or Λ, and we also suppose we have Γ, Δ such that $(\Gamma; \Delta) \vdash (X; Y)$. Then by the induction hypothesis we obtain $(\Gamma; \Delta) \vdash \exists x\, \phi$. Also by Proposition 5 (and weakening) there is a derivation D' of $(X'; Y' \cup \Lambda \cup \{\phi\}) \Rightarrow \psi$ of significant length at most n with X', Y', finite subsets of X and Y respectively. Then by Proposition 5 we can find finite subsets Γ' and Δ' of Γ and Δ such that $(\Gamma'; \Delta') \vdash (X'; Y')$. Then we can find a variable y which does not appear in D' or any element of Γ' or Δ', and then by Proposition 6 we obtain a derivation of $(X'; Y' \cup \Lambda \cup \{\phi[y|x]\}) \Rightarrow \psi$ of significant length at most n. Then by the induction hypothesis we have $(\Gamma'; \Delta' \cup \Lambda \cup \{\phi[y|x]\}) \vdash \psi$, and thus $(\Gamma; \Delta \cup \Lambda \cup \{\phi[y|x]\}) \vdash \psi$ by weakening. But then since $(\Gamma; \Delta) \vdash \exists x\, \phi$, using $\rightarrow$-elimination together with the fact that

$$(\varnothing; \varnothing) \vdash (\exists x \phi) \rightarrow (\exists y\, \phi[y|x])$$

(which is easily verified) gives $(\Gamma; \Delta) \vdash \exists y\, \phi[y|x]$, so that by $\exists$-elimination we obtain $(\Gamma; \Delta) \vdash \psi$, as required.

For the substitution rule, we suppose the induction hypothesis holds for derivations of significant length n, and we have a derivation D of significant length $n+1$, whose final line is $(X; Y \cup \Lambda) \Rightarrow \phi[\psi(\vec{x})|P]$, deduced from previous line $(X; Y \cup \Lambda) \Rightarrow \phi$ using the substitution rule, with P not appearing in any element of Y or Λ, and suppose also we have Γ, Δ such that $(\Gamma; \Delta) \vdash (X; Y)$. By Proposition 5 (and weakening) there is a derivation D' of $(X'; Y' \cup \Lambda) \Rightarrow \phi$ of significant length at most n with X', Y', finite subsets of X and Y respectively. Then by Proposition 5 we can find finite subsets Γ' and Δ' of Γ and Δ such that $(\Gamma'; \Delta') \vdash (X'; Y')$. Then we can find a placeholder variable Q which does not appear in D' or any element of Γ' or Δ', and then by Proposition 7 we obtain a derivation of

$$(\{\chi[Q|P] \mid \chi \in X'\}; Y' \cup \Lambda) \Rightarrow \phi[Q|P]$$

of significant length at most n. Then for each $\chi \in X'$, since $\Gamma' \vdash \chi$ by assumption we have $\Gamma' \vdash \chi[Q|P]$ by Proposition 8, so that

$$(\Gamma'; \Delta') \vdash (\{\chi[Q|P] \mid \chi \in X'\}; Y')$$

and thus by the induction hypothesis $(\Gamma'; \Delta' \cup \Lambda) \vdash \phi[Q|P]$. Then the substitution rule gives $(\Gamma'; \Delta' \cup \Lambda) \vdash \phi[Q|P][\psi(\vec{x})|Q]$, i.e. $(\Gamma'; \Delta' \cup \Lambda) \vdash \phi[\psi(\vec{x})|P]$. Finally weakening gives $(\Gamma; \Delta \cup \Lambda) \vdash \phi[\psi(\vec{x})|P]$ as required. □

We introduce a relation $\vdash_1$ for derivability in the first order portion of the deductive system. To be precise, if Γ is a set of formulae and ϕ a formula (which may contain placeholder variables) then $\Gamma \vdash_1 \phi$ is defined to hold if $(\varnothing; \Gamma) \vdash \phi$. Thus $\Gamma \vdash_1 \phi$ implies $\Gamma \vdash \phi$, by Proposition 4. By Proposition 9, if $\Lambda \vdash_1 \phi$, and for all $\theta \in \Lambda$ we have $(\Gamma; \Delta) \vdash \theta$, then $\Gamma; \Delta \vdash \phi$.

For formulae of first order logic, this derivability relation $\vdash_1$ is complete (it is a standard system of first order natural deduction). To be more precise, we use $\vDash_1$ to denote normal first order entailment. Then for Γ a set of placeholder free formulae and ϕ another placeholder free formula, $\Gamma \vDash_1 \phi$ implies $\Gamma \vdash_1 \phi$.

6. Soundness

We now prove a soundness theorem for schematic logic, which justifies its use in an ontologically innocent way. The idea is that when we derive a conclusion ϕ from general premises Γ in schematic logic, we are implicitly giving meanings to placeholders with each use of the substitution rule—setting up a way that the meaning of the formula χ being substituted in depends on the meaning stipulated for the result $\chi[\psi|P]$. Combining these dependencies with a stipulation for any placeholders in the conclusion ϕ, we obtain stipulations for every placeholder that occurs in the derivation, and in every premise that is used (at some points we may also have to make some arbitrary stipulations for surplus placeholders). The soundness theorem states that if the premises in Γ are true under these meaning stipulations (there may be multiple meaning stipulations applying to each premise), then the conclusion ϕ is also true under its meaning stipulation. The theorem deals entirely with meaning stipulations that arise in the course of a derivation, and to apply it in any particular case we do not need to quantify over an infinite totality of meaning stipulations, or formulae, or properties.

A single use of the substitution rule, to obtain $\chi[\psi|P]$ from χ, may well not be enough to give a meaning stipulation for χ. There may be other placeholders in χ than P, and ψ may itself contain placeholders. These may have meanings given by later uses of the substitution rule, perhaps deducing $\chi[\psi|P][\theta|Q]$ from $\chi[\psi|P]$, providing a meaning for Q as well as P. As a result we are interested in meanings that χ picks up on a path through the derivation. By path here we mean a sequence $(\phi_1, \ldots, \phi_n)$ of lines of a derivation (not necessarily consecutive) such that for each $i < n$ ϕ_{i+1} is deduced by a rule of inference with ϕ_i one of the premises used. We will call such a sequence a **deductive chain**.

So let $\Phi = (\Phi_1, \ldots, \Phi_n)$ be a deductive chain in some derivation, and suppose we have a meaning stipulation λ which covers every placeholder that appears in Φ. We define by induction downwards along the chain a meaning stipulation $M_i^{\Phi,\lambda}$ for each Φ_i. This is done via the uses of the substitution rule in Φ: each of these can be considered an act setting up a meaning dependence of the premise of the substitution on the result, which put together gives a meaning stipulation for each line of the chain in terms of λ.

We let $M_n^{\Phi,\lambda}$ be λ. For each i if Φ_{i+1} is deduced from Φ_i by any inference rule other than substitution, then $M_i^{\Phi,\lambda}$ is just defined to be $M_{i+1}^{\Phi,\lambda}$. If Φ_{i+1} is deduced from Φ_i by substitution, with Φ_{i+1} being $\Phi_i[\psi(\vec{x})|P]$, then $M_i^{\Phi,\lambda}$ is built using the meaning dependency described in Section 4: it is defined to be $g_{\psi(\vec{x})|P}(M_{i+1}^{\Phi,\lambda})$. At every point in the chain $M_i^{\Phi,\lambda}$ covers all the placeholders covered by $M_{i+1}^{\Phi,\lambda}$, so since $M_n^{\Phi,\lambda} = \lambda$

covers every placeholder appearing in the deductive chain, so does each $M_i^{\Phi,\lambda}$. In particular every $M_i^{\Phi,\lambda}$ is a meaning stipulation for Φ_i.

We are interested in the meaning stipulations of this kind that a formula picks up from different paths through a derivation. So suppose we have a derivation D with conclusion χ. We let C_D be the set of maximal deductive chains in D which end in χ; thus each Φ in C_D starts with a premise and ends in χ (we require that they end in χ since there might be dead ends in D — lines of the proof not used at any later point — and we want to ignore these). Now a meaning stipulation λ for χ may not be enough to define the $M_i^{\Phi,\lambda}$ as above: we also need that λ covers every placeholder in a given path Φ. Thus we extend λ arbitrarily to cover any surplus placeholders.

Suppose λ is a meaning stipulation with domain A^p, and let $\mathcal{P}$ be a finite set of placeholders. An **extension of λ to $\mathcal{P}$** is just a meaning stipulation $\overline{\lambda}$ with domain A^p, which covers $\mathcal{P}$ and every placeholder covered by λ, and which agrees with λ on the placeholders covered by λ (if λ covers Q and $a_1, \ldots, a_p \in A^p$ then $\overline{\lambda}(a_1, \ldots, a_p)(Q) = \lambda(a_1, \ldots, a_p)(Q)$). Obviously every meaning stipulation has an extension to any finite set of placeholders. We can if we like suppose that we have stipulated some canonical manner for forming extensions — that we have at some point stipulated arbitrary subsets of A^m to serve as meanings for surplus placeholders of arity m (this could take place once and for all the first time schematic logic is used for the language in question).

Now suppose that D is a derivation with conclusion χ and that $\overline{\lambda}$ is a meaning stipulation which covers every placeholder in D. For each line ϕ of D, we let $S^{\overline{\lambda}}(\phi)$ be the set of meaning stipulations ϕ obtains in the course of the derivation when $\overline{\lambda}$ is used as a meaning stipulation for the conclusion χ:

$$S^{\overline{\lambda}}(\phi) = \{M_i^{\Phi,\overline{\lambda}} \mid \Phi \in C_D,\ \Phi_i = \phi\}.$$

These $S^{\overline{\lambda}}(\phi)$ are obviously finite sets. Note that to talk about these meanings we do not need to believe in any abstract objects. Although the chains are modelled as finite sequences of lines of a derivation, this is just the appropriate set theoretic representative. One would normally have no problem talking about a sequence of lines of a physically realised proof; if one wants an analysis of this, one could talk in terms of pluralities or fusions of lines.

We make some quick notes about $S^{\overline{\lambda}}$ before proceeding. Suppose that ϕ is deduced by some inference rule apart from substitution using previous line θ, and let $\mu \in S^{\overline{\lambda}}(\phi)$. By definition there is a chain Φ in C_D with $\Phi_i = \phi$, $M_i^{\Phi,\lambda} = \mu$ for some i. Then there must also exist a chain Ψ which goes through θ to ϕ and agrees with Φ from that point on: i.e. there is some j such that Ψ_j is ϕ, Ψ_{j-1} is θ and $\Psi_k = \Phi_{k+i-j}$ for $k \geqslant j$. By induction down from the end of these chains, we have that $M_k^{\Psi,\lambda} = M_{k+i-j}^{\Phi,\lambda}$ for $k \geqslant j$. In particular $M_j^{\Psi,\lambda} = M_i^{\Phi,\lambda} = \mu$; so $M_{j-1}^{\Psi,\lambda} = \mu$ by definition, so $\mu \in S^{\overline{\lambda}}(\theta)$.

Now suppose that ϕ is deduced by substitution from θ, with $\phi = \theta[\psi(\vec{x})|P]$. Again if $\mu \in S^{\overline{\lambda}}(\phi)$ then there is a chain Φ in C_D with $\Phi_i = \phi$, $M_i^{\Phi,\lambda} = \mu$. Thus θ must be Φ_{i-1}, and by the definition of M we have $M_{i-1}^{\Phi,\lambda} = g_{\psi(\vec{x})|P}(\mu)$, so $g_{\psi(\vec{x})|P}(\mu) \in S^{\overline{\lambda}}(\theta)$.

To prove the soundness theorem we need an entailment relation for formulae with meaning stipulations over a given structure. Let A be a structure. Suppose that K is a set of pairs where if $(\phi, \mu) \in K$ then ϕ is a formula and μ a meaning stipulation for ϕ over A. For a variable assignment v over A, we write $A, v, \vDash K$ if $A, v, \mu \vDash \phi$ for all (ϕ, μ) in K. If (θ, λ) is another pair with θ a formula and λ a meaning stipulation for θ then we write $K \vDash_A (\theta, \lambda)$ if for every variable assignment v such that $A, v, \vDash K$, we have $A, v, \lambda \vDash \theta$.

Finally if Λ is a set of lines of D, we let

$$S^{\overline{\lambda}}(\Lambda) = \{(\phi, \mu) \mid \phi \in \Lambda,\ \mu \in S^{\overline{\lambda}}(\phi)\}.$$

We use this notation in particular for $S^{\overline{\lambda}}(\Gamma)$ where $(\Gamma; \Delta) \Rightarrow \chi$ is a line in a derivation — the set Γ of general premises being a set of lines of the derivation, as discussed in Section 5.

We can now state and prove the soundness theorem. The proof is very much like soundness for first order natural deduction, except for having to keep track of the meaning stipulations in use, and the appeal to Lemma 3.

Theorem 10 (Soundness of Schematic Logic). *Let D be a derivation in schematic logic with final line $(\Gamma; \Delta) \Rightarrow \chi$. Let λ be a meaning stipulation for χ, and let $\overline{\lambda}$ be an extension of λ to the set of placeholders occurring in D. Then $S^{\overline{\lambda}}(\Gamma) \cup (\Delta \times \{\overline{\lambda}\}) \vDash_A (\chi, \lambda)$.*

Proof. We show by induction on i that for every line $(\Gamma; \Delta) \Rightarrow \phi_i$ of D, and for every $\mu \in S^{\overline{\lambda}}(\phi_i)$, we have $S^{\overline{\lambda}}(\Gamma) \cup (\Delta \times \{\mu\}) \vDash_A (\phi_i, \mu)$. This proves the theorem since $\overline{\lambda} \in S^{\overline{\lambda}}(\chi)$, and λ agrees with $\overline{\lambda}$ on the placeholders of χ.

If $(\{\phi_i\};\varnothing) \Rightarrow \phi_i$ is a general premise then for any $\mu \in S^{\overline{\lambda}}(\phi_i)$ we have $(\phi,\mu) \in S^{\overline{\lambda}}(\{\phi_i\})$ so this is straightforward. If $(\varnothing;\{\phi_i\}) \Rightarrow \phi_i$ is a specific premise then for any $\mu \in S^{\overline{\lambda}}(\phi_i)$ we have $(\phi_i,\mu) \in \{\phi_i\} \times \{\mu\}$ so this is again straightforward.

If line $(\Gamma;\Delta) \Rightarrow \phi_i$ is deduced from line $(\Gamma';\Delta') \Rightarrow \phi_j$ by weakening then $S^{\overline{\lambda}}(\phi_j) = S^{\overline{\lambda}}(\phi_i)$. Thus by the induction hypothesis if $\mu \in S^{\overline{\lambda}}(\phi_i)$ then $S^{\overline{\lambda}}(\Gamma') \cup (\Delta' \times \{\mu\}) \vDash_A (\phi_j,\mu)$, so that since $\phi_i = \phi_j$, $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$ we obtain $S^{\overline{\lambda}}(\Gamma) \cup (\Delta \times \{\mu\}) \vDash_A (\phi_i,\mu)$ as required.

Suppose line $(\Gamma;\Delta) \Rightarrow \phi_i$ is deduced from ϕ_j and ϕ_k by $\wedge$-introduction, with $\phi_i = \phi_j \wedge \phi_k$, and let $\mu \in S^{\overline{\lambda}}(\phi_i)$. Then also $\mu \in S(\phi_j)$ and $\mu \in S^{\overline{\lambda}}(\phi_k)$, so by the induction hypothesis we have $S^{\overline{\lambda}}(\Gamma)\cup(\Delta\times\{\mu\}) \vDash_A (\phi_j,\mu)$ and $S^{\overline{\lambda}}(\Gamma) \cup (\Delta \times \{\mu\}) \vDash_A (\phi_k,\mu)$. Thus $S^{\overline{\lambda}}(\Gamma) \cup (\Delta \times \{\mu\}) \vDash_A (\phi_j \wedge \phi_k,\mu)$.

The cases of $\wedge$-elimination and $\vee$-introduction are immediate.

Suppose line $(\Gamma;\Delta) \Rightarrow \phi_i$ is deduced by $\vee$-elimination from lines j, k and l (with the premises in this order), where $\phi_j = \theta \vee \psi$, $\phi_k = \phi_i$ and $\phi_l = \phi_i$. Let $\mu \in S^{\overline{\lambda}}(\phi_i)$, so $\mu \in S^{\overline{\lambda}}(\phi_j)$, $\mu \in S^{\overline{\lambda}}(\phi_k)$ and $\mu \in S(\phi_l)$. By the induction hypothesis we have $S^{\overline{\lambda}}(\Gamma)\cup(\Delta\times\{\mu\}) \vDash_A (\theta\vee\psi,\mu)$, $S^{\overline{\lambda}}(\Gamma)\cup((\Delta \cup \{\theta\}) \times \{\mu\}) \vDash_A (\phi_i,\mu)$ and $S^{\overline{\lambda}}(\Gamma) \cup ((\Delta \cup \{\psi\}) \times \{\mu\}) \vDash_A (\phi_i,\mu)$. Suppose that $A, v \vDash S^{\overline{\lambda}}(\Gamma)\cup(\Delta\times\{\mu\})$. Then $A, v, \mu \vDash \theta\vee\psi$, so $A, v, \mu \vDash \theta$ or $A, v, \mu \vDash \phi$. If the former then $A, v \vDash S^{\overline{\lambda}}(\Gamma) \cup ((\Delta \cup \{\theta\}) \times \{\mu\})$, so $A, v \vDash (\phi_i,\mu)$. Similarly if $A, v, \mu \vDash \psi$ then $A, v \vDash (\phi_i,\mu)$. Thus either way, $A, v \vDash (\phi_i,\mu)$, as required.

Suppose that line $(\Gamma;\Delta) \Rightarrow \phi_i$ is deduced from ϕ_j by $\rightarrow$-introduction, with $\phi_i = \psi \rightarrow \phi_j$. Let $\mu \in S^{\overline{\lambda}}(\phi)$, so $\mu \in S^{\overline{\lambda}}(\phi_j)$. By the induction hypothesis $S^{\overline{\lambda}}(\Gamma) \cup ((\Delta \cup \{\psi\}) \times \{\mu\}) \vDash_A (\phi_j,\mu)$. Suppose that v is a variable assignment such that $A, v \vDash S^{\overline{\lambda}}(\Gamma) \cup (\Delta \times \{\mu\})$. Then if $A, v, \mu \vDash \psi$ then $A, v \vDash S^{\overline{\lambda}}(\Gamma) \cup ((\Delta \cup \{\psi\}) \times \{\mu\})$ so $A, v, \mu \vDash \phi_j$. Thus for any v such that $A, v \vDash S^{\overline{\lambda}}(\Gamma) \cup (\Delta \times \{\mu\})$, we have $A, v, \mu \vDash \psi \rightarrow \phi_j$. Thus $S^{\overline{\lambda}}(\Gamma) \cup (\Delta \times \{\mu\}) \vDash_A (\phi_i,\mu)$, as required.

Suppose that line $(\Gamma;\Delta) \Rightarrow \phi_i$ is deduced from ϕ_j and $\phi_k = \phi_j \rightarrow \phi_i$ by $\rightarrow$-elimination. Let $\mu \in S^{\overline{\lambda}}(\phi_i)$, so $\mu \in S^{\overline{\lambda}}(\phi_j)$ and $\mu \in S^{\overline{\lambda}}(\phi_k)$. Thus if v is a variable assignment such that $A, v \vDash S^{\overline{\lambda}}(\Gamma) \cup (\Delta \times \{\mu\})$, then by the induction hypothesis $A, v, \mu \vDash \phi_j$ and $A, v, \mu \vDash \phi_j \rightarrow \phi_i$. Thus $A, v, \mu \vDash \phi_i$, as required.

Suppose that line $(\Gamma;\Delta) \Rightarrow \phi_i$ is deduced from previous lines ϕ_j and $\phi_k = \neg\phi_j$ by the $\neg$-rule, with $\phi_i = \neg\psi$. Let $\mu \in \overline{S^\lambda}(\phi_i)$, so $\mu \in \overline{S^\lambda}(\phi_j)$ and $\mu \in \overline{S^\lambda}(\phi_k)$. Let v be a variable assignment such that $A, v \vDash \overline{S^\lambda}(\Gamma) \cup (\Delta \times \{\mu\})$. Then if $A, v, \mu \vdash \psi$, then $A, v \vDash \overline{S^\lambda}(\Gamma) \cup ((\Delta \cup \{\psi\}) \times \{\mu\})$ so by the induction hypothesis $A, v, \mu \vDash \phi_j$ and $A, v, \mu \vDash \neg\phi_j$ which is a contradiction. Thus $A, v, \mu \nvdash \psi$, so $A, v, \mu \vdash \neg\psi$, so $A, v, \mu \vdash \phi_i$ as required.

Suppose that line $(\Gamma;\Delta) \Rightarrow \phi_i$ is deduced from $\phi_j = \neg\neg\phi_i$ by double negation elimination. Let $\mu \in \overline{S^\lambda}(\phi_i)$, so $\mu \in \overline{S^\lambda}(\phi_j)$. Let v be a variable assignment such that $A, v \vDash \overline{S^\lambda}(\Gamma) \cup (\Delta \times \{\mu\})$. Thus by the induction hypothesis $A, v, \mu \vDash \neg\neg\phi_i$. Thus it's not true that $A, v, \mu \nvDash \phi_i$, so $A, v, \mu \vDash \phi_i$ as required.

Suppose that line $(\Gamma;\Delta) \Rightarrow \phi_i$ is deduced from ϕ_j by $\forall$-introduction, $\phi_i = \forall x\, \phi_j$. We have that x is not free in any element of Γ or Δ. Let $\mu \in \overline{S^\lambda}(\phi_i)$, so $\mu \in \overline{S^\lambda}(\phi_j)$. Let v be a variable assignment such that $A, v \vDash \overline{S^\lambda}(\Gamma) \cup (\Delta \times \{\mu\})$. Then if a is any element of A, we also have $A, v(a|x) \vDash \overline{S^\lambda}(\Gamma) \cup (\Delta \times \{\mu\})$. Thus by the induction hypothesis $A, v(a|x), \mu \vDash \phi_j$. But a was arbitrary, so $A, v, \mu \vDash \forall x\, \phi_j$ as required.

Suppose that line $(\Gamma;\Delta) \Rightarrow \phi_i$ is deduced from ϕ_j by $\forall$-elimination, with $\phi_j = \forall x\, \psi$ and $\phi_i = \psi[t|x]$. Let $\mu \in \overline{S^\lambda}(\phi_i)$, so $\mu \in \overline{S^\lambda}(\phi_j)$. Let v be a variable assignment such that $A, v \vDash \overline{S^\lambda}(\Gamma) \cup (\Delta \times \{\mu\})$. Thus by the induction hypothesis $A, v, \mu \vDash \forall x\, \psi$, so $A, v(v(t)|x), \mu \vDash \psi$, so $A, v, \mu \vDash \psi[t|x]$, as required.

The case of $\exists$-introduction is similar.

Suppose that line $(\Gamma;\Delta) \Rightarrow \phi_i$ is deduced by $\exists$-elimination from lines j and k (with the premises in this order), where $\phi_j = \exists x\, \theta$ and $\phi_k = \phi_i$. We have that x is not free in ϕ_i or any member of Γ or Δ. Let $\mu \in \overline{S^\lambda}(\phi)$, so $\mu \in \overline{S^\lambda}(\phi_j)$ and $\mu \in S(\phi_k)$. Let v be a variable assignment such that $A, v \vDash \overline{S^\lambda}(\Gamma) \cup (\Delta \times \{\mu\})$. Then by the induction hypothesis $A, v, \mu \vDash \exists x\, \theta$, so there is some $a \in A$ with $A, v(a|x), \mu \vDash \theta$. Then $A, v(a|x) \vDash \overline{S^\lambda}(\Gamma) \cup ((\Delta \cup \{\theta\}) \times \{\mu\})$ so again by the induction hypothesis $A, v(a|x), \mu \vDash \phi_i$. But x is not free in ϕ_i so $A, v, \mu \vDash \phi_i$ as required.

Suppose finally that line $(\Gamma;\Delta) \Rightarrow \phi_i$ is deduced from ϕ_j by substitution, $\phi_i = \phi_j[\psi(\vec{x})|P]$. Let $\mu \in \overline{S^\lambda}(\phi_i)$, so (by the discussion before the theorem) $g_{\psi(\vec{x})|P}(\mu) \in \overline{S^\lambda}(\phi_j)$. Let v be a variable assignment such that $A, v \vDash \overline{S^\lambda}(\Gamma) \cup (\Delta \times \{\mu\})$. Then for any $\delta \in \Delta$ we have $A, v, \mu \vDash \delta$, and P does not appear in δ, so $g_{\psi(\vec{x})|P}(\mu)$ and μ agree on the placeholders of δ, so $A, v, g_{\psi(\vec{x})|P}(\mu) \vDash \delta$. Thus $A, v \vDash S^{\overline{\lambda}}(\Gamma) \cup (\Delta \times \{g_{\psi(\vec{x})|P}(\mu)\})$, so by the induction hypothesis $A, v, g_{\psi(\vec{x})|P}(\mu) \vDash \phi_j$. We can then apply Lemma 3 and deduce that $A, v, \mu \vDash \phi_j[\psi(\vec{x})|P]$, as required. □

Thus — essentially — if we stipulate meanings for the placeholders in the conclusion of a derivation, and the premises of the derivation are true under the corresponding meaning stipulations picked up on paths through the derivation, then the conclusion of the derivation is true under that meaning stipulation. This justifies the use of schematic logic in an ontologically innocent way. We do not need any totality of properties, pluralities, formulae or meaning stipulations for the placeholder variables to range over: they just take on meanings given in the course of a derivation.

7. Definable meaning stipulations and conservativeness

We now focus on particular meaning stipulations, the *definable meaning stipulations*. These are meaning stipulations which assign meanings in terms of open formulae of the language in use. We also introduce *stipulation blueprints*, which are syntactic objects that give rise to definable meaning stipulations. We will see that we can require that all meaning stipulations that appear in the proof of the soundness theorem are definable, so we could have used the notion of definable meaning stipulation or stipulation blueprint throughout — that would have complicated the exposition somewhat though. We define the notions now as once we have established their basic properties we can prove a simple conservativeness theorem for schematic logic, as a corollary to the soundness theorem. In the next section we will also see how using stipulation blueprints allows us to prove a straightforward completeness theorem for schematic logic.

Suppose we have placeholders $P_1^{n_1}, \ldots, P_m^{n_m}$ and placeholder free formulae $\psi_1(x_1^1, \ldots, x_{n_1}^1), \ldots, \psi_m(x_1^m, \ldots, x_{n_m}^m)$, and also variables $y_1, \ldots, y_p$ distinct from each other and all of the x_j^i, such that for each i, $(\mathrm{FV}(\psi_i) \setminus \{x_1^i, \ldots x_{n_i}^i\}) \subseteq \{y_1, \ldots, y_p\}$. If the P_i, $\psi_i(\vec{x^i})$ and y_j satisfy these conditions then we call

$$(\{\psi_1(\vec{x^1})|P_1, \ldots \psi_m(\vec{x^m})|P_m\}, (y_1, \ldots, y_p))$$

a **stipulation blueprint**, where we write $\psi_i(\vec{x^i})|P_i$ for the pair $(\psi_i(\vec{x^i}), P_i)$. We use the abbreviation $\vec{\psi}|\vec{P}$ for $\{\psi_1(\vec{x^1})|P_1, \ldots \psi_m(\vec{x^m})|P_m\}$.

Any stipulation blueprint $\sigma = (\vec{\psi}|\vec{P}, (y_1, \ldots, y_p))$ has a meaning dependence $g^{\vec{y}}_{\vec{\psi}|\vec{P}}$ associated with it, as defined in Section 4. This is a meaning dependence of $\{P_1, \ldots, P_m\}$ on the empty set of placeholders (since the ψ_i are placeholder free). For such a stipulation blueprint we define the meaning stipulation $\eta^\sigma = \eta^{y_1,\ldots,y_p}_{\vec{\psi}|\vec{P}}$ which has domain A^p by

$$\eta^{\vec{y}}_{\vec{\psi}|\vec{P}} = g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda_\varnothing),$$

where $\lambda_\varnothing$ is the trivial unparametrized meaning stipulation, which covers no placeholders (so $\lambda_\varnothing$ is just the empty function). A stipulation of the form η^σ for some stipulation blueprint σ is called a **definable meaning stipulation**.

Unpacking the definition, we have that the set of placeholders $\eta^{\vec{y}}_{\vec{\psi}|\vec{P}}$ covers is just $\{P_1, \ldots, P_m\}$, and for each i we have

$$\begin{aligned}\eta^{\vec{y}}_{\vec{\psi}|\vec{P}}(b_1, ..., b_p)(P_i) &= g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda_\varnothing)(b_1, ..., b_p)(P_i) = \delta^{\vec{y},\lambda_\varnothing}_{\psi_i(\vec{x_i})|P_i}(b_1 \ldots b_p)(P_i)\\ &= \{(a_1, \ldots, a_{n_i}) \in A^{n_i} \mid A,\, x^i_j \mapsto a_j, y_k \mapsto b_k,\, \lambda_\varnothing \vDash \psi_i\}\\ &= \{(a_1, \ldots, a_{n_i}) \in A^{n_i} \mid A,\, x^i_j \mapsto a_j, y_k \mapsto b_k \vDash_1 \psi_i\}.\end{aligned}$$

It will be useful to know that if λ is a definable meaning stipulation for ϕ, then λ is actually defined by a stipulation blueprint which is particularly suited to substitution in ϕ. For a formula ϕ we say that a stipulation blueprint $(\{\psi_1(\vec{x^1})|P_1, \ldots \psi_m(\vec{x^m})|P_m\}, \vec{y})$ is **suited to** ϕ if all placeholders in ϕ are among the P_i, and each $\psi_i(\vec{x^i})$ is substitutable for P_i in ϕ, and no y_j occurs free in ϕ.

Note that from the second condition it follows that the repeated substitution $\phi[\psi_1(\vec{x^1})|P_1], \ldots [\psi_m(\vec{x^m})|P_m]$ is defined, and is independent of the order the substitution are made, and equal to the multiple substitution $\phi[\psi_1(\vec{x^1})|P_1, \ldots \psi_m(\vec{x^m})|P_m]$.

LEMMA 11. *Suppose that λ is a definable meaning stipulation for ϕ. Let the domain of λ be A^p, and let $\{P_1, \ldots, P_m\}$ be the set of placeholders it covers. Then there is a stipulation blueprint $(\vec{\psi}|\vec{P}, (y_1, \ldots, y_p))$ suited to ϕ such that λ is $\eta^{\vec{y}}_{\vec{\psi}|\vec{P}}$.*

PROOF. We have a stipulation blueprint $(\{\theta_1(\vec{u^1})|P_1, \ldots \theta_m(\vec{u^m})|P_m\}, (v_1, \ldots, v_p))$ such that λ is $\eta^{\vec{v}}_{\vec{\theta}|\vec{P}}$, and can let n_i be the arity of P_i. We just need to do some variable relabelling. Let $y_1, \ldots, y_p$ be distinct variables which do not occur in ϕ or and θ_i. For each i let $x^i_1, \ldots, x^i_{n_i}$ be variables distinct from each other and the y_j, and which also do not occur in ϕ or in any θ_i. Finally for each i let $w^i_1, \ldots, w^i_{l_i}$ be the variables in θ_i which are not among the u^i_j or the v_j, and let $z^i_1, \ldots, z^i_{l_i}$ be variables distinct from each other, from the x^j_k and from the y_j and which also do not occur in ϕ or in any θ_j. Let ψ_i be θ_i but with every occurrence of u^i_j replaced by x^i_j, every occurrence of v_j replaced by y_j, and every occurrence of w^i_j replaced by z^i_j. Then ψ_i is just θ_i with variables relabelled so for all $a_1, \ldots, a_{n_i}, b_1, \ldots, b_p$ we have

$$A, u^i_j \mapsto a_j, v_k \mapsto b_k \vDash \theta_i \text{ iff } A, x^i_j \mapsto a_j, y_k \mapsto b_k \vDash \psi_i.$$

Thus $\eta^{\vec{v}}_{\vec{\theta}|\vec{P}} = \eta^{\vec{y}}_{\vec{\psi}|\vec{P}}$. Also ψ_i has no variables in common with ϕ, so certainly $\psi_i(\vec{x^i})$ is substitutable for P_i in ϕ; and the $y_1, \ldots, y_p$ are not free in ϕ. Thus the stipulation blueprint $(\{\psi_1(\vec{x^1})|P_1, \ldots, \psi_m(\vec{x^m})|P_m\}, \vec{y})$ is suited to ϕ. $\square$

We now relate the meaning stipulation $\eta^{\vec{y}}_{\vec{\psi}|\vec{P}}$ to the result of carrying out the substitution.

LEMMA 12. *Suppose that* $(\{\psi_1(\vec{x^1})|, P_1, \ldots \psi_m(\vec{x^m})|P_m\}, y_1, \ldots, y_p)$ *is suited to* ϕ. *Let* $z_1, \ldots, z_k$ *be the free variables of* ϕ. *Then for any* $a_1, \ldots, a_k, b_1, \ldots, b_p$ *we have*

$$A, z_i \mapsto a_i, y_j \mapsto b_j \vDash_1 \phi[\vec{\psi}|\vec{P}]$$

if and only if $A, z_i \mapsto a_i, \eta^{\vec{y}}_{\vec{\psi}|\vec{P}}(b_1, \ldots, b_p) \vDash \phi$.

PROOF. All the placeholders of ϕ are among $P_1, \ldots, P_m$, and the ψ_i are placeholder free, so the trivial meaning stipulation $\lambda_\varnothing$ is a meaning stipulation for $(\mathrm{Plc}(\phi) \setminus \{P_1, \ldots, P_m\}) \cup \bigcup_i \mathrm{Plc}(\psi_i)$. Thus by Lemma 3 we have

$$A,\ v,\ \lambda_\varnothing \vDash \phi[\vec{\psi}|\vec{P}] \text{ iff } A,\ v, g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda_\varnothing)(v(y_1), \ldots, v(y_p)) \vDash \phi.$$

Thus we obtain

$$\begin{aligned} A, z_i \mapsto a_i, y_j \mapsto b_j \vDash_1 \phi[\vec{\psi}|\vec{P}] &\text{ iff } A, z_i \mapsto a_i, y_j \mapsto b_j, \lambda_\varnothing \vDash \phi[\vec{\psi}|\vec{P}] \\ &\text{ iff } A, z_i \mapsto a_i, y_j \mapsto b_j, g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda_\varnothing)(b_1, \ldots, b_p) \vDash \phi \\ &\text{ iff } A, z_i \mapsto a_i, g^{\vec{y}}_{\vec{\psi}|\vec{P}}(\lambda_\varnothing)(b_1, \ldots, b_p) \vDash \phi \\ &\text{ iff } A, z_i \mapsto a_i, \eta^{\vec{y}}_{\vec{\psi}|\vec{P}}(b_1, \ldots, b_p) \vDash \phi \end{aligned}$$

as claimed. $\square$

We use this lemma to show that a meaning dependence $g^{\vec{z}}_{\chi(\vec{x})|Q}$ maps definable meaning stipulations to definable meaning stipulations.

LEMMA 13. *For any meaning dependence $g^{\vec{z}}_{\chi(\vec{w})|Q}$, if λ is a definable meaning stipulation for χ then $g^{\vec{z}}_{\chi(\vec{w})|Q}(\lambda)$ is definable.*

PROOF. Let $\vec{w} = w_1, \ldots, w_l$, $\vec{z} = z_1, \ldots, z_q$.

λ is a definable meaning stipulation for χ so there is a stipulation blueprint $(\{\psi_1(\vec{x^1})|P_1, \ldots \psi_m(\vec{x^m})|P_m\}, (y_1, \ldots, y_p))$ suited to χ with $\lambda = \eta^{\vec{y}}_{\vec{\psi}|\vec{P}}$. Then we have for all $b_1 \ldots b_q, c_1 \ldots c_p$ that

$$g^{\vec{z}}_{\chi(\vec{w})|Q}(\eta^{\vec{y}}_{\vec{\psi}|\vec{P}})(\vec{b}, \vec{c})(Q)$$

is $\{(a_1, \ldots, a_l) \mid A, w_i \mapsto a_i, z_j \mapsto b_j, \eta^{\vec{y}}_{\vec{\psi}|\vec{P}}(\vec{c}) \vDash \chi\}$ which by the previous lemma is $\{(a_1, \ldots, a_l) \mid A, w_i \mapsto a_i, z_j \mapsto b_j, y_k \mapsto c_k \vDash_1 \chi[\psi_1(\vec{x^1})|P_1, \ldots, \psi_m(\vec{x^m})|P_m]\}$ which is $\eta^{z_1,\ldots,z_q,y_1,\ldots,y_p}_{\chi[\vec{\psi}|\vec{P}]|Q}(\vec{b}, \vec{c})(Q)$.

For $P_i \neq Q$ we have that $g^{\vec{z}}_{\chi(\vec{w})|Q}(\eta^{\vec{y}}_{\vec{\psi}|\vec{P}})(\vec{b}, \vec{c})(P_i)$ is $\eta^{\vec{y}}_{\vec{\psi}|\vec{P}}(\vec{c})(P_i)$, which is $\eta^{z_1,\ldots,z_q,y_1,\ldots,y_p}_{\vec{\psi}|\vec{P}}(\vec{b}, \vec{c})(P_i)$.

Thus $g^{\vec{z}}_{\chi(\vec{w})|Q}(\eta^{\vec{y}}_{\vec{\psi}|\vec{P}})$ is η^{σ}, where σ is

$$(\{(\psi_i(\vec{x^i}), P_i) : P_i \neq Q\} \cup \{(\chi[\vec{\psi}|\vec{P}], Q)\}, (z_1, \ldots, z_q, y_1, \ldots, y_p)),$$

so $g^{\vec{z}}_{\chi(\vec{w})|Q}(\eta^{\vec{y}}_{\vec{\psi}|\vec{P}})$ is a definable meaning stipulation. $\square$

Thus we could have stated the soundness theorem in terms of definable meaning stipulations. If we require that the meaning stipulation λ for the final formula of a derivation is definable, then we can extend it to a definable meaning stipulation $\overline{\lambda}$ which covers every placeholder occurring in the derivation. It then follows from this lemma that every $M^{\Phi,\overline{\lambda}}_i$ is definable (by induction down each chain). Thus all the elements of each $S^{\overline{\lambda}}(\phi)$ are definable.

It will be helpful to state a version of the soundness theorem in terms of stipulation blueprints though. We need a notion of extension of stipulation blueprints corresponding to the notion of extension of meaning stipulations. So suppose $\sigma = (\{\psi_1(\vec{x^1})|P_1, \ldots \psi_m(\vec{x^m})|P_m\}, \vec{y})$ is a stipulation blueprint, and $\mathcal{P}$ a set of placeholders. A stipulation blueprint τ is an **extension of σ to $\mathcal{P}$** if τ is of the form $(\{\psi_1(\vec{x^1})|P_1, \ldots \psi_m(\vec{x^m})|P_m\} \cup$

$S, \vec{y})$, with every placeholder Q in $\mathcal{P}$ occurring in a pair $(\chi(\vec{x})|Q)$ in S. Note that this requires that every pair $(\theta(\vec{v})|Q')$ in S has that every free variable in θ beyond the v_i is among the y_j. Obviously every stipulation blueprint has an extension to any finite set of placeholders.

We define a notion of entailment for pairs (ϕ, σ) with σ a stipulation blueprint suited to ϕ. Recall that η^σ denotes the definable meaning stipulation obtained from stipulation blueprint σ. Then for a structure A and variable assignment v over A we define $A, v \vDash (\phi, \sigma)$ to hold when $A, v, \eta^\sigma \vDash \phi$ holds. If K is a set of such pairs, then we define $K \vDash (\phi, \sigma)$ to hold when for all structures A and variable assignments v over A such that $A, v \vDash (\theta, \tau)$ for all $(\theta, \tau) \in K$, we have $A, v \vDash (\phi, \sigma)$.

If ϕ is a formula we let $\mathcal{D}(\phi)$ be the set of pairs (ϕ, σ) where σ is a stipulation blueprint suited to ϕ. We write $\mathcal{D}_{\mathcal{L}}$ when we want to specify the language in use. If Γ is a set of formulae then we let $\mathcal{D}(\Gamma)$ be $\bigcup_{\gamma \in \Gamma} \mathcal{D}(\gamma)$.

Now we can state a version of the soundness theorem in terms of these stipulation blueprints.

THEOREM 14. *Let D be a derivation in schematic logic with final line $(\Gamma; \Delta) \Rightarrow \chi$. Let σ be a stipulation blueprint suited to χ. Let $\overline{\sigma}$ be an extension of σ to the set of placeholders which appear in D. Then $\mathcal{D}(\Gamma) \cup (\Delta \times \{\overline{\sigma}\}) \vDash (\chi, \sigma)$.*

PROOF. Let D be a derivation of χ from Γ. Pick a structure A, and let λ be η^σ with respect to A. Then $\overline{\lambda} = \eta^{\overline{\sigma}}$ is an extension of λ to the set of placeholders in A, so we obtain by Theorem 10 that $S^{\overline{\lambda}}(\Gamma) \cup (\Delta \times \{\overline{\lambda}\}) \vDash_A (\chi, \lambda)$. As remarked after Lemma 13, we have that all meaning stipulations in each $S^{\overline{\lambda}}(\gamma)$ are definable: thus $\{(\gamma, \kappa) \mid \gamma \in \Gamma$, κ a definable meaning stipulation for $\gamma\} \cup \Delta\{\overline{\lambda}\} \vDash_A (\chi, \lambda)$, so

$$\{(\gamma, \eta^\tau) \mid \gamma \in \Gamma,\, \tau \in \mathcal{D}(\gamma)\} \cup \Delta\{\eta^{\overline{\sigma}}\} \vDash_A (\chi, \eta^\sigma)$$

so since A was arbitrary, we obtain

$$\mathcal{D}(\Gamma) \cup \Delta\{\overline{\sigma}\} \vDash (\chi, \sigma). \qquad \square$$

We are now approaching the conservativeness theorem, but we need some more preliminary definitions. First we need a notation for certain universally closed first order substitution instances of a formula. Let ϕ be a formula of schematic logic, and let $(\{\psi_1(\vec{x^1})|P_1, \dots \psi_m(\vec{x^m})|P_m\}, \vec{y})$ be a stipulation blueprint suited to ϕ. Then we define $\forall \vec{y}\, \phi[\vec{\psi}|\vec{P}]$ to be

$$\forall y_1, \dots, \forall y_p\, (\phi[\psi_1(x_1^1, \dots, x_{n_1}^1)|P_1, \dots, \psi_m(x_1^m, \dots, x_{n_m}^m)|P_m]).$$

If $\sigma = (\{\psi_1(\vec{x^1})|P_1, \dots \psi_m(\vec{x^m})|P_m\}, \vec{y})$, we will also use the notation $\phi[\sigma]$ for this.

We let $F(\phi)$ be the set of first order formulae of this form for given ϕ, written $F_{\mathcal{L}}(\phi)$ when we want to specify the language $\mathcal{L}$ in use. If Γ is a set of formulae then we let $F(\Gamma)$ be $\bigcup_{\gamma \in \Gamma} F(\gamma)$.

This $F(\phi)$ is exactly the normal first order scheme associated with ϕ: for instance if ϕ_{ind} is $(P(0) \wedge \forall x\,(P(x) \rightarrow P(Sx)) \rightarrow \forall x\, P(x)$ then $F(\phi_{\text{ind}})$ is exactly the normal first order induction scheme (as described for example in Cori and Lascar, 2001, p. 65).

Note that for any $\forall \vec{y}\, \phi[\vec{\psi}|\vec{P}] \in F(\phi)$ we have $(\{\phi\}; \varnothing) \vdash \forall \vec{y}\, \phi[\vec{\psi}|\vec{P}]$: one can deduce $\phi[\psi_1(x_1^1, \dots, x_{n_1}^1)|P_1, \dots, \psi_m(x_1^m, \dots, x_{n_m}^m)|P_m]$ by repeated substitutions, and then deduce $\forall \vec{y}\, \phi[\vec{\psi}|\vec{P}]$ by repeated $\forall$-introduction, since by definition the y_j are not free in ϕ.

The following lemma is simple but useful.

LEMMA 15. *Let ϕ be a formula, σ a stipulation blueprint suited to it. Then for any variable assignment v, we have $A, v \vDash (\phi, \sigma)$ iff $A, v \vDash_1 \phi[\sigma]$.*

PROOF. Let $\sigma = (\{\psi_1(\vec{x^1})|P_1, \dots \psi_m(\vec{x^m})|P_m\}, \vec{y})$ and let $z_1, \dots, z_k$ be the free variables of ϕ. Using Lemma 12, we have for any $a_1, \dots, a_k \in A$ that

$$A, z_i \mapsto a_i, \eta^{\vec{y}}_{\vec{\psi}|\vec{P}} \vDash \phi$$

$$\text{iff } A, z_i \mapsto a_i, \eta^{\vec{y}}_{\vec{\psi}|\vec{P}}(b_1, \dots, b_p) \vDash \phi \text{ for all } b_1, \dots, b_p$$

$$\text{iff } A, z_i \mapsto a_i, y_j \mapsto b_j \vDash_1 \phi[\psi_1(\vec{x^1})|P_1, \dots, \psi_m(\vec{x^m})|P_m] \text{ for all } b_1, \dots, b_p$$

$$\text{iff } A, z_i \mapsto a_i \vDash_1 \forall y_1, \dots, \forall y_p\, (\phi[\psi_1(\vec{x^1})|P_1, \dots, \psi_m(\vec{x^m})|P_m])$$

which is the desired equivalence. □

Immediate, from Proposition 15 we obtain:

COROLLARY 16. *Let K be a set of pairs (θ, τ) with τ a stipulation blueprint suited to θ. Let (ϕ, σ) be another such pair. Then $K \vDash (\phi, \sigma)$ if and only if $\{\theta[\tau] \mid (\theta, \tau) \in K\} \vDash_1 \phi[\sigma]$.*

Immediate, from Corollary 16 we have:

COROLLARY 17. *Let Γ and Δ be sets of formulae of schematic logic, ϕ another formula. Let σ be a stipulation blueprint suited to ϕ and to every element of Δ. Then*

$$\mathcal{D}(\Gamma) \cup (\Delta \times \{\sigma\}) \vDash (\phi, \sigma) \textit{ if and only if } F(\Gamma) \cup \{\delta[\sigma] \mid \delta \in \Delta\} \vDash_1 \phi[\sigma]$$

Note that this lemma can only apply to a set Δ of formulae if Δ contains finitely many placeholders, since otherwise no stipulation blueprint will be suited to all its members.

We now obtain our conservativeness theorem. Recall the relation $\vdash_1$, which for first order formulae is a deducibility relation in a standard system of first order natural deduction.

THEOREM 18 (Conservativeness of Schematic Logic). *Let Γ be a set of formulae of schematic logic, Δ a set of first order formulae and ϕ a first order formula (all of the same language). If $(\Gamma; \Delta) \vdash \phi$, then $F(\Gamma) \cup \Delta \vdash_1 \phi$.*

PROOF. Suppose that $(\Gamma; \Delta) \vdash \phi$. There are finite subsets Γ' of Γ and Δ' of Δ such that there is some derivation D of $(\Gamma'; \Delta') \Rightarrow \phi$. Let

$$\sigma = (\{\psi_1(\vec{x^1})|P_1, \ldots \psi_m(\vec{x^m})|P_m\}, \vec{y})$$

be a stipulation blueprint which covers every placeholder in D, where each ψ_i actually has no free variables beyond the x_i, and $\vec{y}$ is actually the empty list of variables. Then σ is suited to ϕ, and to every element of Δ'.

By Theorem 14, we have

$$\mathcal{D}(\Gamma) \cup (\Delta \times \{\sigma\}) \vDash (\phi, \sigma)$$

so by the previous lemma

$$F(\Gamma) \cup \{\delta[\sigma] \mid \delta \in \Delta\} \vDash_1 \phi[\sigma]$$

so by first order completeness

$$F(\Gamma) \cup \{\delta[\sigma] \mid \delta \in \Delta\} \vdash_1 \phi[\sigma].$$

But ϕ contains no placeholders so for each $\theta \in \Delta \cup \{\phi\}$, $\forall \vec{y}\, \theta[\vec{\psi}|\vec{P}]$ is just a universal closure of θ, prefixed by the quantifiers $\forall y_1 \ldots \forall y_n$ corresponding to the list $\vec{y}$: since the list $\vec{y}$ is the empty list, this means $\forall \vec{y}\, \theta[\vec{\psi}|\vec{P}]$ is just θ. Thus

$$F(\Gamma) \cup \Delta \vdash_1 \phi$$

as claimed. $\square$

Otherwise one can give a syntactic proof of this, in which one can show that every derivation in schematic logic has a similar derivation in first order logic, following the same argument except that one may have to repeat pieces of reasoning in the first order logic: the substitution rule for schematic logic allows one to derive a result using a placeholder and then draw various different conclusions from it immediately as substitution instances, whereas in first order logic one may have to repeat the argument for each desired conclusion. Feferman (1991, p. 7) gives a quick syntactic proof for the case of a language with a single schematic variable.

The converse to Theorem 18 also holds. For any $f \in F(\gamma)$ we have that f is derivable from γ in schematic logic, by the remark following the definition of F. Thus if $F(\Gamma) \vdash_1 \phi$, then $\Gamma \vdash \phi$, by Proposition 9.

8. Completeness

We now define an entailment relation for formulae of schematic logic, and prove that the deductive system is complete with respect to this notion of entailment. As far as the author is aware, this is the first completeness theorem that has been proved for the logic.

We will not worry as much about ontological innocence here as above. Proving a completeness theorem requires the existence of sufficient structures for the theorem to hold, so one generally cannot proceed in a completely ontologically innocent way. We will quantify over languages as well as structures. The resources needed are similar to those required for proving the completeness of first order logic (essentially a countable infinity of objects is required), so in as much as one can believe the completeness of first order logic without incurring significant ontological commitments the same is true for schematic logic.

For the purposes of the completeness theorem we need the possibility of making meaning stipulations for infinitely many placeholders. A **blueprint limit** is a sequence $(\sigma_n | n \in \mathbb{N})$ of blueprints such that if $j \geqslant i$ then σ_j extends σ_i, and such that every placeholder is covered by some σ_i. We say a blueprint limit $(\sigma_n \mid n \in \mathbb{N})$ extends a stipulation blueprint τ if for every n we have that σ_n extends τ (equivalently if σ_0 extends τ). It is easy to see that for every stipulation blueprint there is a blueprint limit extending it. For a blueprint limit σ and formula ϕ we let $\sigma(\phi)$ be σ_n where n is minimal such that σ_n covers the placeholders of ϕ. Wet let $\phi[\sigma]$ be $\phi[\sigma(\phi)]$, i.e. $\phi[\sigma_n(\phi)]$ where n is minimal such that σ_n covers the placeholders of ϕ. For a set Δ of formulae we let $\Delta * \sigma$ be $\{(\phi, \sigma(\phi)) \mid \phi \in \Delta\}$. Immediate from Corollary 16 we have:

PROPOSITION 19. *Let* Γ *and* Δ *be formulae of schematic logic,* ϕ *a formula of schematic logic,* σ *a stipulation blueprint suited to* ϕ*, and* τ *a blueprint limit extending* σ*. Then*

$$\mathcal{D}(\Gamma) \cup (\Delta * \tau) \vDash (\phi, \sigma) \text{ if and only if } F(\Gamma) \cup \{\delta[\tau] \mid \delta \in \Delta\} \vDash_1 \phi[\sigma]$$

In the previous section we defined an entailment relation for pairs (ϕ, σ) with σ a stipulation blueprint suited to ϕ. We use this to define a notion of entailment for bare formulae of schematic logic. For this we need to quantify over languages.

Fix a language $\mathcal{L}$ for schematic logic. Let Γ and Δ be sets of formulae of $\mathcal{L}$, and let ϕ be another formula of $\mathcal{L}$. We define $(\Gamma; \Delta) \vDash \phi$ to hold when for every $\mathcal{L}'$ extending $\mathcal{L}$, for every $\sigma \in \mathcal{D}_{\mathcal{L}'}(\phi)$, and every blueprint limit τ (over $\mathcal{L}'$) extending σ we have

$$\mathcal{D}_{\mathcal{L}'}(\Gamma) \cup \Delta * \tau \vDash (\phi, \sigma).$$

It is easy to see that the deductive system is sound with respect to this notion of entailment. Indeed, suppose that $(\Gamma; \Delta) \vdash \phi$, and let $\mathcal{L}'$ extend $\mathcal{L}$. There are finite subsets Γ' of Γ and Δ' of Δ such that $(\Gamma'; \Delta') \Rightarrow \phi$ is derivable via some derivation D in $\mathcal{L}$, and thus in $\mathcal{L}'$. Let $\sigma \in \mathcal{D}_{\mathcal{L}'}$, and let τ be a blueprint limit extending σ. Pick τ_n such that every placeholder in D is covered by τ_n. Then by Theorem 14 we have $\mathcal{D}_{\mathcal{L}'}(\Gamma) \cup (\Delta' \times \{\tau_n\}) \vDash (\phi, \sigma)$. For any $\delta \in \Delta'$ we have that $\tau(\delta)$ agrees with τ_n on the placeholders in δ; thus if A is a structure and v a variable assignment over A then $A, v \vDash (\Delta' \times \{\tau_n\})$ if and only if $A, v \vDash \Delta' * \tau$. Hence we obtain $\mathcal{D}_{\mathcal{L}'}(\Gamma) \cup \Delta' * \tau \vDash (\phi, \sigma)$, so $\mathcal{D}_{\mathcal{L}'}(\Gamma) \cup \Delta * \tau \vDash (\phi, \sigma)$. Since $\mathcal{L}'$, σ and τ were arbitrary, it follows that $(\Gamma; \Delta) \vDash \phi$ as claimed.

The completeness theorem is the converse to this.

THEOREM 20 (Completeness of Schematic Logic). *Suppose that* $(\Gamma; \Delta) \vDash \phi$ *in schematic logic. Then* $(\Gamma; \Delta) \vdash \phi$*.*

PROOF. Fix a language $\mathcal{L}$, with Γ and Δ sets of formulae of $\mathcal{L}$ and ϕ a formula of $\mathcal{L}$ such that $\Gamma; \Delta \vDash \phi$. Pick some enumeration $P_1, P_2, \ldots$ of all the placeholders, and let $\mathcal{L}'$ be the language obtained be extending $\mathcal{L}$ with a countably infinite list $R_1, R_2, \ldots$ of new relation symbols, with R_i having the same arity as P_i. Let n_i be the arity of P_i (and R_i) and for each i let $x^i_1, \ldots, x^i_{n_i}$ be any distinct variables. Let ψ_i be $R_i(x^i_1, \ldots, x^i_{n_i})$.

Then as in Proposition 8, for any θ, $\psi_i(\vec{x^i})$ is substitutable for P_i in θ, with $\theta[\psi_i(\vec{x^i})|P_i] = \theta[R_i|P_i]$, the formula obtained from θ by replacing every occurrence of P_i with R_i (this is the point where having the wide version of the substitution rule is useful).

Pick c such that every placeholder P_i in ϕ has $i \leqslant c$, and let τ_k be the stipulation blueprint $(\{\psi_1(\vec{x^1})|P_1, \ldots, \psi_{k+c}(\vec{x^{k+c}})|P_{k+c}\}, \vec{y})$ where $\vec{y}$ is the empty list of variables. Then each τ_k is suited to every formula θ whose placeholders it covers. For any such θ we have $\theta[\tau_k] = \theta[\vec{R}|\vec{P}]$, the result of replacing every occurrence of each P_i in θ by R_i. In particular, for any θ we have $\theta[\tau(\theta)] = \theta[\vec{R}|\vec{P}]$.

Also if $l \geqslant k$ then τ_l extends τ_k, and every P_i is covered by some τ_k, so the sequence τ is a stipulation blueprint. Indeed it is a stipulation blueprint extending τ_0, which is a stipulation blueprint suited to ϕ.

Since $(\Gamma; \Delta) \vDash \phi$ we thus have

$$\mathcal{D}_{\mathcal{L}'}(\Gamma) \cup \Delta * \tau \vDash (\phi, \tau_0).$$

By Proposition 19 we deduce

$$F(\Gamma) \cup \{\delta[\tau] \mid \delta \in \Delta\} \vDash_1 \phi[\tau_0].$$

In other words

$$F(\Gamma) \cup \{\delta[\vec{R}|\vec{P}] \mid \delta \in \Delta\} \vDash_1 \phi[\vec{R}|\vec{P}],$$

so by first order completeness

$$F(\Gamma) \cup \{\delta[\vec{R}|\vec{P}] \mid \delta \in \Delta\} \vdash_1 \phi[\vec{R}|\vec{P}].$$

Let D be any first order derivation of $\phi[\vec{R}|\vec{P}]$ from

$$F(\Gamma) \cup \{\delta[\vec{R}|\vec{P}] \mid \delta \in \Delta\}.$$

Replacing every occurrence of each symbol R_i in D by P_i gives us a valid derivation of $\phi = \phi[\vec{R}|\vec{P}][\vec{P}|\vec{R}]$ from

$$\begin{aligned}&\{f[\vec{P}|\vec{R}] \mid f \in F_{\mathcal{L}'}(\Gamma)\} \cup \{\delta[\vec{R}|\vec{P}][\vec{P}|\vec{R}] \mid \delta \in \Delta\}\\ &= \{f[\vec{P}|\vec{R}] \mid f \in F_{\mathcal{L}'}(\Gamma)\} \cup \Delta,\end{aligned}$$

in the first order portion of our deductive system. This derivation only uses symbols from $\mathcal{L}$, so we can consider it to be a derivation over $\mathcal{L}$. Thus we have

$$\{f[\vec{P}|\vec{R}] \mid f \in F_{\mathcal{L}'}(\gamma) \text{ for some } \gamma \in \Gamma\} \cup \Delta \vdash_1 \phi$$

over $\mathcal{L}$.

Let $\gamma[\rho]$ be an element of $F_{\mathcal{L}'}(\Gamma)$. Then $\gamma[\rho][\vec{P}|\vec{R}]$ is of the form

$$\forall z_1, \ldots, \forall z_p\, (\gamma[\chi_1(\vec{w^1})|Q_1, \ldots, \chi_k(\vec{w^k})|Q_k][\vec{P}|\vec{R}]),$$

with the z_i not free in γ, the Q_i being all the placeholders in γ and the χ_i containing no placeholders (but possibly containing the relation symbols R_i). We will show that this can be deduced from general premise γ.

Letting χ'_i be $\chi_i[\vec{P}|\vec{R}]$, χ'_i is substitutable for Q_i in γ since χ_i is. Thus the multiple substitution $\gamma[\chi'_1(\vec{w^1})|Q_1, \ldots, \chi'_k(\vec{w^k})|Q_k]$ is defined. γ is a formula of $\mathcal{L}$ so contains none of the R_i, so in fact we have

$$\gamma[\chi_1(\vec{w^1})|Q_1, \ldots, \chi_k(\vec{w^k})|Q_k][\vec{P}|\vec{R}] = \gamma[\chi'_1(\vec{w^1})|Q_1, \ldots, \chi'_k(\vec{w^k})|Q_k],$$

which by Proposition 2 can be obtained from γ by a series of uses of the (single) substitution rule. We then obtain $\gamma[\rho][\vec{P}|\vec{R}]$ by repeated $\forall$-introduction, since none of the z_i are free in γ. Thus $(\{\gamma\}; \varnothing) \vdash \gamma[\rho][\vec{P}|\vec{R}]$, as claimed.

This means that for all $\chi \in \{f[\vec{P}|\vec{R}] \mid f \in F_{\mathcal{L}'}(\Gamma)\} \cup \Delta$ we have $(\Gamma; \Delta) \vdash \chi$. Since $\{f[\vec{P}|\vec{R}] \mid f \in F_{\mathcal{L}'}(\Gamma)\} \cup \Delta \vdash_1 \phi$, by Proposition 9, we have $(\Gamma; \Delta) \vdash \phi$. $\square$

By putting soundness and completeness together, we can see that the above definition of the entailment relation $\Gamma \vDash \phi$ for schematic logic can be replaced by an apparently stronger definition. We have that $\Gamma \vDash \phi$ over $\mathcal{L}$ if and only for any extension $\mathcal{L}'$ of $\mathcal{L}$, for any stipulation blueprint σ suited to ϕ, there is a finite set K of pairs (γ, τ) with τ suited to $\gamma \in \Gamma$ such that $K \vDash (\phi, \sigma)$. In other words, $\Gamma \vDash \phi$ if and only if however we extend our language, and whatever (definable) meaning we stipulate for the conclusion of the derivation, under a finite number of meaning stipulations for the premises of the derivations those premises entail the conclusion.

9. Schematic logic as the logic of mathematics

Since we have an ontologically innocent justification of the use of schematic logic, and a completeness theorem for the logic, it has potential for being the logic we should look to when formalizing our mathematical arguments. The main objections to using second order logic in mathematics are that it comes with an ontological burden, and that its standard semantics is not complete; neither objection applies here. Plural logic is another option, but again under its standard semantics it is not complete, and it does not natively allow for second order variables of arity above 1. Schematic logic does not have the same expressive power as second order or plural logic, but it does have some extra expressive power over first order logic, which can come in useful when formalizing some mathematical results: it allows what would normally be axiom schemes and theorem schemes, dealt with in the metalanguage, to be expressed natively within the logic.

Our first example is from arithmetic. Many mathematical theories can be naturally phrased in schematic logic, and arithmetic is one of them. As mentioned earlier in Section 5, in the context of arithmetic we could give an argument for induction as follows: if P holds for 0, and it holds for $S(x)$ whenever it holds for x, then it holds for 0, and thus also for $S(0)$, and thus also for $S(S(0))$ and so on, and thus for all natural numbers. Thus we can establish the statement

$$(P(0) \wedge \forall x\, (P(x) \rightarrow P(S(x)))) \rightarrow \forall x P(x)$$

involving placeholder P with unspecified meaning. Together with the other usual axioms, this gives us $\mathrm{PA}_{\mathrm{Schem}}$, the schematic theory of arithmetic. The end of Section 2 addresses possible worries one might have about the Sorites paradox here (taking P to be a vague predicate such as "small").

Thanks to the conservativeness theorem, we see that the first order consequences of $\mathrm{PA}_{\mathrm{Schem}}$ are just the normal theorems of PA. However it also has other consequences, which are best expressed as theorems in schematic logic. For instance strong induction

$$(\forall x(\forall y\, y < x \rightarrow Q(y)) \rightarrow Q(x)) \rightarrow \forall x\, Q(x)$$

is a theorem of $\mathrm{PA}_{\mathrm{Schem}}$. The derivation parallels the normal first order derivations of instances of strong induction. We can draw instances of strong induction as consequences of the strong induction theorem by substitution. In the first order setting, if we had a proof in which strong induction was used multiple times, then in formalizing the proof we would have to derive individually each instance of strong induction we want to use: we would have to repeat the schematic derivation of strong induction for each substitution instance, with something different substituted for Q each time. The point of schematic logic is that it allows us to reuse reasoning in cases like this, and to avoid having to repeat ourselves. It thus gives a better formalization of mathematical practice than first order logic does. From the first order point of view one has to see mathematicians as working in the metalanguage whenever they prove or use theorem schemes like this, so the logic itself does not handle this standard aspect of mathematical practice.

Set theory is another theory which has a natural axiomatization in schematic logic. When justifying the axioms of set theory one generally starts by telling a story about a hierarchy of stages (as in Boolos, 1971, or Shoenfield, 1982). One then justifies individual axioms based on this. For instance as Boolos says (about a particular ϕ):

> For any stage s [and set z], there is a set of all sets formed at earlier stages, which belong to z and to which ϕ applies. Let s be the stage at which z is formed. All members of z are formed before s. So, for any z, there is a set of just those members of z to which ϕ applies[.]
>
> (Boolos, 1971, p. 226)

This works just as well as a statement with a placeholder P:

> For any stage s and set z, there is a set of all sets x formed at earlier stages, which belong to z and such that $P(x)$. Let s be the stage at which z is formed. All members of z are formed before s. So, for any z, there is a set of just those members x of z such that $P(x)$.

Thus we obtain the axiom of separation

$$\forall z\, \exists y\, \forall x\, (x \in y \leftrightarrow (x \in z \wedge P(x)))$$

containing placeholder P with unspecified meaning. Similarly justifications for the axiom scheme of replacement can generally be rephrased as justifications for a placeholder axiom of replacement. Together with the other usual axioms of ZFC, the placeholder axioms of separation and replacement give ZFC_{Schem} — schematic set theory.

As with arithmetic, there are results of set theory that are best phrased in schematic logic. For instance $\in$-induction

$$\forall x\, (\forall\, y(y \in x \to P(y)) \to P(x)) \to \forall x P(x) \qquad (*)$$

is a theorem of ZFC_{Schem}; there is no need to phrase it as a theorem scheme, i.e. as a metatheorem. Again, schematic logic gives a better formalization of the reasoning mathematicians actually employ. In first order logic, one would have to individually derive every instance of $\in$-induction one wanted to use, repeating a substituted version of the schematic derivation each time; or one could move to the metatheory (in which case one is not really working in first order logic any more).

As these examples show, the extra expressive power of schematic logic over first order logic is useful in various situations in mathematics; and since the logic can be justified in an ontologically innocent way and comes with a completeness theorem, the usual objections to second order logic do not apply. Also it seems that the use of axiom and theorem schemes is the main way that mathematicians leave the confines of first order logic and use reasoning that is most naturally seen as second order (involving Π^1_1 second order statements): more of second order logic beyond the fragment seen in schematic logic does not appear to be needed.

To use the logic as the language of mathematics, we do not need to actually mention any of the issues discussed in this paper involving placeholders or meaning stipulations. Mathematicians just need to be able to reason informally using the logic, which is very easy — the only extra rule is the substitution rule. We do not require that mathematicians be fluent with the semantics for first order logic before we formalize their proofs in it.

Because of examples like the above we can in some cases give much quicker proofs in schematic logic than in first order logic. For instance if we had an argument which required multiple instances of $\in$-induction, then in schematic logic we could prove this single theorem ($*$) of ZFC_{Schem}, and then draw each instance of $\in$-induction needed as a consequence in a single line via the substitution rule; whereas if we were to strictly work in first order logic, we might have to repeat the derivation of $\in$-induction for each substitution instance of ($*$) we required. Thus it is possible that there is non trivial speed up when moving from first order logic to schematic logic, a question for future investigation.

10. Open-endedness

One important aspect of the use of placeholders and schematic logic is that it naturally leads us to see that our axioms involving placeholders are and should be open-ended, in the sense that if we expand our language to include new vocabulary then we in the same way expand the range of substitutions we allow for placeholder letters. This is a consequence of the way placeholder letters are used: we do not specify in advance any particular range for them, instead first giving an argument involving them and only later deciding to take them to mean certain things. Thus if we give an argument involving placeholders and then expand our language to include new vocabulary, we have just as much right to take the placeholders to mean items of (or built out of) the new vocabulary as we do to take them to mean items of the original language. We do need to be careful of running into problems such as the Sorites paradox, as discussed earlier; but a problem like this can arise for vocabulary of the original language just as much as it can for an expanded language.

This open-endedness is already built into schematic logic. If we establish a statement of the logic involving a placeholder, such as the statement of induction

$$(P(0) \wedge \forall x\,(P(x) \to P(S(x)))) \to \forall x P(x),$$

and we expand our language to include new vocabulary, then the statement of induction is still a statement of the expanded language; and now by applying the substitution rule in the expanded language we can substitute formulae involving new vocabulary (as well as old vocabulary) for the placeholder P. The range of substitutions allowed for placeholders depends on the language being used, as it should given the informal motivation. Formally, if we establish an entailment $(\Gamma; \Delta) \vdash \phi$ in a language, and then expand this language to include new vocabulary, then the entailment still holds in the expanded language; and we can then take placeholders involved to mean formulae involving new vocabulary as well as old. As seen in Section 8 it is in terms of this open-ended sense of entailment that the logic is complete, showing how suited the rules of the logic are to this open-ended interpretation.

11. Conclusion

We have seen that the placeholder interpretation of schematic logic allows us to give an ontologically innocent soundness theorem for the logic, which only requires the existence of formulae we actually write down in the course of a derivation.

This means that the logic is available as a setting in which to do mathematics, with minimal cost and with the advantage over first order logic of being able to natively formalize axiom and theorem schemes, without having to resort to the metatheory.

Additionally, the approach to the logic seen here, and the completeness theorem, naturally support a view of schematic statements as open-ended. This supports the work of various authors (Lavine 1998, pp. 224–240; McGee 1997, pp. 56–62; Parsons 2007, pp. 290–293) who have appealed to this open-ended interpretation of the logic to address issues of determinacy.

Acknowledgements. I wish to thank Leon Horsten for his helpful and insightful comments on versions of this paper. This work was supported by AHRC grant AH/L503939/1.

References

Boolos, George, 1971, "The iterative conception of set", *The Journal of Philosophy* 68 (8): 215–231. URL http://www.jstor.org/stable/2025204. DOI: 10.2307/2025204

Burgess, John P., and Gideon Rosen, 1997, *A Subject With No Object: Strategies for Nominalistic Interpretation of Mathematics*, Oxford University Press: Oxford, New York.

Cori, René, and Daniel Lascar, 2001, *Mathematical Logic: A Course with Exercises Part II*, OUP Oxford: Oxford.

Feferman, Solomon, 1991, "Reflecting on incompleteness", *The Journal of Symbolic Logic* 56 (1): 1–49. URL http://www.jstor.org/stable/2274902. DOI: 10.2307/2274902

Field, Hartry, 1980, *Science Without Numbers*, Princeton University Press: Princeton, N.J.

Heck, Richard Kimberly, 2011, *Frege's Theorem*, OUP Oxford: Oxford, New York. (Originally published under the name "Richard G. Heck, Jr".)

Lavine, Shaughan, 1998, *Understanding the Infinite*, new edition, Harvard University Press: Cambridge, Mass.

Maddy, Penelope, 1997, *Naturalism in Mathematics*, Clarendon Press.

McGee, Vann, 1997, "How we learn mathematical language", *The Philosophical Review* 106 (1): 35–68. URL http://www.jstor.org/stable/2998341. DOI: 10.2307/2998341

Parsons, Charles, 2007, *Mathematical Thought and its Objects*, Cambridge University Press.

Quine, W. V., 1999, "Reply to Charles Parsons", in pages 396–403 Lewis Edwin Hahn and Paul Arthur Schilpp (eds.), *The Philosophy of W. V. Quine, Volume 18*, Open Court Publishing Co: La Salle, Ill.

Shapiro, Stewart, 2000, *Foundations without Foundationalism: A Case for Second-order Logic*, new edition, Oxford University Press: Oxford, USA.

Shoenfield, J., 1982, "Axioms of set theory", in *Handbook of Mathematical Logic*, North Holland: Amsterdam, New York.

Oliver Tatton-Brown
Department of Philosophy
University of Bristol, United Kingdom
o.tatton-brown@bristol.ac.uk