On Height and Happiness

Melvin Fitting
Department of Philosophy
CUNY Graduate Center
365 Fifth Avenue
New York, NY 10016

June 3, 2015

Abstract

Predicate abstraction provides very simple formal machinery that represents some of the ideas connecting intension and extension. It has been investigated and applied for many years now. Here we do two things with it. First, we present some natural examples, and show how predicate abstraction can be applied to them. Second, we give a formal proof procedure for predicate abstraction, based on *nested sequents*. Soundness and completeness proofs are sketched.

1 Introduction

One simple and natural way of enhancing propositional modal logic is via *predicate abstraction*, which goes back to [17, 18]. It disambiguates many of the perceived syntactic and semantic problems with modal logic, and does so with a certain amount of intuitive appeal. The basic idea behind predicate abstraction in modal logics is elementary, but like first-order logic itself, there are many concrete versions of it. With first-order logic(s) one can have constant symbols, or not, function symbols, or not, multiple sorts, or not. A similar range of choices is available for predicate abstraction. Here we introduce the minimal machinery to handle the kinds of problems we discuss.

We begin with some examples of commonly perceived modal problems, and discuss how predicate abstraction deals with them. Though there are domains, predicate symbols, and function symbols, there is no quantification. Domains are constant. Type complexity is minimal. Think of this presentation as something that can be expanded on, as needed. All the really important ideas are already present. After the examples, there is a formal presentation of the syntax and semantics. Then we go on to formulate a nice nested sequent calculus for predicate abstraction. Up to this point the work presented is not new, though we hope the discussion is simple and enlightening. The nested sequent calculus is both new and not new—a remark that will be elaborated then and not now.

In 2010, Kai Wehmeier gave a talk at the CUNY Graduate Center. That talk (and presumably many others) crystalized into *Subjunctivity and Cross-World Predication*, [19]. His presentation, and the discussion it initiated, were the origins of the present paper. His way of handling the problems discussed here is very different than ours. In a sense, ours is more traditional. His approach involves the introduction of symbols that are designed to represent relations between objects across worlds. It is very natural, especially from a linguistic point of view, and his paper is highly recommended. The two approaches are not in competition, but are rather complementary. It would not be surprising if there were natural embeddings between the two versions, but an investigation must wait until another day.

2 Examples

Suppose we are talking about the King (or Queen) of Sweden. We can think of the monarch as represented by a non-rigid constant symbol, say m, designating different persons at different times, or nobody at some times too. This could be elaborated into a definite description, but such machinery isn't really needed for what is discussed here, and would require the introduction of quantifiers anyway. Suppose also that we wish to express the idea that someday the monarch might be taller than now. We can give a modal operator a temporal reading, so that $\Diamond X$ asserts that X will be true at some future time. Also suppose we have a two-place predicate symbol T whose intended reading is: both arguments are people, and the first is taller than the second. With standard modal machinery, the best formalization we can manage is the absurd $\Diamond T(m, m)$.

It isn't m, the monarchical role or office, that we want to measure for height, it's the particular monarch at a particular time. We want to compare what m designates now with what m designates at some future time. In $\Diamond T(m, m)$, the first m must be understood as designating a person now; the second as designating at a possible future. Predicate abstraction gives us precisely that ability. A formal definition will come later, but the idea is that $\langle \lambda x.X \rangle(c)$ should be true at a state if X is true at that state when x is given the value that c designates at that state. (We'll worry about possible non-designation later on.) Then we can express that someday the monarch will be taller than now as follows.

$$\langle \lambda y.\Diamond \langle \lambda x.T(x, y) \rangle(m) \rangle(m) \tag{1}$$

This says it is true of the person who is presently King or Queen (the value of y) that at some possible future state the taller-than predicate will hold between the person who then holds the position (the value of x) and the present person.

This seems straightforward. But now suppose we want to say about some person, Alice, that she might have been taller than she is (say given different nourishment as a baby than she actually had). We can shift our intended reading of $\Diamond$ so that $\Diamond X$ asserts that X is true in some other possible world, whatever we may mean by possible. And we can represent

Alice by a constant symbol, say a, similarly to our use of m above. But the following sentence, corresponding to (1), won't work.

$$\langle \lambda y. \Diamond \langle \lambda x. T(x, y) \rangle (a) \rangle (a) \tag{2}$$

The problem is that 'Alice' is a name and so a is commonly understood to be a *rigid* constant symbol. We are talking about the same individual a in the present world and in alternative circumstances, and in effect we are saying a person might be taller than themselves. More technically, if a is rigid we have the truth of the following.

$$\langle \lambda y. \Box \langle \lambda x. x = y \rangle (a) \rangle (a) \tag{3}$$

And from (2) and (3) it follows that

$$\langle \lambda x. \Diamond T(x, x) \rangle (a) \tag{4}$$

and this is clearly wrong. (We will give a derivation of (4) from (2) and (3) in Section 8, after we have introduced nested sequents for predicate abstraction.)

There is a way around the problem however, one that applies to the two cases just discussed, though it is overkill for the King of Sweden. When we say that Alice is represented by a rigid designator, a, what does that mean? In some sense a designates the same person in different possible worlds, of course, but in different possible worlds that person might have different inessential properties. In particular, the height of a might be different in different possible worlds, even though a is the same person.

Suppose we introduce a non-rigid function symbol, h, intended to map a person to that person's height, thought of as a number. What h assigns to a depends on the possible world in which $h(a)$ is evaluated. (Of course we need to assume some constancy in our standard of measuring, but let us do so for purposes of the present discussion.) The point is that even though a is rigid, $h(a)$ can vary from possible world to possible world. This happens because an inessential attribute of a can vary from world to world, and this is reflected in the non-rigidity of h, rather than of a. If we assume h is a non-rigid function symbol, and G is the two-place greater-than relation on numbers, our assertion that Alice might have been taller than she is, formalizes as follows.

$$\langle \lambda y. \Diamond \langle \lambda x. G(x, y) \rangle (h(a)) \rangle (h(a)) \tag{5}$$

For the height example above, we don't need the whole real number system as part of our model. In practice we can only distinguish a finite set of height measurements, and these can certainly be built into formal *finite* models. But now, suppose we move from height to happiness. We can say that someday the King of Sweden will be happier than the King is now, rather similarly to (1). But how can we express that Alice might have been happier than she is? If we introduce a function symbol as we did above, we would have to think of it as designating a mapping of individuals to some kind of degrees of happiness, and

assume that these degrees can (often) be compared. The ordering might not be linear—a partial order seems more plausible. Perhaps psychologists really do something like this, but for most of us it seems to expand our horizons a bit. On the other hand, since we generally can make sense out of saying that Alice is happier than Bob, we do seem to be implicitly making use of such machinery anyway. At any rate, there is certainly no bar to a formalization along these lines.

3 Predicate Abstraction Syntax

We begin the precise formulation of predicate abstraction. It is, in a sense, intermediate between propositional and first-order machinery. It makes use of relation symbols, variables, constant and function symbols, but quantifiers need not be involved. A formulation can be given in a variety of ways that differ in some technical details, but these details don't matter all that much. Below we will say something about what could be modified without changing anything essential.

The basic distinction is between *objects* and *intensions*. Very loosely, objects (or extensions) are the entities we bump into: a person, the number 4 (which we bump into in a soft sort of way), and so on. Intensions, on the other hand, are non-rigid designators, often specified by definite descriptions: the number of nations in the UN, my best friend, and so on. Semantically, intensions pick out objects in possible worlds, though perhaps not in all of them. This is actually part of Carnap's method of intension and extension, [3]. As formulated here, equality relates objects, while synonymy relates intensions (though we will not discuss this now). This is enough for us to get started.

There is a family of *object variables*, typically $x, y, \ldots$, and *intension constants*, $a, b, \ldots$. (In [7] we allowed intension variables, but did not allow any constants—it makes no deep difference.) We also have *intension function symbols*, $a, b, \ldots$, of various arities, which take object variables as arguments. Finally, we have *relation symbols*, $P, Q, \ldots$ of various arities, also taking object variables as arguments in the usual way. (In [6] we allowed relation symbols to have intension variables as arguments as well, and positions in relations were typed—see also [7, 8]. A similar generalization could be extended to the arguments of function symbols as well.) We will make use of a special two-place relation symbol, $=$, intended to denote equality. For ease in reading, we will write $x = y$ instead of $= (x, y)$.

An *intension function term* is $a(x_1, \ldots, x_n)$ where $x_1, \ldots, x_n$ are object variables and a is an n-ary intension function symbol. It is convenient for us to assume that intension constants are 0-place intension function symbols, so they are also function terms. Note that intension functions are not allowed to be nested—arguments are object variables—but there is a way around this, discussed below.

An *atomic formula* is $P(x_1, \ldots, x_n)$ where $x_1, \ldots, x_n$ are object variables and P is an n-ary relation symbol. Note that intension function terms don't come into things here. There is a way around this too, also discussed below.

Formulas (and *predicate abstracts*) are defined by recursion. Similarly for *free variable occurrences*.

- An atomic formula is a formula. All variable occurrences are free occurrences.
- If X and Y are formulas, so are $(X \wedge Y)$, $(X \vee Y)$, $(X \supset Y)$. Free variable occurrences in these formulas are the free variable occurrences of X and of Y.
- If X is a formula, so are $\neg X$, $\Box X$, and $\Diamond X$. Free variable occurrences in these formulas are the free variable occurrences of X.
- If X is a formula, x is an object variable, and t is an intension function term, $\langle \lambda x.X \rangle(t)$ is a formula, called a predicate abstract. Free variable occurrences are those of X, except for occurrences of x, together with any free variable occurrences in t.

We finish with some comments and examples. Both $\langle \lambda x.\Diamond P(x) \rangle(a)$ and $\Diamond\langle \lambda x.P(x) \rangle(a)$ are formulas, where a is an intension constant and $P(x)$ is atomic. Neither formula has any free variable occurrences. Predicate abstracts impose a scope distinction, which is something that goes all the way back to Russell in [16, 20]. For him a crucial issue was non-designating terms, in particular definite descriptions. If $X(x)$ is a formula with object variable x free, and a is an intension constant, how should $\neg X(a)$ be read? Does it say that the object designated by a has the $\neg X$ property, or that it fails to have the X property? If a doesn't designate, these come out differently, as the present King of France would know. In predicate abstract notation the difference is syntactically represented by $\langle \lambda x.\neg X(x) \rangle(a)$ and $\neg\langle \lambda x.X(x) \rangle(a)$. Loosely the first says the a object, if any, has the $\neg X$ property, and the second says it fails to have the X property. We will assume that nothing can be correctly predicated of the object designated by a non-designating term, and so $\langle \lambda x.\neg X(x) \rangle(a)$ is false if a does not designate. But then $\langle \lambda x.X(x) \rangle(a)$ is also false, and hence $\neg\langle \lambda x.X(x) \rangle(a)$ is true.

We did not allow intension function symbols to appear in atomic formulas. This simplifies some things, without restricting expressiveness. Suppose a is an intension constant. Instead of $P(a)$, which is not legal, we can write $\langle \lambda x.P(x) \rangle(a)$, which is. We will consider the first as a convenient abbreviation for the second. But we must be careful, since this does not extend to formulas generally, or we would run into an ambiguity with, say, $\Diamond P(a)$. The familiar broad scope/narrow scope distinction is at work here, but by itself it is not sufficient—consider $\Diamond\Diamond P(a)$ for instance, where broad scope and narrow scope are not all the readings there are. Generally, we will not abbreviate unless an explicit proper version is also specified.

We also did not allow nesting of function symbols. This is not a problem because we can understand something line $P(f(g(a)))$ as an abbreviation for

$$\langle \lambda z.\langle \lambda y.\langle \lambda x.P(x) \rangle(f(y)) \rangle(g(z)) \rangle(a)$$

where substitution is made explicit using λ. Note that the two formulas

$$\Diamond\langle\lambda y.\langle\lambda x.P(x)\rangle(f(y))\rangle(a) \tag{6}$$

$$\langle\lambda y.\Diamond\langle\lambda x.P(x)\rangle(f(y))\rangle(a) \tag{7}$$

are not syntactically the same, and will be seen to differ semantically as well. We might abbreviate (6) as $\Diamond P(f(a))$, using a narrow scope reading, but (7) has no corresponding abbreviation.

Finally, in the interest of readability we will sometimes abbreviate $\langle\lambda x.\langle\lambda y.X\rangle(b)\rangle(a)$ as $\langle\lambda x, y.X\rangle(a, b)$, and similarly for larger numbers of nested abstracts.

4 Predicate Abstract Semantics

Domains, as in first-order models, must be part of the semantic machinery since we have relation and function symbols to model. We are not considering quantifiers here, so a distinction between constant and varying domain models really cannot be seen. We confine things to constant domain models for simplicity (and other reasons as well). Think of members of domains as the objects—extensions—that intensions pick out at possible worlds.

A *model* is a structure $\mathcal{M} = \langle\mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I}\rangle$ meeting the following conditions.

- $\langle\mathcal{G}, \mathcal{R}\rangle$ is a *frame*, with $\mathcal{G}$ the set of possible worlds and $\mathcal{R}$ a binary accessibility relation on $\mathcal{G}$.
- $\mathcal{D}$ is a non-empty set called the *object domain*.
- $\mathcal{I}$ is an *interpretation* function. It must cover relation and intension function symbols.

  If P is an n-ary relation symbol then $\mathcal{I}(P)$ is a mapping from $\mathcal{G}$ to subsets of $\mathcal{D}^n$. It is required that $\mathcal{I}(=)$ be the constant function mapping each world to the identity relation on $\mathcal{D}$. (Thus equality is interpreted rigidly, and is total.)

  If a is an n-ary intension function symbol then $\mathcal{I}(a) : S \to (\mathcal{D}^n \to \mathcal{D})$, for some $S \subseteq \mathcal{G}$. (The idea is that $\mathcal{I}(a)$ provides an n-ary function from $\mathcal{D}$ to itself for some, but perhaps not all, of the possible worlds in $\mathcal{G}$. Such a function is provided for those worlds in S, but not for worlds outside S.) As a special case, a 0-ary intension function symbol, or intension constant, is simply a partial function from $\mathcal{G}$ to $\mathcal{D}$. If $\mathcal{I}(a) : S \to (\mathcal{D}^n \to \mathcal{D})$, we say *a designates* at the worlds in S.

A *valuation* in model $\mathcal{M}$ is a map assigning to each object variable a member of $\mathcal{D}$. We write $\mathcal{M}, w \Vdash_v X$ to symbolize that formula X is true at possible world w of model $\mathcal{M}$ with respect to valuation v. The conditions for this are as follows.

Atomic $\mathcal{M}, w \Vdash_v P(x_1, \ldots, x_n) \Leftrightarrow \langle v(x_1), \ldots, v(x_n)\rangle \in \mathcal{I}(P)(w)$.

Propositional $\mathcal{M}, w \Vdash_v X \supset Y \Leftrightarrow \mathcal{M}, w \nVdash_v X$ or $\mathcal{M}, \Gamma \Vdash_v Y$

And similarly for other propositional connectives.

Necessity $\mathcal{M}, w \Vdash_v \Box X \Leftrightarrow \mathcal{M}, w' \Vdash_v X$ for all $w' \in \mathcal{G}$ such that $w\mathcal{R}w'$.

Possibility $\mathcal{M}, w \Vdash_v \Diamond X \Leftrightarrow \mathcal{M}, w' \Vdash_v X$ for some $w' \in \mathcal{G}$ such that $w\mathcal{R}w'$.

Predicate Abstraction, n-ary function symbol

If intension function symbol a designates at w, $\mathcal{M}, w \Vdash_v \langle \lambda y.X \rangle(a(x_1, \ldots, x_n))$ if
$\mathcal{M}, w \Vdash_{v'} X$ where v' is like v except that $v'(y) = \mathcal{I}(a)(w)(v(x_1), \ldots, v(x_n))$.

If a does not designate at w, $\mathcal{M}, w \nVdash_v \langle \lambda x.X \rangle(a(x_1, \ldots, x_n))$.

The special case of predicate abstraction where a is 0-place, that is, an intension constant, is of particular interest, so we state it explicitly.

Predicate Abstraction, intension constant

If a designates at w, $\mathcal{M}, w \Vdash_v \langle \lambda y.X \rangle(a)$ if
$\mathcal{M}, w \Vdash_{v'} X$ where v' is like v except that $v'(y) = \mathcal{I}(a)(w)$.

If a does not designate at w, $\mathcal{M}, w \nVdash_v \langle \lambda x.X \rangle(a)$.

A formula X is valid in a model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$ if $\mathcal{M}, w \Vdash_v X$ for every $w \in \mathcal{G}$ and every valuation v, and is valid if it is valid in every model. We have placed no restrictions on accessibility, so this is a version of K with predicate abstracts. Reflexivity, transitivity, and so on, can be imposed in the familiar way.

The first predicate abstraction condition says that $\mathcal{M}, w \Vdash_v \langle \lambda x.X \rangle(t)$ if the object designated by t at w has the property specified by X at w. The second condition says that no property can correctly be ascribed to the object designated by a term that does not designate. Then intension symbols enter into things via what they designate, if anything. Other things are possible as well, and were considered in [6].

5 Examples Continued

In (1) we gave a formula intended to express the idea that someday the King/Queen of Sweden might be taller than now. We repeat the formula here for convenience.

$$\langle \lambda y. \Diamond \langle \lambda x. T(x, y) \rangle(m) \rangle(m)$$

Now we use this formula to illustrate the way the semantics works. Here are a few historical facts we can make use of.

The list of Swedish royalty commonly begins with Eric the Victorious, who died around 995. But to be on the safe side, the first vague reference to a Swedish King is in Tacitus,

around the year 100. So we can plausibly assume that in the year 50 there was no such King.

It is hard to get data about heights before modern times, but an examination of skeletons shows that, in the 13^{th} century the average adult male Swede was 174.3 cm (68.6 in), [13]. King Magnus III Barnlock (who we will represent as mb) died in 1290, and we will assume his height was exactly average.

Finally, King Carl XVI Gustaf, king at the time of writing this, is 179 cm (70.5 in) tall. We will use cg to represent him. It is a small difference, but he is taller than Magnus III.

Figure 1 graphically shows a model, $\mathcal{M}$. For this model the set of possible worlds is $\mathcal{G} = \{w_{50}, w_{1289}, w_{2013}\}$, intended to represent states of the world in the years 50, 1289, and 2013. Accessibility is shown using right pointing arrows, and is intended to represent passage to a future state. (We may assume transitivity, but it is not shown in the figure and is not needed for the points we wish to make.) The object domain is $\mathcal{D} = \{\mathsf{mb}, \mathsf{cg}\}$, intended to represent the two Swedish kings discussed above. The interpretation function $\mathcal{I}$ is such that $\mathcal{I}(T)(w_{50}) = \mathcal{I}(T)(w_{1289}) = \mathcal{I}(T)(w_{2013}) = \{(\mathsf{cg}, \mathsf{mb})\}$. (Thus T is interpreted rigidly, and informally says that cg is taller than mb.) We assume m designates at $\{w_{1289}, w_{2013}\}$, and $\mathcal{I}(m)(w_{1289}) = \mathsf{mb}$ and $\mathcal{I}(m)(w_{2013}) = \mathsf{cg}$. That is, $\mathcal{I}(m)$ picks out the King of Sweden at each time state we consider, at which a king exists.

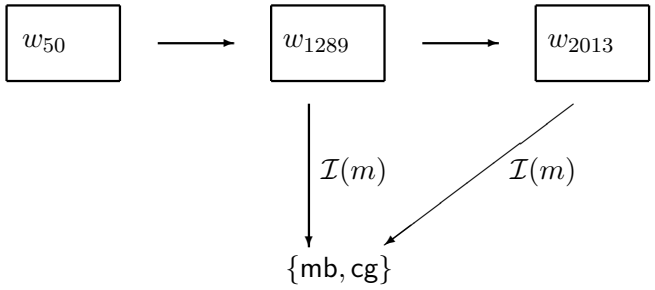

Figure 1: Model $\mathcal{M}$ With Swedish Kings

We check that $\mathcal{M}, w_{1289} \Vdash_v \langle\lambda y.\Diamond\langle\lambda x.T(x,y)\rangle(m)\rangle(m)$, thus showing that (1) is formally true under the expected circumstances. (v is an arbitrary valuation. In fact, the formula has no free variables, so a choice of v won't matter.) Now, $\mathcal{M}, w_{1289} \Vdash_v \langle\lambda y.\Diamond\langle\lambda x.T(x,y)\rangle(m)\rangle(m)$ just in case $\mathcal{M}, w_{1289} \Vdash_{v'} \Diamond\langle\lambda x.T(x,y)\rangle(m)$, where $v'(y) = \mathcal{I}(m)(w_{1289}) = \mathsf{mb}$. This holds just when $\mathcal{M}, w_{2013} \Vdash_{v'} \langle\lambda x.T(x,y)\rangle(m)$, and in turn we have this just in case $\mathcal{M}, w_{2013} \Vdash_{v''} T(x,y)$ where $v''(y) = v'(y) = \mathsf{mb}$ and $v''(x) = \mathcal{I}(m)(w_{2013}) = \mathsf{cg}$. And this is the case, because $(\mathsf{cg}, \mathsf{mb}) \in \mathcal{I}(T)(w_{2013})$.

Next $\mathcal{M}, w_{50} \nVdash_v \langle\lambda y.\Diamond\langle\lambda x.T(x,y)\rangle(m)\rangle(m)$, because m does not designate at w_{50}. Also $\mathcal{M}, w_{2013} \nVdash_v \langle\lambda y.\Diamond\langle\lambda x.T(x,y)\rangle(m)\rangle(m)$ because w_{2013} is a possible world with no future. The first is what we would expect. The second really reflects an inadequacy in our model—we have not represented time correctly (we dearly hope).

We move on to the example in which it is said, of Alice, that she might have been taller than she is, initially formalized in (5) which we repeat here.

$$\langle \lambda y.\Diamond\langle \lambda x.G(x,y)\rangle(h(a))\rangle(h(a))$$

The immediate difficulty is that this is not a legal formula as we have set things up—see the discussion at the end of Section 3. What takes its place is the appalling looking formula (8), which is actually simpler in a sense, because there is no function/constant symbol nesting.

$$\langle \lambda w.\langle \lambda y.\Diamond\langle \lambda z.\langle \lambda x.G(x,y)\rangle(h(z))\rangle(a)\rangle(h(w))\rangle(a) \tag{8}$$

We want the interpretation of a to be rigid, but the function interpreting h to be non-rigid. For simplicity, let us suppose we measure heights to the nearest centimeter, and we only need heights from, say, 0 to 400. This keeps things finite, though nothing essential depends on it.

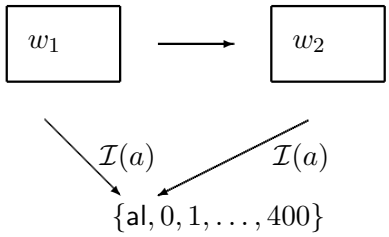

Figure 2: Model $\mathcal{N}$ With Alice

Figure 2 shows a model, $\mathcal{N}$. The set of possible worlds $\mathcal{G}$ consists of w_1 and w_2 (how things are, and how they might have been), with $w_1 \mathcal{R} w_2$. The object domain $\mathcal{D}$ is $\{\mathsf{al}, 0, 1, \ldots, 400\}$, where al is intended to be the 'Alice object.'

Next we have the interpretation function $\mathcal{I}$. We set $\mathcal{I}(G)(w_1) = \mathcal{I}(G)(w_2) = \{(x,y) \mid x, y$ are integers from 0 to 400 and $x > y\}$. Intension constant a designates at both w_1 and w_2, and $\mathcal{I}(a)(w_1) = \mathcal{I}(a)(w_2) = \mathsf{al}$. Thus a is interpreted rigidly. h is interpreted non-rigidly, though it designates at both possible worlds; $\mathcal{I}(h) : \mathcal{G} \to (\mathcal{D} \to \mathcal{D})$. We set $\mathcal{I}(h)(w_1)(\mathsf{al}) = 165$ and $\mathcal{I}(h)(w_2)(\mathsf{al}) = 180$. (Values on other members of $\mathcal{D}$ are quite irrelevant, so we won't bother to specify them.)

It can now be verified that we have

$$\mathcal{N}, w_1 \Vdash_v \langle \lambda w.\langle \lambda y.\Diamond\langle \lambda z.\langle \lambda x.G(x,y)\rangle(h(z))\rangle(a)\rangle(h(w))\rangle(a)$$

and so (8) is satisfiable, pretty much as expected.

Equality is a relation on objects, and not on intensions, and is independent of the particular possible world. We have the validity of $(x = y) \supset \Box(x = y)$ and, $\neg(x = y) \supset \Box\neg(x = y)$. On the other hand, we do *not* have validity of $\langle\lambda x, y.(x = y)\rangle(a, b) \supset \Box\langle\lambda x, y.(x = y)\rangle(a, b)$, which says that if a and b designate the same object in the present world, they must designate the same object in all accessible worlds. We do, however, have the validity of $\langle\lambda x, y.(x = y)\rangle(a, b) \supset \langle\lambda x, y.\Box(x = y)\rangle(a, b)$. Think about it.

Suppose a is a one-place intension function symbol and b is zero-place, that is, an intension constant. The expression $a(b) = a(b)$ is not a legal formula, and there is more than one way of unabbreviating it, but all versions have similar behavior. Let us examine the simplest.

$$\langle\lambda x.\langle\lambda y.y = y\rangle(a(x))\rangle(b) \tag{9}$$

Suppose $\mathcal{M} = \langle\mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I}\rangle$ is a model, $w \in \mathcal{G}$, and v is a valuation. Under what circumstances do we have $\mathcal{M}, w \Vdash_v (9)$?

1. Suppose b does not designate at w. Then directly, $\mathcal{M}, w \nVdash_v (9)$.

2. Now suppose b designates at w. Let v' be like v except that $v'(x) = \mathcal{I}(b)(w)$. Then $\mathcal{M}, w \Vdash_v (9)$ just in case $\mathcal{M}, w \Vdash_{v'} \langle\lambda y.y = y\rangle(a(x))$. If a does not designate at w, again directly $\mathcal{M}, w \nVdash_{v'} \langle\lambda y.y = y\rangle(a(x))$, and hence $\mathcal{M}, w \nVdash_v (9)$.

3. Now suppose b designates, and also a designates. Let v'' be like v' except that $v''(y) = \mathcal{I}(a)(w)(v'(x))$. Then $\mathcal{M}, w \Vdash_{v'} \langle\lambda y.y = y\rangle(a(x))$ just in case $\mathcal{M}, w \Vdash_{v''} y = y$, and this is so.

Conclusion: (9) is true at a possible world exactly when both a and b designate at that world—essentially what we would expect.

6 Propositional Nested Sequents

Nested sequent systems are proof systems that allow sequents to be nested inside of other sequents. They were invented several times with some variation, [1, 2, 14, 15]. Usually they are for modal logics and generalize one-sided sequents, though there are exceptions. There is, for instance, a two-sided version for intuitionistic logic in [11], including constant domain logic. It turns out that nested sequents bear the same relationship to prefixed tableaus that Gentzen sequents do to ordinary tableaus, namely one is the other 'upside down,' [10]. Nested sequent systems are forward reasoning, while tableaus are backward reasoning systems.

We sketch a nested sequent system for the modal logic K with predicate abstraction and equality. Ultimately it derives from a prefixed tableau system for the same logic, a relationship discussed in [10], though there are significant modifications. We think of a sequent as a set, so there are no structural rules. We do not allow the empty sequent,

though it would not change things much if we did. The definition of nested sequent that we use is a recursive one.

Definition 6.1 A *nested sequent* is a non-empty finite set of formulas and nested sequents.

A one-sided sequent can be thought of as a disjunction, while nesting corresponds to necessitation. Formally, let $\Gamma = \{X_1, \ldots, X_n, \Delta_1, \ldots, \Delta_k\}$ be a nested sequent, where each X_i is a formula and each Δ_j is a nested sequent. This translates to an ordinary modal formula $\Gamma^\dagger$, which can be thought of as the 'meaning' of Γ.

$$\Gamma^\dagger = X_1 \vee \ldots \vee X_n \vee \Box\Delta_1^\dagger \vee \ldots \vee \Box\Delta_k^\dagger$$

We follow common notational conventions for nested sequents. Enclosing outer set brackets are generally omitted. A nested sequent that is a member of another nested sequent has its members listed in square brackets, and is called a *boxed sequent*. For example, $A, B, [C, [D, E], [F, G]]$, is the conventional way of writing $\{A, B, \{C, \{D, E\}, \{F, G\}\}\}$. We use Γ, Δ, ... for nested sequents, boxed or top level.

Suppose that $\Gamma(P)$ is a nested sequent in which propositional letter P (technically, a zero-place relation symbol) occurs exactly once, as a direct member of some subsequent of $\Gamma(P)$ and not as part of a more complex formula. Then we may write $\Gamma(X)$ to denote the result of replacing P in $\Gamma(P)$ with X. Similarly for $\Gamma(X, Y)$, $\Gamma(\Delta)$, and so on. Think of $\Gamma(P)$ as supplying a 'context.' In other presentations the role of P is played by a 'hole,' but the idea is the same.

The rules for propositional connectives and modal operators are given in Figure 3. These are standard, and are for the logic K. In stating them we assume that $\Gamma(P)$ is some nested sequent with one occurrence of propositional letter P. Also we use $[\ldots]$ to stand for a *non-empty* nested sequent, and $[Z, \ldots]$ is $[\ldots]$ but with Z added.

Sequent proofs start with axioms and end with the nested sequent being proved. More precisely, they are trees with axioms at the leaves, the item being proved at the root, with each non-leaf node labeled with a nested sequent that follows from the nested sequents labeling its child nodes using one of the nested sequent rules. Proof of a formula is a derivative notion: a proof of the nested sequent consisting of just the formula X is taken to be a proof of X itself. For illustration, Figure 4 displays a proof of $\Box(A \supset B) \supset (\Box A \supset \Box B)$. Formulas A and B are atomic. Later, atomic formulas will have internal structure, but the details don't matter now. The two subsequents shown at the top are axioms, and otherwise reasons are displayed for each inference.

7 Nested Sequents and Predicate Abstraction

From this point on relation and function symbols play a significant role. They were suppressed in the previous propositional section. It is common in proof systems for first-order

Axioms $\Gamma(A, \neg A)$,

A an atomic formula

Double Negation Rule $$\frac{\Gamma(X)}{\Gamma(\neg\neg X)}$$

α **Rule** $$\frac{\Gamma(X) \quad \Gamma(Y)}{\Gamma(X \wedge Y)} \qquad \frac{\Gamma(\neg X) \quad \Gamma(\neg Y)}{\Gamma(\neg(X \vee Y))} \qquad \frac{\Gamma(X) \quad \Gamma(\neg Y)}{\Gamma(\neg(X \supset Y))}$$

β **Rule** $$\frac{\Gamma(X, Y)}{\Gamma(X \vee Y)} \qquad \frac{\Gamma(\neg X, \neg Y)}{\Gamma(\neg(X \wedge Y))} \qquad \frac{\Gamma(\neg X, Y)}{\Gamma(X \supset Y)}$$

ν **Rule** $$\frac{\Gamma([X])}{\Gamma(\Box X)} \qquad \frac{\Gamma([\neg X])}{\Gamma(\neg\Diamond X)}$$

π **Rule** $$\frac{\Gamma(\Diamond X, [X, \ldots])}{\Gamma(\Diamond X, [\ldots])} \qquad \frac{\Gamma(\neg\Box X, [\neg X, \ldots])}{\Gamma(\neg\Box X, [\ldots])}$$

Figure 3: Nested Sequent Rules for K

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\neg\Box(A \supset B), \neg\Box A, [\neg A, A, B] \quad \neg\Box(A \supset B), \neg\Box A, [\neg A, \neg B, B]}{\neg\Box(A \supset B), \neg\Box A, [\neg A, \neg(A \supset B), B]}\,\alpha\textbf{ Rule}}{\neg\Box(A \supset B), \neg\Box A, [\neg(A \supset B), B]}\,\pi\textbf{ Rule}}{\neg\Box(A \supset B), \neg\Box A, [B]}\,\pi\textbf{ Rule}}{\neg\Box(A \supset B), \neg\Box A, \Box B}\,\nu\textbf{ Rule}}{\neg\Box(A \supset B), \Box A \supset \Box B}\,\beta\textbf{ Rule}}{\Box(A \supset B) \supset (\Box A \supset \Box B)}\,\beta\textbf{ Rule}$$

Figure 4: Proof of $\Box(A \supset B) \supset (\Box A \supset \Box B)$

logic to introduce special symbols, just for use in proofs—new free variables, special constant symbols, parameters. They play the role of existential witnesses. We do a similar thing here. Just for use in proofs, we introduce into the formal language *extension function symbols*. We might have allowed such things in formulas from the start, but in the present treatment their use is confined to proofs—they will not appear in formulas being proved. For convenience, we refer to them as parameters. We also introduce a new kind of atomic formula, connecting these new extension function symbols with the intension function symbols we have been using.

Definition 7.1 (Parameters and Object Terms) *Parameters* are a new family of symbols, typically written as p, q, r, $\ldots$, each having an *arity*, 0-place, 1-place, and so on.

A new kind of atomic formula is introduced, $a \not\rightarrow p$, where a is an intension function

symbol and p is a parameter, both having the same *arity*.

An *object term* is an expression built up entirely from parameters (but not from free variables). Object terms are allowed to appear in proofs directly as arguments of intension function symbols and of atomic formulas.

Informally, a parameter represents the extensional function picked out at a particular possible world by an intension function symbol. The new atomic formula $a \not\rightarrow p$ is intended to assert that, at a particular possible world, a *does not* designate extensional function p. (It is more convenient to introduce $\not\rightarrow$ directly, than to introduce $\rightarrow$ and combine it with negation. The point is a minor one.) As to allowing object terms to appear directly as arguments, parameters are to represent functions that are not world-dependent, so the sort of ambiguity that can arise with nested intension function symbols cannot arise with parameters.

Only closed formulas may appear in proofs where a closed formula has no free variable occurrences, though parameters may be present. We only prove closed formulas not containing parameters.

In Figure 5 we give rules for predicate abstracts. In them $X(x)$ is a formula that may contain free occurrences of variable x, and $X(p(t_1, \ldots, t_n))$ is the result of replacing all free occurrences of x with occurrences of object term $p(t_1, \ldots, t_n)$. For Negative Abstract Rule 2 two restrictive conditions are imposed. The first is needed to ensure rule soundness. The second somewhat simplifies our completeness proof. Of course we would still have completeness without it since any proof meeting a restriction is still a proof if the restriction is relaxed.

Positive Abstract Rule
$$\frac{\Gamma(X(p(t_1, \ldots, t_n)), a \not\rightarrow p)}{\Gamma(\langle \lambda x.X(x)\rangle(a(t_1, \ldots, t_n)), a \not\rightarrow p)}$$

Negative Abstract Rule 1
$$\frac{\Gamma(\neg X(p(t_1, \ldots, t_n)), a \not\rightarrow p)}{\Gamma(\neg\langle \lambda x.X(x)\rangle(a(t_1, \ldots, t_n)), a \not\rightarrow p)}$$

Negative Abstract Rule 2
$$\frac{\Gamma(\neg X(p(t_1, \ldots, t_n)), a \not\rightarrow p)}{\Gamma(\neg\langle \lambda x.X(x)\rangle(a(t_1, \ldots, t_n)))}$$
p not in conclusion
$a \not\rightarrow q$ not in conclusion for any q

In the rules above $t_1, \ldots, t_n$ are object terms, and n could be 0.

Figure 5: Predicate Abstract Rules

Figure 6 displays a proof of $\langle \lambda x.\neg P(x)\rangle(a) \supset \neg\langle \lambda x.P(x)\rangle(a)$ where P is a one-place relation symbol and a is an intension constant, that is, a 0-place function symbol. The converse, $\neg\langle \lambda x.P(x)\rangle(a) \supset \langle \lambda x.\neg P(x)\rangle(a)$ is not valid. One can easily construct a model

in which a does not designate at a world, and so $\langle\lambda x.P(x)\rangle(a)$ fails there, and hence the antecedent is true, but of course the consequent is false. Once soundness is established, this tells us the converse is not provable. The reader may find it instructive to see what goes wrong with a direct attempt at constructing a proof. On the other hand, $[\neg\langle\lambda x.P(x)\rangle(a) \wedge \langle\lambda x.Q(x)\rangle(a)] \supset \langle\lambda x.\neg P(x)\rangle(a)$ is provable, and constructing a proof is a good exercise.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(p), a \not\to p, \neg P(p)}{\neg\neg P(p), a \not\to p, \neg P(p)}\ \neg\neg\ \textbf{Rule}}{\neg\neg P(p), a \not\to p, \neg\langle\lambda x.P(x)\rangle(a)}\ \textbf{Neg Abs 1}}{\neg\langle\lambda x.\neg P(x)\rangle(a), \neg\langle\lambda x.P(x)\rangle(a)}\ \textbf{Neg Abs 2}}{\langle\lambda x.\neg P(x)\rangle(a) \supset \neg\langle\lambda x.P(x)\rangle(a)}\ \beta\ \textbf{Rule}$$

Figure 6: Proof of $\langle\lambda x.\neg P(x)\rangle(a) \supset \neg\langle\lambda x.P(x)\rangle(a)$

8 Nested Sequents and Equality

In the semantics from Section 4, equality is a relation on the object domain, $\mathcal{D}$, of a model. This is reflected in nested sequent proofs, in which the symbol $=$ may appear between object terms as defined in the previous section. Now we give rules for $=$, and this can be done in two ways—reflexivity is the key point. We could introduce a rule as follows.

$$\dfrac{\Gamma(\neg t = t, \ldots)}{\Gamma(\ldots)}$$

This gives a complete system when combined with the rules from Figure 7. It is easier to prove completeness than it is for the system actually adopted here. But it has the annoying property of possibly letting a relation symbol (equality) and function terms (those in t) disappear during a rule application. To avoid this, we adopt an axiom instead, loosely basing our approach on Section 9.8 of [4].

Definition 8.1 A formula is *simple* if it is the negation of an atomic formula, or it is an atomic formula whose relation symbol is $=$.

Figure 7 displays the nested sequent rules for equality. In stating the Substitution Rule $X(t)$ is a (closed, simple) formula and $X(u)$ is the result of replacing *some* (as opposed to *every*) occurrence of object term t in it with an occurrence of object term u.

In Section 2 we noted that formula (4) followed from (2) and (3). We now show this using a nested sequent proof, in Figure 8.

Formula (10) is a related example one might practice on. Part of the antecedent is $\langle\lambda x.\Box\langle\lambda y.x = y\rangle(a)\rangle(a)$. In [12] this was called a *local rigidity* condition—it expresses that

Equality Axioms	$\Gamma(t = t)$
Equality Up Rule	$\dfrac{\Gamma([\neg t = u, \ldots], \neg t = u)}{\Gamma([\ldots], \neg t = u)}$
Equality Down Rule	$\dfrac{\Gamma([\neg t = u, \ldots], \neg t = u)}{\Gamma([\neg t = u, \ldots])}$
Equality Left-Right Substitution Rule	$\dfrac{\Gamma(\neg t = u, X(u), X(t))}{\Gamma(\neg t = u, X(u))}$
Equality Right-Left Substitution Rule	$\dfrac{\Gamma(\neg t = u, X(u), X(t))}{\Gamma(\neg t = u, X(t))}$

In the rules above, t and u are object terms,
X is a simple formula
and $\ldots$ is allowed to be empty in the Down Rule,
but not in the Up Rule.

Figure 7: Equality Rules

the intension constant a designates in the current world, and must designate the same value in all related worlds. The second condition, $\Box\langle\lambda x.\langle\lambda y.y = y\rangle(h(x))\rangle(b)$ can only be true if h designates a function in all related worlds. Of course it also requires that b designates in all related worlds, but the values designated play no significant role.

$$\begin{array}{l}[\langle\lambda x.\Box\langle\lambda y.x = y\rangle(a)\rangle(a) \wedge \Box\langle\lambda x.\langle\lambda y.y = y\rangle(h(x))\rangle(b)] \\ \supset \\ \langle\lambda x.\Box\langle\lambda y.\langle\lambda z.\langle\lambda w.z = w\rangle(h(y))\rangle(h(x))\rangle(a)\rangle(a)\end{array} \tag{10}$$

9 Soundness

We begin with an obvious problem that affects our completeness argument as well. Parameters can occur in proofs, but were not mentioned in the discussion of semantics in Section 4. They must be given a semantic meaning. Similarly for the atomic formula $a \not\rightarrow p$.

Let $\mathcal{M} = \langle\mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I}\rangle$ be a model. We extend the definition of interpretation so that values are supplied for parameters too. If p is an n-argument parameter, $\mathcal{I}(p) : \mathcal{D}^n \rightarrow \mathcal{D}$. Note that this is not world dependent, unlike with intension function symbols. This in turn gives a natural semantic meaning to each *object term*, Definition 7.1. We write $\mathcal{I}(t)$ for the member of $\mathcal{D}$ assigned to object term t. Definition is by recursion: $\mathcal{I}(p(t_1, \ldots, t_n)) =$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\neg T(q,p), \neg T(p,p), a \not\to q, \neg q = p, T(p,p)], a \not\to p, \neg\Box\langle\lambda x.x = p\rangle(a), \Diamond T(p,p)}{[\neg T(q,p), a \not\to q, \neg q = p, T(p,p)], a \not\to p, \neg\Box\langle\lambda x.x = p\rangle(a), \Diamond T(p,p)}\ \textbf{Eq R-L Subs}}{[\neg T(q,p), a \not\to q, \neg\langle\lambda x.x = p\rangle(a), T(p,p)], a \not\to p, \neg\Box\langle\lambda x.x = p\rangle(a), \Diamond T(p,p)}\ \textbf{Neg Abs 1}}{[\neg\langle\lambda x.T(x,p)\rangle(a), \neg\langle\lambda x.x = p\rangle(a), T(p,p)], a \not\to p, \neg\Box\langle\lambda x.x = p\rangle(a), \Diamond T(p,p)}\ \textbf{Neg Abs 2}}{[\neg\langle\lambda x.T(x,p)\rangle(a), \neg\langle\lambda x.x = p\rangle(a)], a \not\to p, \neg\Box\langle\lambda x.x = p\rangle(a), \Diamond T(p,p)}\ \pi\ \textbf{Rule}}{[\neg\langle\lambda x.T(x,p)\rangle(a)], a \not\to p, \neg\Box\langle\lambda x.x = p\rangle(a), \Diamond T(p,p)}\ \pi\ \textbf{Rule}}{\neg\Diamond\langle\lambda x.T(x,p)\rangle(a), a \not\to p, \neg\Box\langle\lambda x.x = p\rangle(a), \Diamond T(p,p)}\ \nu\ \textbf{Rule}}{\neg\Diamond\langle\lambda x.T(x,p)\rangle(a), a \not\to p, \neg\Box\langle\lambda x.x = p\rangle(a), \langle\lambda x.\Diamond T(x,x)\rangle(a)}\ \textbf{Pos Abs}}{\neg\Diamond\langle\lambda x.T(x,p)\rangle(a), a \not\to p, \neg\langle\lambda y.\Box\langle\lambda x.x = y\rangle(a)\rangle(a), \langle\lambda x.\Diamond T(x,x)\rangle(a)}\ \textbf{Neg Abs 1}}{\neg\langle\lambda y.\Diamond\langle\lambda x.T(x,y)\rangle(a)\rangle(a), \neg\langle\lambda y.\Box\langle\lambda x.x = y\rangle(a)\rangle(a), \langle\lambda x.\Diamond T(x,x)\rangle(a)}\ \textit{Neg Abs 2}}{\dfrac{\dfrac{\neg 2, \neg 3, 4}{\neg(2 \wedge 3), 4}\ \alpha\ \textbf{Rule}}{(2 \wedge 3) \supset 4}\ \alpha\ \textbf{Rule}}\ \textit{Abbreviating}$$

Figure 8: Equality Proof Example

$\mathcal{I}(p)(\mathcal{I}(t_1), \ldots, \mathcal{I}(t_n))$, with $\mathcal{I}(p)$ defined directly as part of the condition for being an interpretation now.

We have a new atomic formula allowed in proofs. $\not\to$ is interpreted as a two-place relation symbol, across types, so that $\mathcal{M}, w \Vdash_v a \not\to p$ if either a does not designate at w, or else a does designate, but $\mathcal{I}(a)(w) \neq \mathcal{I}(p)$. More formally, but less intelligibly, $\mathcal{I}(\not\to)(a,p) = \{w \in \mathcal{G} \mid a \text{ does not designate at } w\} \cup \{w \in \mathcal{G} \mid a \text{ designates at } w \text{ and } \mathcal{I}(a)(w) \neq \mathcal{I}(p)\}$.

Finally, the interpretation of predicate abstraction must be modified a bit, from Section 4. We now want the following, which agrees with the earlier definition in which no parameters are present.

Predicate Abstraction, n-ary function symbol Suppose u_i is either a variable or an object term. Let $\overline{u_i}$ be $v(u_i)$ if u_i is a variable, and $\mathcal{I}(u_i)$ if u_i is an object term.

If intension function symbol a designates at w, $\mathcal{M}, w \Vdash_v \langle\lambda y.X\rangle(a(u_1, \ldots, u_n))$ if $\mathcal{M}, w \Vdash_{v'} X$ where v' is like v except that $v'(y) = \mathcal{I}(a)(w)(\overline{u_1}, \ldots, \overline{u_n})$.

If a does not designate at w, $\mathcal{M}, w \nVdash_v \langle\lambda x.X\rangle(a(x_1, \ldots, x_n))$.

Detailed arguments are lengthy, but the basic ideas are rather straightforward. We sketch them, and leave it to the reader to work out the specifics.

In Section 6 is given the usual translation of nested sequents into formulas. If one shows the translate of each axiom is valid, and the rules preserve validity of translations, then soundness is an immediate consequence. There are just a few general cases. First, we have

the following.

Let A be a closed formula, allowing object terms and $\not\to$. If A is valid, so is $\Gamma(A)^\dagger$. (11)

Item (11) is shown by induction on the nesting level of A in $\Gamma(A)$. We leave this to the reader. Once it has been established, validity of the nested sequent axioms follows.

Next is another general result whose proof also can be shown by induction on nesting depth.

Let A and B be closed formulas, allowing object terms and $\not\to$.
If $A \supset B$ is valid, so is $\Gamma(A)^\dagger \supset \Gamma(B)^\dagger$. (12)

This allows easy treatment of most of the single-premise rules. Consider, for example, the Positive Abstract Rule. First one shows validity of the following formula. We omit the verification.

$$[X(p(t_1, \ldots, t_n)) \vee a \not\to p] \supset [\langle \lambda x.X(x)\rangle(a(t_1, \ldots, t_n)) \vee a \not\to p]$$

Then soundness of the Positive Abstract Rule follows using (12). In a similar way one handles Double Negation, β, ν, π, Positive Abstract, Negative Abstract 1, and all the Equality Rules.

Double premise rules can be handled once the following is shown.

Let A, B, and C be closed formulas allowing object terms and $\not\to$.
If $(A \wedge B) \supset C$ is valid, so is $[\Gamma(A)^\dagger \wedge \Gamma(B)^\dagger] \supset \Gamma(C)^\dagger$. (13)

Using (13) one easily gets soundness of the β rules.

We still have not verified soundness of Negative Abstract Rule 2, which is more complicated. For this we need something that is related to the usual treatment of quantification. We say a model $\mathcal{M}' = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I}' \rangle$ is a *p-variant* of model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$ just in case $\mathcal{I}'$ agrees with $\mathcal{I}$ except possibly on parameter p.

Assume $\Gamma(\neg\langle \lambda x.(x)\rangle(a(t_1, \ldots, t_n)))$ does not contain parameter p. Let $\mathcal{M}$ be a model, and w be an arbitrary member of $\mathcal{G}$. If $\mathcal{M}', w \Vdash_v \Gamma(\neg X(p(y_1, \ldots, y_n)) \vee a \not\to p)$ for every p variant $\mathcal{M}'$ of $\mathcal{M}$, then $\mathcal{M}, w \Vdash_v \Gamma(\neg\langle \lambda x.X(x)\rangle(a(t_1, \ldots, t_n)))$. (14)

As with earlier items in this section, the proof of (14) is by induction on nesting depth. For the ground case where there is no nesting of sequents, what needs to be shown is: if $\mathcal{M}', w \Vdash_v A_1 \vee \ldots \vee A_k \vee \neg X(p(t_1, \ldots, t_n)) \vee a \not\to p$ for every p variant $\mathcal{M}'$ of $\mathcal{M}$, then $\mathcal{M}, w \Vdash_v A_1 \vee \ldots \vee A_k \vee \neg\langle \lambda x.X(x)\rangle(a(t_1, \ldots, t_n))$, where A_1, …, A_k, $X(x)$ do not contain p, and k might be 0.

This is most easily argued in the contrapositive direction. Assume $\mathcal{M}, w \Vdash_v \neg A_i$ for $i = 1, \ldots, k$, and $\mathcal{M}, w \Vdash_v \langle \lambda x.X(x)\rangle(a(t_1, \ldots, t_n))$. Then a must designate at w, and

$\mathcal{M}, w \Vdash_{v'} X(x)$ where $v'(x) = \mathcal{I}(a)(w)(\mathcal{I}(t_1), \ldots, \mathcal{I}(t_n))$, and otherwise v' and v agree. Now introduce a new parameter, p, and define $\mathcal{I}'$ to be like $\mathcal{I}$ except that $\mathcal{I}'(p) = \mathcal{I}(a)(w)$, and let $\mathcal{M}' = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I}' \rangle$. Then $\mathcal{M}', w \Vdash_v X(p(t_1, \ldots, t_n))$, and $\mathcal{M}', w \Vdash_v \neg(a \not\to p)$. Of course $\mathcal{M}', w \Vdash_v \neg A_i$ since A_i does not contain p so $\mathcal{M}$ and $\mathcal{M}'$ behave alike on it.

We have now finished the argument for the ground case. We leave the induction step to the reader. With (14) available, soundness of Negative Abstract Rule 2 follows, completing the proof of the soundness of the nested sequent system.

10 Completeness

Let Z be a closed parameter-free formula, fixed for this section. Assume Z is unprovable; we show a counter model for Z exists..

It is likely that this could be done using maximal consistency as described in [9], but here we follow a more traditional route. With sequent systems one often reasons backwards from the formula to be proved, in order to discover a proof. A backward search for a proof should be done systematically, but there is much variation possible. A search must be *fair*, in that any applicable rule must eventually be brought into the proof. Beyond this we omit specifics, and simply assume some fair search procedure can be formulated.

Apply a fair proof search procedure, attempting to prove formula Z. This constructs a tree, with the formula being proved at the bottom, the tree branching upward, with branching associated with α Rule applications. Maximum nesting depth of nested sequents cannot exceed the modal depth of the formula being proved, as a look at the ν and π rules in Figure 3 makes clear. If a proof is found during the course of a search, each branch terminates with an Axiom, either $\Gamma(A, \neg A)$ or $\Gamma(t = t)$. If a proof is not found there must be a branch—call it *open*—without an axiom occurrence. Let us say $\mathcal{B}(Z)$ is a specific such branch, arising during a fair search for a proof of Z. Like Z, this too is fixed for the rest of this section.

Suppose, as an example, that on branch $\mathcal{B}(Z)$ the following step appears:

$$\begin{array}{c} \vdots \\ A, [B, C], [D, X], [E, [F]] \\ A, [B, C], [D, \neg\neg X], [E, [F]] \\ \vdots \end{array}$$

Most of the structure remains the same in this step, but $[D, X]$ 'turns into' $[D, \neg\neg X]$. We can think of these as the *same* boxed sequent, but we are seeing different stages of its history. Then a boxed sequent, dynamically, is really a sequence of boxed sequents appearing in $\mathcal{B}(Z)$, where the terms of the sequence are the stages through which the sequent passes. We will call such a sequence a *dynamic sequent*. This can be made mathematically precise, but an informal approach should suffice for our purposes: for the example above, we say that

D, X, and $\neg\neg X$ all appear in the same dynamic sequent, though X and $\neg\neg X$ appear at different stages. If w is a dynamic sequent, we use the special notation $A \dot{\in} w$ to indicate that A is a member at some stage. Members of a nested sequent can be other nested sequents, which have their own histories, but the general idea should be clear enough for present purposes.

We are about to use $\mathcal{B}(Z)$ to construct a model. We need a few results first, and it is easiest to state them if we say at this point what the possible worlds of our model will be. Making use of the discussion about stage-of-construction above, *let $\mathcal{G}$ be the set of dynamic boxed sequents that appear on $\mathcal{B}(Z)$, together with the top level dynamic sequent.*

Lemma 10.1 *Suppose A and B are closed atomic or negated atomic formulas, including those of the form $a \not\rightarrow p$. Also suppose $w \in \mathcal{G}$. If $A \dot{\in} w$ and $B \dot{\in} w$, then both A and B are present simultaneously at some stage of the construction of w.*

Proof An inspection of the nested sequent rules shows that (applied forward) none introduces either an atomic formula or its negation. Then when rules are applied backwards during a proof search, atomic formulas and their negations are not eliminated. It follows that during a proof search, if A and B are both present at different stages of the construction of w, both must be present at whichever stage is uppermost. ■

Lemma 10.2 *If P is atomic we cannot have both $P \dot{\in} w$ and $\neg P \dot{\in} w$ for any $w \in \mathcal{G}$.*

Proof Immediate from Lemma 10.1 and the fact that $\mathcal{B}(Z)$ is open. ■

Lemma 10.3 *Suppose $w \in \mathcal{G}$. Then for each intension function symbol a, $a \not\rightarrow p \dot{\in} w$ for at most one parameter p.*

Proof Suppose both $a \not\rightarrow p \dot{\in} w$ and $a \not\rightarrow q \dot{\in} w$, where $p \neq q$. Then both must be present at the same stage of the construction of w, by Lemma 10.1. But during the fair proof search that constructs $\mathcal{B}(Z)$, working from Z upward, the only rule that can introduce an expression $a \not\rightarrow p$ into a nested sequent is Negative Abstract Rule 2, from Figure 5. This rule can only be applied if $a \not\rightarrow q$ is not already present, for any q, so it is impossible to have both present at the same stage. ■

The remaining items all have to do with the behavior of equality.

Lemma 10.4 *Suppose t_1 and t_2 are object terms, and $\neg t_1 = t_2 \dot{\in} w$ for some $w \in \mathcal{G}$. Then $\neg t_1 = t_2 \dot{\in} w$ for every $w \in \mathcal{G}$.*

Proof This is an easy consequence of the Equality Up Rule and the Equality Down Rule. ■

Lemma 10.5 *Suppose $w \in \mathcal{G}$ and t_1, t_2, and t_3 are object terms.*

1. *If* $\neg t_1 = t_2 \mathrel{\dot\in} w$ *then* $\neg t_2 = t_1 \mathrel{\dot\in} w$.

2. *If* $\neg t_1 = t_2 \mathrel{\dot\in} w$ *and* $\neg t_2 = t_3 \mathrel{\dot\in} w$, *then* $\neg t_1 = t_3 \mathrel{\dot\in} w$.

Proof We show 1; item 2 is similar. Suppose $\neg t_1 = t_2 \mathrel{\dot\in} w$. Recall, we are conducting a backwards search for a proof. Now, $\neg t_1 = t_2$ can be inferred (in any context), from $\neg t_1 = t_2, \neg t_2 = t_2$ using the Equality Right-Left Substitution Rule (and deleting duplicates, since we are dealing with sets). Since our proof search procedure is a fair one, at some point this rule must be tried. In turn, $\neg t_1 = t_2, \neg t_2 = t_2$ can be inferred from $\neg t_1 = t_2, \neg t_2 = t_2, \neg t_2 = t_1$ using the Equality Left-Right Substitution Rule, and fairness says this will be tried. But then $\neg t_2 = t_1 \mathrel{\dot\in} w$. ■

Definition 10.6 We say object term t_1 $\mathcal{B}(Z)$-*rewrites* to object term t_2 if t_2 results from the replacement of some subterm u_1 of t_1 with u_2, where $\neg u_1 = u_2 \mathrel{\dot\in} w$ for some (any) $w \in \mathcal{G}$. We also say t_1 and t_2 are $\mathcal{B}(Z)$-*equivalent* if t_1 can be turned into t_2 via a sequence of $\mathcal{B}(Z)$-rewrites (possibly 0).

Lemma 10.7 *Suppose* X_1 *is a closed simple formula containing an occurrence of object term* t_1, *and* X_2 *is like* X_1 *but with some occurrence of* t_1 *replaced with an occurrence of object term* t_2. *If* $X_1 \mathrel{\dot\in} w$ *for some* $w \in \mathcal{G}$, *and* t_1 *and* t_2 *are* $\mathcal{B}(Z)$*-equivalent, then* $X_2 \mathrel{\dot\in} w$.

Proof A direct consequence of the Equality Substitution Rules and our fair search procedure. ■

Lemma 10.8 *Suppose* t_1 *is* $\mathcal{B}(Z)$*-equivalent to* t_2. *Then it is not the case that* $t_1 = t_2 \mathrel{\dot\in} w$ *for any* $w \in \mathcal{G}$.

Proof Suppose $t_1 = t_2 \mathrel{\dot\in} w$ and t_1 is $\mathcal{B}(Z)$-equivalent to t_2. Then using Lemma 10.7, $t_2 = t_2 \mathrel{\dot\in} w$, contradicting the fact that $\mathcal{B}(Z)$ is not closed and so contains no axiom occurrence. ■

Now we are ready to define our model. Actually we construct two models because, as in most approaches to equality, it is first modeled by an equivalence relation, then equivalence classes are introduced.

Construction of Model One

We have already said what our possible worlds are, but we repeat the definition for convenience. Let $\mathcal{G}$ be the set of dynamic boxed sequents that appear on $\mathcal{B}(Z)$, together with the top level dynamic sequent. Continuing, if $w_1, w_2 \in \mathcal{G}$ set $w_1 \mathcal{R} w_2$ just in case $w_2 \mathrel{\dot\in} w_1$. And let $\mathcal{D}$ be the Herbrand universe consisting of object terms (terms built up from parameters, Definition 7.1). We thus have $\mathcal{G}$, $\mathcal{R}$, and $\mathcal{D}$. The interpretation $\mathcal{I}$ is a bit more complicated.

Since parameters are involved, $\mathcal{I}$ must cover them. We do the thing usual with Herbrand universes. If p is 0-place, $\mathcal{I}(p) = p$, and if p is n-ary and $h_1, \ldots, h_n \in \mathcal{D}$, $\mathcal{I}(p)(h_1, \ldots, h_n)$ is the object term $p(h_1, \ldots, h_n)$. Then $\mathcal{I}(t) = t$ for any object term t.

Let a be an n-ary intension function symbol. We take a to designate at $w \in \mathcal{G}$ if $a \nrightarrow p \mathrel{\dot\in} w$ for some p (p is unique by Lemma 10.3). Then we set $\mathcal{I}(a) : S \to (\mathcal{D}^n \to \mathcal{D})$ where S is the subset of $\mathcal{G}$ at which a designates, and if $w \in S$, then $\mathcal{I}(a)(w)(h_1, \ldots, h_n) = p(h_1, \ldots, h_n)$, where p is the unique parameter such that $a \nrightarrow p \mathrel{\dot\in} w$, and $p(h_1, \ldots, h_n)$ is a Herbrand term.

Suppose P is an n-ary relation symbol other than $=$, and $w \in \mathcal{G}$. Set $\mathcal{I}(P)(w) = \{\langle h_1, \ldots, h_n\rangle \mid \neg P(h_1, \ldots, h_n) \mathrel{\dot\in} w\}$. For equality a little more is needed. We set $\mathcal{I}(=)(w)$ to be $\{(h_1, h_2) \mid \neg(h_1 = h_2) \mathrel{\dot\in} w\}$ together with $\{(h_1, h_2) \mid h_1$ is $\mathcal{B}(Z)$ equivalent to $h_2\}$. Finally we model $a \nrightarrow p$ as we did in the Soundness section, so that $\mathcal{M}, w \Vdash_v a \nrightarrow p$ if a does not designate at w, or else a designates at w, but doesn't designate p.

This completes the definition of a model, $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I}\rangle$. The only thing that still needs checking is that equality is interpreted rigidly—the same at all members of $\mathcal{D}$. But this is an easy consequence of the Equality Up and Equality Down Rules.

Now we prove the following key result. It is the opposite of what one might have expected.

Lemma 10.9 (Truth Lemma) *Let $w \in \mathcal{G}$ and let A be any closed formula, allowing parameters. If $A \mathrel{\dot\in} w$ then $\mathcal{M}, w \nVdash_v A$. (Choice of v is not important because A is closed. In the proof v will be arbitrary unless otherwise specified.)*

Proof By induction on the complexity of A. We show the result simultaneously for A and $\neg A$. Assume the result is known for formulas simpler than A. There are several cases to consider.

Atomic A is atomic, so there are no simpler formulas.

> We begin with the conventional non-equality case. Suppose $\neg P(h_1, \ldots, h_n) \mathrel{\dot\in} w$. Then $\langle h_1, \ldots, h_n\rangle \in \mathcal{I}(P)(w)$, so $\mathcal{M}, w \Vdash_v P(h_1, \ldots, h_n)$, and of course $\mathcal{M}, w \nVdash_v \neg P(h_1, \ldots, h_n)$.
>
> Suppose $P(h_1, \ldots, h_n) \mathrel{\dot\in} w$. Then we do not have $\neg P(h_1, \ldots, h_n) \mathrel{\dot\in} w$ by Lemma 10.2. But then $\langle h_1, \ldots, h_n\rangle \notin \mathcal{I}(P)(w)$, so $\mathcal{M}, w \nVdash_v P(h_1, \ldots, h_n)$.
>
> Next we consider equality. If $\neg h_1 = h_2 \mathrel{\dot\in} w$ the argument is as in the previous case. Now suppose $h_1 = h_2 \mathrel{\dot\in} w$. Since $\mathcal{B}(Z)$ is not closed, we do not have $\neg(h_1 = h_2) \mathrel{\dot\in} w$. By Lemma 10.8, h_1 is not $\mathcal{B}(Z)$-equivalent to h_2. But then $(h_1, h_2) \notin \mathcal{I}(=)(w)$ and so $\mathcal{M}, w \nVdash_v h_1 = h_2$.
>
> Finally we have the rather special case of $a \nrightarrow p$. Now $\mathcal{M}, w \nVdash_v a \nrightarrow p$ exactly when a designates at w, and it is p that it designates. But a look at the definitions earlier shows that these conditions obtain exactly when $a \nrightarrow p \mathrel{\dot\in} w$.

Propositional We consider only $A = X \wedge Y$, the other propositional cases are similar.

Suppose $X \wedge Y \mathrel{\dot\in} w$. Since a fair proof search was made, at some point an α rule application was introduced, and so either $X \mathrel{\dot\in} w$ or $Y \mathrel{\dot\in} w$. Then by the induction hypothesis, either $\mathcal{M}, w \nVdash_v X$ or $\mathcal{M}, w \nVdash_v Y$, and hence $\mathcal{M}, w \nVdash_v X \wedge Y$.

The $\neg A = \neg(X \wedge Y)$ argument is along the same lines, but using the β rule.

Modal Suppose $A = \Box X$, the case of $\Diamond X$ is similar.

Suppose $\Box X \mathrel{\dot\in} w$. Since a fair proof search was made, at some point a ν Rule application was introduced, and a new dynamic nested box, call it w', was created using X. Then $w' \in \mathcal{G}$, $w \mathcal{R} w'$, and $X \mathrel{\dot\in} w'$. By the induction hypothesis, $\mathcal{M}, w' \nVdash_v X$, and so $\mathcal{M}, w \nVdash_v \Box X$.

The $\neg A = \neg\Box X$ case is similar, but using the π Rule.

Predicate Abstract Suppose $A = \langle \lambda x.X(x) \rangle (a(t_1, \ldots, t_n))$.

This time we examine the negated case. Suppose $\neg\langle \lambda x.X(x) \rangle (a(t_1, \ldots, t_n)) \mathrel{\dot\in} w$. At some point in our fair proof search an application of one of the Negative Abstract Rules 1 or 2 is introduced concluding this. Either way, we have a unique p so that both $a \not\twoheadrightarrow p \mathrel{\dot\in} w$ and $\neg X(p(t_1, \ldots, t_n)) \mathrel{\dot\in} w$. By the induction hypothesis, $\mathcal{M}, w \nVdash_v \neg X(p(t_1, \ldots, t_n))$, and so $\mathcal{M}, w \Vdash_v X(p(t_1, \ldots, t_n))$. Since $a \not\twoheadrightarrow p \mathrel{\dot\in} w$, we have that a designates at w, and $\mathcal{I}(a)(w)(\mathcal{I}(t_1), \ldots, \mathcal{I}(t_n)) = \mathcal{I}(a)(w)(t_1, \ldots, t_n) = p(t_1, \ldots, t_n)$. Let v' be like v except that $v'(x) = p(t_1, \ldots, t_n)$; then $\mathcal{M}, w \Vdash_{v'} X(x)$, and so $\mathcal{M}, w \Vdash_v \langle \lambda x.X(x) \rangle (a(t_1, \ldots, t_n))$, and hence $\mathcal{M}, w \nVdash_v \neg\langle \lambda x.X(x) \rangle (a(t_1, \ldots, t_n))$.

The case where $A = \langle \lambda x.X(x) \rangle (a(t_1, \ldots, t_n))$ is similar.

■

Let t be the top level sequent of $\mathcal{B}(Z)$. Then $Z \mathrel{\dot\in} t$, so by the Truth Lemma, $\mathcal{M}, t \nVdash_v Z$. Thus $\mathcal{M}$ is a counter-model.

Construction of Model Two

The interpretation of the equality symbol in the model $\mathcal{M}$, constructed above, is not by the equality relation. However, it turns out that $\mathcal{I}(=)(w)$ is the same for all w in $\mathcal{G}$, is an equivalence relation on $\mathcal{D}$, and is a congruence with respect to all relation and function symbols. Once this is established, a 'factor' model can be constructed with an object domain consisting of equivalence classes from $\mathcal{D}$. This construction is quite standard, and we omit all details. It is necessary, however, to establish the facts about $\mathcal{I}(=)$ just mentioned.

First we show that $\mathcal{I}(=)(w)$ is the same for all $w \in \mathcal{G}$. Suppose $(h_1, h_2) \in \mathcal{I}(=)(w)$. The definition of interpretation for equality has two parts. It could be that $\neg h_1 = h_2 \mathrel{\dot\in} w$. Then Lemma 10.4 gives us what we need. Or else it could be that h_1 is $\mathcal{B}(Z)$-equivalent to

h_2. But Definition 10.6 does not depend on a particular choice of possible world, relying on Lemma 10.4.

Next we show that the interpretation of equality, which we now know is world independent, is an equivalence relation. It is reflexive because h is $\mathcal{B}(Z)$-equivalent to h since it $\mathcal{B}(Z)$-rewrites to itself with 0 rewritings. Now consider symmetry. Suppose $(h_1, h_2) \in \mathcal{I}(=)(w)$. There are two possibilities. First, $\neg h_1 = h_2 \dot{\in} w$, and we appeal to Lemma 10.5. Second, h_1 is $\mathcal{B}(Z)$-equivalent to h_2, and clearly $\mathcal{B}(Z)$-rewriting is bidirectional, making use of Lemma 10.5 again. We leave transitivity to the reader.

We need to establish we have a congruence with respect to all relation symbols. Let us assume P is not the equality symbol, and leave the equality case to the reader. Suppose $(h_1, \ldots, h_n) \in \mathcal{I}(P)(w)$ and $(h_1, h_1') \in \mathcal{I}(=)(w)$. By the first condition, and the definition of $\mathcal{I}$, $\neg P(h_1, \ldots, h_n) \dot{\in} w$. As usual, there are two possibilities for the second condition. We could have $\neg h_1 = h_1' \dot{\in} w$. In this case, by Lemma 10.1, both $\neg P(h_1, \ldots, h_n)$ and $\neg h_1 = h_1'$ must be present at some same stage in the construction of w, and a fair construction procedure will eventually bring in $\neg P(h_1', \ldots, h_n)$ using the Equality Left-Right Substitution Rule. The other possibility is that h_1 is $\mathcal{B}(Z)$ equivalent to h_1'. In this case we apply Lemma 10.7.

Finally we need to show congruence properties with respect to function symbols. To keep notation simple we work with a unary function symbol. Suppose $(h_1, h_2) \in \mathcal{I}(=)(w)$; we show $(f(h_1), f(h_2)) \in \mathcal{I}(=)(w)$. There are the usual two cases. We might have that $\neg h_1 = h_2 \dot{\in} w$, or we might have that h_1 is $\mathcal{B}(Z)$-equivalent to h_2. But either way, h_1 is $\mathcal{B}(Z)$-equivalent to h_2, and it follows from the definition that $f(h_1)$ is $\mathcal{B}(Z)$-equivalent to $f(h_2)$. giving us what we need.

This establishes what was needed, and ends our discussion of Model Two.

11 Conclusion

All formal work here has been for systems based on the simplest normal modal logic K. Propositional nested sequent rules for other modal logics can be found in [1, 10] and elsewhere. Using rules for other modal logics, together with the Predicate Abstract and Equality Rules, gives appropriate formal systems for a family of modal logics containing the machinery discussed here. Our completeness and soundness methods carry over.

Predicate Abstract machinery is intermediate between propositional and first-order, in an obvious sense. Which way it leans is not always obvious. For instance, it is shown in [5] that logics between $\mathsf{K4}$ and $\mathsf{S5}$, with predicate abstraction and equality are undecidable, and for $\mathsf{S5}$, equality is not needed for undecidability. For K, T, and other simpler modal logics with predicate abstraction and equality, we have decidability if no function symbols are present—a fair proof search procedure must terminate. What the effect of function symbols might be is not known.

In Sections 2 and 5 we discussed a few natural examples, and saw how predicate abstrac-

tion machinery dealt with them. This is not the only formalization possible. Quantifiers can be made use of. Cross-world predication can be used, as in [19]. A multiplicity of methods is not a drawback. Each approach has its own virtues and provides its own insights. Relationships between approaches needs investigation, and we encourage readers to think about this.

References

[1] Kai Brünnler. Deep sequent systems for modal logic. *Archive for Mathematical Logic*, 48(6):551–577, 2009.

[2] Kai Brünnler. *Nested Sequents*. Habilitation thesis, Institut für Informatik und angewandte Mathematik, Universität Bern, 2010.

[3] Rudolph Carnap. *Meaning and Necessity*. University of Chicago Press, Chicago, enlarged edition 1956 edition, 1947.

[4] Melvin C. Fitting. *First-Order Logic and Automated Theorem Proving*. Springer-Verlag, 1996. First edition, 1990. Errata at `http://comet.lehman.cuny.edu/fitting/errata/errata.html`.

[5] Melvin C. Fitting. Modal logics between propositional and first-order. *Journal of Logic and Computation*, 12:1017–1026, 2002.

[6] Melvin C. Fitting. First-order intensional logic. *Annals of Pure and Applied Logic*, 127:171–193, 2004.

[7] Melvin C. Fitting. FOIL axiomatized. *Studia Logica*, 84(1):1–22, 2006. Electronic version at `http://dx.doi.org/10.1007/s11225-006-9000-2`. See Correction, [8].

[8] Melvin C. Fitting. Correction to *FOIL Axiomatized*. *Studia Logica*, 85(2):275, 2007. Electronic version at `http://dx.doi.org/10.1007/s11225-007-9032-2`.

[9] Melvin C. Fitting. Proving completeness for nested sequent calculi. In Jean-Yves Béziau and Marcelo Esteban Coniglio, editors, *Logic Without Frontiers, Festschrift for Walter Alexandre Carnielli on the occasion of his 60th Birthday*, Tributes, pages 145–154. College Publications, 2011.

[10] Melvin C. Fitting. Prefixed tableaus and nested sequents. *Annals of Pure and Applied Logic*, 163:291–313, 2012. Available on-line at http://dx.doi.org/10.1016/j.apal.2011.09.004.

[11] Melvin C. Fitting. Nested sequents for intuitionistic logics. Forthcoming in Notre Dame Journal of Formal Logic, 2013.

[12] Melvin C. Fitting and Richard Mendelsohn. *First-Order Modal Logic*. Kluwer, 1998. Paperback, 1999. Errata at `http://comet.lehman.cuny.edu/fitting/errata/errata.html`.

[13] Christie Ward (Gunnvôr). The Viking Answer Lady. `http://www.vikinganswerlady.com`, 2013. Online; accessed August 2, 2013.

[14] R. Kashima. Cut-free sequent calculi for some tense logics. *Studia Logica*, 53:119–135, 1994.

[15] F. Poggiolesi. The tree-hypersequent method for modal propositional logic. In David Makinson, Jacek Malinowski, and Heinrich Wansing, editors, *Trends in Logic: Towards Mathematical Philosophy*, pages 31–51. Springer, 2009.

[16] Bertrand Russell. On denoting. *Mind*, 14 (new series):479–493, 1905.

[17] Robert Stalnaker and Richmond Thomason. Abstraction in first-order modal logic. *Theoria*, 34:203–207, 1968.

[18] Richmond Thomason and Robert Stalnaker. Modality and reference. *Nous*, 2:359–372, 1968.

[19] Kai F. Wehmeier. Subjunctivity and cross-world predication. *Philosophical Studies*, 159(1):107–122, 2012.

[20] Alfred North Whitehead and Bertrand Russell. *Principia Mathematica*. Cambridge University Press, Cambridge, 2nd edition, 1927. First edition, 1910.