Bertrand Russell, Herbrand's theorem, and the assignment statement

Melvin Fitting

Dept. Mathematics and Computer Science
Lehman College (CUNY), Bronx, NY 10468
fitting@alpha.lehman.cuny.edu
http://math240.lehman.cuny.edu/fitting

Abstract. While propositional modal logic is a standard tool, first-order modal logic is not. Indeed, it is not generally understood that conventional first-order syntax is insufficiently expressible. In this paper we sketch a natural syntax and semantics for first-order modal logic, and show how it easily copes with well-known problems. And we provide formal tableau proof rules to go with the semantics, rules that are, at least in principle, automatable.

1 Introduction

Propositional modal logic has become a fairly standard item in certain areas of artificial intelligence and computer science. Computational states are often thought of as possible worlds, while knowledge is frequently modeled using Hintikka's idea that a multiplicity of possible worlds reflects our ignorance about the actual one. But *first-order* modal logic is less familiar. It is often seen as a labyrinth full of twists and problems (see [5], for instance). Indeed, standard first-order syntax is actually of inadequate expressive power for modal logic. This is behind many of the well-known "paradoxes" of modal logic. But, a solution to the expressiveness problems of first-order modal logic exists, though it is still not as well-known as it deserves to be. It was explicitly introduced to modal logic in [8, 9], though as we will see, the underlying ideas are much earlier.

It is always interesting to find that problems in disparate areas have a common solution. In our case the areas we wish to look at include: the theory of definite descriptions of Bertrand Russell; certain classical philosophical problems; inadequate expressibility of logics of knowledge; and the treatment of the assignment statement in dynamic logic. The source of difficulty in all these cases is the same: classical symbolic logic is taken as the standard. In classical logic assumptions are made that preclude certain kinds of problems—they cannot arise. Since these problems do not arise in classical logic, machinery for a solution is missing as well. Since the problems are significant, and the solution is trivial, the issues should be better known.

The material in this paper is a drastically abbreviated version of a book-length presentation forthcoming soon [2]. While our book is primarily aimed at philosophers, it contains much discussion of formal semantics and tableau-based proof procedures. Automation of these proof procedures still awaits.

2 Frege's Puzzle

The philosopher Frege introduced a famous distinction between *sense* (sinn) and *denotation* (bedeutung), [3]. In part, this was to deal with problems like the following. The morning star is the evening star. Let us symbolize this by $m = e$, a statement of identity. When the ancients came to know the truth of this statement it was a discovery of astronomy. But why could it not have been established by pure logic as follows. Certainly the ancients knew objects were self-identical; in particular, $K(m = m)$—we are using K as a knowledge modality. Since $m = e$ in fact, using substitutivity of equals for equals, occurrences of m can be replaced by occurrences of e. Doing so with the second occurrence in $K(m = m)$ gives us $K(m = e)$. Yet this does not appear to be how the Babylonians did it.

On the basis of this and other examples, Frege came to insist that terms such as "morning star" have both a denotation (in this case, a particular astronomical object) and a sense (roughly, how the object is designated). Identity of denotation is expressed by $m = e$, but in a non truth-functional context, such as $K(m = e)$, sense is what matters. This distinction has given rise to a vast literature, but it is enough here to point out that in *mathematical* logic, all contexts are truth-functional, and the problem fades into the background.

3 Russell, On Denoting

Bertrand Russell, in a famous paper [7], gave a theory of definite descriptions—he showed how to treat expressions like "the positive square root of 3" in a formal language. This was to be of special significance later in *Principia Mathematica* since it allowed classes to be introduced via definite descriptions. Like Frege, Russell too had a guiding example. In his case it was how to assign a truth value to "the King of France is bald," given that France has no King.

His solution involved a distinction between grammatical form and logical form. In this case the grammatical form is $B(f)$ (where B is the bald predicate and f is the King of France). But this cannot be the logical structure since f does not designate. According to Russell's theory the expression should be expanded to the following, where F is a predicate specifying what it means to be King of France: $(\exists x)\{F(x) \wedge (\forall y)(F(y) \supset y = x) \wedge B(x)\}$. Something Kings France, only one thing does so, and that thing is bald. Clearly, then, the sentence is false.

Then what about the sentence, "the King of France is not bald." Russell noted an ambiguity here. Should this be taken to assert the King of France has the non-baldness property, or should it be taken to be the negation of the assertion that the King of France has the baldness property? Using the Russell translation, do

we negate the predicate $B(x)$, or do we negate the entire sentence? These are not equivalent formally, and the original English sentences do not seem equivalent informally.

To deal with this, Russell at first introduced a *narrow scope/broad scope* distinction, which I won't discuss further. It turned out to be inadequate, and eventually he was led to introduce an explicit scoping mechanism, one that is used systematically in *Principia Mathematica*. Russell's notation is somewhat fierce, and I won't reproduce it here. But the underlying idea is critical, and it's a wonder it came with such difficulty. It amounts to this. When definite descriptions are translated away, an existential quantifier is introduced. That quantifier has a formal scope, given by the usual quantifier rules. Then it must be that definite descriptions also have scopes within sentences. Generally that scope is implicit in natural language constructs, but machinery can be introduced to make it explicit. In short, *terms* of a formal language can have scopes, just as quantified variables do.

Today Russell's treatment of definite descriptions is well-known, at least among those to whom it is well-known. But the explicit introduction of a scoping mechanism is pushed into the background and generally ignored. In fact, it is a central point.

4 Modal Issues

Modal logic is a standard formal tool today in many areas of computer science and artificial intelligence. Propositional issues are well-known, while first-order ones may be less familiar. It is in the modal setting that the problems with scoping become particularly clear, using the familiar machinery of Kripke possible-world semantics.

Suppose we have a first-order modal frame $\langle \mathcal{G}, \mathcal{R}, \mathcal{D} \rangle$, consisting of a collection $\mathcal{G}$ of *possible worlds*, an *accessibility relation* $\mathcal{R}$ between worlds, and a *domain function* $\mathcal{D}$ assigning non-empty domains for quantification to possible worlds. No special assumptions are needed for the points I am about to make. $\mathcal{R}$ can be an equivalence relation, or transitive only, or have no particular properties. Likewise $\mathcal{D}$ can be the constant function, entirely arbitrary, or something inbetween.

Now, consider the behavior of the formula $\Diamond P(c)$, where c is a constant symbol and P is a predicate symbol. We want to allow c to be *non-rigid*, perhaps designating different things at different worlds. For instance, if "the King of France" were to be represented by a constant symbol in a temporal model, it would designate different people at different times. Of course sometimes it would not designate at all—ignore this point for now. Formally, let us say we have an *interpretation* $\mathcal{I}$ that assigns to constant symbols such as c, and to each possible world $\Gamma \in \mathcal{G}$ some object $\mathcal{I}(c, \Gamma)$—the "meaning" of c at Γ. We also assume $\mathcal{I}$ assigns to each relation symbol and each possible world some actual relation. We thus have a *model* $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$, based on the original frame. Finally let us symbolize by $\mathcal{M}, \Gamma \Vdash X$ the notion that the closed formula X is true at

world Γ of the model $\mathcal{M}$. Now, just what should

$$\mathcal{M}, \Gamma \Vdash \Diamond P(c) \tag{1}$$

mean? I assume you are generally familiar with the behavior of modal operators in possible-world settings.

First Possibility The formula $\Diamond P(c)$ asserts that, whatever c means, it has the "$\Diamond P$" property. A reasonable way of formalizing this in models is to allow the occurrence of free variables, together with a *valuation* function to assign values to them. Thus we broaden the machinery a bit, and write $\mathcal{M}, \Gamma \Vdash_v X$ to mean: X (which may contain free variables) is true at world Γ of model $\mathcal{M}$, with respect to valuation v which assigns values to the free variables of X. With this extra machinery, (1) could be taken to mean $\mathcal{M}, \Gamma \Vdash_v \Diamond P(x)$ where $v(x) = \mathcal{I}(c, \Gamma)$. That is, whatever c designates at Γ is something for which $\Diamond P(x)$ is true (at Γ).

Since $\Diamond P(x)$ is a formula whose main connective is $\Diamond$, according to the usual behavior of modal models, there is some accessible world $\Delta \in \mathcal{G}$, that is $\Gamma \mathcal{R} \Delta$, such that $\mathcal{M}, \Delta \Vdash_v P(x)$ and this is the case provided $v(x)$ is in the relation $\mathcal{I}(P, \Delta)$, the "meaning" of P at Δ. Tracing back, for (1) to be true, we should have $\mathcal{I}(c, \Gamma) \in \mathcal{I}(P, \Delta)$.

Second Possibility The formula $\Diamond P(c)$ has $\Diamond$ as its main connective, so (1) says there is a possible world $\Omega \in \mathcal{G}$ with $\Gamma \mathcal{R} \Omega$ such that $\mathcal{M}, \Omega \Vdash P(c)$ and, most reasonably, this should be so if $\mathcal{I}(c, \Omega)$ is in the relation $\mathcal{I}(P, \Omega)$. Thus (1) means $\mathcal{I}(c, \Omega) \in \mathcal{I}(P, \Omega)$.

We got two ways of reading (1). The world Δ of the First Possibility and the world Ω of the Second Possibility need not be the same, but for simplicity suppose they are, $\Delta = \Omega$. Then we still have the following alternate readings for (1):

1. $\mathcal{I}(c, \Gamma) \in \mathcal{I}(P, \Omega)$
2. $\mathcal{I}(c, \Omega) \in \mathcal{I}(P, \Omega)$

and these are *not* equivalent. They are not equivalent because we allowed c to be non-rigid, so $\mathcal{I}(c, \Gamma)$ and $\mathcal{I}(c, \Omega)$ can be different.

Once the problem is seen, the solution is obvious. We need Russell's scoping mechanism, and just such a device was introduced into modal logic in [8, 9]. What it amounts to is separating the notion of formula and predicate. Using notation based on the Lambda-calculus, we *abstract* from a formula $\varphi(x)$ a *predicate* $\langle \lambda x.\varphi(x) \rangle$. A formal treatment of this is given in the next section, but for now we observe that it separates $\Diamond P(c)$ into the following two distinct formulas:

1. $\langle \lambda x.\Diamond P(x) \rangle (c)$
2. $\Diamond \langle \lambda x.P(x) \rangle (c)$

5 Formal Semantics

Based on the sketchy ideas above, we set out a formal modal semantics. And because of space limitations, we go directly to the most general version—in [2] the treatment is much more leisurely. We have already said what a modal *frame* was above, but the present notion of *model* is considerably broadened.

First, in a modal frame $\langle \mathcal{G}, \mathcal{R}, \mathcal{D} \rangle$, recall that we place no restrictions on the domain function $\mathcal{D}$—it assigns to each world some non-empty set, the *domain* of that world. Members of the domain of a world should be thought of as the things "actually" existing at that world. If something exists at a world Δ, at a different world Γ we can think of that thing as a "possible existent." Also, by the *domain of the frame* we mean $\cup_{\Gamma \in \mathcal{G}} \mathcal{D}(\Gamma)$. Thus the members of the domain of the frame are the things that are actually or possibly existent at every world.

A modal *model* is a structure $\langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$ where $\langle \mathcal{G}, \mathcal{R}, \mathcal{D} \rangle$ is a modal frame and $\mathcal{I}$ is an *interpretation* such that:

1. For each relation symbol P and each $\Gamma \in \mathcal{G}$, $\mathcal{I}(P, \Gamma)$ is a relation on the domain of the frame.
2. For some, not necessarily all, constant symbols c and $\Gamma \in \mathcal{G}$, $\mathcal{I}(c, \Gamma)$ is a member of the domain of the frame.

This can be extended to include function symbols, though because of space limitations in this paper, we do not do so.

Think of $\mathcal{I}(P, \Gamma)$ as the "meaning" of the relation symbol P at the world Γ. Note the important point that it may include things that do not exist at Γ. It is true to say, "Pegasus is a mythological beast," and we interpret this to mean that "Pegasus" designates something in a world other than this one (a make-believe world, if you will), the thing designated does not exist in this world, but in this world the property of being mythological correctly applies to it.

Think of $\mathcal{I}(c, \Gamma)$ as what c designates at the world Γ. As with relation symbols, what is designated need not exist. Thus "the first president of the United States" designates George Washington, who in a temporal sense once existed but no longer does. Note the added complication that for constant symbols $\mathcal{I}$ is allowed to be partial. This gives us the start of a mechanism to deal with "the present King of France."

Before we say which formulas are true at which worlds, we should indicate just what things we take as formulas.

1. As atomic formulas we take expressions of the form $R(x_1, \ldots, x_n)$ where R is an n-place relation symbol and $x_1, \ldots, x_n$ are *variables*. (Recall, formulas are not properties, but rather, properties are abstracted from formulas.)
2. Complex formulas are built up from simpler ones using $\wedge$, $\vee$, $\neg$, $\supset$, $\equiv$, $\Box$, $\Diamond$, $\forall$, and $\exists$ in the usual way, with the usual conventions about free variables.
3. If t is a term (which here means a variable or a constant symbol), φ is a formula, and x is a variable, $\langle \lambda x.\varphi \rangle(t)$ is a formula. Its free variables are those of φ, except for occurrences of x, together with the free variables of t.

Think of $\langle \lambda x.\varphi \rangle(t)$ as asserting of the object t designates that it has the property $\langle \lambda x.\varphi \rangle$, the property abstracted from the formula φ.

Let $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$ be a model. A *valuation* in $\mathcal{M}$ is a mapping v assigning to each variable x some member $v(x)$ in the domain of the frame underlying the model. Note that valuations are not world dependent.

We say a term t *designates at* Γ if t is a variable, or if t is a constant symbol and $\mathcal{I}(t, \Gamma)$ is defined. If t designates at Γ we use the following notation:

$$(v * \mathcal{I})(t, \Gamma) = \begin{cases} v(x) & \text{if } t \text{ is the variable } x \\ \mathcal{I}(c, \Gamma) & \text{if } t \text{ is the constant symbol } c \end{cases}$$

Finally we must define $\mathcal{M}, \Gamma \Vdash_v \varphi$: formula φ is true at world Γ of model $\mathcal{M}$ with respect to valuation v. Here is the truth definition, much of which is straightforward.

1. For an atomic formula $R(x_1, \ldots, x_n)$, $\mathcal{M}, \Gamma \Vdash_v R(x_1, \ldots, x_n)$ just in case $\langle v(x_1), \ldots, v(x_n) \rangle \in \mathcal{I}(R, \Gamma)$.
2. $\mathcal{M}, \Gamma \Vdash_v X \wedge Y$ if and only if $\mathcal{M}, \Gamma \Vdash_v X$ and $\mathcal{M}, \Gamma \Vdash_v Y$ (and similarly for the other Boolean connectives).
3. $\mathcal{M}, \Gamma \Vdash_v \Box X$ if and only if $\mathcal{M}, \Delta \Vdash_v X$ for every $\Delta \in \mathcal{G}$ such that $\Gamma \mathcal{R} \Delta$ (and similarly for $\Diamond$).
4. $\mathcal{M}, \Gamma \Vdash_v (\forall x)\varphi$ if and only if $\mathcal{M}, \Gamma \Vdash_{v'} \varphi$ for every valuation v' that is like v except that v' is some arbitrary member of $\mathcal{D}(\Gamma)$ (and similarly for $\exists$).
5. If t does not designate at Γ, $\mathcal{M}, \Gamma \nVdash_v \langle \lambda x.\varphi \rangle(t)$.
6. If t designates at Γ, $\mathcal{M}, \Gamma \Vdash_v \langle \lambda x.\varphi \rangle(t)$ if and only if $\mathcal{M}, \Gamma \Vdash_{v'} \varphi$ where v' is like v except that $v'(x)$ is what t designates at Γ.

Item 4 makes explicit the idea that quantifiers quantify over what actually exists—over the domain of the particular possible world only. Such quantifiers are called "actualist" by philosophers. They are not the only version available, but a choice among quantifiers is not the issue just now. Item 5 is the formal counterpart of the informal notion that no assertion about what is designated by a term that fails to designate can be correct. Note that the issue is designation by the term, not existence of the object designated. And item 6 expresses the idea that properties are properties of objects, and so we must know what object a term designates before knowing if a property applies.

One last item before turning to examples. We will always assume our models are *normal*: there is a relation symbol $=$, written in infix position, and $\mathcal{I}(=, \Gamma)$ is the equality relation on the domain of the model, for every world Γ.

6 A Few Examples

We discuss several examples to show the richness these simple ideas provides us. We give more examples once tableau proof machinery has been introduced.

Example 1. We begin with Frege's morning star/evening star problem. Suppose we consider an epistemic model in which the various possible worlds are those

compatible with the knowledge possessed by the early Babylonians, with the actual situation among them, of course. The constant symbols m and e are intended to designate the morning and the evening stars respectively. (They designate the same object in the actual world, but need not do so in every possible world.) Also, let us read $\Box$ as "the ancients knew that." How can we have $m = e$ without, by substitutivity of equality, also having $\Box(m = e)$?

There is a certain amount of deception in the paragraph above. Neither $m = e$ nor $\Box(m = e)$ is a formula in our formal system. (Recall, constants cannot appear in atomic formulas, rather they enter via predicate abstraction.) The incorrect $m = e$ should be replaced with $\langle\lambda x.\langle\lambda y.x = y\rangle(e)\rangle(m)$, which we abbreviate as $\langle\lambda x, y.x = y\rangle(m, e)$. More significantly, for $\Box(m = e)$ we have a choice of replacements: $\Box\langle\lambda x, y.x = y\rangle(m, e)$, or $\langle\lambda x, y.\Box(x = y)\rangle(m, e)$, or even $\langle\lambda x.\Box\langle\lambda y.x = y\rangle(e)\rangle(m)$. These do not behave the same. And, as a matter of fact, the formula

$$\langle\lambda x, y.x = y\rangle(m, e) \supset \langle\lambda x, y.\Box(x = y)\rangle(m, e) \tag{2}$$

is valid (true at all worlds of all models) while

$$\langle\lambda x, y.x = y\rangle(m, e) \supset \Box\langle\lambda x, y.x = y\rangle(m, e) \tag{3}$$

is not. (We leave the demonstration to you.) A little thought shows that in formula (2), $\langle\lambda x, y.\Box(x = y)\rangle(m, e)$ asserts that the ancients knew, of the objects denoted by m and e (in the actual world) that they were identical. This, in fact, is so, since they certainly knew that an object is self-identical. But in formula (3), $\Box\langle\lambda x, y.x = y\rangle(m, e)$ asserts the ancients knew that m and e designated the same object, and at one time they did not know this. The use of predicate abstraction serves to disambiguate $(m = e) \supset \Box(m = e)$ into a valid version and an invalid version, corresponding roughly to Frege's use of reference and sense.

Example 2. The introduction of modal operators is not essential to see the effects of predicate abstraction. Consider the formula

$$\langle\lambda x.\neg\varphi\rangle(c) \equiv \neg\langle\lambda x.\varphi\rangle(c) \tag{4}$$

If we evaluate the truth of this at a possible world at which c designates, the equivalence is valid. But if we are at a world at which c does not designate, the left side of (4) is false, since no abstract correctly applies to a term that fails to designate. But for the same reason, $\langle\lambda x.\varphi\rangle(c)$ is false, so its negation, the right side of (4), is true. Thus if c fails to designate, (4) is false. This epitomizes precisely the distinction Russell made between the King of France having the non-baldness property, and the King of France failing to have the baldness property.

Note that if c does designate, (4) is true. And, whether c designates or not, $\langle\lambda x.\varphi\wedge\psi\rangle(c) \equiv (\langle\lambda x.\varphi\rangle(c)\wedge\langle\lambda x.\psi\rangle(c))$ is true at each possible world. In classical logic, it is assumed that terms designate, so the consequence of (4) and this is to make the effects of predicate abstraction invisible. Only when Russell tried to treat definite descriptions, that may lack designation, or when Frege considered non-truth functional contexts, did such effects turned up.

Example 3. A term t may or may not designate at a particular possible world Γ. If it does, $\langle \lambda x.x = x \rangle(t)$ is true there, since whatever t designates is self-identical. But also, if t does not designate at a world, $\langle \lambda x.x = x \rangle(t)$ is false there, since predicate abstracts are false when applied to non-designating terms. Therefore, let us define a "designation" abstract:

$$\mathbf{D} \text{ abbreviates } \langle \lambda x.x = x \rangle.$$

This allows us to move a semantic notion into syntax—as we have seen, $\mathbf{D}(t)$ is true at a world if and only if t designates at that world.

Using our designation abstract, let us return to Example 2. We saw that formula (4) is true if c designates. Consequently we have the validity of the following.

$$\mathbf{D}(c) \supset [\langle \lambda x.\neg\varphi \rangle(c) \equiv \neg\langle \lambda x.\varphi \rangle(c)] \tag{5}$$

Remarkably enough, the equivalence holds the other way around as well. That is, the following is valid.

$$\mathbf{D}(c) \equiv [\langle \lambda x.\neg\varphi \rangle(c) \equiv \neg\langle \lambda x.\varphi \rangle(c)] \tag{6}$$

Example 4. In exactly the same way that the semantic notion of designation could be expressed syntactically, we can express existence as well. We introduce the following abbreviation.

$$\mathbf{E} \text{ abbreviates } \langle \lambda x.(\exists y)(y = x) \rangle.$$

It is easy to show that t designates something that exists at world Γ if and only if $\mathbf{E}(t)$ is true at Γ.

We do not have the validity of: $(\forall x)\varphi \supset \langle \lambda x.\varphi \rangle(t)$. But if we assume that t not only designates, but designates something that exists, things become better. The following is valid: $\mathbf{E}(t) \supset [(\forall x)\varphi \supset \langle \lambda x.\varphi \rangle(t)]$.

In classical logic one cannot even talk about things that do not exist, and the $\mathbf{E}(t)$ antecedent above is unnecessary.

Example 5. Suppose p is a constant symbol intended to designate the President of the United States. Assume we have a temporal model in which possible worlds represent instants of time, and $\Box X$ means X is and will remain true, that is, X is true in the present and all future states. Dually, $\Diamond X$ means X is or will become true. For simplicity, let us assume there will always be a President of the United States, so p designates at all times. Now, consider the following formula.

$$\langle \lambda x.\Diamond\neg\langle \lambda y.x = y \rangle(p) \rangle(p) \tag{7}$$

To say this is true at the current world asserts, of the current President of the United States, that at some future time he will not be identical to the individual who is then the President of the United States. That is, (7) asserts: "someday the President of the United States will not be the President of the United States." This is not a valid formula, but it is satisfiable.

7 Herbrand's Theorem

In classical logic, Herbrand's theorem provides a reduction of the first-order provability problem to the propositional level, plus an open-ended search. It is often considered to be the theoretical basis of automated theorem proving for classical logic. Unfortunately, Herbrand's theorem does not extend readily to non-classical logics. Fortunately, the use of predicate abstraction allows us to prove a reasonable version for first-order modal logics. Since the full statement of the resulting theorem is somewhat complex, I only raise a few of the issues, and refer to [1] for a fuller treatment.

Classically, the first step of formula processing in the Herbrand method involves the introduction of Skolem functions. To cite the simplest case, the formula $(\exists x)P(x)$ is replaced with $P(c)$, where c is a new constant symbol. It is not the case that the two formulas are equivalent, but they are equi-satisfiable—if either is, both are.

One would like to Skolemize modally as well, but consider the following formula: $\Box(\exists x)P(x)$. If this is true at a possible world Γ of some modal model, then $(\exists x)P(x)$ is true at each world accessible from Γ. Say Δ and Ω are two such accessible worlds. Then at Δ, some member of the domain associated with Δ satisfies $P(x)$—let the new constant symbol c designate such an object at Δ. The situation is similar with Ω, so let c designate some member of the domain associated with Ω that satisfies $P(x)$ there. In general, c will designate different things at Δ and Ω, and so will be non-rigid. Thus the Skolemization of $\Box(\exists x)P(x)$ seems to be $\Box P(c)$, where c is a new non-rigid constant symbol. But, as we have seen, once non-rigid constant symbols are admitted, conventional syntax becomes ambiguous. Indeed, it would appear that $\Box P(c)$ should also be the Skolemization of $(\exists x)\Box P(x)$, and this seems quite unreasonable, since the two quantified formulas having a common Skolemization behave quite differently.

Of course, the solution involves using predicate abstraction. The proper Skolemization for $\Box(\exists x)P(x)$ is $\Box\langle\lambda x.P(x)\rangle(c)$, while $(\exists x)\Box P(x)$ has a Skolemization of $\langle\lambda x.\Box P(x)\rangle(c)$, which is behaviorally distinct.

Similar results apply to more complex formulas, but function symbols must be involved, and we do not attempt to give a full presentation here. Suffice it to say that the full force of Herbrand's theorem can be brought to bear, even in a modal setting.

8 Dynamic Logic

One of the interesting applications of multi-modal logic is *dynamic logic*, a logic of programs, [6]. In addition to the usual machinery of modal logic, a class of *actions* is introduced, with the class of actions closed under various operations, such as sequencing, repetition, and so on. For each action α there is a corresponding modal operator, generally written $[\alpha]$. The formula $[\alpha]X$ is informally read: after action α is completed, X will be true. (Since non-determinism is allowed, there may be several ways of completing α.) There is a corresponding semantics in

which possible worlds are possible states of a computation. Likewise there is a proof theory, at least for the propositional case. A typical principle of dynamic logic is $[\alpha;\beta]X \equiv [\alpha][\beta]X$, where the semicolon corresponds to sequencing of actions.

Dynamic logic provides an elegant treatment of compound actions, but what about atomic ones? Consider the assignment statement $c := c + 1$—what is its dynamic characterization? We are all familiar with the before/after behavior of assignment statements, where the right-hand side uses the current value of c, while the left-hand side reflects the new value it acquires. To explain $c := c + 1$ in English, we would say something like: "after execution, the value of c is one more than its value before execution."

To formalize this, it is enough to recognize that c is non-rigid—it designates different things at different computational states. Then, assuming arithmetic behaves in the expected way, the essential feature of the assignment statement in question is captured by the following, in which we use $\Box$ as shorthand for $[c := c + 1]$.

$$\langle \lambda x.\Box\langle \lambda y.y = x + 1\rangle(c)\rangle(c) \tag{8}$$

What this expresses is: it is true of the current value of c that, after $c := c + 1$ is executed, the value of c will be that plus 1.

If we assume, about arithmetic, only that incrementing a number gives us a result unequal to the number, then it is easily shown to be a logical consequence of (8) that

$$\langle \lambda x.\Box\neg\langle \lambda y.(y = x)\rangle(c)\rangle(c) \tag{9}$$

This should be compared with (7). Indeed, both simply amount to assertions that p in the one case and c in the other are non-rigid.

Issues of designation and existence are relevant in dynamic logic as well. Saying c designates at a computational state amounts to saying it has been initialized, in the standard programming sense. Saying c exists at a computational state says something about c's availability—we are in the scope of c. Formal notions here are somewhat unclear, but it would be interesting to work them out fully. Finally, saying that c is *rigid* is simply saying that c is *const*, as in C or C^{++}, or *final* as in Java.

Full first-order dynamic logic is not axiomatizable. What we propose is that the addition of terms, equality, and predicate abstraction to propositional dynamic logic, without adding quantifiers, might serve as a satisfactory strengthening of propositional dynamic logic. It is a subject worth investigating.

9 Tableau Proof Methods

Formal proof rules based on *prefixed* tableaus are quite natural for the constructs discussed above. Here I give rules for varying domain S5—versions for other standard modal logics are presented in [2], but that for S5 is simplest to present.

Because of space limitations, rather than giving examples as we go along, I'll reserve them all until the following section.

For S5, by a *prefix* we simply mean a positive integer—think of it as informally designating a possible world. (For other modal logics besides S5 the structure of prefixes is more complex.) A *prefixed formula* is an expression of the form $n\,X$, where n is a prefix and X is a formula—think of it as saying X holds in world n. A *signed prefixed formula* is an expression of the form $T\ n\,X$ or $F\ n\,X$—the first asserts $n\,X$ and the second denies it.

A *tableau proof* of a formula X (without free variables) is a *closed tableau* for $F\ 1\,X$. A *tableau* for a signed prefixed formula is a tree, with that signed prefixed formula at the root, and constructed using the various *branch extension rules* to be given below. A branch of a tableau is *closed* if it contains an explicit contradiction: both $T\ k\,Z$ and $F\ k\,Z$, for some k and Z. If every branch is closed, the tableau itself is *closed*.

Intuitively, when we begin a tableau with $F\ 1\,X$ we are supposing there is some possible world, designated by 1, at which X fails to hold. A closed tableau represents an impossible situation. So the intuitive understanding of a tableau proof is that X cannot fail to hold at any world—X must be valid. Proper soundness and completeness proofs can be based on this intuition but there is not space here to present them. Now it remains to give the various branch extension rules—the rules for "growing" a tableau. The propositional connective rules are easily described, and we begin with them.

If $T\ n\,X \wedge Y$ occurs on a tableau branch, $X \wedge Y$ intuitively is true at world n, hence both X and Y must also be true there. Consequently the tableau branch can be extended twice, with the first node labeled $T\ n\,X$ and the second $T\ n\,Y$. If $F\ n\,X \wedge Y$ occurs on a branch, $X \wedge Y$ is false at n, so either X or Y must be false at n. This gives rise to two cases, and so the branch "splits." That is, the last node is given two children, the left labeled $F\ n\,X$ and the right $F\ n\,Y$.

Rules for other binary connectives are similar, while those for negation are simpler. In summary form, here they are.

Negation:

$$\frac{T\ n\,\neg X}{F\ n\,X} \qquad \frac{F\ n\,\neg X}{T\ n\,X}$$

Binary:

$$\frac{T\ n\,X \wedge Y}{\begin{array}{c} T\ n\,X \\ T\ n\,Y \end{array}} \qquad \frac{F\ n\,X \wedge Y}{F\ n\,X \mid F\ n\,Y} \qquad \frac{T\ n\,X \vee Y}{T\ n\,X \mid T\ n\,Y} \qquad \frac{F\ n\,X \vee Y}{\begin{array}{c} F\ n\,X \\ F\ n\,Y \end{array}}$$

$$\frac{T\ n\,X \supset Y}{F\ n\,X \mid T\ n\,Y} \qquad \frac{F\ n\,X \supset Y}{\begin{array}{c} T\ n\,X \\ F\ n\,Y \end{array}}$$

Modal rules are quantifier-like in nature. Here the prefixes play a significant role.

Necessity: In these, k is *any* prefix.

$$\frac{T\ n\,\Box X}{T\ k\,X} \qquad \frac{F\ n\,\Diamond X}{F\ k\,X}$$

Possibility: In these, k is a prefix *that is new to the branch.*

$$\frac{T\ n\,\Diamond X}{T\ k\,X} \qquad \frac{F\ n\,\Box X}{F\ k\,X}$$

For quantifiers we need some additional machinery. For each prefix n we introduce an infinite alphabet of *parameters* associated with n—typically we write p_n, q_n, etc., for parameters associated with n. Think of the parameters associated with n as (designating) the things that exist at world n. From now on we allow parameters to appear in proofs (though not in the formulas being proved). They follow the syntax rules of free variables, though they are never quantified. Now, here are the quantifier rules.

Universal: In these, p_n is *any* parameter associated with n.

$$\frac{T\ n\,(\forall x)\varphi(x)}{T\ n\,\varphi(p_n)} \qquad \frac{F\ n\,(\exists x)\varphi(x)}{F\ n\,\varphi(p_n)}$$

Existential: In these, p_n is a parameter associated with n that is *new* to the branch.

$$\frac{T\ n\,(\exists x)\varphi(x)}{T\ n\,\varphi(p_n)} \qquad \frac{F\ n\,(\forall x)\varphi(x)}{F\ n\,\varphi(p_n)}$$

The rules so far are fairly standard. To deal with non-rigid constant symbols (and this can be extended to include function symbols too), we again need to extend the machinery. If c is a constant symbol and n is a prefix, we allow c_n to occur in a proof (though again, not in the formula being proved). Think of c_n intuitively as the object that c designates at world n. We have allowed partial designation, that is, a constant symbol may not designate at every world. All this is incorporated rather easily into our rules for predicate abstracts, which we give in a moment.

First, however, a little more notation. For each term t we define $t@n$, which we can think of as what t designates at n. (This gets more complicated when function symbols are present.) For a prefix n:

1. For a parameter p_i, let $p_i@n = p_i$.
2. For a subscripted constant symbol c_i, let $c_i@n = c_i$.
3. For an unsubscripted constant symbol c, let $c@n = c_n$.

Now, here are the abstraction rules.

Positive Abstraction:

$$\frac{T\ n\,\langle\lambda x.\varphi(x)\rangle(t)}{T\ n\,\varphi(t@n)}$$

Negative Abstraction: If $t@n$ already occurs on the branch,

$$\frac{F\ n\ \langle\lambda x.\varphi(x)\rangle(t)}{F\ n\ \varphi(t@n)}$$

Finally we have the rules for equality, and these are quite straightforward. Let us say a term is *grounded on a branch* if it is a parameter or a subscripted constant symbol, and it already occurs on the branch.

Reflexivity: If t is grounded on the branch, $T\ n\ t = t$ can be added to the end, for any prefix n. Briefly,

$$\frac{}{T\ n\ t = t}$$

Substitutivity: If t and u are grounded on the branch, and $T\ k\ t = u$ occurs on the branch, any occurrences of t can be replaced with occurrences of u. Again briefly,

$$\frac{T\ k\ t = u \quad T\ n\ \varphi(t)}{T\ n\ \varphi(u)} \qquad \frac{T\ k\ t = u \quad F\ n\ \varphi(t)}{F\ n\ \varphi(u)}$$

This completes the full set of tableau rules.

10 More Examples

We give several simple examples of tableau proofs. More can be found in [2].

Example 6. In Section 6 we gave formula (6) as an interesting example of a valid formula involving designation. We now give a tableau proof of part of this,

$$\mathbf{D}(c) \supset [\neg\langle\lambda x.P(x)\rangle(c) \supset \langle\lambda x.\neg P(x)\rangle(c)]$$

Line numbers are for explanation purposes only. Since no modal operators are present, only world 1 is involved throughout.

$F\ 1\ \mathbf{D}(c) \supset [\neg\langle\lambda x.P(x)\rangle(c) \supset \langle\lambda x.\neg P(x)\rangle(c)]$ 1.
$T\ 1\ \mathbf{D}(c)$ 2.
$F\ 1\ \neg\langle\lambda x.P(x)\rangle(c) \supset \langle\lambda x.\neg P(x)\rangle(c)$ 3.
$T\ 1\ \neg\langle\lambda x.P(x)\rangle(c)$ 4.
$F\ 1\ \langle\lambda x.\neg P(x)\rangle(c)$ 5.
$F\ 1\ \langle\lambda x.P(x)\rangle(c)$ 6.
$T\ 1\ \langle\lambda x.x = x\rangle(c)$ $2'$.
$T\ 1\ c_1 = c_1$ 7.
$F\ 1\ \neg P(c_1)$ 8.
$F\ 1\ P(c_1)$ 9.
$T\ 1\ P(c_1)$ 10.

In this, 2 and 3 are from 1, and 4 and 5 are from 3 by an implication rule; 6 is from 4 by a negation rule; $2'$ is line 2 unabbreviated; 7 is from $2'$ by a positive abstraction rule; then 8 is from 5 and 9 is from 6 by negative abstraction; and 10 is from 8 by negation. The single branch is closed because of 9 and 10.

Example 7. We give a tableau proof of a formula discussed in Example 4 of Section 6. Again this does not involve modal operators.

$F\ 1\ \mathbf{E}(c) \supset [(\forall x)P(x) \supset \langle \lambda x.P(x)\rangle(c)]$ 1.
$T\ 1\ \mathbf{E}(c)$ 2.
$F\ 1\ (\forall x)P(x) \supset \langle \lambda x.P(x)\rangle(c)$ 3.
$T\ 1\ (\forall x)P(x)$ 4.
$F\ 1\ \langle \lambda x.P(x)\rangle(c)$ 5.
$T\ 1\ \langle \lambda x.(\exists y)(y = x)\rangle(c)$ 2′.
$T\ 1\ (\exists y)(y = c_1)$ 6.
$T\ 1\ p_1 = c_1$ 7.
$T\ 1\ P(p_1)$ 8.
$T\ 1\ P(c_1)$ 9.
$F\ 1\ P(c_1)$ 10.

In this, 2 and 3 are from 1 and 4 and 5 are from 3 by an implication rule; 2′ is 2 unabbreviated; 6 is from 2′ by positive abstraction; 7 is from 6 by an existential rule (p_1 is a new parameter at this point); 8 is from 4 by a universal rule (note that a parameter is involved, as the rule requires); 9 is from 7 and 8 by substitutivity of equality; 10 is from 5 by negative abstraction (note that c_1 already occurs on the tableau branch). Closure is by 9 and 10.

Example 8. Our final example is an interesting modal example. It is a proof of

$$\langle \lambda x.\Box\langle \lambda y.x = y\rangle(c)\rangle(c) \supset [\langle \lambda x.\Box P(x)\rangle(c) \supset \Box\langle \lambda x.P(x)\rangle(c)]$$

In [2] we make a case that the antecedent of this expresses rigidity of c. The consequent asserts that a *de re* usage of c implies the corresponding *de dicto* version. There is no space here to discuss just what this is all about. Just take it as providing an illustrative tableau proof.

$F\ 1\ \langle \lambda x.\Box\langle \lambda y.x = y\rangle(c)\rangle(c) \supset [\langle \lambda x.\Box P(x)\rangle(c) \supset \Box\langle \lambda x.P(x)\rangle(c)]$ 1.
$T\ 1\ \langle \lambda x.\Box\langle \lambda y.x = y\rangle(c)\rangle(c)$ 2.
$F\ 1\ \langle \lambda x.\Box P(x)\rangle(c) \supset \Box\langle \lambda x.P(x)\rangle(c)$ 3.
$T\ 1\ \langle \lambda x.\Box P(x)\rangle(c)$ 4.
$F\ 1\ \Box\langle \lambda x.P(x)\rangle(c)$ 5.
$T\ 1\ \Box P(c_1)$ 6.
$F\ 2\ \langle \lambda x.P(x)\rangle(c)$ 7.
$T\ 2\ P(c_1)$ 8.
$T\ 1\ \Box\langle \lambda y.c_1 = y\rangle(c)$ 9.
$T\ 2\ \langle \lambda y.c_1 = y\rangle(c)$ 10.
$T\ 2\ c_1 = c_2$ 11.
$F\ 2\ P(c_2)$ 12.
$T\ 2\ P(c_2)$ 13.

In this, 2 and 3 are from 1 and 4 and 5 are from 3 by an implication rule; 6 is from 4 by positive abstraction; 7 is from 5 by a possibility rule (the prefix 2 is new to the branch at this point); 8 is from 6 by a necessity rule; 9 is from 2 by positive

abstraction; 10 is from 9 by a necessity rule; 11 is from 10 by positive abstraction; 12 is from 7 by negative abstraction (c_2 occurs on the branch already); and 13 is from 8 and 11 by substitutivity of equality.

We leave it to you to provide a proof of

$$\langle\lambda x.\Box\langle\lambda y.x = y\rangle(c)\rangle(c) \supset [\Box\langle\lambda x.P(x)\rangle(c) \supset \langle\lambda x.\Box P(x)\rangle(c)]$$

11 Conclusions

The work above should begin to make it clear that first-order modal logic, with its syntax and semantics properly enhanced, is a very expressive formalism. It relates well to natural language constructs and to those of computer science. And it has a simple proof procedure which is automatable (though this has not been done). Keep it in mind.

References

1. M. C. Fitting. A modal Herbrand theorem. *Fundamenta Informaticae*, 28:101–122, 1996.
2. M. C. Fitting and R. Mendelsohn. *First-Order Modal Logic*. Kluwer, 1998. Forthcoming.
3. G. Frege. Uber Sinn und Bedeutung. *Zeitschrift fur Philosophie und philosophische Kritik*, 100:25–50, 1892. "On Sense and Reference" translated in [4].
4. G. Frege. *Translations from the Philosophical Writings of Gottlob Frege*. Basil Blackwell, Oxford, 1952. P. Geach and M. Black editors.
5. J. W. Garson. Quantification in modal logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2, pages 249–307. D. Reidel, 1984.
6. D. Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2, pages 497–604. D. Reidel, 1984.
7. B. Russell. On denoting. *Mind*, 14:479–493, 1905. Reprinted in Robert C. Marsh, ed., *Logic and Knowledge: Essays 1901-1950, by Bertrand Russell*, Allen & Unwin, London, 1956.
8. R. Stalnaker and R. Thomason. Abstraction in first-order modal logic. *Theoria*, 34:203–207, 1968.
9. R. Thomason and R. Stalnaker. Modality and reference. *Nous*, 2:359–372, 1968.