A Modal Herbrand Theorem*

Melvin Fitting
mlflc@cunyvm.cuny.edu
Dept. Mathematics and Computer Science
Lehman College (CUNY), Bronx, NY 10468
Depts. Computer Science, Philosophy, Mathematics
Graduate Center (CUNY), 33 West 42nd Street, NYC, NY 10036

August 2, 1995

Abstract

We state and prove a modal Herbrand theorem that is, we believe, a more natural analog of the classical version than has appeared before. The statement itself requires the enlargement of the usual machinery of first-order modal logic — we use the device of *predicate abstraction*, something that has been considered elsewhere as well. This expands the expressive power of modal logic in a natural way. Our proof of the modal version of Herbrand's theorem uses a tableau system that takes predicate abstraction into account. It is somewhat simpler than other systems for the same purpose that have previously appeared.

1 Introduction

In classical logic, Herbrand's famous theorem of 1930 plays many roles. Herbrand seems to have thought of it as something like a constructive completeness theorem [12, 13]. Robinson cited it as the foundation of automated theorem proving [15]. It has been applied to derive results on decidability [3]. But despite its fundamental nature, it has remained remarkably classical. Completeness results, with suitable generalizations of Tarskian semantics, have been extended to a rich variety of non-classical logics. The same is true of interpolation and compactness theorems, cut-free sequent calculi, ultraproduct constructions, and many other tools originally developed for classical logic. Such generalizations not only provide us with machinery for working with non-classical logics, they also help us understand the tools themselves in a deeper way. But Herbrand's theorem, by and large, has always remained confined to its original setting. To be sure, there have been attempts at broadening it [5, 14, 1], but these have been not entirely satisfactory for a variety of reasons. While it was constructive in nature, [5] was not really a modal analog of Herbrand's theorem, but rather of a related result of Smullyan, in which expansions have quite a different form. On the other hand, [1] was a true generalization of Herbrand's theorem to the modal setting, but rather than an expansion being a formula, it is, in effect, a set of formulas, and the notion has more the nature of a process than a static entity. The treatment of [14] is, in many ways, closest to ours, making use of tableaus and expanding the machinery of first-order modal logic, but our approach is almost orthogonal to that one. Where [14] modifies the structure of terms using a "bullet" operator, we modify the structure of formulas using predicate abstraction which we feel is a more natural modification.

In this paper we present what we think is a close modal version of Herbrand's theorem. We give it for the modal logic **K** without the Barcan formula. In order to do this the basic machinery of modal logic must

*I want to thank a referee for pointing out a serious error in an earlier version of this paper and for being generally uncompromising.

be enriched — in ways that are natural and useful for other purposes. *Non-rigid designators* are essential, and with these comes what we call *predicate abstraction*, something that has no role in classical logic, for reasons we will see below. This provides a conservative extension of first-order modal logic as it is ordinarily formulated, and gives us what we need to state and prove a Herbrand theorem that is a natural analog of the classical one, and which reduces to a classical version if no modal operators are present. The notion of predicate abstraction is an intrinsically interesting one, enriching the expressive power of first-order modal logic in useful ways. It has been explored in other publications and has a lengthy, if spotty, history.

2 What Is the Problem?

Often, classical Herbrand expansion is applied to formulas in prenex form, but such a normal form is not available in modal logic. No matter. As originally presented, and as formulated in [2], the Herbrand expansion process applies to arbitrary formulas, so this is no modal obstacle.

Classically Skolemization is involved, and here our troubles begin. Actually, there are two broad versions of Skolemization, depending on whether existential or universal quantifiers are removed. The first version preserves satisfiability, the second preserves validity. It is the second that will be of importance here. In this, for instance, the *validity functional form* of $(\forall x)Px$, where Px is atomic, is Pc, where c is a new constant symbol. But now, in the modal setting, what should be the validity functional form of $\Diamond(\forall x)Px$, and what should be the validity functional form of $(\forall x)\Diamond Px$? Both would seem to be $\Diamond Pc$, but it is not reasonable to have both formulas, with quite different meanings, Skolemize to the same thing.

Carrying the discussion a little further, when the validity functional form of $(\forall x)Px$ is created, the intuition is that if $(\forall x)Px$ fails in a model, c should designate something in the domain of the model for which Px is not true — thus if Pc is valid it follows that $(\forall x)Px$ must also be valid. Carrying this intuition over to the modal setting, consider $\Diamond(\forall x)Px$, which we tentatively Skolemize by $\Diamond Pc$. If $\Diamond(\forall x)Px$ is not **K**-valid, there must be a **K**-model, and a world p of it, at which $\Diamond(\forall x)Px$ fails. But then, in every world accessible from p, $(\forall x)Px$ must also fail, and so in each world accessible from p, Px must be false of some item in the domain of that world. But, there is no reason why that item should be the same in every world accessible from p. Consequently, if we are to have $\Diamond Pc$ be the Skolemization of $\Diamond(\forall x)Px$, c must be allowed to vary its designation from world to world — it must be a *non-rigid designator*.

Unfortunately, the need for non-rigid designators brings a new set of problems, because the act of designation and the act of passing to an alternate world do not commute. Consider again the formula $\Diamond Pc$, where c is non-rigid. What might it mean to say this is true at world p? One possibility: it could mean the formula Pc is true at some world q_1 that is alternate to p, which in turn means that the designation of c at q_1 is in the set assigned by the model as the meaning of P at q_1. But there is another possibility: it could mean the $\Diamond P$ "property" holds at p of the designation of c at p, which in turn means that for some world q_2 alternate to p, the designation of c at p is in the set assigned as the meaning of P at q_2. Even if q_1 and q_2 turn out to be the same, these two versions need not coincide, since in the first case c designates at q_1, after the move to an alternate world, and in the second case it designates at p, before the move. If c is non-rigid, these are not necessarily equivalent. Sometimes these two versions are referred to as "narrow scope" and "broad scope" and one or the other is disallowed. Unfortunately, we need to interpret $\Diamond Pc$ one way for it to serve as the Skolemization of $\Diamond(\forall x)Px$, and the other way for $(\forall x)\Diamond Px$ — neither can be disallowed.

The problems are not over. After Skolemization, Herbrand expansion itself can be discussed. Now the remaining quantifiers — all essentially existential — are replaced by disjunctions of instances. But these instances involve function and constant symbols, and since these can be non-rigid, we once again run into the same kind of difficulties we saw above.

3 Predicate Abstraction

In the lambda-calculus a distinction is made between a term, such as $x + 3$, and the function it determines, $\langle\lambda x.x+3\rangle$. We make a similar distinction here between a formula of first order logic, say φ, and the predicate abstracted from it, which we denote $\langle\lambda x.\varphi\rangle$. (We use angle brackets here to assist the eye.) The machinery of first-order modal logic will be expanded both syntactically and semantically to allow for predicate abstraction. Once done, the two ways of reading the formula $\Diamond Pc$, discussed in the previous section, are represented by two distinct formulas, $\Diamond\langle\lambda x.Px\rangle(c)$ for the narrow scope reading, and $\langle\lambda x.\Diamond Px\rangle(c)$ for the broad scope reading. We will see that this device solves our various difficulties, and does so in quite a natural way.

The device of predicate abstraction was introduced into modal logic in [16, 17] where an axiomatic formulation (involving equality) was presented. In [4, 6] it was used to give a proof system for modal logic in the style of Hilbert's classical epsilon-calculus (epsilon terms in modal logic are non-rigid). In [5] a modal analog of a theorem of Smullyan was proved, using predicate abstraction in an essential way. This theorem is Herbrand-like, but the expansion specifics are quite different. After this, interest in predicate abstraction seems to have disappeared for a time, reviving more recently in [10, 11], where prefixed tableau system formulations can be found. We will give yet another tableau version here.

Syntax We follow the presentation in [11]. We have an alphabet with infinitely many variables, constant symbols, function symbols and relation symbols, $\wedge$, $\vee$, $\neg$, $\supset$ as propositional connectives, $\Box$, $\Diamond$ as modal operators, and $\forall$, $\exists$ as quantifiers, along with parentheses and a comma as punctuation. In addition the symbol λ is present, as a predicate abstraction former. Terms are defined in the usual way, except that we write $ft_1 \ldots t_n$ instead of the more usual $f(t_1, \ldots, t_n)$ in order to minimize parentheses and make formula reading easier. (We do a similar thing with atomic formulas.)

We have a somewhat restricted notion of atomic formula: $Rx_1 \ldots x_n$ is an atomic formula provided R is an n-place relation symbol and $x_1, \ldots, x_n$ are *variables*. Then formulas are built up from atomic formulas, and free occurrences of variables are defined, all in the usual way, but with the following additional item.

- If φ is a formula, x is a variable, and t is a term, $\langle\lambda x.\varphi\rangle(t)$ is a formula. Its free variable occurrences are those of φ, except for occurrences of x, together with all variable occurrences in t.

The restriction on atomic formulas is simply for uniformity's sake — we could allow, say, Pt where P is a one-place relation symbol and t is a term, treating it semantically as if it were $\langle\lambda x.Px\rangle(t)$.

Example The following is a formula, assuming that P is a one-place relation symbol and t is a term:

$$(\forall y)\Diamond\langle\lambda x.Px\rangle(y) \supset \langle\lambda x.\Diamond Px\rangle(t).$$

The only free variable occurrences are those variable occurrences (if any) in t.

Semantics The version of Kripke model we use is essentially standard, except for the machinery to treat non-rigidity.

Definition 3.1 A *first-order frame* is a structure $\langle\mathcal{G}, \mathcal{R}, \mathcal{D}\rangle$, where $\mathcal{G}$ is a non-empty set (of possible worlds), $\mathcal{R}$ is a binary relation on $\mathcal{G}$ (of accessibility), and $\mathcal{D}$ is a domain function, from members of $\mathcal{G}$ to non-empty sets, meeting the *monotonicity condition*: $p\mathcal{R}q$ implies $\mathcal{D}(p) \subseteq \mathcal{D}(q)$.

An *interpretation* in a first-order frame $\langle\mathcal{G}, \mathcal{R}, \mathcal{D}\rangle$ is a mapping $\mathcal{I}$ that assigns:

1. to each constant symbol c and each $p \in \mathcal{G}$ some member $\mathcal{I}(p, c) \in \mathcal{D}(p)$;

2. to each n-place function symbol f and each $p \in \mathcal{G}$ some n-place function $\mathcal{I}(p, f) : (\mathcal{D}(p))^n \to \mathcal{D}(p)$;

3. to each n-place relation symbol R and each $p \in \mathcal{G}$ some n-place relation $\mathcal{I}(p, R) \subseteq (\mathcal{D}(p))^n$.

A first-order frame, together with an interpretation, $\langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$, is a *non-rigid model*.

In order to deal with truth in non-rigid models we need to assign values to free variables, just as in the classical case.

Definition 3.2 An *assignment* in a non-rigid model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$ is a mapping s that assigns to every variable x some member $s(x)$ of $\cup\{\mathcal{D}(p) \mid p \in \mathcal{G}\}$. By $s\left[\begin{smallmatrix} x \\ a \end{smallmatrix}\right]$ we mean the assignment that is like s on all variables except x, and that maps x to a.

Note that unlike interpretations, assignments do not depend on worlds. Also their values are not required to exist in the domains of all worlds. Next we extend the action of assignments, in models, to arbitrary terms at worlds.

Definition 3.3 Let $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$ be a non-rigid model, and s be an assignment in it. We define a function, also denoted by s, on worlds and terms, as follows. For $p \in \mathcal{G}$:

1. if x is a variable, $s(p, x) = s(x)$;

2. if c is a constant symbol, $s(p, c) = \mathcal{I}(p, c)$;

3. if f is an n-place function symbol and $t_1, \ldots, t_n$ are terms,

$$s(p, ft_1 \ldots t_n) = \mathcal{I}(p, f)(s(p, t_1), \ldots, s(p, t_n)).$$

Notice that meanings of terms are not always defined. For instance, if $s(x)$ is not in $\mathcal{D}(p)$, and f is a one-place function symbol, since $\mathcal{I}(p, f)$ is a mapping on $\mathcal{D}(p)$, $s(p, f(x))$ is undefined. It can be shown, however, that this situation never arises when evaluating the truth of closed formulas at worlds, and this is the only case we are interested in.

Now, finally, we characterize the fundamental notion, $\mathcal{M}, p \Vdash \varphi[s]$, intended to mean: formula φ is true at world p in model $\mathcal{M}$, under the assignment s of values to free variables.

Definition 3.4 Let $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$ be a non-rigid model and s be an assignment in it.

1. If R is an n-place relation symbol, $\mathcal{M}, p \Vdash Rx_1 \ldots x_n[s]$ iff $\langle s(p, x_1), \ldots, s(p, x_n) \rangle \in \mathcal{I}(p, R)$. (Recall that all atomic formulas are of this form, involving only variables and not more complex terms. Also note that $s(p, x_i) = s(x_i)$ may not be in the domain $\mathcal{D}(p)$ for some i, in which case $\mathcal{M}, p \Vdash Rx_1 \ldots x_n[s]$ is simply false.)

2. $\mathcal{M}, p \Vdash (\varphi \vee \psi)[s]$ iff $\mathcal{M}, p \Vdash \varphi[s]$ or $\mathcal{M}, p \Vdash \psi[s]$.

3. $\mathcal{M}, p \Vdash (\varphi \wedge \psi)[s]$ iff $\mathcal{M}, p \Vdash \varphi[s]$ and $\mathcal{M}, p \Vdash \psi[s]$.

4. $\mathcal{M}, p \Vdash \neg\varphi[s]$ iff not-$\mathcal{M}, p \Vdash \varphi[s]$

5. $\mathcal{M}, p \Vdash (\varphi \supset \psi)[s]$ iff not-$\mathcal{M}, p \Vdash \varphi[s]$ or $\mathcal{M}, p \Vdash \psi[s]$.

6. $\mathcal{M}, p \Vdash \Box\varphi[s]$ iff $\mathcal{M}, q \Vdash \varphi[s]$ for all $q \in \mathcal{G}$ such that $p\mathcal{R}q$.

7. $\mathcal{M}, p \Vdash \Diamond\varphi[s]$ iff $\mathcal{M}, q \Vdash \varphi[s]$ for some $q \in \mathcal{G}$ such that $p\mathcal{R}q$.

8. $\mathcal{M}, p \Vdash (\forall x)\varphi[s]$ iff $\mathcal{M}, p \Vdash \varphi[s\left[\begin{smallmatrix} x \\ a \end{smallmatrix}\right]]$ for all $a \in \mathcal{D}(p)$.

9. $\mathcal{M}, p \Vdash (\exists x)\varphi[s]$ iff $\mathcal{M}, p \Vdash \varphi[s\begin{bmatrix}x\\a\end{bmatrix}]$ for some $a \in \mathcal{D}(p)$.
10. $\mathcal{M}, p \Vdash \langle \lambda x.\varphi\rangle(t)[s]$ iff $\mathcal{M}, p \Vdash \varphi[s\begin{bmatrix}x\\a\end{bmatrix}]$ where $a = s(p, t)$.

The last item is the only one that is in any way non-standard. It says that for $\langle \lambda x.\varphi\rangle(t)$ to be true at a world, φ must be true when we think of x as standing for whatever t designates at that world. This is exactly what one might expect.

Example We leave it to you to verify that $\mathcal{M}, p \Vdash \Diamond\langle \lambda x.Px\rangle(c)[s]$ iff for some $q \in \mathcal{G}$ such that $p\mathcal{R}q$, $\mathcal{I}(q, c) \in \mathcal{I}(q, P)$. And likewise, $\mathcal{M}, p \Vdash \langle \lambda x.\Diamond Px\rangle(c)[s]$ iff for some $q \in \mathcal{G}$ such that $p\mathcal{R}q$, $\mathcal{I}(p, c) \in \mathcal{I}(q, P)$. Since c is non-rigid, $\mathcal{I}(p, c)$ and $\mathcal{I}(q, c)$ need not be the same.

It is easy to verify that for a closed formula φ the choice of assignment does not matter. For closed φ we write $\mathcal{M}, p \Vdash \varphi$ to mean $\mathcal{M}, p \Vdash \varphi[s]$ for some (all) s. Finally, we say a closed formula φ is *valid* in $\mathcal{M}$ if $\mathcal{M}, p \Vdash \varphi$ for all worlds p of the model $\mathcal{M}$, and φ is simply **K**-valid if it is valid in all models.

4 Basic Properties

Predicate abstraction plays no role in classical logic, and the reason is quite simple — it is because universal closures of the following are **K**-valid formulas, something we leave to you to verify.

1. Propositional Equivalences
   (a) $\langle \lambda x.(\varphi \wedge \psi)\rangle(t) \equiv (\langle \lambda x.\varphi\rangle(t) \wedge \langle \lambda x.\psi\rangle(t))$
   (b) $\langle \lambda x.(\varphi \vee \psi)\rangle(t) \equiv (\langle \lambda x.\varphi\rangle(t) \vee \langle \lambda x.\psi\rangle(t))$
   (c) $\langle \lambda x.(\varphi \supset \psi)\rangle(t) \equiv (\langle \lambda x.\varphi\rangle(t) \supset \langle \lambda x.\psi\rangle(t))$
   (d) $\langle \lambda x.\neg\varphi\rangle(t) \equiv \neg\langle \lambda x.\varphi\rangle(t)$
2. First-Order Equivalences. Assume x and y are different, and y does not occur in t.
   (a) $\langle \lambda x.(\forall y)\varphi\rangle(t) \equiv (\forall y)\langle \lambda x.\varphi\rangle(t)$
   (b) $\langle \lambda x.(\exists y)\varphi\rangle(t) \equiv (\exists y)\langle \lambda x.\varphi\rangle(t)$

Thus the classical connectives and quantifiers are "transparent" to predicate abstraction. This is not the case for the modal operators — we have seen that $\Diamond\langle \lambda x.Px\rangle(c)$ and $\langle \lambda x.\Diamond Px\rangle(c)$ need not be equivalent.

The machinery of predicate abstraction, while adding power, is a conservative extension. The following is immediate since if constant symbols, function symbols, and predicate abstraction do not play a role, the definition of model from section 3 agrees with the usual one.

Theorem 4.1 *If φ is a closed modal formula with no constant or function symbols, and no occurrences of predicate abstraction, then φ is* **K***-valid in the present sense if and only if φ is valid in all first-order* **K** *models, as usually defined.*

Actually this result can be strengthened. Suppose $\varphi(c_1, \ldots, c_n)$ is a closed modal formula, *defined in the usual way*, allowing the constant symbols $c_1, \ldots, c_n$ to occur within atomic formulas. Let $\varphi(x_1, \ldots, x_n)$ be the result of substituting new variables for occurrences of $c_1, \ldots, c_n$, which we assume are the only constant symbols occurring (we also assume there are no function symbols). Then $\varphi(c_1, \ldots, c_n)$ is valid in all first-order **K** models, in the customary sense, if and only if $\langle \lambda x_1. \cdots \langle \lambda x_n.\varphi(x_1, \ldots, x_n)\rangle(c_n) \cdots\rangle(c_1)$ is **K**-valid in the sense used here. Thus, in effect, "top level" predicate abstractions give the effect of rigid designation.

Finally, the replacement theorem carries over in a direct way.

Theorem 4.2 *Let φ, X and Y be formulas, and let φ' be like φ except that an occurrence of X as a subformula has been replaced with an occurrence of Y. If the universal closure of $X \equiv Y$ is* **K***-valid, so is the universal closure of $\varphi \equiv \varphi'$.*

5 Skolemization

The usual method of Skolemization applies in the modal setting, provided we use non-rigid function symbols and predicate abstraction. And as in the classical setting, there are two versions, one preserving satisfiability, one preserving validity. In stating the following, we assume the notion of positive and negative subformula is understood. We call a quantified subformula of a formula φ *essentially universal* if it is a positive subformula of the form $(\forall x)\psi$, or if it is a negative subformula of the form $(\exists x)\psi$. The notion of *essentially existential* is defined dually. Now, the following is more or less from [10].

Theorem 5.1 *Suppose φ is a closed modal formula, and $(Qy)\psi$ is a quantified subformula of it that occurs within the scope of quantifiers whose variables are $x_1, \ldots, x_n$. Let φ^* be the result of replacing $(Qy)\psi$ in φ with $\langle\lambda y.\psi\rangle(f x_1 \ldots x_n)$, where f is an n-place function symbol not occurring in φ.*

1. *(Satisfiability Version) If $(Qy)\psi$ is essentially existential in φ, and all the quantifiers within whose scope it lies are essentially universal, then φ is satisfiable in some* **K** *model if and only if φ^* is satisfiable in some* **K** *model.*

2. *(Validity Version) If $(Qy)\psi$ is essentially universal in φ, and all the quantifiers within whose scope it lies are essentially existential, then φ is* **K***-valid if and only if φ^* is* **K***-valid.*

The proof of this is a straightforward extension of the classical one. Now, by repeatedly applying part 2 of the theorem above, all essentially universal quantifiers can be eliminated from a given formula φ. We call the result a *validity functional form* of φ.

6 A Herbrand Theorem

When constructing a Herbrand expansion in classical logic we first Skolemize, then having specified a non-empty domain D of terms, the Herbrand expansion essentially results by replacing each (positively occurring) existential quantifier by a disjunction of instances over D, and each (negatively occurring) universal quantifier by a conjunction of instances over D. This can be turned into a recursive definition, based on the complexity of the formula in question. We now give such a definition as a lead-in to the modal version that follows. We have made one modification to the conventional notion: we allow quantifiers to be replaced by conjunctions or disjunctions of various lengths, which means that even with respect to a single domain, Herbrand expansions are not unique. Our version counts, as Herbrand expansions, anything that is usually counted as such, but also allows expansions that may have shorter proofs since "irrelevant" subformulas need not be present.

Since Herbrand expansions have lost their uniqueness, instead of a functional definition, we give a relational one. The notation we use is: $X \rightarrow X'$, which is intended to be read, "X' is a classical Herbrand expansion (over D) of X." This is determined by the calculus below. (We include rules for $\supset$, and omit those for $\wedge$ and $\vee$ which are similar.)

Literal If A is atomic, $A \rightarrow A$ and $\neg A \rightarrow \neg A$.

Propositional

$$\frac{X \to X'}{\neg\neg X \to \neg\neg X'}\ Neg$$

$$\frac{\neg X \to \neg X' \quad Y \to Y'}{X \supset Y \to X' \supset Y'}\ +Imp \qquad \frac{X \to X' \quad \neg Y \to \neg Y'}{\neg(X \supset Y) \to \neg(X' \supset Y')}\ -Imp$$

Quantification For closed terms $t_1, \ldots, t_n$ in D,

$$\frac{\varphi(x) \to \varphi'(x)}{(\exists x)\varphi(x) \to \varphi'(t_1) \vee \ldots \vee \varphi'(t_n)}\ +Quant \qquad \frac{\neg\varphi(x) \to \neg\varphi'(x)}{\neg(\forall x)\varphi(x) \to \neg[\varphi'(t_1) \wedge \ldots \wedge \varphi'(t_n)]}\ -Quant$$

Not surprisingly, the statement of a modal Herbrand theorem is more complex than in the classical case. This complexity arises because the act of substituting a term for a free variable is no longer as simple. Consider, for example, the formula $\Box Px$, where Px is atomic. If we want to "substitute" the closed term fc for x in a modal context, we must first introduce the machinery of predicate abstraction, and for this formula there are essentially different ways of doing so. Either $\langle\lambda x.\Box Px\rangle(fc)$ or $\Box\langle\lambda x.Px\rangle(fc)$ will serve as ways of binding x to fc. We could even have a "two-level" abstraction process, leading to $\langle\lambda y.\Box\langle\lambda x.Px\rangle(fy)\rangle(c)$. A layer of complexity is thus added because of the introduction of predicate abstraction.

In the calculus that follows, the propositional rules are the same as in the classical version, and straightforward modal and abstraction rules have been added. The essential change is that the quantifier rules have been replaced by more complex ones (introducing variables, rather than closed terms), together with rules for binding variables to terms. Since variables can be present, we do not use the terminology "Herbrand expansion," reserving this for cases where formulas are closed.

Definition 6.1 We say X' is a *modal Herbrand transform* of the formula X if $X \to X'$ is derivable in the calculus that follows.

Literal If A is atomic, $A \to A$ and $\neg A \to \neg A$.

Propositional

$$\frac{X \to X'}{\neg\neg X \to \neg\neg X'}\ Neg$$

$$\frac{\neg X \to \neg X' \quad Y \to Y'}{X \supset Y \to X' \supset Y'}\ +Imp \qquad \frac{X \to X' \quad \neg Y \to \neg Y'}{\neg(X \supset Y) \to \neg(X' \supset Y')}\ -Imp$$

Modal

$$\frac{X \to X'}{\Box X \to \Box X'}\ +Nec \qquad \frac{\neg X \to \neg X'}{\neg\Box X \to \neg\Box X'}\ -Nec$$

Abstraction

$$\frac{X \to X'}{\langle\lambda x.X\rangle(t) \to \langle\lambda x.X'\rangle(t)}\ +Lambda \qquad \frac{\neg X \to \neg X'}{\neg\langle\lambda x.X\rangle(t) \to \neg\langle\lambda x.X'\rangle(t)}\ -Lambda$$

Quantification For new variables $x_1, \ldots, x_n$,

$$\frac{\varphi(x) \to \varphi_1(x) \quad \ldots \quad \varphi(x) \to \varphi_n(x)}{(\exists x)\varphi(x) \to \varphi_1(x_1) \vee \ldots \vee \varphi_n(x_n)}\ +Quant$$

$$\frac{\neg\varphi(x) \to \neg\varphi_1(x) \quad \ldots \quad \neg\varphi(x) \to \neg\varphi_n(x)}{\neg(\forall x)\varphi(x) \to \neg[\varphi_1(x_1) \wedge \ldots \wedge \varphi_n(x_n)]} -Quant$$

Binding For x not free in X,

$$\frac{X \to X'}{X \to \langle\lambda x.X'\rangle(t)} +Bind \qquad \frac{\neg X \to \neg X'}{\neg X \to \neg\langle\lambda x.X'\rangle(t)} -Bind$$

Note that in applying the rules, if we show, say, that $\varphi(x) \to \varphi'(x)$, and y is a new variable, we can also show $\varphi(y) \to \varphi'(y)$. This observation is sometimes useful.

Example Here is a derivation in the calculus above, beginning with a Literal axiom.

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\neg Py \to \neg Py}{\neg(\forall y)Py \to \neg(Py_1 \wedge Py_2)} -Quant}{\neg(\forall y)Py \to \neg\langle\lambda y_2.Py_1 \wedge Py_2\rangle(fz)} -Bind}{\Box\neg(\forall y)Py \to \Box\neg\langle\lambda y_2.Py_1 \wedge Py_2\rangle(fz)} +Nec}{\Box\neg(\forall y)Py \to \langle\lambda y_1.\Box\neg\langle\lambda y_2.Py_1 \wedge Py_2\rangle(fz)\rangle(c)} +Bind}{\Box\neg(\forall y)Py \to \langle\lambda z.\langle\lambda y_1.\Box\neg\langle\lambda y_2.Py_1 \wedge Py_2\rangle(fz)\rangle(c)\rangle(d)} +Bind$$

Note that while two variables were introduced at the step eliminating the universal quantifier, a third variable was introduced by one of the binding steps. This cannot be ruled out if completeness is to be achieved, and is a source of the complexity of first-order modal logic.

Definition 6.2 Let φ be a closed modal formula, and let φ^* be a validity functional form for φ. If φ^{**} is a modal Herbrand transform of φ^* and is a closed formula, we say φ^{**} is a *modal Herbrand expansion* of φ.

If X has only essentially existential quantifiers, and X' is a modal Herbrand transform of X, X' must be quantifier free. It follows that any modal Herbrand expansion of a closed formula must be quantifier free.

Theorem 6.3 (A Modal Herbrand Theorem) *Let φ be a closed modal formula. φ is* **K***-valid if and only if some modal Herbrand expansion of φ is* **K***-valid.*

The classical Herbrand theorem reduces validity from a first-order problem to an infinite set of propositional problems. So does the theorem above, in the sense that modal Herbrand expansions contain no quantifiers. On the other hand, the **K**-semantics still involves the machinery of domain functions, since non-rigid designators must designate something. The following says that, nonetheless, things are essentially propositional in nature.

Theorem 6.4 *There is a decision procedure for the* **K***-validity of closed formulas that are quantifier free.*

Proofs of these two theorems will be found in the next several sections.

Example Consider the (**K**-valid) formula

$$\Box(\forall y)[Py \supset Qy] \supset (\exists z)[(\exists x)\Diamond Px \supset \Diamond Qz].$$

In this, $(\exists x)$ is essentially universal, while both the other quantifiers are essentially existential. Consequently a validity functional form for it is the following, where f is a function symbol.

$$\Box(\forall y)[Py \supset Qy] \supset (\exists z)[\langle\lambda x.\Diamond Px\rangle(fz) \supset \Diamond Qz].$$

Now, another derivation using the calculus above,

$$\dfrac{\dfrac{\dfrac{\dfrac{Py \to Py \quad \neg Qy \to \neg Qy}{\neg[Py \supset Qy] \to \neg[Py \supset Qy]}\ -Imp}{\neg(\forall y)[Py \supset Qy] \to \neg[Py_1 \supset Qy_1]}\ -Quant}{\neg\Box(\forall y)[Py \supset Qy] \to \neg\Box[Py_1 \supset Qy_1]}\ -Nec}{\neg\Box(\forall y)[Py \supset Qy] \to \neg\langle\lambda y_1.\Box[Py_1 \supset Qy_1]\rangle(fc)}\ -Bind$$

In a straightforward way we can derive $[\langle\lambda x.\Diamond Px\rangle(fz) \supset \Diamond Qz] \to [\langle\lambda x.\Diamond Px\rangle(fz) \supset \Diamond Qz]$ — we omit the steps (the rules for $\Diamond$ are analogous to those for $\Box$). Then we can proceed as follows.

$$\dfrac{[\langle\lambda x.\Diamond Px\rangle(fz) \supset \Diamond Qz] \to [\langle\lambda x.\Diamond Px\rangle(fz) \supset \Diamond Qz]}{\begin{array}{l}(\exists z)[\langle\lambda x.\Diamond Px\rangle(fz) \supset \Diamond Qz] \to \\ \quad [\langle\lambda x.\Diamond Px\rangle(fz_1) \supset \Diamond Qz_1] \vee [\langle\lambda x.\Diamond Px\rangle(fz_2) \supset \Diamond Qz_2]\end{array}}\ +Quant$$

$$\dfrac{\begin{array}{l}(\exists z)[\langle\lambda x.\Diamond Px\rangle(fz) \supset \Diamond Qz] \to \\ \quad [\langle\lambda x.\Diamond Px\rangle(fz_1) \supset \Diamond Qz_1] \vee [\langle\lambda x.\Diamond Px\rangle(fz_2) \supset \Diamond Qz_2]\end{array}}{\begin{array}{l}(\exists z)[\langle\lambda x.\Diamond Px\rangle(fz) \supset \Diamond Qz] \to \\ \quad \langle\lambda z_1.[\langle\lambda x.\Diamond Px\rangle(fz_1) \supset \Diamond Qz_1] \vee [\langle\lambda x.\Diamond Px\rangle(fz_2) \supset \Diamond Qz_2]\rangle(c)\end{array}}\ +Bind$$

$$\dfrac{\begin{array}{l}(\exists z)[\langle\lambda x.\Diamond Px\rangle(fz) \supset \Diamond Qz] \to \\ \quad \langle\lambda z_1.[\langle\lambda x.\Diamond Px\rangle(fz_1) \supset \Diamond Qz_1] \vee [\langle\lambda x.\Diamond Px\rangle(fz_2) \supset \Diamond Qz_2]\rangle(c)\end{array}}{\begin{array}{l}(\exists z)[\langle\lambda x.\Diamond Px\rangle(fz) \supset \Diamond Qz] \to \\ \quad \langle\lambda z_2.\langle\lambda z_1.[\langle\lambda x.\Diamond Px\rangle(fz_1) \supset \Diamond Qz_1] \vee [\langle\lambda x.\Diamond Px\rangle(fz_2) \supset \Diamond Qz_2]\rangle(c)\rangle(fc)\end{array}}\ +Bind$$

Now, combining the items above using + *Imp*,

$$\begin{array}{l}\Box(\forall y)[Py \supset Qy] \supset (\exists z)[\langle\lambda x.\Diamond Px\rangle(fz) \supset \Diamond Qz] \\ \quad \to \\ \quad \langle\lambda y_1.\Box[Py_1 \supset Qy_1]\rangle(fc) \supset \\ \qquad \langle\lambda z_2.\langle\lambda z_1.[\langle\lambda x.\Diamond Px\rangle(fz_1) \supset \Diamond Qz_1] \vee [\langle\lambda x.\Diamond Px\rangle(fz_2) \supset \Diamond Qz_2]\rangle(c)\rangle(fc)\end{array}$$

So

$$\langle\lambda y_1.\Box[Py_1 \supset Qy_1]\rangle(fc) \supset \langle\lambda z_2.\langle\lambda z_1.[\langle\lambda x.\Diamond Px\rangle(fz_1) \supset \Diamond Qz_1] \vee [\langle\lambda x.\Diamond Px\rangle(fz_2) \supset \Diamond Qz_2]\rangle(c)\rangle(fc)$$

is a Herbrand expansion of $\Box(\forall y)[Py \supset Qy] \supset (\exists z)[(\exists x)\Diamond Px \supset \Diamond Qz]$ and is, in fact, valid.

7 Soundness

In this section we show the easy half of Theorem 6.3.

Proposition 7.1 *Let φ be a closed modal formula. If there is a* **K***-valid modal Herbrand expansion of φ then φ itself is* **K***-valid.*

Since a closed formula φ is **K**-valid if and only if its validity functional form φ^* is **K**-valid, the Proposition above is an immediate consequence of the following.

Proposition 7.2 *Let A be a closed modal formula with all its quantifiers essentially existential, and let B be a modal Herbrand expansion of A. Then $B \supset A$ is* **K***-valid.*

Proof In order to show this we must prove something more general. Suppose X is a modal formula with all its quantifiers essentially existential (not necessarily closed) and suppose $X \to X'$ is derivable in the calculus of section 6. Then the universal closure of $X' \supset X$ is **K**-valid. This follows by induction on the length of the derivation of $X \to X'$. Among more obvious things, the induction uses the fact that universal closures of the following formulas are **K**-valid.

1. $(X \supset Y) \equiv (\neg Y \supset \neg X)$
2. $(\psi(x_1) \vee \cdots \vee \psi(x_n)) \supset (\exists x)\psi(x)$
3. $(\forall x)\psi(x) \supset (\psi(x_1) \wedge \cdots \wedge \psi(x_n))$
4. $[(X \supset X') \wedge (Y' \supset Y)] \supset [(X' \supset Y') \supset (X \supset Y)]$

It also uses the fact that **K**-validity of the universal closure is preserved under the following transformations.

6. $X \supset Y \Rightarrow \Box X \supset \Box Y$
7. $X \supset Y \Rightarrow \langle\lambda x.X\rangle(t) \supset \langle\lambda x.Y\rangle(t)$
8. $X \supset Y \Rightarrow X \supset \langle\lambda z.Y\rangle(t)$ provided z does not occur in X.
9. $X \supset Y \Rightarrow \langle\lambda z.X\rangle(t) \supset Y$ provided z does not occur in Y.

We leave details to you. ■

8 A Tableau System

We will show a **K**-valid closed formula in validity functional form has a **K**-valid modal Herbrand expansion by extracting one from a cut-free tableau proof. A tableau system admitting predicate abstraction, using *prefixed* formulas, was given in [10, 11], but this is not best suited to our present purposes. Instead we give a new tableau formulation, extending the *destructive* tableau systems of [7, 8]. The formulation we present is designed for machine implementation, and is not particularly convenient for people. We sketch the system, followed by a proof of completeness and soundness, then use it to finish our proof of a modal Herbrand theorem.

We find it convenient to use *signed* formulas: if X is a formula, $T\ X$ and $F\ X$ are signed formulas. It is signed formulas that appear in our proofs. Intuitively, $T\ X$ says X is true at some world, and $F\ X$ says X is false at some world.

Customarily tableaus are presented as trees, but this is not particularly useful for present purposes. Following [8] we take a tableau to be a set (or list) of its branches, and a branch to be a set (or list) of signed formulas. We will use traditional terminology and refer to a signed formula as being *on* a branch, or a branch as being *of* a tableau, when what is really meant is that it is in it.

Classically, tableaus are refutation systems. To prove a closed formula φ, we begin a tableau consisting of a single branch, that containing only $F\ \varphi$. Then we apply *branch extension rules*, to "grow" branches. A branch is called *closed* if it contains a direct contradiction: $T\ X$ and $F\ X$ for some formula X. If each branch is closed, the tableau itself is said to be *closed*. A closed tableau beginning with $F\ \varphi$ constitutes a proof of φ — intuitively it shows the assumption that φ could be false leads to a contradiction.

In the kind of tableau we are presenting there are two broad classes of branch extension rules — *destructive* and *non-destructive*. Non-destructive rules make small changes to branches; destructive rules replace branches by entirely new ones. Modal operators require destructive rules — their application corresponds semantically to moving from one world to another in a model. Non-destructive rules are appropriate for the classical connectives and quantifiers. (See [9] for a more extended treatment of the non-destructive rules.)

In order to treat predicate abstraction we add several new pieces of machinery to that which is customary with tableaus. First, associated with each branch of a tableau will be a non-negative integer, called a *level number*. Tableau proofs will begin with a level number of 0. Syntactically, the level number of a branch represents how many times a modal branch extension rule has been applied on that branch. Semantically, if we think of a branch as a partial description of a possible world, its level number represents the distance between the world being described and a world at which the original root formula is satisfied.

Next, signed formulas containing free variables will be allowed in tableaus, and *substitutions* play a role. Associated with each tableau will be a substitution, giving values for some (not necessarily all) of the free variables present. We use σ, σ_1, σ_2, etc. to denote substitutions. For a term t, by $t\sigma$ we mean the result of applying the substitution σ to t — similarly for formulas. Composition of substitutions is denoted by concatenation: $\sigma_1\sigma_2$. The substitution replacing x_1 by t_1, ..., x_n by t_n is denoted $\{x_1/t_1, \ldots, x_n/t_n\}$. All this is standard, and we assume it is understood. There is one item we use that is somewhat non-standard, though. In proving tableau completeness we will make use of substitutions that have infinite domains. We do not restrict substitutions to have finite support, as is often done. Of course, the substitutions that arise in the construction of tableau proofs will have finite support.

A basic issue is: what kind of objects do substitutions assign to free variables. For this purpose we enlarge the language considerably. Proofs will be *of* closed formulas from the basic language, but will *use* formulas and terms from the enlarged language.

First we expand the alphabet: for each variable, constant, and function symbol we add infinitely many copies, one for each level number. Syntactically, if f is a function symbol, its level n counterpart will be denoted f^n, and similarly for constant symbols and variables. Intuitively, think of f^n as the function that the function symbol f denotes at a particular possible world. Function and constant symbols with levels will not appear in formulas, but are used by substitutions. Variables with levels can appear in formulas in tableaus.

Definition 8.1 *Terms of level n* are characterized as follows.

1. A variable or a constant symbol of level k is a term of level n for all $n \geq k$.

2. If f^k is a function symbol of level k, and $t_1, \ldots, t_m$ are terms of level k, then $f^k t_1, \ldots, t_m$ is a term of level n for all $n \geq k$.

A term of level n for some n will be called an *object expression*.

For example, $f^3 c^2 x^3$ is a term of level 3, and also of levels 4, 5, ... Any term of level n is also a term of level m for all $m \geq n$. This is how the monotonicity condition on model domains comes in.

Definition 8.2 By a *level substitution* we mean a mapping σ that assigns to each variable of level n in its domain some term of level n.

It is easy to verify that the composition of level substitutions is another level substitution. Formulas appearing in tableaus may contain free variables — these will all be variables with level numbers. Constant and function symbols with level numbers will not occur in formulas, but they will appear in level substitution ranges.

Definition 8.3 If t is a term with function and constant symbols from the original language (i.e. without level numbers), and n is a non-negative integer, by $t@n$ we mean the result of replacing each function and constant symbol of t by its level n counterpart.

Now to define the modal tableau machinery.

Definition 8.4

1. A *tableau* is a pair $\langle \sigma, \mathcal{T} \rangle$ where σ is a level substitution (the level substitution of the tableau) and $\mathcal{T}$ is a finite set of branches.

2. A *branch* is a pair $\langle n, \mathcal{B}\rangle$ where n is a non-negative integer (the level number of the branch) and $\mathcal{B}$ is a finite set of signed formulas.

Next we present the branch extension rules, and for this purpose we make use of the more-or-less standard grouping of signed formulas into general types. We begin with the propositional connectives.

α	α_1	α_2
$T\ X \wedge Y$	$T\ X$	$T\ Y$
$F\ X \vee Y$	$F\ X$	$F\ Y$
$F\ X \supset Y$	$T\ X$	$F\ Y$

β	β_1	β_2
$T\ X \vee Y$	$T\ X$	$T\ Y$
$F\ X \wedge Y$	$F\ X$	$F\ Y$
$T\ X \supset Y$	$F\ X$	$T\ Y$

Now the propositional rules, which are non-destructive, are the following, familiar from other contexts. Suppose $\langle \sigma, \mathcal{T}\rangle$ is a tableau and $\langle n, \mathcal{B}\rangle$ is a branch of it. We say a tableau $\langle \sigma, \mathcal{T}'\rangle$ is a *propositional successor* of $\langle \sigma, \mathcal{T}\rangle$ if

1. $\alpha \in \mathcal{B}$, $\mathcal{B}'$ is $\mathcal{B}$ with α removed and α_1, α_2 added, and $\mathcal{T}'$ is $\mathcal{T}$ with $\langle n, \mathcal{B}\rangle$ removed and $\langle n, \mathcal{B}'\rangle$ added.

2. $T\ \neg X \in \mathcal{B}$, $\mathcal{B}'$ is $\mathcal{B}$ with $T\ \neg X$ removed and $F\ X$ added, and $\mathcal{T}'$ is $\mathcal{T}$ with $\langle n, \mathcal{B}\rangle$ removed and $\langle n, \mathcal{B}'\rangle$ added.

3. $F\ \neg X \in \mathcal{B}$, $\mathcal{B}'$ is $\mathcal{B}$ with $F\ \neg X$ removed and $T\ X$ added, and $\mathcal{T}'$ is $\mathcal{T}$ with $\langle n, \mathcal{B}\rangle$ removed and $\langle n, \mathcal{B}'\rangle$ added.

4. $\beta \in \mathcal{B}$, $\mathcal{B}_1$ is $\mathcal{B}$ with β removed and β_1 added, $\mathcal{B}_2$ is $\mathcal{B}$ with β removed and β_2 added, and $\mathcal{T}'$ is $\mathcal{T}$ with $\langle n, \mathcal{B}\rangle$ removed and both $\langle n, \mathcal{B}_1\rangle$ and $\langle n, \mathcal{B}_2\rangle$ added.

These rules can be given schematically in the following more familiar form. Notice that they do not change either tableau level substitutions or branch level numbers.

$$\frac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}} \qquad \frac{\beta}{\beta_1 \mid \beta_2} \qquad \frac{T\ \neg X}{F\ X} \qquad \frac{F\ \neg X}{T\ X}$$

We will only be proving formulas that have been put into validity functional form, and hence that only contain essentially existential quantifiers. Since a tableau begins with F applied to the formula being proved, it follows that in our tableaus all quantifiers behave like universal ones. Consequently we only have one category of quantifiers.

γ	$\gamma(z)$
$T\ (\forall x)\varphi(x)$	$T\ \varphi(z)$
$F\ (\exists x)\varphi(x)$	$F\ \varphi(z)$

In applications of this z must be an variable with a level number, and $\varphi(z)$ denotes the result of substituting occurrences of z for all free occurrences of x in $\varphi(x)$. Now the quantifier rule, which is again non-destructive, is this.

Suppose $\langle \sigma, \mathcal{T}\rangle$ is a tableau and $\langle n, \mathcal{B}\rangle$ is a branch of it. We say a tableau $\langle \sigma, \mathcal{T}'\rangle$ is a *quantificational successor* of $\langle \sigma, \mathcal{T}\rangle$ if $\gamma \in \mathcal{B}$, $\mathcal{B}'$ is $\mathcal{B}$ with γ removed and $\gamma(x_1^n), \ldots, \gamma(x_k^n)$ added, where $x_1^n, \ldots, x_k^n$ are k variables of level n *that are new to the tableau*, and $\mathcal{T}'$ is $\mathcal{T}$ with $\langle n, \mathcal{B}\rangle$ removed and $\langle n, \mathcal{B}'\rangle$ added.

This rule is non-deterministic in its choice of k. Again neither level substitutions nor level numbers change. The rule can be given schematically as follows.

$$\frac{\gamma}{\begin{array}{c}\gamma(x_1^n)\\ \vdots \\ \gamma(x_k^n)\end{array}}$$

Where n is the branch level number, and the variables are new to the tableau.

For the modal rule we again use uniform notation.

ν	ν_0
$T\,\Box X$	$T\,X$
$F\,\Diamond X$	$F\,X$

π	π_0
$T\,\Diamond X$	$T\,X$
$F\,\Box X$	$F\,X$

We also use a "sharp" operator appropriate for **K**. For a set S of signed formulas, $S^{\#} = \{\nu_0 \mid \nu \in S\}$.

The modal rule is *destructive*. Branches get considerably modified, and lose information. For this rule branch level numbers change.

Suppose $\langle\sigma, \mathcal{T}\rangle$ is a tableau and $\langle n, \mathcal{B}\rangle$ is a branch of it. We say a tableau $\langle\sigma, \mathcal{T}'\rangle$ is a *modal successor* of $\langle\sigma, \mathcal{T}\rangle$ if $\pi \in \mathcal{B}$, $\mathcal{B}'$ is $\mathcal{B}^{\#} \cup \{\pi_0\}$, and $\mathcal{T}'$ is $\mathcal{T}$ with $\langle n, \mathcal{B}\rangle$ removed and $\langle n+1, \mathcal{B}'\rangle$ added.

Again the rule can be given schematically as follows.

$$\frac{S, \pi}{S^{\#}, \pi_0}$$

If the level number of a branch is n, and this rule is applied to the branch, the level number changes to $n+1$.

Finally, the rules for predicate abstraction. These change the tableau level substitution.

Suppose $\langle\sigma, \mathcal{T}\rangle$ is a tableau and $\langle n, \mathcal{B}\rangle$ is a branch of it. We say a tableau $\langle\sigma', \mathcal{T}'\rangle$ is an *abstraction successor* of $\langle\sigma, \mathcal{T}\rangle$ if

1. $T\,\langle\lambda x.\varphi(x)\rangle(t) \in \mathcal{B}$, $\mathcal{B}'$ is $\mathcal{B}$ with $T\,\langle\lambda x.\varphi(x)\rangle(t)$ removed and $T\,\varphi(z^n)$ added, where z^n is a variable of level n that is new to the tableau, $\sigma' = \{z^n/t@n\}\sigma$, and $\mathcal{T}'$ is $\mathcal{T}$ with $\langle n, \mathcal{B}\rangle$ removed and $\langle n, \mathcal{B}'\rangle$ added.

2. $F\,\langle\lambda x.\varphi(x)\rangle(t) \in \mathcal{B}$, $\mathcal{B}'$ is $\mathcal{B}$ with $F\,\langle\lambda x.\varphi(x)\rangle(t)$ removed and $F\,\varphi(z^n)$ added, where z^n is a variable of level n that is new to the tableau, $\sigma' = \{z^n/t@n\}\sigma$, and $\mathcal{T}'$ is $\mathcal{T}$ with $\langle n, \mathcal{B}\rangle$ removed and $\langle n, \mathcal{B}'\rangle$ added.

These too can be given schematically.

$$\frac{T\,\langle\lambda x.\varphi(x)\rangle(t)}{T\,\varphi(z^n) \quad \{z^n/t@n\}} \qquad\qquad \frac{F\,\langle\lambda x.\varphi(x)\rangle(t)}{F\,\varphi(z^n) \quad \{z^n/t@n\}}$$

Where z^n is new to the tableau.

Definition 8.5 Let S be a finite set of signed formulas, σ a level substitution, and n a non-negative integer. Then $\langle\sigma, \{\langle n, S\rangle\}\rangle$ is a tableau. By a *tableau for S and σ at level n* we mean any tableau that results from this by the application of 0 or more of the various successor rules: propositional, quantificational, modal, or abstraction.

Definition 8.6 Let $\langle\sigma, \mathcal{T}\rangle$ be a tableau, and $\langle n, \mathcal{B}\rangle$ be a branch of it. We say a level substitution τ *closes* the branch provided there are atomic formulas A and B, with $T\ A$ and $F\ B$ both on $\mathcal{B}$, and τ unifies $A\sigma$ and $B\sigma$.

We say the level substitution τ closes the tableau if it closes each branch.

We say a tableau is *closed* if some level substitution closes it.

Note: if a tableau is closed, a level substitution that closes it can be found by a simple modification of the unification algorithm, applied to A and B. Instead of starting the algorithm with the empty substitution, we begin with σ, the substitution associated with the tableau. And at further stages of the algorithm, in addition to verifying that a binding does not violate the occurs check, it must be verified that it does not bind to a variable of level n a term of a higher level.

Definition 8.7 A *proof* of a closed formula X is a closed tableau for $\{F\ X\}$ and the null substitution at level 0.

Example We give a tableau proof of the following:

$$(\forall y)\Diamond(\forall z)\langle\lambda x.Rxz\rangle(y) \supset \langle\lambda x.\Diamond\langle\lambda z.Rxz\rangle(fx)\rangle(c).$$

The tableaus constructed only have single branches, so to keep notation simple we display the set of signed formulas on it, and give the level number and the tableau level substitution separately. Initially, of course, the substitution is null, {}, the level number is 0, and the set of signed formulas on the only branch is just

$$\{F\ (\forall y)\Diamond(\forall z)\langle\lambda x.Rxz\rangle(y) \supset \langle\lambda x.\Diamond\langle\lambda z.Rxz\rangle(fx)\rangle(c)\}.$$

The propositional α rule changes this to

$$\{T\ (\forall y)\Diamond(\forall z)\langle\lambda x.Rxz\rangle(y),\ F\ \langle\lambda x.\Diamond\langle\lambda z.Rxz\rangle(fx)\rangle(c)\}.$$

Next an abstraction rule (with v_1^0 being a new variable of level 0) turns this into

$$\{T\ (\forall y)\Diamond(\forall z)\langle\lambda x.Rxz\rangle(y),\ F\ \Diamond\langle\lambda z.Rv_1^0z\rangle(fv_1^0)\},$$

and changes the tableau level substitution to $\{v_1^0/c^0\}$. Next a quantificational rule (introducing one new variable, v_2^0, of level 0) produces

$$\{T\ \Diamond(\forall z)\langle\lambda x.Rxz\rangle(v_2^0),\ F\ \Diamond\langle\lambda z.Rv_1^0z\rangle(fv_1^0)\}.$$

Now the modal rule can be applied. This leaves the tableau level substitution unaltered, but changes the branch level to 1, and the set of signed formulas into

$$\{T\ (\forall z)\langle\lambda x.Rxz\rangle(v_2^0),\ F\ \langle\lambda z.Rv_1^0z\rangle(fv_1^0)\}.$$

Applying the abstraction rule (with v_3^1 being a new variable of level 1) gives

$$\{T\ (\forall z)\langle\lambda x.Rxz\rangle(v_2^0),\ F\ Rv_1^0v_3^1\},$$

and changes the tableau level substitution to $\{v_1^0/c^0,\ v_3^1/f^1c^0\}$. A quantificational rule, again introducing one variable (v_4^1, of level 1), produces

$$\{T\ \langle\lambda x.Rxv_4^1\rangle(v_2^0),\ F\ Rv_1^0v_3^1\}.$$

Finally the abstraction rule gives

$$\{T\ Rv_5^1v_4^1,\ F\ Rv_1^0v_3^1\},$$

and turns the level substitution into $\sigma = \{v_1^0/c^0,\ v_3^1/f^1c^0,\ v_5^1/v_2^0\}$.

This tableau closes, using the level substitution $\tau = \{v_2^0/c^0,\ v_4^1/f^1c^0\}$, and thus we have a proof.

9 Tableau Soundness

Tableau soundness arguments all have the same form, with the central items being that tableau rules preserve satisfiability, but closed tableaus can't be satisfiable. The key to such an approach is finding a suitable notion of satisfiability. We begin with this, sketch the main argument, and finally show how soundness is a consequence. We must add to the classical tableau machinery something that can take level numbers and level substitutions into account.

Since variables with levels can occur in formulas in tableau proofs, for this section we assume each assignment in a model gives values for such variables, as well as for variables without levels.

Definition 9.1 Let $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$ be a non-rigid model.

1. A *world assignment of level n* in $\mathcal{M}$ is a mapping Λ that assigns to every non-negative integer $\leq n$ some member of $\mathcal{G}$ such that, for all $i < n$, $\Lambda(i)\mathcal{R}\Lambda(i+1)$.

2. Let σ be a level substitution. A function symbol f^k or a constant symbol c^k, of level k, is in the *range* of σ if it appears as part of the term $x^i\sigma$ for some variable x^i.

3. Again let σ be a level substitution, and let Λ be a world assignment of level n in $\mathcal{M}$. We say Ω is an *object assignment* relative to σ and Λ if: for each constant symbol c^k of level $k \leq n$ in the range of σ, $\Omega(c^k)$ is a member of $\mathcal{D}(\Lambda(k))$; and for each function symbol f^k of level $k \leq n$ in the range of σ, $\Omega(f^k)$ is a function from $\mathcal{D}(\Lambda(k))$ to itself.

Now we extend an earlier definition to cover terms with levels in a straightforward way.

Definition 3.3 Continued Assume σ is a level substitution, Λ is a world assignment of level n, in $\mathcal{M}$, and Ω is an object assignment relative to σ and Λ.

4. if c^k is a constant symbol (with $k \leq n$), $s(p, c^k) = \Omega(c^k)$;

5. if f^k is an m-place function symbol (with $k \leq n$),

$$s(p, f^k t_1 \ldots t_m) = \Omega(f^k)(s(p, t_1), \ldots, s(p, t_m));$$

6. if x^k is a variable in the domain of σ, $s(p, x^k) = s(p, x^k\sigma)$ (if x^k is not in the domain of σ, $s(p, x^k) = s(x^k)$, as before).

Thus we interpret each constant and function symbol with level k as always having the meaning it has according to the object assignment Ω. Given this extended definition of s, Definition 3.4 retains its form, though with a broadened meaning, since s now covers symbols with levels. Now we can give the definition of satisfiability that is needed. The idea is, signed formulas on a branch with level number n are to be considered only in the world assigned to n.

Definition 9.2 Let $\langle \sigma, \mathcal{T} \rangle$ be a tableau, $\langle n, \mathcal{B} \rangle$ be a branch of it, $\mathcal{M}$ be a model, Λ be a world assignment of level n in $\mathcal{M}$, Ω be an object assignment relative to σ and Λ. We say the branch $\langle n, \mathcal{B} \rangle$ of $\langle \sigma, \mathcal{T} \rangle$ is *satisfied* in $\mathcal{M}$ with respect to σ, Λ, and Ω if, for every assignment s in $\mathcal{M}$ (with the action of s extended as above):

1. $\mathcal{M}, \Lambda(n) \Vdash X[s]$ for every $T\ X$ on $\mathcal{B}$, and

2. $\mathcal{M}, \Lambda(n) \nVdash X[s]$ for every $F\ X$ on $\mathcal{B}$.

The tableau $\langle \sigma, \mathcal{T} \rangle$ is *satisfied* in $\mathcal{M}$ with respect to Λ and Ω if some branch of it is. Finally the tableau is *satisfiable* if it is satisfied in $\mathcal{M}$ with respect to Λ and Ω for some $\mathcal{M}$, Λ, and Ω.

As remarked above, the key to proving soundness is to show satisfiability is a loop invariant — that is, if any branch extension rule is applied to a satisfiable tableau, the result is a satisfiable tableau. We leave the verification that this is so to the reader. The proof has a number of cases, but none of them are difficult.

Let X be some closed formula. If X has a tableau proof there is a closed tableau for $\{F\ X\}$ and the empty substitution at level 0. A closed tableau cannot be satisfiable. It follows from the loop invariance of satisfiability that the initial single-branch, single-formula tableau for $\{F\ X\}$ and the empty substitution at level 0 cannot be satisfiable. It is easy to see that if X were false at any world of any model, this initial tableau would be satisfiable. Consequently if X has a tableau proof, X can not be false at any world of any model, and hence X must be **K**-valid. We thus have proved the following.

Theorem 9.3 (Tableau Soundness) *If the closed formula X has a tableau proof, X is* **K**-*valid.*

10 Tableau Completeness

Tableau completeness also follows a familiar pattern, with additions to treat the extra machinery we have introduced. In the tableau system as presented, we Skolemize before beginning a tableau construction. Consequently there is no δ rule. Let us call a signed formula *universal* if it is of the form $T\ \varphi$ and all quantifiers of φ are essentially universal, or it is of the form $F\ \varphi$ and all quantifiers of φ are essentially existential. All signed formulas that occur in a tableau proof are universal in this sense.

Definition 10.1 We say the triple $\langle S, \sigma, n \rangle$ is *worldly* if: S is a set of universal signed formulas, n is a non-negative integer, all free variables occurring in S have levels that are $\leq n$, σ is a level substitution, and σ assigns a ground term to every variable occurring free in S.

We say the worldly triple $\langle S, \sigma, n \rangle$ is *consistent* provided, for every finite subset S_0 of S, no tableau for S_0 and σ at level n is closed

We say the worldly triple $\langle S, \sigma, n \rangle$ is *downward saturated* if:

1. $\langle S, \sigma, n \rangle$ is consistent;
2. α in S implies α_1 and α_2 are in S;
3. β in S implies one of β_1 or β_2 is in S;
4. $T\ \neg X$ in S implies $F\ X$ is in S;
5. $F\ \neg X$ in S implies $T\ X$ is in S;
6. γ in S implies that for each closed term t of level n there is some level n variable z^n, not in γ, such that $\gamma(z^n)$ is in S, and $z^n\sigma = t$;
7. $T\ \langle \lambda x.\varphi(x) \rangle(t)$ in S implies $T\ \varphi(z^n)$ is in S for some level n variable z^n, not in $\langle \lambda x.\varphi(x) \rangle(t)$, such that $z^n\sigma = t@n$;
8. $F\ \langle \lambda x.\varphi(x) \rangle(t)$ in S implies $F\ \varphi(z^n)$ is in S for some level n variable z^n, not in $\langle \lambda x.\varphi(x) \rangle(t)$, such that $z^n\sigma = t@n$.

Loosely speaking, downward saturation means closure under all tableau rules except the modal ones. Now, the key step in proving completeness is the following.

Proposition 10.2 *Assume $\langle S, \sigma, n\rangle$ is consistent, and there are infinitely many variables of level n that are not in the domain of σ. Then there is a set S' that extends S, and a substitution σ' that extends σ, such that $\langle S', \sigma', n\rangle$ is downward saturated.*

We omit the proof of this proposition. Essentially it amounts to a systematic expansion of S and σ very much like the systematic construction of tableaus that is generally at the center of completeness proofs for classical tableaus. Given this proposition, a completeness proof proceeds as follows. First we construct a kind of canonical model.

- Let $\mathcal{G}$ consist of worldly triples that are downward saturated.
- For $\langle S_1, \sigma_1, n\rangle$ and $\langle S_2, \sigma_2, k\rangle$ in $\mathcal{G}$, set $\langle S_1, \sigma_1, n\rangle \mathcal{R} \langle S_2, \sigma_2, k\rangle$ if:
  1. $S_1^\# \subseteq S_2$;
  2. σ_2 extends σ_1;
  3. $k = n + 1$.
- $\mathcal{D}(\langle S, \sigma, n\rangle)$ is the set of closed terms of level n.
- For each constant symbol c and $p = \langle S, \sigma, n\rangle$, $\mathcal{I}(p, c) = c^n$.
- For each k-place function symbol f and $p = \langle S, \sigma, n\rangle$, $\mathcal{I}(p, f)$ is the function such that for level n terms $t_1, \ldots, t_k$, $\mathcal{I}(p, f)(t_1, \ldots, t_k)$ is the level n term $f^n t_1 \ldots t_k$.
- For each k-place relation symbol R and $p = \langle S, \sigma, n\rangle$, $\mathcal{I}(p, R)$ is the relation such that, for level n terms $t_1, \ldots, t_k$, $\mathcal{I}(p, R)(t_1, \ldots, t_k)$ is true provided the signed formula $T\ Rz_1^n, \ldots, z_k^n \in S$ for some level n variables $z_1^n, \ldots, z_k^n$ where $z_1^n\sigma = t_1, \ldots, z_k^n\sigma = t_k$.

We have thus defined a non-rigid model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I}\rangle$ which we call a *canonical* model..

Theorem 10.3 (Truth Theorem) *Let $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I}\rangle$ be a canonical model. Let $p = \langle S, \sigma, n\rangle$ be a member of $\mathcal{G}$, and let s be any assignment in $\mathcal{M}$ that, as a mapping, extends σ.*

1. *If $T\ \varphi \in S$ then $\mathcal{M}, p \Vdash \varphi[s]$;*
2. *If $F\ \varphi \in S$ then not-$\mathcal{M}, p \Vdash \varphi[s]$.*

Proof This is shown by induction on the complexity of φ. The atomic cases are directly by the definition of $\mathcal{M}$ (the consistency requirement on downward saturation comes in for the F-signed half). Most other cases are covered by the various downward saturation closure conditions. We give only the modal cases in detail. We consider formulas of the form $\Box\varphi$ — those of the form $\Diamond\varphi$ are treated dually.

Suppose first that the result is known for φ, $T\ \Box\varphi \in S$, and s extends σ. Let $p' = \langle S', \sigma', k\rangle$ be an arbitrary member of $\mathcal{G}$ and assume that $p\mathcal{R}p'$.

Since $p\mathcal{R}p'$, $S^\# \subseteq S'$ so $T\ \varphi \in S'$. Then by the induction hypothesis, if s' extends σ', $\mathcal{M}, p' \Vdash \varphi[s']$. Now, the free variables of φ are the same as those of $\Box\varphi$, and σ assigns ground terms to all the variables of $\Box\varphi$ hence to those of φ. Since $p\mathcal{R}p'$, σ' must extend σ, and since s' extends σ', s' also extends σ. It follows that s' and σ must agree on the variables of φ, and hence so must s and s'. Since $\mathcal{M}, p' \Vdash \varphi[s']$, we must also have $\mathcal{M}, p' \Vdash \varphi[s]$. Since p' was arbitrary, it follows from Definition 3.4 that $\mathcal{M}, p \Vdash \Box\varphi[s]$.

Finally, suppose that the result is known for φ, $F\ \Box\varphi \in S$, and s extends σ. It is not hard to check that if $S^\# \cup \{F\ \varphi\}$ were not consistent with respect to σ and $n + 1$, then S would not have been consistent with respect to σ and n, contrary to assumption. Then by Proposition 10.2, there is a set S' extending $S^{\#} \cup \{F\ \varphi\}$ and a level substitution σ' extending σ such that S' is downward saturated with respect to σ' and $n+1$. Then $p' = \langle S', \sigma', n+1\rangle$ is in $\mathcal{G}$ and $p\mathcal{R}p'$. By the induction hypothesis, not-$\mathcal{M}$, $p' \Vdash \varphi[s']$ for every s' extending σ'. As above, it follows that not-$\mathcal{M}$, $p' \Vdash \varphi[s]$, and so not-$\mathcal{M}$, $p \Vdash \Box\varphi[s]$. ∎

Now all is in place for the following.

Theorem 10.4 (Tableau Completeness) *If the closed formula X, in validity functional form, is* **K**-*valid, then X has a* **K**-*tableau proof.*

Proof Suppose X does not have a **K**-tableau proof. Then $\{F\ X\}$ is consistent with respect to the empty level substitution and level number 0. Using Proposition 10.2, there is an S extending $\{F\ X\}$ and a level substitution σ (extending the empty substitution, of course) such that $\langle S, \sigma, 0\rangle$ is downward saturated. Then $p = \langle S, \sigma, 0\rangle \in \mathcal{G}$, and by Theorem 10.3, not-$\mathcal{M}$, $p \Vdash X[s]$ for any s extending σ (actually, since X is closed, this is for *any* s). It follows that X is not **K**-valid. ∎

If no quantifiers are present in a closed formula X, there can be only a finite number of different tableaus for $F\ X$ (up to a choice of free variables). It follows that we have a decision procedure for provability, and hence validity, of quantifier-free closed formulas. Thus we have verified Theorem 6.4 as well.

11 Proof of Herbrand's Theorem

Let X be a closed modal formula, which we can assume is in validity functional form. If X is **K**-valid, it has a **K**-tableau proof. In this section we describe how to extract from such a proof a modal Herbrand expansion of X, together with a tableau proof of it, thus completing our verification of the modal Herbrand theorem. We do this in two phases. First we produce a modal Herbrand *transform* of X, possibly containing free variables. Then we show how to instantiate these variables — something that is much more complicated than its classical counterpart. We begin by extending the notion of Herbrand transform to signed formulas and to entire tableaus. Recall that universal formulas are signed formulas in which all quantifiers act universally.

Definition 11.1 A *modal Herbrand transform* of the universal formula $F\ X$ is any signed formula of the form $F\ X'$ where X' is a modal Herbrand transform of X. A *modal Herbrand transform* of the universal formula $T\ X$ is any signed formula of the form $T\ X'$ where $\neg X'$ is a modal Herbrand transform of $\neg X$. A *modal Herbrand transform* of the tableau $\langle\sigma, \mathcal{T}\rangle$ is any tableau $\langle\sigma, \mathcal{T}'\rangle$ that results by replacing each signed formula of $\mathcal{T}$ by some modal Herbrand transform of it.

Now for phase one of our proof of the modal Herbrand theorem. Suppose the closed formula X, in validity functional form, has a tableau proof. Then there is a sequence of tableaus, $\langle\sigma_1, \mathcal{T}_1\rangle, \langle\sigma_2, \mathcal{T}_2\rangle, \ldots, \langle\sigma_k, \mathcal{T}_k\rangle$ where: $\langle\sigma_k, \mathcal{T}_k\rangle$ is closed, σ_1 is the empty substitution; $\mathcal{T}_1 = \{\langle 0, \{F\ X\}\rangle\}$; and each tableau except the first results from its predecessor by the application of some tableau successor rule. We replace this sequence by a sequence of modal Herbrand transforms that yield a tableau proof of a modal Herbrand transform of X. We do so in such a way that while formulas on branches are modified, level substitutions remain the same.

We work our way backward, beginning with $\langle\sigma_k, \mathcal{T}_k\rangle$. This is closed; say the level substitution τ closes it. Then on each branch of $\mathcal{T}_k$ there must be members $T\ A$ and $F\ B$, where A and B are atomic, such that $A\sigma_k\tau = B\sigma_k\tau$. Now, $T\ A$ itself is the only modal Herbrand transform of $T\ A$, and $F\ B$ is the only modal Herbrand transform of $F\ B$, so in this case the situation is simple — replace all universal formulas present by any modal Herbrand transforms for them, and keep unchanged the level substitution σ_k. This yields a tableau that τ also closes.

Now suppose we have dealt with $\langle \sigma_{i+1}, \mathcal{T}_{i+1} \rangle$ — we say what to do with $\langle \sigma_i, \mathcal{T}_i \rangle$. There are several cases, depending on what tableau rule was applied to turn the i^{th} tableau into the $i+1^{st}$. To begin, say that a propositional rule was applied — they are all dealt with in a similar way so we pick a representative case. Say $\langle n, \mathcal{B} \rangle$ is a branch of $\langle \sigma_i, \mathcal{T}_i \rangle$, $F\ X \supset Y$ is on it, and the α rule was applied to produce $\langle \sigma_{i+1}, \mathcal{T}_{i+1} \rangle$, where the branch has been replaced by $\langle n, \mathcal{B}' \rangle$, where $\mathcal{B}'$ is $\mathcal{B}$ with $F\ X \supset Y$ removed and $T\ X$ and $F\ Y$ added. (σ_{i+1} will be the same as σ_i, though this plays no special role.)

By assumption, a modal Herbrand transform $\langle \sigma_{i+1}, \mathcal{T}'_{i+1} \rangle$ has been constructed for $\langle \sigma_{i+1}, \mathcal{T}_{i+1} \rangle$ (note that the level substitutions are the same). In this modal Herbrand transform, say $T\ X$ was replaced by its modal Herbrand transform $T\ X'$, and $F\ Y$ was replaced by its modal Herbrand transform $F\ Y'$. Then $\neg X \rightarrow \neg X'$ and $Y \rightarrow Y'$ are both derivable in the calculus of section 6, and it follows that $(X \supset Y) \rightarrow (X' \supset Y')$ is also derivable, so $F\ X' \supset Y'$ is a modal Herbrand transform of $F\ X \supset Y$. Let $\langle \sigma_i, \mathcal{T}'_i \rangle$ be the tableau that results when all the transforms that converted $\langle \sigma_{i+1}, \mathcal{T}_{i+1} \rangle$ into $\langle \sigma_{i+1}, \mathcal{T}'_{i+1} \rangle$ are applied to $\langle \sigma_i, \mathcal{T}_i \rangle$, and also $F\ X \supset Y$ is replaced by $F\ X' \supset Y'$. Note that an α rule application to $\langle \sigma_i, \mathcal{T}'_i \rangle$ turns it into $\langle \sigma_{i+1}, \mathcal{T}'_{i+1} \rangle$.

The other propositional rules are treated similarly. The quantifier rules are not much different. Say in $\langle \sigma_i, \mathcal{T}_i \rangle$ there is a branch $\langle n, \mathcal{B} \rangle$ containing $T\ (\forall x)\varphi(x)$, and this signed formula was removed from the branch and $T\ \varphi(x_1^n), \ldots, T\ \varphi(x_k^n)$ were added to produce $\langle \sigma_{i+1}, \mathcal{T}_{i+1} \rangle$ (with $\sigma_{i+1} = \sigma_i$). Again by assumption, a modal Herbrand transform $\langle \sigma_{i+1}, \mathcal{T}'_{i+1} \rangle$ has been constructed for $\langle \sigma_{i+1}, \mathcal{T}_{i+1} \rangle$ — say each of $T\ \varphi(x_1^n), \ldots, T\ \varphi(x_k^n)$ were replaced by their modal Herbrand transforms, $T\ \varphi_1(x_1^n), \ldots, T\ \varphi_k(x_k^n)$ respectively. Then each of $\neg\varphi_j(x) \rightarrow \neg\varphi_j(x)$ will be derivable in the calculus of section 6. It follows that $\neg(\forall x)\varphi(x) \rightarrow \neg[\varphi_1(x_1^n) \wedge \ldots \wedge \varphi_k(x_k^n)]$ is derivable, so $T\ \varphi_1(x_1^n) \wedge \ldots \wedge \varphi_k(x_k^n)$ is a modal Herbrand transform of $T\ (\forall x)\varphi(x)$.

Now transform the tableau $\langle \sigma_i, \mathcal{T}_i \rangle$ as we did in the propositional case, but replacing $T\ (\forall x)\varphi(x)$ by $T\ \varphi_1(x_1^n) \wedge \ldots \wedge \varphi_k(x_k^n)$. The transition from $\langle \sigma_i, \mathcal{T}'_i \rangle$ to $\langle \sigma_{i+1}, \mathcal{T}'_{i+1} \rangle$ is now by a sequence of α rule applications, in place of the original γ rule application. We omit details.

Suppose the transition from $\langle \sigma_i, \mathcal{T}_i \rangle$ to $\langle \sigma_{i+1}, \mathcal{T}_{i+1} \rangle$ was via an abstraction rule — say $T\ \langle \lambda x.\varphi(x) \rangle(t)$ was removed from branch $\langle n, \mathcal{B} \rangle$, $T\ \varphi(z^n)$ was added, and $\sigma_{i+1} = \{z^n/t@n\}\sigma_i$. And again, assume a modal Herbrand transform $\langle \sigma_{i+1}, \mathcal{T}'_{i+1} \rangle$ for $\langle \sigma_{i+1}, \mathcal{T}_{i+1} \rangle$ has been constructed. In it, say $T\ \varphi'(z^n)$ is the modal Herbrand transform of $T\ \varphi(z^n)$. Then $\neg\varphi(z^n) \rightarrow \neg\varphi'(z^n)$ is derivable in the calculus of section 6, hence so is $\neg\langle \lambda x.\varphi(x) \rangle(t) \rightarrow \neg\langle \lambda x.\varphi'(x) \rangle(t)$, and so $T\ \langle \lambda x.\varphi'(x) \rangle(t)$ is a modal Herbrand transform of $T\ \langle \lambda x.\varphi(x) \rangle(t)$. Now transform $\langle \sigma_i, \mathcal{T}_i \rangle$ into $\langle \sigma_i, \mathcal{T}'_i \rangle$ by replacing $T\ \langle \lambda x.\varphi(x) \rangle(t)$ with $T\ \langle \lambda x.\varphi'(x) \rangle(t)$ and otherwise making the changes that turned $\langle \sigma_{i+1}, \mathcal{T}_{i+1} \rangle$ into $\langle \sigma_{i+1}, \mathcal{T}'_{i+1} \rangle$. The transition from $\langle \sigma_i, \mathcal{T}'_i \rangle$ to $\langle \sigma_{i+1}, \mathcal{T}'_{i+1} \rangle$ is still by an abstraction rule.

The modal case is left to the reader.

Example Before going on to phase two of the proof, we give an example illustrating things thus far.

$$(\forall x)\Box\Box(\forall y)Rxy \supset \Box(\exists z)\Box(\forall w)Rzw$$

is a **K**-valid closed formula. A validity functional form for it is

$$(\forall x)\Box\Box(\forall y)Rxy \supset \Box(\exists z)\Box\langle \lambda w.Rzw \rangle(fz).$$

We give a closed **K**-tableau for this, then apply the process described above to produce a provable modal Herbrand transform for it, together with a tableau proof. First, here is a tableau proof of the validity functional form. We begin with the empty substitution and a single branch with a level number of 0. We display only the set of formulas on the branch, rather than the whole tableau structure. At the start we have a single formula

$$\{F\ (\forall x)\Box\Box(\forall y)Rxy \supset \Box(\exists z)\Box\langle \lambda w.Rzw \rangle(fz)\}.$$

An application of the α rule replaces this by

$$\{T\ (\forall x)\Box\Box(\forall y)Rxy,\ F\ \Box(\exists z)\Box\langle\lambda w.Rzw\rangle(fz)\}.$$

A γ rule application turns this into

$$\{T\ \Box\Box(\forall y)Rx^0y,\ F\ \Box(\exists z)\Box\langle\lambda w.Rzw\rangle(fz)\}.$$

Next, the modal rule changes the level number to 1, and the branch contents to

$$\{T\ \Box(\forall y)Rx^0y,\ F\ (\exists z)\Box\langle\lambda w.Rzw\rangle(fz)\}.$$

The γ rule again gives us

$$\{T\ \Box(\forall y)Rx^0y,\ F\ \Box\langle\lambda w.Rz^1w\rangle(fz^1)\}.$$

Now the modal rule changes the level number to 2, and the branch contents to

$$\{T\ (\forall y)Rx^0y,\ F\ \langle\lambda w.Rz^1w\rangle(fz^1)\}.$$

The predicate abstraction rule changes the tableau substitution to $\{w^2/f^2z^1\}$ and the branch contents to

$$\{T\ (\forall y)Rx^0y,\ F\ Rz^1w^2\}.$$

Finally, the γ rule turns this set into

$$\{T\ Rx^0y^2,\ F\ Rz^1w^2\}.$$

A substitution that closes this is $\tau = \{y^2/f^2x^0, z^1/x^0\}$.

Now, apply the transformations given in the proof of the theorem above — we omit the steps. At the end, the modal Herbrand transform we arrive at for

$$F\ (\forall x)\Box\Box(\forall y)Rxy \supset \Box(\exists z)\Box\langle\lambda w.Rzw\rangle(fz)$$

is

$$F\ \Box\Box Rx^0y^2 \supset \Box\Box\langle\lambda w.Rz^1w\rangle(fz^1).$$

There is a closed tableau for the set consisting of this formula and the empty substitution at level 0, and indeed, τ is the substitution that closes the tableau.

We have now described how a tableau proof of X can be converted into a tableau proof of a modal Herbrand transform of X. But this is a “hybrid” result since the transform will contain free variables with levels, and thus not be a formula of the original language but of the enlarged language introduced for the purposes of the tableau proof procedure. Indeed, the soundness proof we gave for the modal tableau system only established soundness for closed formulas!

The variables with levels that occur in the transformed tableau $\langle\sigma_1, \mathcal{T}_1'\rangle$ above are those that arose from γ rule applications in the original proof of X, since the transformation process we gave eliminates variables with levels that were introduced by abstraction rule applications. Phase two consists of removing all these γ rule free variables by instantiating them and, as remarked earlier, this is somewhat more complex than it is classically.

We explicitly note the obvious fact that if a variable x^n of level n is introduced into the tableau sequence $\langle\sigma_1, \mathcal{T}_1\rangle, \langle\sigma_2, \mathcal{T}_2\rangle, \ldots, \langle\sigma_k, \mathcal{T}_k\rangle$ by a γ rule application, it must be from a rule application on a branch of level n. A slightly less obvious fact is that, if $T\ Z$ or $F\ Z$ is present on a branch of level n in a tableau proof of X, Z must occur as a subformula of X *within the scope of n nested modalities*. This is an easy consequence of the form of the tableau modal rule. Carrying this observation over to the sequence of modal Herbrand transforms, $\langle \sigma_1, \mathcal{T}_1' \rangle, \langle \sigma_2, \mathcal{T}_2' \rangle, \ldots, \langle \sigma_k, \mathcal{T}_k' \rangle$, which yield a tableau proof of the modal Herbrand transform X' of X, it follows that if the variable x^n of level n occurs in X', it does so in a subformula that is within the scope of n nested modalities.

Now we give details of the second phase of the proof of the modal Herbrand theorem — eliminating free variables. First, as "preprocessing," there may be variables in X' that are not in the domain of τ. These we simply instantiate, more-or-less arbitrarily, as follows. Say the level n variable z^n occurs in X', but it is not in the domain of τ. Choose an arbitrary constant symbol c, and let $\tau' = \tau\{z^n/c^n\}$. Since τ is a more general substitution than τ', τ' will close any tableau that τ closes, so we can use it in place of τ. From now on we assume the level substitution τ assigns a ground term to each free variable of X'.

From here on the details are somewhat complex to present in general. We discuss a representative special case, to keep subscripts and superscripts to a minimum. Let us say the level 3 variable x^3 occurs in X'. Also, the level substitution τ closes both $\langle \sigma_k, \mathcal{T}_k \rangle$ and $\langle \sigma_k, \mathcal{T}_k' \rangle$ — say $x^3\tau = f^2c^1$, where f is a one-place function symbol and c is a constant symbol. We say how to instantiate occurrences of x^3 in X'. The idea is, loosely, to mimic the action of τ with predicate abstractions.

As noted above, each occurrence of x^3 in X' must be within the scope of 3 nested modal operators — to keep things simple, say these operators are all $\Box$, the process is similar if some of them are $\Diamond$. Pick one occurrence of x^3 in X' – we say how to instantiate it. The occurrence of x^3 that we picked is within the scope of 3 nested modal operators, hence there is a subformula of X', not within the scope of any further modal operators, that schematically has the following form:

$$\Box \cdots (\Box \cdots (\Box \cdots x^3 \cdots) \cdots) \cdots .$$

Now, let u and v be new variables (without levels), and replace this subformula of X' with the following:

$$\Box \langle \lambda u. \cdots \Box \langle \lambda v. \cdots (\Box \cdots v \cdots) \cdots \rangle (fu) \cdots \rangle (c).$$

In a tableau construction for X' after this replacement has been made, abstraction rules will cause the level substitutions associated with tableaus to, in effect, instantiate u to c^1, and subsequently v to f^2c^1. Thus some of the behavior of the substitution τ has been "built into" the formula itself, and one free variable occurrence in X' has been eliminated.

Continuing in this way, each free variable in X' can be removed, producing a closed formula that still has a tableau proof — indeed, almost the same tableau proof.

Finally, X' is a modal Herbrand transform of X. Using the Binding Rule of the calculus in section 6 (which has played no role in this section thus far), it is not hard to see that the alteration described above to X' produces yet another modal Herbrand transform of X. Since a closed formula is finally produced, we have a modal Herbrand expansion of X, and it is provable.

Example Continued At the end of phase one of the proof we gave an example. The **K**-valid closed formula $(\forall x)\Box\Box(\forall y)Rxy \supset \Box(\exists z)\Box(\forall w)Rzw$ was converted into its modal Herbrand transform $\Box\Box Rx^0y^2 \supset \Box\Box\langle \lambda w.Rz^1w \rangle (fz^1)$, and we saw there was a tableau for the F-signed version of this that was closed using the level substitution $\tau = \{y^2/f^2z^1, z^1/x^0\}$. We now instantiate the free variables, as outlined above.

We preprocess by modifying τ to deal with the fact that it assigns no value to x^0. Let c be some constant symbol, and compose the binding $\{x^0/c^0\}$ with τ, converting it into $\tau' = \{y^2/f^2c^0, z^1/c^0, x^0/c^0\}$.

Now begin by eliminating the occurrence of y^2 in $\Box\Box Rx^0y^2 \supset \Box\Box\langle \lambda w.Rz^1w \rangle (fz^1)$. Doing so yields the formula:

$$\langle \lambda u.\Box\Box\langle \lambda v.Rx^0v \rangle (fu) \rangle (c) \supset \Box\Box\langle \lambda w.Rz^1w \rangle (fz^1).$$

Eliminating x^0 produces

$$\langle\lambda r.\langle\lambda u.\Box\Box\langle\lambda v.Rrv\rangle(fu)\rangle(c)\rangle(c) \supset \Box\Box\langle\lambda w.Rz^1w\rangle(fz^1)$$

and finally, eliminating z^1 gives

$$\langle\lambda r.\langle\lambda u.\Box\Box\langle\lambda v.Rrv\rangle(fu)\rangle(c)\rangle(c) \supset \langle\lambda s.\Box\Box\langle\lambda w.Rsw\rangle(fs)\rangle(c).$$

This is a closed formula, is a modal Herbrand expansion of $(\forall x)\Box\Box(\forall y)Rxy \supset \Box(\exists z)\Box(\forall w)Rzw$, and is provable. We leave the checking of this to you.

12 Conclusion

We did not discuss equality above. Predicate abstraction lends itself naturally to the incorporation of equality into modal logic (see [10, 11]). Our modal Herbrand theorem extends directly to admit equality. We omit the rather straightforward details.

If no modal operators are present, predicate abstraction plays no essential role — the classical connectives and quantifiers are transparent with respect to it. If it is eliminated from modal-free formulas in the obvious way, the modal Herbrand theorem and its proof as given above turn into a classical version.

Finally, the modal Herbrand theorem as we have given it offers no assistance to those interested in automated theorem proving. We derived the existence of a valid Herbrand expansion from a tableau proof, and tableau proofs themselves are natural candidates for automation. However, Herbrand's original proof of his result, as corrected in subsequent years by others [13], is along quite different lines. Perhaps a study of his methods might yield something applicable to the automation of proof search in the modal area. Also, we found it necessary to introduce the mechanism of predicate abstraction. As we have urged elsewhere, this device enlarges the expressive power of first-order modal logic in useful ways. We encourage the theorem proving community to devote some effort to implementing proof methods for it.

References

[1] Marta Cialdea and Luis Fariñas del Cerro. A modal Herbrand property. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 32:523–530, 1986.

[2] Burton Dreben and John Denton. Herbrand-style consistency proofs. In A. Kino, J. Myhill, and R. E. Vesley, editors, *Intuitionism and Proof Theory, Proceedings of the Summer Conference at Buffalo N. Y. 1968*, Studies in Logic and the Foundations of Mathematics, pages 419–433, Amsterdam, 1970. North-Holland.

[3] Burton Dreben and Warren D. Goldfarb. *The Decision Problam. Solvable cases of quantificational formulas*. Advanced Book Program. Addison-Wesley, Reading, MA, 1979.

[4] Melvin C. Fitting. An epsilon-calculus system for first-order S4. In Wilfred Hodges, editor, *Conference in Mathematical Logic, London '70*, pages 103–110, 1972. *Springer Lecture Notes in Mathematics, No. 255*.

[5] Melvin C. Fitting. A modal logic analog of Smullyan's fundamental theorem. *Zeitschrift für mathematische Logik und Gründlagen der Mathematik*, 19:1–16, 1973.

[6] Melvin C. Fitting. A modal logic epsilon-calculus. *Notre Dame Journal of Formal Logic*, 16:1–16, 1975.

[7] Melvin C. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. D. Reidel Publishing Co., Dordrecht, 1983.

[8] Melvin C. Fitting. First-order modal tableaux. *Journal of Automated Reasoning*, 4:191–213, 1988.

[9] Melvin C. Fitting. *First-Order Logic and Automated Theorem Proving*. Springer-Verlag, 1990.

[10] Melvin C. Fitting. Modal logic should say more than it does. In Jean-Louis Lassez and Gordon Plotkin, editors, *Computational Logic, Essays in Honor of Alan Robinson*, pages 113–135. MIT Press, Cambridge, MA, 1991.

[11] Melvin C. Fitting. Basic modal logic. In Dov M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 1, pages 368–448. Oxford University Press, 1993.

[12] Jacques Herbrand. Investigations in proof theory. 1930. English translation in [18].

[13] Jacques Herbrand. *Logical Writings*. Harvard University Press, Cambridge, MA, 1971. Translation of 'Écrits logiques', edited by Jean Van Heijenoort.

[14] Kurt Konolige. *A deduction model of belief*. Morgan Kaufman, Los Altos, CA, 1986.

[15] J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12:23–41, 1965.

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

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

[18] Jean van Heijenoort. *From Frege to Gödel*. Harvard University Press, Cambridge, MA, 1967.