

Tableaux for Quantified Hybrid Logic^{*}

Patrick Blackburn¹ and Maarten Marx²

¹ INRIA, Lorraine, Nancy, France. patrick@aplog.org

² ILLC, University of Amsterdam, the Netherlands. marx@science.uva.nl

Abstract. We present a (sound and complete) tableau calculus for Quantified Hybrid Logic (*QHL*). *QHL* is an extension of orthodox quantified modal logic: as well as the usual \Box and \Diamond modalities it contains names for (and variables over) states, operators $@_s$ for asserting that a formula holds at a named state, and a binder \downarrow that binds a variable to the current state. The first-order component contains equality and rigid and non-rigid designators. As far as we are aware, ours is the first tableau system for *QHL*.

Completeness is established via a variant of the standard translation to first-order logic. More concretely, a valid *QHL*-sentence is translated into a valid first-order sentence in the correspondence language. As it is valid, there exists a first-order tableau proof for it. This tableau proof is then converted into a *QHL* tableau proof for the original sentence. In this way we recycle a well-known result (completeness of first-order logic) instead of a well-known proof.

The tableau calculus is highly flexible. We only present it for the constant domain semantics, but slight changes render it complete for varying, expanding or contracting domains. Moreover, completeness with respect to specific frame classes can be obtained simply by adding extra rules or axioms (this can be done for every first-order definable class of frames which is closed under and reflects generated subframes).

1 Introduction

Hybrid logic is an extension of modal logic in which it is possible to name states and to assert that a formula is true at a named state. Hybrid logic uses three fundamental tools to do this: nominals, satisfaction operators, and the \downarrow -binder. Nominals are special propositional symbols that are true at precisely one state in any model: nominals ‘name’ the unique state they are true at. A satisfaction operator has the form $@_s$ where s is a nominal. A formula of the form $@_s\phi$ asserts that ϕ is true at the state named by the nominal s . Finally, a formula of the form $\downarrow s.\phi$ binds all occurrences of the nominal s in ϕ to the current state of evaluation — that is, it makes s a name for the current state. (Actually, so that we don’t have to worry about accidental binding in the course of tableau proofs, we shall distinguish between ordinary nominals, which cannot be bound, and ‘state variables’ which are essentially bindable nominals.)

^{*} This research was supported by the Netherlands Organization for Scientific Research (NWO, grants# 612.000.106 and 612.062.001). This work was carried out as part of the INRIA funded partnership between LIT (Language and Inference Technology, ILLC, University of Amsterdam) and LED (Langue et Dialogue, LORIA, Nancy).

Hybrid logic has a lengthy history (see the web page www.hylo.net for further information), and over the years it has become clear that adding the hybrid apparatus of nominals (and state variables), satisfaction operators, and \downarrow to modal logic often results in systems with better logical properties than the original. But most previous work on hybrid logic has examined the effects of hybridizing *propositional* modal logics. What about *quantified* (first-order) hybrid logic?

In fact, strong evidence already exists that quantified hybrid logic (*QHL*) is also better behaved logically than orthodox quantified modal logic. In [2], the only recent paper devoted to the topic, it is shown that a very general interpolation theorem holds in *QHL* (as is well known interpolation almost never holds in orthodox quantified modal logic [3]). The purpose of the present paper is to show that *QHL* is well behaved in another respect: just as in the propositional case, it is possible to define simple and intuitive tableau systems. We shall present a tableau system for *QHL* which handles equality, and rigid and non-rigid designators.

Our method for proving completeness is very simple and inspired by Jerry Seligman's paper [10]. Instead of redoing a proof we use existing results. Correspondence theory and its notion of a standard translation $ST(\cdot)$ places the model theory of (propositional and first-order) modal logic firmly into first-order logic [12, 13]. Our plan is the following. We prove completeness for our tableaux calculus by taking a proof P for $ST\phi$ in a proven complete first-order calculus, and transform P into a proof P' for ϕ in our calculus. The tableaux system we use is by Fitting, in particular the one presented in [4]. This strategy works in hybrid logic because it has an equivalent expression for every subformula which might occur in a first-order proof of a translated formula.

Outline of paper. The paper starts with a definition of first-order hybrid logic. Then we present the tableau system in three natural parts. The fourth section is devoted to completeness issues. Again we split them up into three natural parts. This section ends with a very general completeness result. Finally we draw conclusions.

2 Quantified Hybrid Logic

We first define the syntax of *QHL*. We have a set *NOM* of nominals, a set *SVAR* of state variables, a set *FVAR* of first-order variables, a set *CON* of first-order constants, a set *IC* of unary function symbols, and predicates of any arity (note that predicates of nullary arity are simply propositional variables). The *terms* of the language are the constants from *CON*, the first-order variables from *FVAR* and the terms generated by the rule

$$\text{if } q \in \text{IC and } s \in \text{NOM} \cup \text{SVAR, then } @_s q \text{ is a term.}$$

(For readers familiar with propositional hybrid logic, this notation may come as a surprise: we are combining a satisfaction operator with a term to make a new term. But as the semantics defined below will show, overloading the $@$ notation in this way is quite natural: $@_s q$ will be the value of the non-rigid term q at the world named by s .)

The *atomic formulas* are all symbols in *NOM* and *SVAR* together with the usual first-order atomic formulas generated from the predicate symbols and equality using the terms. *Complex formulas* are generated from these according to the rules

$$\neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \rightarrow \psi \mid \exists x\phi \mid \forall x\phi \mid \diamond\phi \mid \square\phi \mid @_n\phi \mid \downarrow w.\phi.$$

Here $x \in \text{FVAR}$, $w \in \text{SVAR}$ and $n \in \text{NOM} \cup \text{SVAR}$.

These formulas are interpreted in first-order modal models with constant domains. A *QHL* model is a structure $(W, R, D, I_{nom}, I_w)_{w \in W}$ such that

- (W, R) is a modal frame;
- I_{nom} is a function assigning members of W to nominals;
- for every $w \in W$, (D, I_w) is an ordinary first-order model such that
 - $I_w(c) = I_{w'}(c)$, for all $w, w' \in W$ and constants c ;
 - $I_w(q) \in D$, for q a unary function symbol;
 - $I_w(P) \subseteq {}^k D$, for P a k -ary predicate symbol.

To interpret formulas with free variables we use special two-sorted assignments. A *QHL assignment* is a function g from $\text{SVAR} \cup \text{FVAR}$ to $W \cup D$ which sends state variables to members of W and first-order variables to elements of D . Given a model and an assignment g , the interpretation of terms t , denoted by \bar{t} , is defined as

$$\begin{aligned} \bar{x} &= g(x) && \text{for } x \text{ a variable} \\ \bar{c} &= I_w(c) && \text{for } c \text{ a constant and some } w \in W \\ \overline{@_n q} &= \begin{cases} I_{I_{nom}(n)}(q) & \text{if } n \text{ a nominal} \\ I_{g(n)}(q) & \text{if } n \text{ a state variable.} \end{cases} && \text{for } q \text{ a unary function symbol} \end{aligned}$$

Formulas are now interpreted as usual. With g_d^x we denote the assignment which is just like g except that $g(x) = d$. $\mathfrak{M}, g, s \Vdash \phi$ means that ϕ holds in model \mathfrak{M} at state s under the assignment g . The inductive definition is

$$\begin{aligned} \mathfrak{M}, g, s \Vdash P(t_1, \dots, t_n) &\iff \langle \bar{t}_1, \dots, \bar{t}_n \rangle \in I_s(P) \\ \mathfrak{M}, g, s \Vdash t_i = t_j &\iff \bar{t}_i = \bar{t}_j \\ \mathfrak{M}, g, s \Vdash n &\iff I_{nom}(n) = s, \text{ for } n \text{ a nominal} \\ \mathfrak{M}, g, s \Vdash w &\iff g(w) = s, \text{ for } w \text{ a state variable} \\ \mathfrak{M}, g, s \Vdash \neg \phi &\iff \mathfrak{M}, g, s \not\Vdash \phi \\ \mathfrak{M}, g, s \Vdash \phi \wedge \psi &\iff \mathfrak{M}, g, s \Vdash \phi \text{ and } \mathfrak{M}, g, s \Vdash \psi \\ \mathfrak{M}, g, s \Vdash \phi \vee \psi &\iff \mathfrak{M}, g, s \Vdash \phi \text{ or } \mathfrak{M}, g, s \Vdash \psi \\ \mathfrak{M}, g, s \Vdash \phi \rightarrow \psi &\iff \mathfrak{M}, g, s \Vdash \phi \text{ implies } \mathfrak{M}, g, s \Vdash \psi \\ \mathfrak{M}, g, s \Vdash \exists x \phi &\iff \mathfrak{M}, g_d^x, s \Vdash \phi, \text{ for some } d \in D \\ \mathfrak{M}, g, s \Vdash \forall x \phi &\iff \mathfrak{M}, g_d^x, s \Vdash \phi, \text{ for all } d \in D \\ \mathfrak{M}, g, s \Vdash \diamond \phi &\iff \mathfrak{M}, g, t \Vdash \phi \text{ for some } t \in W \text{ such that } Rst \\ \mathfrak{M}, g, s \Vdash \square \phi &\iff \mathfrak{M}, g, t \Vdash \phi \text{ for all } t \in W \text{ such that } Rst \\ \mathfrak{M}, g, s \Vdash @_n \phi &\iff \mathfrak{M}, g, I_{nom}(n) \Vdash \phi \text{ for } n \text{ a nominal} \\ \mathfrak{M}, g, s \Vdash @_w \phi &\iff \mathfrak{M}, g, g(w) \Vdash \phi \text{ for } w \text{ a state variable} \\ \mathfrak{M}, g, s \Vdash \downarrow w. \phi &\iff \mathfrak{M}, g_s^w, s \Vdash \phi. \end{aligned}$$

3 The tableau calculus

The tableau system can be divided into three natural pieces: **(A)** the propositional rules, the \diamond and \square rules and the rules for $@$; **(B)** the rule for \downarrow ; **(C)** the rules for (first-order) quantification and equality. The blocks of rules taken separately form a complete calculus for the appropriate reducts. In particular:

1. \mathbf{A} is complete for the propositional modal language expanded with nominals and $\@$. (We name this system $\mathcal{HL}(\@)$; in the literature it is often called the *basic hybrid language*.)
2. $\mathbf{A} \cup \mathbf{B}$ is complete for $\mathcal{HL}(\@, \downarrow)$, the expansion of $\mathcal{HL}(\@)$ with state variables and the \downarrow binder;
3. $\mathbf{A} \cup \mathbf{B} \cup \mathbf{C}$ is complete for QHL .

Some terminology. As usual, a tableau branch is *closed* if it contains ϕ and $\neg\phi$, where ϕ is a formula. A tableau is closed if each branch is closed. A branch is *atomically closed* if it closes on an atom and its negation. A (*tableau*) *proof* of a hybrid sentence ϕ is a closed tableau beginning with $\neg\@_s\phi$, where s is a nominal not occurring in ϕ .

3.1 Tableau for $\mathcal{HL}(\@)$

A key feature of our tableau is that all modal formulas occurring in a proof are grounded to a named world by their label. (This same feature also occurs in labelled tableau for propositional modal logic [8, 7].)

Grounding to a named state is implemented in our system by ensuring that all formulas occurring in proofs are of the form $\@_s\phi$ or $\neg\@_s\phi$ for s a nominal. Thus the propositional rules become

Conjunctive rules		
$\frac{\@_s(\phi \wedge \psi)}{\@_s\phi \quad \@_s\psi}$	$\frac{\neg\@_s(\phi \vee \psi)}{\neg\@_s\phi \quad \neg\@_s\psi}$	$\frac{\neg\@_s(\phi \rightarrow \psi)}{\@_s\phi \quad \neg\@_s\psi}$
Disjunctive rules		
$\frac{\@_s(\phi \vee \psi)}{\@_s\phi \mid \@_s\psi}$	$\frac{\neg\@_s(\phi \wedge \psi)}{\neg\@_s\phi \mid \neg\@_s\psi}$	$\frac{\@_s(\phi \rightarrow \psi)}{\neg\@_s\phi \mid \@_s\psi}$
Negation rules		
$\frac{\neg\@_s\neg\phi}{\@_s\phi}$	$\frac{\@_s\neg\phi}{\neg\@_s\phi}$	

To these we add rules for diamond and box. In the diamond rules, t is a nominal which does not occur on the branch.

Diamond rules	
$\frac{\@_s\Diamond\phi}{\@_s\Diamond t \quad \@_t\phi}$	$\frac{\neg\@_s\Box\phi}{\@_s\Diamond t \quad \neg\@_t\phi}$
Box rules	
$\frac{\@_s\Box\phi, \@_s\Diamond t}{\@_t\phi}$	$\frac{\neg\@_s\Diamond\phi, \@_s\Diamond t}{\neg\@_t\phi}$

Finally the rules for $\@$. There are two rewrite rules to delete nestings of $\@$. Next, as $\@_s t$ really means that s and t are equal, there are rules to handle equality. These three rules are direct analogues of the reflexivity and replacement rules in Fitting's first-order tableau system [4]. As we will use them often, we gave them separate names.

@ rules		
$\frac{\@_s \@_t \phi \quad \neg \@_s \@_t \phi \quad [s \text{ on the branch}]}{\@_t \phi \quad \neg \@_t \phi} \quad \text{[Ref]}$	$\frac{\@_s t \quad \@_s \varphi}{\@_t \varphi} \quad \text{[Nom]}$	$\frac{\@_s t \quad \@_r \diamond s}{\@_r \diamond t} \quad \text{[Bridge]}$

The following rules can be derived: $\frac{\@_s t}{\@_t s} \text{ [Sym]}$ $\frac{\@_s t \quad \@_t r}{\@_s r} \text{ [Trans]}$ $\frac{\@_s t \quad \@_t \varphi}{\@_s \varphi} \text{ [Nom}^{-1}\text{]}$

Example. Below we give a tableau proof for $(\diamond p \wedge \diamond \neg p) \rightarrow (\Box(q \rightarrow n) \rightarrow \diamond \neg q)$. Here n is a nominal and p, q are propositional variables. The formula expresses that if a state has two successors, then if it has at most one q successor, it has at least one $\neg q$ successor. Note that this is not expressible in ordinary modal logic. In ordinary modal logic we cannot put an upper bound on the number of successors.

1. $\neg \@_s (\diamond p \wedge \diamond \neg p \rightarrow (\Box(q \rightarrow n) \rightarrow \diamond \neg q))$
2. $\@_s (\diamond p \wedge \diamond \neg p)$
3. $\neg \@_s (\Box(q \rightarrow n) \rightarrow \diamond \neg q)$
4. $\@_s \diamond p$
5. $\@_s \diamond \neg p$
6. $\@_s \Box(q \rightarrow n)$
7. $\neg \@_s \diamond \neg q$
8. $\@_s \diamond t$
9. $\@_t p$
10. $\@_s \diamond r$
11. $\@_r \neg p$
12. $\@_t (q \rightarrow n)$

13.1 $\neg \@_t q$	14. $\@_t n$
13.2 $\neg \@_t \neg q$	15. $\@_r (q \rightarrow n)$
13.3 $\@_t q$	
16.1 $\neg \@_r q$	17. $\@_r n$
16.2 $\neg \@_r \neg q$	18. $\@_n r$
16.3 $\@_r q$	19. $\@_t r$
	20. $\@_r p$
	21. $\neg \@_r p$

In this, 2 and 3 are from 1 by a conjunctive rule; 4,5,6,7 are from 2 and 3 by conjunctive rules; 8,9,10,11 are from 4 and 5 by diamond rules; 12 is from 6 and 8 by box; 13.1 and 14 are from 12 by a disjunctive rule; 13.2 is from 7 and 8 by box; 13.3 is from 13.2 by a negation rule. The branch closes on 13.3 and 13.1.

15 is from 6 and 10 by box; 16.1 and 17 are from 15 by a disjunctive rule; 16.2 is from 10 and 7 by box; 16.3 is from 16.2 by a negation rule. The branch closes on 16.1 and 16.3.

18 is from 17 by the derived Sym rule; 19 is from 18 and 14 by the derived Trans rule; 20 is from 19 and 9 by the Nom rule; 21 is from 11 by a negation rule. The final branch closes on 20 and 21.

3.2 Tableau for $\mathcal{HL}(\downarrow, @)$

To obtain a complete tableau system for the expansion of $\mathcal{HL}(@)$ with variables over states and the binder \downarrow , we only need to add the following two rewrite rules to the rules for $\mathcal{HL}(@)$:

Downarrow rules	
$\frac{@_s \downarrow w. \phi}{@_s \phi[s/w]}$	$\frac{\neg @_s \downarrow w. \phi}{\neg @_s \phi[s/w]}$

Here $[s/w]$ means substitute s for all free occurrences of w in ϕ . Because s is always a nominal, whence cannot be quantified over, we do not have to worry about accidental bindings. As an example the reader can try to prove the validities $\downarrow w. \diamond w \rightarrow (p \rightarrow \diamond p)$ and $\downarrow w. \square \diamond w \rightarrow (\diamond \square p \rightarrow p)$.

3.3 Tableau for QHL

A complete tableau system for quantified hybrid logic consists of the $\mathcal{HL}(\downarrow, @)$ system, plus the (adjusted) rules for the quantifiers and equality from Fitting's system (see [4]) for first-order logic with equality, plus two rules relating equalities across worlds. In the existential rules, c is a parameter which is new to the branch. As parameters are never quantified over, the substitution $[c/x]$ is free for the formula $\phi(x)$. In the universal rules, t is any grounded term on the branch (thus either a first-order constant, a parameter or a grounded definite description). A grounded definite description is a term $@_n q$ for n a nominal and q a non-rigid designator from IC.

Existential rules	
$\frac{@_s \exists x \phi(x)}{@_s \phi(c)}$	$\frac{\neg @_s \forall x \phi(x)}{\neg @_s \phi(c)}$
Universal rules	
$\frac{@_s \forall x \phi(x)}{@_s \phi(t)}$	$\frac{\neg @_s \exists x \phi(x)}{\neg @_s \phi(t)}$

Besides Fitting's [4] Reflexivity (Ref) and Replacement (RR) rules, there are three extra rules for equality. The first (called DD) states that if n and m denote the same state, then $@_n q$ and $@_m q$ denote the same individual. The second and third (both called @=) embody that equality is a rigid predicate: if two terms are the same in one world, they are the same in every world. Because these two rules peel the leading $@_n$ from equalities, reflexivity and replacement can be kept in the old format. In the Replacement rule, $\phi[u]$ denotes $\phi(t)$ with some of the occurrences of t replaced by u .

QHL Equality rules				
$\frac{}{t = t} [\text{Ref}]$	$\frac{t = u, \phi(t)}{\phi[u]} [\text{RR}]$	$\frac{@_n m}{@_n q = @_m q} [\text{DD}]$	$\frac{@_n (t_i = t_j)}{t_i = t_j} [@ =]$	$\frac{\neg @_n (t_i = t_j)}{\neg (t_i = t_j)} [@ =]$

Example. The most interesting examples deal with equality and rigid and non-rigid designators. Consider the sentence *Caroline is Miss America*. When formalising this let c be a rigid designator denoting Caroline and q a non-rigid designator denoting Miss America. Then $\downarrow x.(c = @_x q)$ means *Caroline is the present Miss America*. It is true in a state w if $I_w(c) = I_w(q)$. This formula has the following relation with the \Box operator:

- (1) $\not\models (\downarrow w.c = @_w q) \rightarrow \Box \downarrow w.c = @_w q$
- (2) $\models (\downarrow w.c = @_w q) \rightarrow \downarrow w.\Box c = @_w q$.

A falsifying model for the sentence in (1) is given by two worlds n and m , with Rnm , and a domain $\{a, b\}$ with the interpretation $I_n(c) = I_m(c) = I_n(q) = a$ and $I_m(q) = b$. Then (1) fails at world n . When downarrow has wide scope in the consequent, the formula becomes true. Here is the tableau proof:

1. $\neg @_n((\downarrow w.c = @_w q) \rightarrow \downarrow w.\Box(c = @_w q))$
2. $@_n \downarrow w.c = @_w q$
3. $\neg @_n \downarrow w.\Box(c = @_w q)$
4. $@_n(c = @_n q)$
5. $\neg @_n \Box(c = @_n q)$
6. $@_n \diamond m$
7. $\neg @_m(c = @_n q)$
8. $c = @_n q$
9. $\neg(c = @_n q)$.

In this, 2 and 3 are from 1 by a conjunctive rule; 4 and 5 are from 2 and 3 by a downarrow rule, respectively; 6 and 7 are from 5 by a diamond rule; 8 and 9 are from 4 and 7 by an $@=$ rule, respectively.

4 Soundness and Completeness

The argument to establish soundness follows the familiar pattern: show that satisfiability is preserved by each tableau rule application. This is easy to check and left to the reader. Completeness will be established using the standard translation and a complete first-order inference system. We use the system that is closest to the one presented here: the tableau calculus for first-order logic with equality from Fitting [4] with the reflexivity and replacement rules (restricted to atoms). The main line of the argument is the following. We need to establish that every valid *QHL* sentence has a *QHL* tableau proof. The standard translation preserves validity, thus a *QHL* sentence ϕ is valid if and only if the first-order sentence $ST\phi$ is valid. For valid $ST\phi$, there exists a closed first-order tableau proof T starting with $\neg ST\phi$. Our task is to transform this closed first-order proof T starting with $\neg ST\phi$ into a closed *QHL* tableau proof T' starting with $\neg\phi$.

Most of the work concerns the modalities and the $@$ operator, because with these the standard translation creates the largest change in syntactic structure. For this reason we present the completeness proof for the simplest logic $\mathcal{HL}(@)$ separately. After that, the rest will be easy.

Before we can continue we have to settle two things. We change Fitting’s first-order tableau rules a little bit in order to better cope with translations of modal formulas. Besides that we have to use a modified translation. We start with the former.

In order to save on inductive proofs and definitions, we assume from now on that the *QHL* language contains as primitive logical operators only $\neg, \wedge, \square, @_s, \downarrow w,$ and $\forall v$. Clearly this is without loss of generality because the other operators can be defined in terms of these.

4.1 Tableau rules for relativized quantifiers

The translation of a box modality yields a relativized universal formula of the form $\forall x(A(x) \rightarrow C(x))$, with $A(x)$ an atom. For these relativized universals, a more efficient tableau rule exists than the combination of universal and \rightarrow rule together. In fact it is nothing but Modes Ponens. For t a closed term,

Modes Ponens (MP)	
$A(t), \forall x(A(x) \rightarrow \phi(x))$	$A(t), \neg\exists x(A(x) \wedge \phi(x))$
$\phi(t)$	$\neg\phi(t)$

We change Fitting’s calculus such that on universals relativized by an atom the normal universal rules cannot be applied, but MP can. This is easily seen to be complete (cf., also [11]). We can make a further reduction in complexity in the case the antecedent is an equality. Then the statement just expresses a substitution. We also add the following rules to Fitting’s calculus and make the proviso that universal and existential rules are never applied to quantified sentences relativized by an equality.

Substitution Rules			
$\forall x(x = t \rightarrow \phi(x))$	$\exists x(x = t \wedge \phi(x))$	$\neg\forall x(x = t \rightarrow \phi(x))$	$\neg\exists x(x = t \wedge \phi(x))$
$\phi(t)$	$\phi(t)$	$\neg\phi(t)$	$\neg\phi(t)$

4.2 Translation using predicate abstraction

Unfortunately, the standard translation does not square well with the intention to change one proof into another because it does not preserve syntactic structure. Because we want to transform a proof for the translation of ϕ into a proof for ϕ , we need to translate backwards as well. It is crucial that applying the backwards translation to the translation of ϕ yields ϕ again. This is simply not obtainable by the standard translation or obvious variants.

An example might explain why not. We can read $@_s(p \wedge q)$ as saying that state s has the property $p \wedge q$. As we want to translate proposition letters to one place predicates, in first-order logic we can only say then that s has property p and s has property q . This is of course logically equivalent, but syntactically different. We would like to have machinery which can turn formulas into predicates, so that we can speak about the property “ p and q ”. The lambda calculus provides precisely this: $\langle \lambda x.(Px \wedge Qx) \rangle$ denotes the property of being P and Q . The formula $\langle \lambda x.(Px \wedge Qx) \rangle(s)$ serves then as an excellent proxy for $@_s(p \wedge q)$.

We work in first-order logic with predicate abstraction restricted to variables ranging over individuals. Thus we only add a piece of syntactic sugar. The expressive power of the language remains the same, it is just first-order logic. For a thorough introduction to real predicate abstraction in modal logic we refer to [6].

Suppose ϕ is a first-order formula and x a first-order variable. Then $\langle \lambda x. \phi \rangle$ is a predicate abstract. Its free variable occurrences are the free variable occurrences of ϕ except for x . Predicate abstracts behave as unary predicate symbols; new atomic formulas from predicate abstracts $\langle \lambda x. \phi \rangle$ can be made by the rule

if t is a term, then $\langle \lambda x. \phi \rangle(t)$ is a formula.

Examples are $\langle \lambda x. Px \rangle(t)$ and $\langle \lambda x. Px \wedge Qx \rangle(s)$. The new formulas get their meaning by performing β -reduction:

the β -reduction of $\langle \lambda x. \phi \rangle(t)$ is $\phi[t/x]$.

The meaning of $\langle \lambda x. \phi \rangle(t)$ is simply the meaning of $\phi[t/x]$. This shows that the expressive power remains the same. Our convention is that in λ expressions, the \cdot takes wide scope, thus $\langle \lambda x. \phi \wedge \psi \rangle = \langle \lambda x. (\phi \wedge \psi) \rangle$.

In order to handle predicate abstracts in tableau proofs, we need only add two very simple rules to Fitting's system. The rules just implement β -reduction. Here they are

Abstract rules	
$\frac{\langle \lambda x. \phi \rangle(t)}{\phi[t/x]}$	$\frac{\neg \langle \lambda x. \phi \rangle(t)}{\neg \phi[t/x]}$

Fitting's tableau system with the two abstract rules added is a complete inference system for the expansion of first-order logic with λ abstraction with variables ranging over individuals [5].

We are ready to define the new standard translation AT for the propositional hybrid language, together with its inverse AT^- . In a certain sense, this translation can be traced back to the paper [9] in which McCarthy and Hayes introduce the situation calculus. $AT_y(\phi)$ and $AT_y^-(\phi)$ are defined in the same way but with x and y interchanged, e.g., $AT_y(p) := Py$ and $AT_y(\Box\phi) := \langle \lambda y. \forall x (Ryx \rightarrow AT_x(\phi)) \rangle(y)$.

$$\begin{aligned}
AT_x(p) &:= Px \\
AT_x(n) &:= x = n \\
AT_x(\neg\phi) &:= \langle \lambda x. \neg AT_x(\phi) \rangle(x) \\
AT_x(\phi \wedge \psi) &:= \langle \lambda x. AT_x(\phi) \wedge AT_x(\psi) \rangle(x) \\
AT_x(\Box\phi) &:= \langle \lambda x. \forall y (Rxy \rightarrow AT_y(\phi)) \rangle(x) \\
AT_x(@_n\phi) &:= \langle \lambda x. \forall x (x = n \rightarrow AT_x(\phi)) \rangle(x)
\end{aligned}$$

$$\begin{aligned}
AT_x^-(Px) &:= p \\
AT_x^-(x = n) &:= n \\
AT_x^-(\langle \lambda x. \neg\phi \rangle(x)) &:= \neg AT_x^-(\phi) \\
AT_x^-(\langle \lambda x. \phi \wedge \psi \rangle(x)) &:= AT_x^-(\phi) \wedge AT_x^-(\psi) \\
AT_x^-(\langle \lambda x. \forall y (Rxy \rightarrow \phi) \rangle(x)) &:= \Box AT_y^-(\phi) \\
AT_x^-(\langle \lambda x. \forall x (x = n \rightarrow \phi) \rangle(x)) &:= @_n AT_x^-(\phi)
\end{aligned}$$

The following properties of AT and AT^- hold, for every $\mathcal{HL}(@)$ formula ϕ ,

- (3) $AT_x(\phi)$ is always a formula of the form $\langle \lambda x. \psi \rangle(x)$ or Px or $x = n$.
- (4) $AT_x^-(AT_x(\phi)) = \phi$, and similarly when x is replaced by y .
- (5) ϕ is $\mathcal{HL}(@)$ valid iff $AT_x(\phi)$ is first-order valid.

(3) follows from the definition. (4) is proved by induction on the complexity of the $\mathcal{HL}(@)$ formula. (5) is immediate by performing β -reduction and the well-known result on the standard translation.

4.3 Completeness for $\mathcal{HL}(@)$

Theorem 1. *The $\mathcal{HL}(@)$ tableau calculus is complete.*

We now specify an algorithm for turning a closed Fitting tableau for the formula $AT_x(\phi)[c/x]$ (where c is a parameter) into a closed $\mathcal{HL}(@)$ tableau for $@_c\phi$. Some terminology will be useful. A literal is a grounded formula of the form

$$P(t) \mid t = u \mid Rtu \mid \langle \lambda x. \phi \rangle(t) \mid \langle \lambda y. \phi \rangle(t), \text{ or its negation.}$$

Define the following translation $(\cdot)^*$ from positive literals to $\mathcal{HL}(@)$ sentences

$$\begin{aligned} P(t)^* &:= @_t p \\ (t = u)^* &:= @_t u \\ (Rtu)^* &:= @_t \diamond u \\ (\langle \lambda x. \phi \rangle(t))^* &:= @_t AT_x^-(\langle \lambda x. \phi \rangle(x)) \\ (\langle \lambda y. \phi \rangle(t))^* &:= @_t AT_y^-(\langle \lambda y. \phi \rangle(y)). \end{aligned}$$

For negative literals $(\neg\phi)$, we set $(\neg\phi)^* = \neg\phi^*$.

We recapitulate: AT translates a hybrid formula into a first-order formula and AT^- translates them backwards. The translation $(\cdot)^*$ translates literals occurring in a first-order tableau proof into hybrid formulas. Note that these literals may contain parameters introduced in the proof. The crucial connection between the forward and backward translations is that they preserve syntactic structure: for ϕ a hybrid formula and t a nominal or parameter,

$$(6) \quad (AT_x(\phi)[t/x])^* = @_t \phi \text{ and } (\neg AT_x(\phi)[t/x])^* = \neg @_t \phi.$$

Property (6) follows immediately from the definition of $(\cdot)^*$ and (4).

We are ready to specify the algorithm. Let T be a closed Fitting tableau for the formula $AT_x(\phi)[c/x]$. Without loss of generality we may assume that T is atomically closed. Let T' simply be T with all literals replaced by their $(\cdot)^*$ translation and all other formulas removed.

Claim T' is $\mathcal{HL}(@)$ tableau proof for ϕ .

We first observe that T' starts with $\neg @_c \phi$. This is because T starts with the literal $\neg AT_x(\phi)[c/x]$ whose $*$ translation is $\neg @_c \phi$ by (6).

First-order proof	Corresponding $\mathcal{HL}(\textcircled{\@})$ proof
$\frac{t = u, P(t)}{P(u)} \text{ [RR]}$	$\frac{\textcircled{\@}_t u, \textcircled{\@}_t p}{\textcircled{\@}_u p} \text{ [Nom]}$
$\frac{t = u, v = t}{v = u} \text{ [RR]}$	$\frac{\textcircled{\@}_t u, \textcircled{\@}_v t}{\textcircled{\@}_v u} \text{ [Nom}^{-1}\text{]}$
$\frac{t = u, v R t}{v R u} \text{ [RR]}$	$\frac{\textcircled{\@}_t u, \textcircled{\@}_v \diamond t}{\textcircled{\@}_v \diamond u} \text{ [Bridge].}$

Table 1. Corresponding replacement proofs

Secondly, every branch in T' closes. This is because T branches close on literals, which we all move over to T' , keeping the negation signs in place. We now show that T' is a correct $\mathcal{HL}(\textcircled{\@})$ tableau, i.e. that every formula l^* in T' is derived from $\neg\textcircled{\@}_c \phi$ by a finite number of $\mathcal{HL}(\textcircled{\@})$ rule applications. We prove by induction on the structure of the literals that for all literals l, l' in T , for all literals l_1, l_2 produced from l, l' by applying rules, the literals l_1^*, l_2^* can be obtained from l^*, l'^* by applying a (derived) rule in T' .

There is only one zero premise rule. Ref can introduce literals $t = t$ in T , which can be matched by the hybrid Ref rule producing $(t = t)^* = \textcircled{\@}_t t$.

On literals which are not λ -formulas we can only perform Replacement, which we handle later. Every literal in T which is a λ -formula has the form $\langle \lambda z. AT_z(\psi) \rangle(t)$, for z either x or y , and ψ an $\mathcal{HL}(\textcircled{\@})$ formula. Its $(\cdot)^*$ translation is $\textcircled{\@}_t \psi$ by (6). This gives us with the cases presented in Table 2. This table is read as follows. On the left are first-order proofs with annotations indicating which rule is applied on what to obtain the result. On the right are the $\mathcal{HL}(\textcircled{\@})$ proofs which derive the $(\cdot)^*$ translated results from the $(\cdot)^*$ translated premises, again annotated.

We assumed Replacement only works on positive literals. The possible instantiations of literals in which t is replaced are

$$t = v, v = t, v R t, t R v, P(t) \text{ and } \langle \lambda z. \phi \rangle(t).$$

In Table 1 the application of the replacement rule is given on the left while the corresponding $\mathcal{HL}(\textcircled{\@})$ proof on the $(\cdot)^*$ images of the formulas is on the right. As the cases for $P(t)$, $t = v$, $t R v$ and $\langle \lambda z. \phi \rangle(t)$ are all by applications of Nom, we only show the case for $P(t)$.

We considered all possible applications of all rules on all possible literals. Thus T' is a $\mathcal{HL}(\textcircled{\@})$ tableau.

4.4 Completeness for $\mathcal{HL}(\downarrow, \textcircled{\@})$

Theorem 2. *The tableau system for $\mathcal{HL}(\downarrow, \textcircled{\@})$ is complete.*

Case	FO tableau	$\mathcal{HL}(@)$ tableau
\neg , pos	(1) $\langle \lambda x. \neg AT_x(\phi) \rangle(t)$ (2) $\neg AT_x(\phi)[t/x]$ (1), λ	(1) $@_t \neg \phi$ (2) $\neg @_t \phi$ (1), Neg
\neg neg	(1) $\neg \langle \lambda x. \neg AT_x(\phi) \rangle(t)$ (2) $\neg \neg AT_x(\phi)[t/x]$ (1), $\neg \lambda$ (3) $AT_x(\phi)[t/x]$ (2), $\neg \neg$	(1) $\neg @_t \neg \phi$ (2) $@_t \phi$ (1), Neg
\wedge pos	(1) $\langle \lambda x. AT_x(\phi) \wedge AT_x(\psi) \rangle(t)$ (2) $AT_x(\phi)[t/x] \wedge AT_x(\psi)[t/x]$ (1), λ (3) $AT_x(\phi)[t/x]$ (2), Con (4) $AT_x(\psi)[t/x]$ (2), Con	(1) $@_t(\phi \wedge \psi)$ (2) $@_t \phi$ (1), Con (3) $@_t \psi$ (1), Con
\wedge neg	(1) $\neg \langle \lambda x. AT_x(\phi) \wedge AT_x(\psi) \rangle(t)$ (2) $\neg [AT_x(\phi)[t/x] \wedge AT_x(\psi)[t/x]]$ (1), $\neg \lambda$ (3) $\overline{\neg AT_x(\phi)[t/x] \mid \neg AT_x(\psi)[t/x]}$ (2), Dis	(1) $\neg @_t(\phi \wedge \psi)$ (2) $\neg @_t \phi \mid \neg @_t \psi$, (1), Dis
@ pos	(1) $\langle \lambda x. \forall x(x = n \rightarrow AT_x(\phi)) \rangle(t)$ (2) $\forall x(x = n \rightarrow AT_x(\phi))$ (1), λ (3) $AT_x(\phi)[n/x]$ (2), Sub	(1) $@_t @_n \phi$ (2) $@_n \phi$ (1), @
@ neg	(1) $\neg \langle \lambda x. \forall x(x = n \rightarrow AT_x(\phi)) \rangle(t)$ (2) $\neg \forall x(x = n \rightarrow AT_x(\phi))$ (1), $\neg \lambda$ (3) $\neg AT_x(\phi)[n/x]$ (2), Sub	(1) $\neg @_t @_n \phi$ (2) $\neg @_n \phi$ (1), @
\square pos	(1) $\langle \lambda x. \forall y(Rxy \rightarrow AT_y(\phi)) \rangle(t)$ (2) Rtn (3) $\forall y(Rty \rightarrow AT_y(\phi))$ (1), λ (4) $AT_y(\phi)[n/y]$ (2), (3) MP	(1) $@_t \square \phi$ (2) $@_t n$ (3) $@_n \phi$ (1), (2), \square
\square neg	(1) $\neg \langle \lambda x. \forall y(Rxy \rightarrow AT_y(\phi)) \rangle(t)$ (2) $\neg (\forall y(Rty \rightarrow AT_y(\phi)))$ (1), $\neg \lambda$ (3) Rtc (2), Exi (4) $\neg AT_y(\phi)[c/y]$ (2), Exi	(1) $\neg @_t \square \phi$ (2) $@_t \diamond c$ (1), \diamond (3) $\neg @_c \phi$ (1), \diamond

Table 2. Corresponding proof rules

With all the groundwork done, the proof is very easy. We have to extend the translation to incorporate the variables and downarrow formulas. We assume that x and y are new variables. The translation and its inverse for the state variables and downarrow is simply

$$\begin{aligned}
AT_x(w) &:= x = w \\
AT_x^-(x = w) &:= w \\
AT_x(\downarrow w. \phi) &:= \langle \lambda x. \forall w(w = x \rightarrow AT_x(\phi)) \rangle(x) \\
AT_x^-(\langle \lambda x. \forall w(w = x \rightarrow \phi) \rangle(x)) &:= \downarrow w. AT_x^-(\phi).
\end{aligned}$$

In a straightforward way, the properties (3)–(6) still hold. Then the completeness proof amounts to showing that Fitting’s rules applied to translations of downarrow formulas can be transformed to applications of the downarrow rules. On these translations only substitutions can be applied. This case is similar to the @ case, so we do not spell it out.

4.5 Completeness for *QHL*

Theorem 3. *The tableau system for QHL is complete.*

Again the proof is simple after we made the needed straightforward adjustments. The translation and its inverse for the full *QHL* language is obtained by adding the following rules to the ones already existing:

$$\begin{aligned}
AT_x(P(t_1, \dots, t_k)) &:= P'(x, t_1, \dots, t_k) \\
AT_x(t_i = t_j) &:= \langle \lambda x. t_i = t_j \rangle(x) \\
AT_x(\forall v \phi) &:= \langle \lambda x. \forall v AT_x(\phi) \rangle(x) \\
AT_x^-(P'(x, t_1, \dots, t_k)) &:= P(t_1, \dots, t_k) \\
AT_x^-(\langle \lambda x. t_i = t_j \rangle(x)) &:= t_i = t_j \\
AT_x^-(\langle \lambda x. \forall v \phi \rangle(x)) &:= \forall v AT_x^-(\phi)
\end{aligned}$$

The translation $(\cdot)^*$ is extended for the new literals as follows:

$$\begin{aligned}
P'(s, t_1, \dots, t_k)^* &:= @_s P(t_1, \dots, t_k) \\
(t_i = t_j)^* &:= t_i = t_j.
\end{aligned}$$

We don't translate the *QHL* terms $@_s q$ but just pretend they are first-order terms $q(s)$. Again, properties (3)–(6) still hold. (The first-order tableau calculus has to respect the two sorts of course. For example, $\forall x P'(s, x)$ does not yield the not correctly typed $P'(s, s)$ by universal instantiation.) The atomic hybrid formula $t_i = t_j$ is translated as $\langle \lambda x. t_i = t_j \rangle(x)$. This is done to have a syntactic analogue of $@_s(t_i = t_j)$. In a first-order proof, β -reduction can be applied to $\langle \lambda x. t_i = t_j \rangle(s)$ or its negation, yielding $t_i = t_j$ and $\neg t_i = t_j$, respectively. This proof step corresponds to an application of one of the $@ =$ rules on the $(\cdot)^*$ translations in a *QHL* tableau.

It is immediate that the quantifier rules can be mimicked in *QHL* tableaux (provided they respect the sorts).

For the application of replacement, there are now terms $@_n q$ for q a non-rigid designator and n a nominal. The replacement rule in a first-order proof can then with the premise $n = m$ replace $@_n q$ by $@_m q$ in any atom. But $n = m$ back-translates to $@_n m$ and from that the *QHL* equality rule DD yields $@_n q = @_m q$. Now replacement in *QHL* with this premise on the translated atom yields the translated result.

Thus all first-order rules have a corresponding *QHL* analogue and we are done.

4.6 Completeness for specific frame classes

We only considered the (quantified) hybrid logic of the class of all frames. Here we establish completeness for every elementary first-order definable class of frames which is closed under and reflects generated subframes. A class of frames is closed under generated subframes if all generated subframes of its members are in the class. A class reflects generated subframes if whenever \mathcal{F} is in the class and \mathcal{F}' is a generated subframe of \mathcal{F} , then also \mathcal{F}' is in the class. Note that this implies that the class is closed under

disjoint unions. Closure under and reflection of generated subframes is a requirement which reflects the local evaluation of modal formulas.¹

We recall from [1], that every such elementary class of frames is definable by a first-order sentence $\forall y\gamma(y)$, in which $\gamma(y)$ is equivalent to a pure hybrid $\mathcal{HL}(\@, \downarrow)$ sentence γ' (i.e., without propositional variables nor nominals). As AT preserves meaning we may without loss of generality assume that $\gamma(y) = AT_y(\gamma')$.

Let such a class K be defined by $\forall y\gamma(y)$. Then a QHL sentence ϕ is valid on K iff $\forall y\gamma(y) \rightarrow AT_x(\phi)[c/x]$ for c a new parameter is first-order valid. In that case, there is a first-order tableau proof starting with

1. $\forall y\gamma(y)$
2. $\neg AT_x(\phi)[c/x]$.

Whence the proof will develop almost as for $AT_x(\phi)[c/x]$ except that for any state parameter or nominal s , $\gamma(s)$ may be introduced on the branch. This insight leads to the following rule to be added to the QHL tableau system:

$$\frac{}{\@_s\gamma'} \quad \text{for } s \text{ on the branch.}$$

Now every time a $\gamma(s)$ is added to the branch in the first-order proof, we apply the new rule on s in the QHL proof. Because of the assumption on the form of γ , translating $\gamma(s)$ by $(\cdot)^*$ yields $\@_s\gamma'$. Thus we have shown

Theorem 4. *Let γ a pure nominal free hybrid sentence which axiomatises the class of frames K . Then adding the above rule to the QHL tableau calculus yields completeness for the quantified hybrid logic of the class of frames K .*

¹ **Added in proof.** Balder ten Cate together with the first author of the present paper (from now on referred to as BC) have proposed a proof system for $\mathcal{HL}(\@)$ which is complete for frame classes defined by formulas of the form $\forall\bar{x}\exists\bar{y}\phi(\bar{x}, \bar{y})$, in which ϕ is an $\mathcal{HL}(\@)$ formula starting with $\@_{x_1}$ and the quantifiers bind all nominals \bar{x} and \bar{y} occurring in ϕ . A natural example is $\forall x_1x_2x_3\exists y\@_{x_1}(\@_{x_1}\diamond x_2 \wedge \@_{x_1}\diamond x_3 \rightarrow \@_{x_2}\diamond y \wedge \@_{x_3}\diamond y)$ defining the class of confluent frames.

BC claim that adding the rule (*) below to the $\mathcal{HL}(\@)$ tableaux calculus given here is complete for the class of frames defined by $\forall\bar{x}\exists\bar{y}\phi(\bar{x}, \bar{y})$. This result is most easily proved using the developed theory of translations, as follows: Assume a $\mathcal{HL}(\@)$ formula ψ is valid on the class of frames defined by $\forall\bar{x}\exists\bar{y}\phi(\bar{x}, \bar{y})$. Then $\forall\bar{x}\exists\bar{y}AT_{x_1}(\phi(\bar{x}, \bar{y})) \rightarrow AT_x(\psi)[c/x]$ for c a new parameter is FO valid. Thus there is a first-order tableau proof of it starting with $\forall\bar{x}\exists\bar{y}AT_{x_1}(\phi(\bar{x}, \bar{y})), \neg AT_x(\psi)[c/x]$. Our goal is to turn this tableau into an $\mathcal{HL}(\@)$ tableau as before. The only new thing we have to mimic is an application of universal instantiation followed by existential elimination to $\forall\bar{x}\exists\bar{y}AT_{x_1}(\phi(\bar{x}, \bar{y}))$. This is exactly what rule (*) of BC is doing.

$$(*)\overline{\phi(\bar{s}, \bar{t})} \quad \text{for } \bar{s} \text{ nominals on the branch and all } \bar{t} \text{ nominals new to the branch.}$$

Analogous to the proof in this subsection, completeness now follows.

5 Conclusions

The positive effects of hybridization in propositional logic extend well to the first-order case. In fact, one could argue that the need for hybridization is felt much stronger in first-order modal logic. The field is plagued with failures of desirable properties, and consequently more difficult and obscure than its propositional counterpart. Here we have presented an extremely general completeness theorem (Theorem 4) covering virtually all modally interesting elementary frame classes. In a companion paper we have shown that the calculus can be used to construct interpolants. Interpolation is one of the properties which fail in many quantified modal logics. This theorem also extends to all frame classes from Theorem 4. These very general results indicate that the additions to the syntax are natural and extremely useful.

The paper contained two important ideas. First and foremost is the proof method for showing completeness. An almost standard translation was used in a non-trivial way to transfer a first-order result into the modal setting. In the hybrid language, this was particularly easy, as it contains such first-order proof-elements as parameters. In orthodox modal logic, too many completeness proofs are repeated with only tiny changes. Maybe hybridization is needed to change modal logic into a field in which standard results are recycled instead of proofs. It's worth the price.

The second idea is our treatment of definite descriptions like *Miss America*. In *QHL* it is not possible to write intensional terms as in Montague's *IL*. The hidden variables in intensional terms cause many technical problems and make *IL* mathematically complicated. The use of @ to ground non-rigid designators to states is a simple remedy.

References

1. C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation and complexity. *Journal of Symbolic Logic*, 66(3):977–1010, 2001.
2. C. Areces, P. Blackburn, and M. Marx. Repairing the interpolation lemma in quantified modal logic. Report PP–2001–19, ILLC. To appear in *Annals of Pure and Applied Logic*.
3. K. Fine. Failures of the interpolation lemma in quantified modal logic. *Journal of Symbolic Logic*, 44(2):201–206, 1979.
4. M. Fitting. *First Order Logic and Automated Theorem Proving (second edition)*. Springer Verlag, 1996.
5. M. Fitting. *Types, Tableaus, and Gödel's God*. Unpublished Draft, 1999.
6. M. Fitting and R. Mendelsohn. *First-Order Modal Logic*. Kluwer, 1998.
7. D. Gabbay. *Labelled Deductive Systems*. Oxford University Press, 1996.
8. R. Goré. Tableau methods for modal and temporal logics. In M D'Agostino, et al editors, *Handbook of Tableau Methods*, pages 297–396. Kluwer Academic Publishers, 1999.
9. John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.
10. J. Seligman. Internalization: The case of hybrid logic. *Journal of Logic and Computation*, 11(5):671–689, 2001.
11. R. Smullyan. *First Order Logic*. Springer-Verlag, 1968.
12. J. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, Naples, 1983.
13. J. van Benthem. Correspondence theory. In D.M. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic*, volume 2, pages 167–248. Reidel, Dordrecht, 1984.

References

1. C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation and complexity. *Journal of Symbolic Logic*, 66(3):977–1010, 2001.
2. C. Areces, P. Blackburn, and M. Marx. Repairing the interpolation lemma in quantified modal logic. Technical Report PP–2001–19, Institute for Logic, Language and Computation, University of Amsterdam, 2001. To appear in *Annals of Pure and Applied Logic*.
3. K. Fine. Failures of the interpolation lemma in quantified modal logic. *Journal of Symbolic Logic*, 44(2):201–206, 1979.
4. M. Fitting. *First Order Logic and Automated Theorem Proving (second edition)*. Springer Verlag, 1996.
5. M. Fitting. *Types, Tableaus, and Gödel's God*. Unpublished Draft, 1999.
6. M. Fitting and R. Mendelsohn. *First-Order Modal Logic*. Kluwer Academic Publishers, 1998.
7. D. Gabbay. *Labelled Deductive Systems*. Number 33 in Oxford logic guides. Oxford University Press, 1996.
8. R. Goré. Tableau methods for modal and temporal logics. In M D'Agostino, D Gabbay, R Haenle, and J Posegga, editors, *Handbook of Tableau Methods*, pages 297–396. Kluwer Academic Publishers, 1999.
9. John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969. <http://www-formal.stanford.edu/jmc/mcchay69.html>.
10. J. Seligman. Internalization: The case of hybrid logic. *Journal of Logic and Computation*, 11(5):671–689, 2001.
11. R. Smullyan. *First Order Logic*. Springer-Verlag, 1968.
12. J. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, Naples, 1983.
13. J. van Benthem. Correspondence theory. In D.M. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic*, volume 2, pages 167–248. Reidel, Dordrecht, 1984.