Variants of First-Order Modal Logics

Marta Cialdea Mayer[1] and Serenella Cerrito[2]

[1] Università Roma Tre, Dipartimento di Informatica e Automazione
[2] Université de Paris-Sud, L.R.I.

Abstract. In this paper we study proof procedures for some variants of first order modal logics, where domains may be either cumulative or freely varying and terms may be either rigid or non-rigid, local or non-local. We define both *ground* and *free variable* tableau methods, parametric with respect to the variants of the considered logics. The treatment of each variant is equally simple and is based on the annotation of functional symbols by natural numbers, conveying some semantical information on the worlds where they are meant to be interpreted.

1 Introduction

First order modal logics constitute a delicate and difficult subject (see for example [12] for an overview). In principle, their semantics could simply be obtained by extension of the modal propositional semantics: a first order modal structure can be built on the base of a set of first order classical interpretations (the "possible worlds"), connected by a binary relation (the accessibility relation). However, there are a number of different possibilities that can be considered, concerning for example:

The object domains: is there any relation between the universes of the different worlds? They can be required to be the same for all worlds (*constant domains*), they can bear no relation one to the other (*varying domains*), or the object domains can vary, but monotonically, i.e. if w' is accessible from w, then the object domain of w is included in the domain of w' (*cumulative domains*).

The designation of terms: is the denotation of terms the same in all worlds or can it vary? When the answer is positive, then designation is taken to be *rigid*, otherwise it is *non-rigid* (often, mixed approaches are of interest, too, where the interpretations of some symbols are rigid, others are not).

The existence of objects: does the extension of any ground term belong to every object domain or not? If the answer is positive, then we assume terms to be *local*, otherwise terms are *non-local.*

So, several variants of quantified modal logics (QMLs) are possible, just by choosing different combinations of the cases considered above. We call them DDE variants (Domains/Designation/Existence variants) of QML. Obviously, the logic also depends on its propositional kernel, so that we have a four-dimensional space of possible QMLs. As often happens with modal logics, different choices are appropriate for different applications. A discussion can be found in [12] as well as [11].

Some DDE variants can easily be given a proof theoretic characterization, some are harder to treat. The easiest approach is obviously obtained just by adding the principles of classical logic to a propositional modal system. What we obtain then is a logic with cumulative domains, rigid designation and local terms. In fact, such a logic validates the converse of Barcan Formula (**CBF**), that characterizes cumulative domains: $\Box\forall x A \rightarrow \forall x\Box A$. Rigid designation and locality of terms are consequences of the instantiation rule of classical logic $\forall x A \rightarrow A[t/x]$. For example, $\Box p(c)$ follows from the instantiation rule and $p(c) \wedge \forall x(p(x) \rightarrow \Box p(x))$, but it is not a logical consequence of the latter formula alone, if the denotation of c can vary from world to world (the example is from [16]).

Besides the axiomatic characterization, some first order modal logics with cumulative domains, rigid designation and local terms have been given sequent and tableau calculi [9, 10, 15], natural deduction [5], matrix proof procedures [19], resolution style calculi [1, 7], and translation based procedures [3, 18].

Now, while translation methods are general enough to treat also other DDE variants of QML (for instance, [3] deals with constant domains logics, where designation may be rigid as well as non rigid, and [18] handles all the DDE variants of QML), the direct methods are less flexible. For instance, it can be proved that constant domain logics cannot be treated by standard tableau systems (cut free sequent calculi). The addition of prefixes labelling tableau nodes solves this problem [9], as matrix methods do [19]: in both kinds of calculi in fact it is possible to analyse more than one possible world at a time, and this allows the proof to "go back and forth" (the same mechanism solves the problem of symmetric logics). In [9] all the variants of QML concerning the object domains of possible worlds are treated by prefix tableau methods and some suggestions on how to treat the varying domains case in calculi without prefixes can be found. A direct approach dealing with both varying domains and non-rigid symbols has a representative in [16], which defines a resolution method for epistemic logics, where terms can be annotated by a "bullet" constructor distinguishing rigid terms from non-rigid ones. A different direction is followed by [11], where the language of modal logic is enriched by means of a predicate abstraction operator, in order to capture differences on the denotation of terms, and a tableau proof procedure is presented for such a logic, with no restriction on the domains of possible worlds.

In general, direct proof methods can treat the less easy variants of QML either by modifications of classical and/or modal rules in an often cumbersome manner (if the axiomatic approach is taken), or else by attaching some kind of semantical information to the proof structures.

The aim of this work is to provide a treatment of some variants of QML, where the explicit intrusion of semantics in the proof structures is kept as simple and minimal as possible. We propose calculi in the tableau style, that allow us to treat all the DDE variants of QML (that are formally defined at the end of this section), with the exclusion of constant domain logics, that we believe cannot be captured in the same style. Namely, we deal with the combinations cumulative/varying domains, rigid/non rigid designation, local/non local terms, for the analytic logics **K, D, T, K4, S4**. The treatment of each DDE variant is equally simple and is based on the annotation of functional symbols by natural numbers, resulting in a minimal departure from standard tableau methods (i.e. cut-free sequent systems). A similar mechanism is used in [7, 8], to obtain skolem-like normal forms and define modal resolution proof procedures.

We define both *ground* tableau calculi – called Tab_G (Section 2), dealing with closed formulae – and *free variable* tableau calculi – called Tab_{FV} (Section 3). In both cases, the calculi are parametric with respect to the DDE variants of the considered logic. The treatments differ only in the annotation mechanism and a suitable classification of terms according to the annotation of their symbols, that determines, in Tab_G, which applications of the γ-rule are correct and, in Tab_{FV}, which substitutions are allowed. Both classes of calculi are sound and complete. Due to space restrictions, proofs are only sketched (Section 4).

In the rest of this section, we briefly recall the syntax and semantics of QML. A first order modal language L is constituted by logical symbols (propositional connectives, quantifiers, modal operators and a countable set X of individual variables), a non empty set L_P of predicate symbols and a set L_F of functional symbols with an associated arity. Constants are considered as functional symbols with null arity. Terms are built using symbols from L_F and X. The inductive definition of terms and formulae is as usual.

A first order modal interpretation $\mathcal{M}$ of a language L is a tuple $\langle W, w_0, R, D, \delta, \phi, \pi\rangle$ such that:

- W is a non empty set (the set of "possible worlds");
- w_0 is a distinguished element of W;
- R is a binary relation on W (the *accessibility relation*);
- D is a non empty set (the object domain);
- δ is a function assigning to each $w \in W$ a non empty subset of D, the domain of w: $\delta(w) \subseteq D$;
- ϕ represents the interpretation of constants and functional symbols in the language: for every world $w \in W$ and n-ary functional symbol in the language (with $n \geq 0$), $\phi(w, f) \in D^n \rightarrow D$.
- π is the interpretation of predicate symbols: if p is a n-ary predicate symbol and $w \in W$, then $\pi(w, p) \subseteq D^n$ is a set of n-tuples of elements in D.

The accessibility relation R of a modal structure can be required to satisfy additional properties, characterizing different logics: we consider seriality (**D**), reflexivity (**T**), transitivity (**K4**), both reflexivity and transitivity (**S4**). The logic **K** makes no additional assumption on R.

The interpretation function ϕ is extended to terms in the usual way, and, by an abuse of notation, $\phi(w, t)$ denotes the interpretation of t in w. If $\mathcal{M} = \langle W, w_0, R, D, \delta, \phi, \pi\rangle$ is an interpretation of the language L, the **language of the model** $\mathcal{M}$, L_D, is obtained from L by addition of a "name" for each $d \in D$, i.e. a new constant $\overline{d}$. The interpretation $\mathcal{M}$ is assumed to interpret names in such a way that that for every $d \in D$ and $w \in W$, $\phi(w, \overline{d}) = d$. Note that the interpretation of names $\overline{d}$ is always rigid.

Below we take $\neg$, $\wedge$ and $\vee$ as the only primitive propositional connectives. The relation $\models$ between an interpretation $\mathcal{M} = \langle W, w_0, R, D, \delta, \phi, \pi \rangle$, a world $w \in \mathcal{M}$ and a closed formula in L_D is defined inductively as follows:

1. $\mathcal{M}, w \models p(t_1, ..., t_n)$ iff $\langle \phi(w, t_1), ..., \phi(w, t_n) \rangle \in \pi(w, p)$.
2. $\mathcal{M}, w \models \neg A$ iff $M, w \not\models A$.
3. $\mathcal{M}, w \models A \wedge B$ iff $\mathcal{M}, w \models A$ and $\mathcal{M}, w \models B$.
4. $\mathcal{M}, w \models A \vee B$ iff $\mathcal{M}, w \models A$ or $\mathcal{M}, w \models B$.
5. $\mathcal{M}, w \models \forall x A$ iff for all $d \in \delta(w)$, $\mathcal{M}, w \models A[\overline{d}/x]$
6. $\mathcal{M}, w \models \exists x A$ iff there exists $d \in \delta(w)$ such that $\mathcal{M}, w \models A[\overline{d}/x]$
7. $\mathcal{M}, w \models \Box A$ iff for any $w' \in W$ such that $\langle w, w' \rangle \in R$, $\mathcal{M}, w' \models A$
8. $\mathcal{M}, w \models \Diamond A$ iff there is a $w' \in W$ such that $\langle w, w' \rangle \in R$ and $\mathcal{M}, w' \models A$

A closed formula A is true in $\mathcal{M}$ iff $\mathcal{M}, w_0 \models A$, and it is valid iff it is true in any interpretation $\mathcal{M}$.

Let $\mathcal{M} = \langle W, w_0, R, D, \delta, \phi, \pi \rangle$ be a first order modal interpretation. We shall deal with the following DDE variants of QML:

Cumulative domains: for all $w, w' \in W$, if wRw' then $\delta(w) \subseteq \delta(w')$.
Varying domains: no restrictions on $\delta(w)$.
Rigid designation: for all $w, w' \in \mathcal{M}$ and for every functional symbol f, $\phi(w, f) = \phi(w', f)$.
Non-rigid designation: no restrictions on the relation between $\phi(w, f)$ and $\phi(w', f)$, for any $w \neq w'$ and $f \in L_F$.
Local terms: for all $w \in \mathcal{M}$ and for every functional symbol f, if $d_1, ..., d_n \in \delta(w)$, then $\phi(w, f)(d_1, ..., d_n) \in \delta(w)$.
Non-local terms: no restriction on the closure of domains $\delta(w)$ with respect to the application of the designation of functional symbols.

For the sake of simplicity, we treat each variant as a different logic. However, mixed approaches are also possible. For instance, we can establish that constant symbols denote rigidly and functional symbols (of arity greater than 0) do not. In the calculi described in the rest of this work, each symbol would then be treated according to the rules of the corresponding logic.

Note that, if terms are local, the denotation of every ground term t is in every domain $\delta(w)$. However, if domains are allowed to vary without restrictions, if $\mathcal{M}, w \models \exists x \Diamond p(f(x))$, and d is the element of $\delta(w)$ such that $\mathcal{M}, w \models \Diamond p(f(\overline{d}))$, then the denotation of $f(\overline{d})$ is not necessarily in domain of the world w' accessible from w and such that $\mathcal{M}, w' \models p(f(\overline{d}))$, unless $d \in \delta(w')$.

2 Ground Tableaux

In this section we illustrate the mechanism that allows us to deal with some DDE variants of QML. We show how to treat the cases of cumulative and varying domains, where designation of terms can be either rigid or non-rigid, and terms either local or not. In the following, we assume that a given logic has been fixed, with a propositional basis that is one of the analytic logics **K, D, K4, T**, or **S4**, and a first order variant.

The tableau systems presented in the sequel work with formulae in negation normal form: negation over non-atomic formulae is considered as a defined symbol. In other terms, modal formulae are built out of literals (atoms and negated atoms) by use of $\wedge$, $\vee$, $\Box$, $\Diamond$ and the quantifiers $\forall$ and $\exists$. Negation over non-atomic formulae is defined as usual.

Below, we present the ground calculus Tab_G. Formulae occurring in a ground tableau for a set S of modal formulae are built over an extension L^+ of the language L of S, where constants and functional symbols can be annotated by a numerical superscript, that is called the *level* of the symbol. Symbols with superscripts are called annotated symbols. L^+ extends L with two different infinite sets of constants: a set $\mathcal{A}$ containing a constant for each level $k \in \mathbb{N}$ and a set $\mathcal{C}$, containing an infinite number of constants for each level $k > 0$. Moreover, L^+ contains the annotated versions of the constants and functional symbols in L. Precisely:

Definition 1. *Let L be a modal language, $\mathcal{A} = \{a_0^0, a_1^1, ...\}$ and $\mathcal{C}$ disjoint sets of annotated constants different from any symbol in L, such that $\mathcal{A}$ contains exactly one constant a_k^k for each $k \in \mathbb{N}$, and $\mathcal{C}$ contains a denumerable set of constants $c_{k_1}^k, c_{k_2}^k, ...$ annotated with level k, for each $k \geq 1$. Then the* labelled extension *of L is the language L^+ such that $L_P^+ = L_P$ and L_F^+ is:*

$$L_F \cup \mathcal{C} \cup \mathcal{A} \cup \{f^k \mid f \in L_F \text{ and } k \in \mathbb{N}\}$$

The nodes of a ground tableau for a set of closed formulae in the modal language L are labelled sets of the form $n : S$ where n is a natural number, called the *depth* of the node, and S is a set of closed modal formulae in the language L^+. Similarly to the sequent calculus in [17], in this approach there is no labelling of single formulae, but of whole sets of formulae.

Symbol annotations and node depths are related: let t be the occurrence of a term in a node $n : S$, that we can assume describes a given world w; then the levels of the symbols in t tell us whether t has a denotation in the domain of w, and whether its denotation can be assumed to be the same as the denotation of another occurrence of the same term t (with possibly different annotations).

Next, we define the initial tableau for S and an operation that is used in some propositional expansion rules.

Definition 2.

1. *If S is a set of modal formulae, then the initial tableau for S is the single node $1 : S^*$, where:*
   - **local terms and rigid designation:** *S^* is obtained from S by annotating each functional symbol with level 0.*
   - **local terms and non-rigid designation:** *S^* is obtained from S by annotating each* occurrence *of a functional symbol outside the scope of any modal operator with level 1.*

non local terms: $S^* = S$

2. *The* **base language** *of the tableau rooted at* $1 : S^*$ *is the language* L *of* S. *The* **language of the tableau** *is the labelled extension* L^+ *of* L.
3. *If* S *is a set of modal formulae and* $n \in \mathbb{N}$, *then:*
   local terms: S^n *is obtained from* S *by annotating each non-annotated* occurrence *of a functional symbol outside the scope of any modal operator with level* n;
   non local terms: $S^n = S$.

The main intuition behind the above notions is that symbols annotated with level 0 are always symbols of the original language (and have not been introduced by means of the δ-rule, defined below), their interpretation is the same in every possible world and the domain of every world is closed with respect to the application of the functions they denote. Thus, symbols of level 0 occur only in the calculus for the DDE variant with rigid designation and local terms.

In the case of non-rigid designation and local terms, on the contrary, a symbol of the base language L can occur with different levels in a tableau branch: when occurring with level n, it actually refers to the denotation of the symbol in the world corresponding to the node with depth n in that branch. Symbols occurring outside the scope of a modal operator are initialized with level 1, the depth of the initial node. Symbols of L_F occurring inside a modal operator receive their levels only when, by application of a propositional rule, they come to the surface (by means of operation 3 in Definition 2).

In the case of non local terms, on the contrary, no assumption is made on the denotation of symbols of the base language: they have no level in the initial node of the tableau and never get one.

We consider a simple set of propositional tableau rules (others may be chosen as well), shown in Table 1, where S, S' are sets of modal formulae (in the language of the tableau), $\Box S$ stands for $\{\Box G \mid G \in S\}$, S' is a set of non-boxed modal formulae, comma is set union. The propositional part of the tableau systems we consider consists of the classical rules, and: the rule π_K in systems **K**, **D** and **T**, the rule π_4 in **K4** and **S4**, the rule ν_T in **T** and **S4**, the rule ν_D in **D**.

Adopting a terminology from [13], the rules that do not modify the depth of the node are called *static*, the others are called *dynamic*. So, a non-annotated symbol can obtain a superscript only with the application of dynamic rules.

As can be noted, the depth n of a node $n : S$ in a tableau "counts" the number of dynamic rules applied in the branch before $n : S$. And, in fact, the labelling of nodes is not essential and the depth of a node could be defined as a meta-level concept. For this reason the present approach is not in the style of prefixed tableaux.

The next definition establishes the relation between the depth of a node and levels of symbols, by means of the concept of *domain* of a node, that is needed to formulate the quantifier tableau rules.

Definition 3. *Let* L *be a modal language and* L^+ *its labelled extension.*

1. *A term* t *in* L^+ *is in* **depth** n *iff every symbol in* t *has a level and* t *contains only symbols with:*

Classical rules	
$(\alpha)\ \dfrac{n : A \wedge B, S}{n : A, B, S}$	$(\beta)\ \dfrac{n : A \vee B, S}{n : A, S \qquad n : B, S}$
ν-rules	
$(\nu_D)\ \dfrac{n : \Box S, S'}{n+1 : S^{n+1}}$	$(\nu_T)\ \dfrac{n : \Box A, S}{n : A^n, \Box A, S}$
π-rules	
$(\pi_K)\ \dfrac{n : \Diamond A, \Box S, S'}{n+1 : A^{n+1}, S^{n+1}}$	$(\pi_4)\ \dfrac{n : \Diamond A, \Box S, S'}{n+1 : A^{n+1}, S^{n+1}, \Box S}$

Fig. 1. Propositional expansion rules

varying domains and non-rigid designation: *level* n*;*
varying domains and rigid designation: *level* n *and level* 0*;*
cumulative domains: *level* $k \leq n$*.*

2. *If* $n : S$ *is a tableau node, then its domain is:*

$$dom(n : S) = \{a_p^p\} \cup \{t \mid t \textit{ occurs in } S \textit{ and } t \textit{ is in depth } n\},$$

where

varying domains, rigid designation and local terms: *if there is at least a constant in* L *then* $p = 0$*, otherwise* $p = n$*;*
varying domains and either non-rigid designation or non-local terms: $p = n$*;*
cumulative domains: $p = 1$

Roughly speaking, the object denoted by a term in $dom(n : S)$ (and occurring in a tableau) can be assumed to exist in the domain of the world "corresponding" to the node $n : S$. Such terms are formed by use of symbols of level n itself, symbols of level 0 in the case of rigid designation (i.e. symbols of the base language L), and symbols having lower level in the case of cumulative domains, e.g. constants denoting objects existing in the domain of some previous world, therefore existing also in the considered world (in the ground version of the calculus we are presenting in this section, the only functional symbols with non null arity are those of the base language L).

Note that an annotated term may be in more than one depth. Moreover, if $t \in dom(n : S)$, then t is a ground term, it does not contain any non-annotated symbol and every subterm t' of t is in $dom(n : S)$.

The quantifiers expansion rules in Tab_G are the following:

$(\gamma) \quad \dfrac{n : \forall x A, S}{n : A[t/x], \forall x A, S}$	$(\delta) \quad \dfrac{n : \exists x A, S}{n : A[c^n/x], S}$

In the γ-rule, t is any ground term in $dom(n : \forall x A, S)$. In the δ-rule, c is a constant in $\mathcal{C}$ that does not occur in $\{A\} \cup S$. Note that c needs not be new in the whole tableau: the local restriction on the δ-rule corresponds exactly to the *eigenvariable* condition in sequent calculi.

Since $dom(n : \forall x A, S)$ is never empty, even if $n : \forall x A, S$ does not contain any ground term, the γ-rule can always be applied at least once to such a node. The constants of the set $\mathcal{A}$ (see Definition 1) in fact are used to take into account the fact that the object domains are never empty, in much the same way as an additional constant is used to form Herbrand domains in classical logic when there are no ground terms. The difference, here, is that we may need to have an infinite number of such constants, one for each depth. If domains are cumulative, then one is enough: it denotes any object in the initial world w_0, that exists in any other world accessible from w_0. Also if domains can vary but the base language L contains at least a constant c and terms are assumed to be local, one additional constant is enough: in fact, if terms of L are local, c denotes an object existing in every world, and the constant a_0^0 will be taken to denote such an object. If none of the previous cases applies, then it may be the case that any two object domains have empty intersection, so we need a different constant to name an arbitrary object of each domain.

In Tab_G, once that an occurrence of a symbol gets a level, it never changes. In particular, new constants introduced by the δ-rule never change their level and the only symbols that may occur with different annotations in a tableau node are symbols from the base language, when we are in the case of non-rigid designation and local terms. In that case, f^n is the denotation of f in the world corresponding to the node of depth n, that can be different from the denotation of the same symbol in another world. In consideration of this fact, the notion of complementary literals does not ignore the level of symbols.

Definition 4. *Two literals P and $\neg Q$ in the language L^+ of a tableau are complementary iff P and Q are identical, including their annotations. A node is closed if it contains two complementary literals. A tableau is closed iff all its leaves are closed.*

For example, $p(c^1)$ and $\neg p(c^1)$ are complementary, but $p(c^1)$ and $\neg p(c^2)$ are not.

In the case of cumulative domains, rigid designation and local terms, the calculus is not different from the tableau system described in [9], Chapter 7 (where, however, the language does not contain any functional symbol with non null arity). In fact, every functional symbol in a tableau is annotated and $n : S$ never contains symbols of level greater than n. So every term occurring in $n : S$ is in $dom(n : S)$.

Here follow some examples. The tableau on the left below proves the validity of **CBF** in the cumulative domain logics, using the π_K-rule. Since there are no functional symbols in L_F, there is no distinction to be made about the designation of terms. The last step in the varying domain systems would not be allowed: $\forall x p(x)$ can be instantiated only with terms in depth 2. Moreover, since there are no choice points above and the last expansion can only be replaced by an application of the γ-rule on the constant a_2^2 (there are no constants in L_F), **CBF** cannot be proved valid in the varying domains systems.

$$
\begin{array}{cl}
1 : \neg(\Box\forall x p(x) \to \forall x \Box p(x)) & \\ \hline
1 : \Box\forall x p(x),\ \neg\forall x \Box p(x) & (\alpha) \\ \hline
1 : \Box\forall x p(x),\ \neg\Box p(c^1) & (\delta) \\ \hline
2 : \forall x p(x),\ \neg p(c^1) & (\pi_K) \\ \hline
2 : p(c^1),\ \forall x p(x),\ \neg p(c^1) & (\gamma)
\end{array}
\qquad
\begin{array}{cl}
1 : \neg(\forall x \Box p(x) \to \Box\forall x p(x)) & \\ \hline
1 : \forall x \Box p(x),\ \neg\Box\forall x p(x) & (\alpha) \\ \hline
1 : \Box p(a^1),\ \forall x \Box p(x),\ \neg\Box\forall x p(x) & (\gamma) \\ \hline
2 : p(a^1),\ \neg\forall x p(x) & (\pi_k) \\ \hline
2 : p(a^1),\ \neg p(c^2) & (\delta)
\end{array}
$$

A similar reasoning shows that no instance of the Barcan formula (**BF**) $\forall x \Box A \to \Box\forall x A$, that characterizes constant domains, is provable. The tableau above on the right is an attempt, using again the π_K-rule and the instance where $A = p(x)$. In this case there is no difference between the cumulative and varying domains variants of the calculus. Since with the application of the π_K-rule the universal formula is necessarily lost, the tableau cannot be closed.

Below, on the left, there is a proof of $\Box(\forall x p(x) \to p(c))$ (in any propositional basis) with rigid designation (either varying or cumulative domains: in both cases the term c^0 is in any depth) and local terms, showing that, in this variant of the systems, the rule of instantiation is necessarily valid. The same formula has a different proof (on the right, below) in the case of non rigid designation (and local terms).

$$
\begin{array}{cl}
1 : \neg\Box(\forall x p(x) \to p(c^0)) & \\ \hline
2 : \neg(\forall x p(x) \to p(c^0)) & (\pi) \\ \hline
2 : \forall x p(x),\ \neg p(c^0) & (\alpha) \\ \hline
2 : p(c^0),\ \forall x p(x),\ \neg p(c^0) & (\gamma)
\end{array}
\qquad
\begin{array}{cl}
1 : \neg\Box(\forall x p(x) \to p(c)) & \\ \hline
2 : \neg(\forall x p(x) \to p(c^2)) & (\pi) \\ \hline
2 : \forall x p(x),\ \neg p(c^2) & (\alpha) \\ \hline
2 : p(c^2),\ \forall x p(x),\ \neg p(c^2) & (\gamma)
\end{array}
$$

And it cannot be proved when terms are not necessarily local. Here follows a failed attempt:

$$
\begin{array}{cl}
1 : \neg\Box(\forall x p(x) \to p(c)) & \\ \hline
2 : \neg(\forall x p(x) \to p(c)) & (\pi) \\ \hline
2 : \forall x p(x),\ \neg p(c) & (\alpha) \\ \hline
2 : p(a_2^2),\ \forall x p(x),\ \neg p(c) & (\gamma)
\end{array}
$$

The next example is a (failed) attempt to show that $p(c), \forall x(p(x) \to \Box p(x)) \vdash \Box p(c)$ in the case of non-rigid designation and local terms:

$$\cfrac{\cfrac{1 : \forall x(p(x) \to \Box p(x)),\ p(c^1),\ \neg\Box p(c)}{1 : \forall x(p(x) \to \Box p(x)),\ p(c^1) \to \Box p(c^1),\ p(c^1),\ \neg\Box p(c)}\ (\gamma)}{\begin{array}{c} 1 : \forall x(p(x) \to \Box p(x)), \\ \neg p(c^1),\ p(c^1),\ \neg\Box p(c) \end{array} \qquad \cfrac{\begin{array}{c} 1 : \forall x(p(x) \to \Box p(x)), \\ \Box p(c^1),\ p(c^1),\ \neg\Box p(c) \end{array}}{2 : p(c^1),\ \neg p(c^2)}\ (\pi_k)}\ (\beta)$$

The leftmost leaf is closed, but the rightmost is not, because $p(c^1)$ and $\neg p(c^2)$ are not complementary. And in fact $\Box p(c)$ is not a logical consequence of $p(c)$ and $\forall x(p(x) \to \Box p(x))$ in the case of non-rigid designation and local terms. If terms are not local, even the first inference is not correct. In fact, the tableau is initialized without any annotation on symbols and the γ-rule can never be applied to a term with no levels.

3 Free variable tableaux

In this section we introduce Tab_{FV}, i.e. free-variable tableaux for the considered logics. The definition of quantifier rules is similar to the classical case, but for the annotation on symbols that is taken into account in the definition of modal substitutions. First of all, we define a further extension of the labelled extension of a language (Definition 1), by addition of *parameters* and *skolem functions*:

Definition 5. *Let L be a modal language, $\mathcal{P} = \{z_1^{n_1}, z_2^{n_2}, ...\}$ an infinite denumerable set of new annotated symbols, called parameters, and $\mathcal{F}$ a denumerable set of new annotated function symbols such that, for each $i, j \in \mathbb{N}$, $\mathcal{P}$ contains infinitely many parameters of level i and $\mathcal{F}$ contains infinitely many function symbols of level i and arity j. Then the* labelled free variable extension *of L is the language L^{fv} such that $L_P^{fv} = L_P$ and*

$$L_F^{fv} = L_F^+ \cup \mathcal{P} \cup \mathcal{F}$$

The levels of parameters will be shown only when relevant. The definition of the depth of a term (Definition 3) carries along to terms in L^{fv} with no modification. The next definition introduces the notion of modal substitution, where a parameter z can be mapped to a term t in L^{fv} only if the annotation of symbols in t "agrees" with the level of z. Since the agreement depends on which depths the term is, the different DDE variants of QML correspond to different restrictions on modal substitutions.

Definition 6. *A modal substitution σ is a function from $\mathcal{P}$ to terms in L^{fv}, that is the identity almost everywhere, and such that if $\sigma(z^n) = t$ (and z has level n) then t is in depth n. Substitutions are denoted as usual by expressions of the form $\{t_1/z_1, ..., t_m/z_m\}$.*

Unifiers, i.e. solutions of unification problems, and most general unifiers (*m.g.u.*) are defined like in the classical case. The following definition is also borrowed from classical tableau calculi.

Definition 7. *Let $n_1 : S_1, ..., n_k : S_k$ be all the leaves of a free variable tableau $\mathcal{T}$, and, for each $i = 1, ..., k$, let P_i and $\neg Q_i$ be a pair of literals in S_i. If the modal substitution σ is a solution of the unification problem $P_1 = Q_1, ..., P_k = Q_k$, then σ is an* atomic closure substitution *for $\mathcal{T}$. If it is a most general solution for such a unification problem, then it is a* most general atomic closure substitution *for $\mathcal{T}$.*

Besides the propositional expansion rules, that are the same as in Tab_G (but for the fact that parameters and skolem functions can occur in tableau nodes), the expansion rules of Tab_{FV} are:

Quantifier expansion rules :

$(\gamma) \quad \dfrac{n : \forall xA, S}{n : A[z^n/x], \forall xA, S}$	$(\delta) \quad \dfrac{n : \exists xA, S}{n : A[f^n(z_1^n, ..., {z_k}^n)/x], S}$

In the γ-rule, z is a new parameter. In the δ-rule, f^n is a new skolem function, i.e. a symbol of level n in $\mathcal{F}$ that does not occur in $\{\exists xA\} \cup S$, and ${z_1}^n, ..., {z_k}^n$ are all the parameters with level n in $\exists xA, S$.

Most General Atomic Closure Substitution Rule :
If $\mathcal{T}$ is a tableau for a set S of sentences in L and the modal substitution σ is a most general atomic closure substitution for $\mathcal{T}$, then the tableau $\mathcal{T}\sigma$, obtained by applying σ to $\mathcal{T}$, is a tableau for S.

Note that the skolem term $f^n(z_1^n, ..., {z_k}^n)$ introduced by an application of the δ-rule contains only parameters with level n, and the skolem function f is required to be "locally" new (i.e. new in the node). This is enough to ensure that the rule is sound. In fact, the role of the restriction on f and the parameters in the term is to prevent a modal substitution σ to make $f^n(z_1^n, ..., {z_k}^n)$ equal to some other parameter z occurring in the premise of the δ-rule, $n : \exists xA, S$. If z has level n then $\sigma(f^n(z_1^n, ..., {z_k}^n)) = \sigma(z)$ is impossible, because z is one of $z_1^n, ..., {z_k}^n$. The level m of z cannot be greater than n, since a node never contains any symbol with level greater than its depth. Therefore, if $m \neq n$, then m is necessarily strictly less than n, and, by definition, a modal substitution cannot map a variable with level $m < n$ to a term containing a symbol of level n, that is not in depth m in any DDE variant of QML.

The definition of closed tableau (Definition 4) stays unchanged for Tab_{FV} (modulo the possible presence of parameters and skolem functions in terms). Note that in the calculi Tab_{FV} defined here, closed tableaux are built up by first applying expansion rules, then computing an m.g.u. allowing to simultaneously close all the leaves, and applying the substitution rule to the tableau so far constructed just once. Assuming a single, final application of the substitution rule makes the soundness proof simpler. However, like in the classical case, an alternative version of the rule can be considered, where any m.g.u. of a pair of literals in a single leaf can be applied. Since any most general atomic closure substitution can be found by repeated applications of such an alternative rule, this second version of the calculi is equivalent to the first one.

As an example, below, we give again a proof of the validity of **CBF** in the cumulative domain logics, using the π_K-rule, but this time in the calculus Tab_{FV}.

$$
\dfrac{\dfrac{\dfrac{\dfrac{1 : \neg(\Box\forall x p(x) \to \forall x \Box p(x))}{1 : \Box\forall x p(x), \neg\forall x \Box p(x)}\,(\alpha)}{1 : \Box\forall x p(x), \neg\Box p(c^1)}\,(\delta)}{2 : \forall x p(x), \neg p(c^1)}\,(\pi_K)}{2 : p(z_1^2), \forall x p(x), \neg p(c^1)}\,(\gamma)
$$

Here, c^1 is a (0-ary) skolem function. Since c^1 is in depth 2, it suffices to apply the substitution $\{c^1/z_1^2\}$ to close the tableau. In the case of varying domain systems, such a substitution is illegal, because c^1 is not in depth 2.

As a further example, the tableau below is a Tab_{FV} proof of the validity of the formula $\Box\forall x\exists y p(x,y) \to \exists v\Box\exists w p(v,w)$ in the logic **K** with cumulative domains (since L_F is empty, there is no difference to be made about the designation of terms).

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{1 : \neg(\Box\forall x\exists y p(x,y) \to \exists v\Box\exists w p(v,w))}{1 : \Box\forall x\exists y\ p(x,y),\ \neg\exists v\Box\exists w p(v,w)}\,(\alpha)}{1 : \Box\forall x\exists y\ p(x,y),\ \neg\Box\exists w p(z_1^1,w)}\,(\gamma)}{2 : \forall x\exists y\ p(x,y),\ \neg\exists w p(z_1^1,w)}\,(\pi_K)}{2 : \exists y\ p(z_2^2,y),\ \forall x\exists y\ p(x,y),\ \neg\exists w p(z_1^1,w)}\,(\gamma)}{2 : p(z_2^2, g^2(z_2^2)),\ \forall x\exists y\ p(x,y),\ \neg\exists w p(z_1^1,w)}\,(\delta)}{2 : p(z_2^2, g^2(z_2^2)),\ \forall x\exists y\ p(x,y),\ \neg\exists w p(z_1^1,w),\ \neg p(z_1^1, z_3^2)}\,(\gamma)
$$

Since domains are cumulative, the term z_1^1 is in any depth $k \geq 1$, thus, in particular, it is in depth 2. Hence $\{z_1^1/z_2^2,\ g^2(z_1^1)/z_3^2\}$ is a modal substitution, and constitutes a most general atomic closure substitution. Its application yields a closed tableau. Note that in the varying domains option, no substitution can close the tableau, since z_1^1 is not in depth 2, and neither $\{z_1^1/z_2^2\}$ nor $\{z_2^2/z_1^1\}$ are legal substitutions. Indeed, the considered formula is not valid in this case.

4 Soundness and Completeness of Tab_G and Tab_{FV}

In this section we briefly sketch the main ideas underlying the soundness and completeness proofs for both systems Tab_G and Tab_{FV}.

In order to show that Tab_G is sound, we inductively show that every tableau for a satisfiable set of formulae S has always an open leaf, i.e. a satisfiable leaf, where a node $n : S$ is satisfied by a world w in $\mathcal{M}$ iff $\mathcal{M}, w \models S$ (ignoring annotations) and for every ground term $t \in dom(n : S)$, $\mathcal{M}(w,t) \in \delta(w)$. The latter requirement is exploited to show that if the premise of a γ-rule is satisfiable, then so is its expansion. And it must be guaranteed to hold both in the initial tableau and when a dynamic rule is applied.

In the completeness proof for Tab_G, we first consider a calculus where the γ-rule is replaced by its "wasteful" variant γ^*:

$$(\gamma^*)\ \frac{n : \forall x A, S}{n : A[t/x], \forall x A, S} \qquad \text{where } t \text{ is any ground term in depth } n \text{ of the language of the tableau } L^+$$

A tableau in this modification of the calculus will be called a γ^*-tableau. Completeness is proved by construction of a "syntactical model" of any set of sentences that has no closed γ^*-tableau, and the construction obviously exploits the wasteful γ^*-rule. Then we show that any γ^*-expansion in a closed γ^*-tableau can be replaced by a γ-expansion (using a suitable term from the set of symbols $\mathcal{A}$ introduced in Definition 1), thus obtaining a closed tableau in the original ground calculus Tab_G.

The syntactical model of the unrefutable set of formulae S is built in two steps. First of all an interpretation $\mathcal{M}$ of the extension L^+ of the language L of S is constructed, where however annotations are not ignored: a symbol f^i is considered different from f^j when $i \neq j$. The frame of $\mathcal{M}$ is a (possibly infinite and infinitely branching) tree, whose elements are saturated sets of formulae (the definition of saturation is similar to the standard one, obviously referring to the calculi we are considering). The initial world w_0 is a saturation of S^*, where $1 : S^*$ is the initial tableau for S. Each element w of the frame is associated with a natural number $depth(w)$, that measures its distance from the root.

The elements δ and ϕ of $\mathcal{M}$ make each possible world w a kind of Herbrand interpretation: the domain $\delta(w)$ is the set of ground terms in $dom(depth(w), w)$, where terms that differ (even only) in the annotation of their symbols are different elements of the domain. Annotated functional symbols are always given a rigid interpretation, in such a way that, if a term t contains only annotated symbols, then $\phi(w,t) = t$ for all $w \in W$. The treatment of non-annotated symbols differs in the cases of local and non-local terms. Finally, an atom P is true at w iff $P \in w$. It can be shown that, for all $w \in W$, $\mathcal{M}, w \models A$ for all $A \in w$, where, like before, symbols occurring with different levels are considered different. For this reason, $\mathcal{M}$ is not yet an interpretation of the language L, and we cannot directly conclude that $\mathcal{M} \models S$. From the interpretation of the (possibly infinite) set of symbols $f^1, f^2, f^3, \ldots$ where $f \in L_F$, *the* interpretation of f is extracted for every $w \in W$. Such a new interpretation satisfies all the requirements of the different DDE variants of QML we are considering, and is indeed a model of S.

The soundness and completeness of Tab_{FV} is established by showing the equivalence between the calculi Tab_{FV} and Tab_G. Therefore, since Tab_G is sound and complete, S is unsatisfiable if and only if there is a closed Tab_{FV} tableau for S. The proof is based on a syntactical transformation of Tab_G-tableau proofs into tableau proofs in Tab_{FV} and viceversa.

5 Concluding Remarks

In this paper we provide calculi treating different variants of first order modal logic (varying and cumulative domains, rigid and non rigid designation, local and non local terms), by addition of a minimal semantical annotation to tableau methods obtained by extending propositional modal systems with the classical rules for quantifiers. We give both a ground formulation of the calculi (Tab_G) and free variable tableau systems (Tab_{FV}). In both cases, the different DDE variants are obtained in a simple and modular way, just by choosing among alternative definitions of a few basic operations. In particular, in the case of Tab_{FV} calculi, a first order variant is essentially characterized by the notion of modal substitution. Differently from other approaches such as for instance [15, 2], however, term unification does not depend on the propositional kernel of the logic (i.e. on the accessibility relation) and does not involve any complex or strongly semantically dependent operation: what is additionally needed, with respect to classical unification, is some numerical "greater than" test.

Although we have treated each DDE variant as a different logic, some mixed approaches are possible with respect to the denotation of symbols: for example, it can be established that constant symbols (proper names) are rigid while functional symbols are not, so that definite descriptions do not necessarily denote rigidly. Each symbol is then to be treated according to the corresponding variant in the construction of the initial tableau and in the application of dynamic rules. We believe that equality can easily be treated in the same style.

In the case of the (proof-theoretically) easiest DDE variant assuming cumulative domains, rigid designation and local terms, the calculi Tab_G are actually the same as what is obtained by a simple addition of classical quantifier rules to modal propositional ones (such as, for example, in [9], Chapter 7). However, their Tab_{FV} counterpart is not exactly the same as proposed in [10], where a tableau calculus with free variables, dynamic skolemization and unification is defined, that is correct and complete for this variant of QML. There, the new skolem term $f(z_1, ..., z_n)$ introduced by an application of the δ-rule is such that $z_1, ..., z_n$ are all the parameters occurring in the branch. Exploiting the annotation on symbols, the δ-rule in Tab_{FV} is somewhat more liberal, requiring only parameters having the same level as the depth of the node to be the arguments of the skolem functions. Probably, the δ-rule can be further relaxed, in the style of [14, 6, 4]. The details, however, still have to be worked out.

The (ground) prefixed tableaux presented in Chapter 8 of [9], in the three different variants of constant, cumulative and varying domains, associate prefixes with constants introduced by applications of the δ-rule in a way that bears some resemblance with Tab_G and its labelling mechanism. The present approach, however, is not in the style of prefixed tableaux, where prefixes are associated to each formula (not the whole set of "active" formulae), are more complex structures than integers and play a fundamental role in modal expansion rules. A stricter relation links the present work with [7, 8], where the same kind of annotation is used to restrict unification in the context of modal resolution. However, being the skolem-like transformation static in that case, it works well with logics such as **K** and **D**, where formulae cannot move from their modal context, but requires a preliminary propositional transformation of formulae when dealing with reflexive and/or transitive logics, where the modal degree of a formula is subject to change. The dynamic introduction of parameters and skolem terms in tableau systems makes things easier: a quantifier is treated only when it comes to the surface, i.e. when, after possible changes due to applications of the ν_T or π_4-rule, its modal context is settled.

References

1. M. Abadi and Z. Manna. Modal theorem proving. *CADE-86*, 172–189, 1986.
2. A. Artosi, P. Benassi, G. Governatori, and A. Rotolo. Shakespearian modal logic: A labelled treatment of modal identity. *Advances in Modal Logic*, 1–21, 1998.
3. Y. Auffray and P. Enjalbert. Modal theorem proving: an equational viewpoint. *J. of Logic and Computation*, 2:247–297, 1992.
4. M. Baaz and C.G. Fermüller. Non-elementary sppedups between different versions of tableaux. *Tableaux '95*, 217–230, 1995.
5. D. Basin, M. Matthews, and L. Viganò. Labelled modal logics: Quantifiers. *J. of Logic, Language and Information*, 7(3):237–263, 1998.
6. B. Beckert, R. Hähnle, and P. H. Schmitt. The even more liberalized δ-rule in free variable semantic tableaux. In *Proc. of the 3rd Kurt Gödel Colloquium KGC'93*, 108–119, 1993.
7. M. Cialdea. Resolution for some first order modal systems. *Theoretical Computer Science*, 85:213–229, 1991.
8. M. Cialdea and L. Fariñas del Cerro. A modal Herbrand's property. *Z. Math. Logik Grundlag. Math.*, 32:523–539, 1986.
9. M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. Reidel Publ. Co., 1983.
10. M. Fitting. First-order modal tableaux. *J. of Automated Reasoning*, 4:191–213, 1988.
11. M. Fitting. On quantified modal logic. *Fundamenta Informaticae*, 39, 1999.
12. J. W. Garson. Quantification in modal logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, vol. II, 249–307. Reidel Publ. Co., 1984.
13. R. Goré. Tableau methods for modal and temporal logics. In M. D'Agostino, G. Gabbay, R. Hähnle, and J. Posegga, editors, *Handbook of tableau methods*. Kluwer, 1999.
14. R. Hähnle and P. H. Schmitt. The liberalized δ-rule in free variable semantic tableaux. *J. of Automated Reasoning*, 13:211–222, 1994.
15. P. Jackson and H. Reichgelt. A general proof method for first-order modal logic. IJCAI '87, 942–944, 1987.
16. K. Konolige. Resolution and quantified epistemic logics. *CADE 86*, 199–208, 1986.
17. G. Mints. Indexed systems of sequents and cut-elimination. *J. of Philosophical Logic*, 26:671–696, 1997.
18. H. J. Ohlbach. Semantics based translation methods for modal logics. *J. of Logic and Computation*, 1(5):691–746, 1991.
19. L. A. Wallen. *Automated Deduction in Nonclassical Logics: Efficient Matrix Proof Methods for Modal and Intuitionistic Logics*. MIT Press, 1990.