Beyond Pure Axioms:
Node Creating Rules in Hybrid Tableaux

Patrick Blackburn

Equipe Langue et Dialogue
INRIA Lorraine
Nancy, France

Balder ten Cate

Institute for Logic, Language and Computation
University of Amsterdam
Amsterdam, The Netherlands

Abstract

We present a method of extending the tableau calculus for the basic hybrid language which automatically yields completeness results for many frame classes that cannot be defined by means of pure axioms (for example, Church-Rosser frames). The extended calculus makes use of *node-creating rules*. These rules trade on the idea of using nominals to perform skolemization on formulas of the *strong hybrid language*. Alternatively, viewing them from a Hilbert-style perspective, such rules can be viewed as a systematic generalization of Gabbay's irreflexivity rule. Our completeness result covers all frame classes definable by *pure nominal-free universal existential sentences* of the strong hybrid language. This properly includes all frame classes definable by universal existential first-order sentences.

---

1 Basic Hybrid Logic

Basic hybrid logic is the result of extending modal logic with *nominals* and the *@-operator*. Suppose we are given a set σ of modalities, and two (countably infinite) disjoint sets PROP (whose elements are typically written p, q, and r, possibly subscripted, and called proposition letters) and NOM (whose elements are typically written i, j, k, and l, possibly subscripted, and called nominals). Then the basic hybrid language over σ, PROP and NOM is defined as follows:

$$\phi ::= p \mid i \mid \neg\phi \mid \phi \wedge \psi \mid \triangle(\phi_1, \dots \phi_n) \mid @_i\phi.$$

Here p is a proposition letter, i is a nominal, and $\triangle$ is an n-ary modality (an element of σ). Thus, except for the clauses for i and $@_i\phi$, this is the standard definition of a modal language with arbitrary arity modalities (see, for example, Definition 1.12 in [6]). We follow the usual convention of writing $\Diamond\phi$ rather than $\triangle(\phi)$ when working with unary modalities.

What do the clauses for i and $@_i\phi$ give us? Nominals are special proposition letters that are true at precisely one node in any model: they 'name', or 'label', the unique node they are true at. The @ operator allows us to assert that a formula is true at a named node: $@_i\phi$ says that ϕ is true at the node named by the nominal i. In short, by hybridizing the modal language we make it referential: it can now talk about individual nodes in Kripke models.

Let's be precise. A *model* for the basic hybrid language over σ, PROP, and NOM, is a Kripke model $\mathfrak{M} = (W, (R^{\triangle})_{\triangle\in\sigma}, V)$, such that the valuation V assigns *singleton* subsets of W to nominals; such valuations are sometimes called *hybrid valuations*. Apart from this restriction, everything is standard: W is a non-empty set of nodes, and for all $\triangle$ in σ, if $\triangle$ is an n-ary modality, then $R^{\triangle}$ is an $n+1$-ary relation. Following standard terminology we call the pair $(W, (R^{\triangle})_{\triangle\in\sigma})$ the *frame* underlying the model.

Given such a hybrid model $\mathfrak{M}$, we interpret our language as follows:

$$\mathfrak{M}, w \models a \text{ iff } w \in V(a), \text{ where } a \in \text{PROP} \cup \text{NOM}$$
$$\mathfrak{M}, w \models \neg\phi \text{ iff } \mathfrak{M}, w \not\models \phi$$
$$\mathfrak{M}, w \models \phi \wedge \psi \text{ iff } \mathfrak{M}, w \models \phi \text{ and } \mathfrak{M}, w \models \psi$$
$$\mathfrak{M}, w \models \triangle(\phi_1, \ldots, \phi_n) \text{ iff there are } v_1, \ldots, v_n \in W \text{ such that}$$
$$(w, v_1, \ldots, v_n) \in R^{\triangle} \text{ and } \mathfrak{M}, v_1 \models \phi_1 \ldots \mathfrak{M}, v_n \models \phi_n$$
$$\mathfrak{M}, w \models @_i\phi \text{ iff } \mathfrak{M}, v \models \phi, \text{ where } V(i) = \{v\}$$

Readers unfamiliar with arbitrary arity modalities should note that in the unary case the clause for modalities simplifies down to the more familiar:

$$\mathfrak{M}, w \models \Diamond\phi \text{ iff there is a } v \in W \text{ such that } (w, v) \in R^{\Diamond} \text{ and } \mathfrak{M}, v \models \phi.$$

If $\mathfrak{M}, w \models \phi$ then we say that ϕ is *satisfied* in $\mathfrak{M}$ at w. For any frame $\mathfrak{F}$, if ϕ is satisfied in every model $(\mathfrak{F}, V)$ at every w in $\mathfrak{F}$ no matter which (hybrid) valuation V we choose, then we say that ϕ is *valid on* $\mathfrak{F}$. A formula is *valid* if it is valid on every frame. A formula is *valid on a class of frames* F if it is valid on every frame in F.

As promised, the hybridized language is referential. That nominals name is hardwired into the definition of valuations, and the clause for $@_i\phi$ says "evaluate ϕ at the node that i names". Notice that $@_ij$ says that the nominals i and j name the same node, that $@_i\Diamond j$ means that the node named i has the node named j as an $R^{\Diamond}$-successor, and that $@_i \triangle (j_1, \ldots, j_n)$ means that the $n+1$ nodes named i, j_1,...,j_n, stand in the $R^{\triangle}$ relation.

It's worth mentioning that the language we have just defined is a very simple hybrid language: far stronger hybrid languages have been studied, for example languages in which it is possible to bind nominals using the classical quantifiers $\forall$ and $\exists$ (see, for example, [9,13,8]). But the syntactic simplicity of the basic hybrid language is attractive. Moreover, its syntactic simplicity pays off in terms of computational simplicity: the satisfaction problem of the basic hybrid language (over the class of all models) is decidable in PSPACE (this is proved for the unary case in [1]; the proof extends to the arbitrary arity case). That is, the satisfaction problem for the hybridized language is (up to a polynomial) no more complex than the satisfaction problem for the underlying modal language.

2 Hybrid Tableaux

Hybridization gives us precisely the tools needed to define natural proof systems. This is because the basic hybrid apparatus of @-operators and nominals allows us to reason about what happens at particular nodes, and to extract information from under the scope of the modalities.

In Table 1 we give a sound and complete tableau calculus for basic hybrid logic (the calculus is an arbitrary arity modality version of the calculus of [3] that incorporates improvements to the equality rules introduced in [7]). To prove a formula ϕ, proceed as follows: chose a nominal, say i, that is not in ϕ, and start applying tableaux rules to $\neg @_i \phi$. That is, assert that it is possible to falsify ϕ in some model at a node named i, and use the tableau rules to try and build a falsifying model. If it turns out that this is *not* possible (that is, if the tableau closes) we have proved ϕ.

It is clear from Table 1 that it is the hybrid machinery that propels this calculus. For a start, all the rules are @-driven: we use @ to reason about what must hold at named nodes. And, given the semantics of the @ operator, the import of most of these rules should be clear. For example, the $\neg$ and $\wedge$ rules dismantle the Boolean connectives in the obvious way, and the @ rule lets us drop outermost occurrences of @. The first equality rule says that for any nominal i on a branch we can conclude that i is true at the node named i, which is obviously true. The second equality rule says that from $@_i j$ ("the nominals i and j name the same node") and $@_i \phi$ ("ϕ is true at the node named i") we can conclude $@_j \phi$ ("ϕ is true at the node named j").

But the hybrid apparatus has a second, deeper, role to play. This becomes apparent when we consider the $\triangle$ rules, and in particular the left-hand rule for $\triangle$. It is probably easier to see what is going on in the unary case: Table 2 displays the unary form of all three rules containing occurrences of $\triangle$ (that is, the two $\triangle$ rules and the third equality rule).

It's the left-hand $\Diamond$ rule that is crucial. We are given the assertion $@_i \Diamond \phi$ ("at the node named i, $\Diamond \phi$ is true"). From this we conclude two things: $@_i \Diamond k$ ("the node named i has at least one successor, which we shall call k") and $@_k \phi$ ("at the node named k, ϕ holds"). That is, we have used the nominal k

Table 1
Tableau calculus for the basic hybrid logic

$\neg$	$\dfrac{@_i\neg\phi}{\neg @_i\phi}$	$\dfrac{\neg @_i\neg\phi}{@_i\phi}$
$\wedge$	$\dfrac{@_i(\phi\wedge\psi)}{\begin{array}{c}@_i\phi\\ @_i\psi\end{array}}$	$\dfrac{\neg @_i(\phi\wedge\psi)}{\neg @_i\phi \mid \neg @_i\psi}$
$\triangle$	$\dfrac{@_i \triangle (\phi_1,\ldots,\phi_n)}{\begin{array}{c}@_i \triangle (k_1,\ldots,k_n)\\ @_{k_1}\phi_1\\ \vdots\\ @_{k_n}\phi_n\end{array}}$	$\dfrac{\neg @_i \triangle (\phi_1,\ldots,\phi_n) \quad @_i \triangle (k_1,\ldots,k_n)}{\neg @_{k_1}\phi_1 \mid \ldots \mid \neg @_{k_n}\phi_n}$
$@$	$\dfrac{@_i @_j\phi}{@_j\phi}$	$\dfrac{\neg @_i @_j\phi}{\neg @_j\phi}$
Equality	$\dfrac{(i \text{ occurs on branch})}{@_i i}$	$\dfrac{@_i j \quad @_i\phi}{@_j\phi}$
	$\dfrac{@_i \triangle (k_1,\ldots,k_n) \quad @_{k_1}k'_1 \quad \ldots \quad @_{k_n}k'_n}{@_i \triangle (k'_1,\ldots,k'_n)}$	
Closure	A branch is closed if it contains a formula and its negation. A tableau is closed if all of its branches are closed.	

to decompose the the original information into two simpler parts. Where does the nominal k come from? Nowhere: it's brand new. That is, it is a nominal that hasn't previously occurred on the tableau branch. It is a newly created name for the ϕ-witnessing R-successor node to i that must exist if the original assertion is true.

The general form this rule takes in the left-hand $\triangle$ rule should now be clear: distinct new nominal k_1, …, k_n are introduced to extract information from under the scope of the n-place modality. Incidentally, the convention that nominals only occurring below the bar of a tableau rule are new (and syntactically distinct) is a convention we shall use in Sections 4 and 5 of the

Table 2
Unary form of the rules containing $\triangle$

$\Diamond$	$\dfrac{@_i\Diamond\phi}{\begin{array}{c}@_i\Diamond k\\ @_k\phi\end{array}}$ $\qquad$ $\dfrac{\neg @_i\Diamond\phi \quad @_i\Diamond k}{\neg @_k\phi_1}$
Equality	$\dfrac{@_i\Diamond k \quad @_k k'}{@_i\Diamond k'}$

paper when we discuss node creating rules.

We won't discuss the right-hand $\triangle$ rule, nor the third equality rule; the unary case should make clear what they do. Neither rule introduces new nominals.

3 Adding Pure Axioms

If the first benefit of hybridization is that it is straightforward to define natural proof systems, the second is this: once a base system has been defined, it is easy to extend it to a complete system for many important classes of frames. All we have to do is add *pure axioms*.

A *pure formula* is a formula that contains no propositional letters. Many properties of frames can be defined using pure formulas. For example, transitivity of R is defined by

$$@_i(\Diamond\Diamond j \rightarrow \Diamond j).$$

(This formula is valid on every transitive frame and falsifiable on every non-transitive frame.) Similarly, irreflexivity of R is defined by

$$@_i\neg\Diamond i,$$

and trichotomy of R (that is, $\forall xy(xRy \vee x = y \vee yRx)$ is definable by

$$@_i\Diamond j \vee @_i j \vee @_j\Diamond i.$$

As these examples show, the frame-defining powers of pure formulas are different from those of orthodox modal formulas (that is, formulas built out of ordinary proposition letters): transitivity is definable using an orthodox modal formula, but irreflexivity and trichotomy are not.

But for present purposes the key fact about pure formulas is this: when used as axioms (over a suitable base proof system) they are guaranteed to be complete with respect to the class of frames they define (for a detailed proof, see Chapter 7.3 of [6]). For example, adding the pure axiom for transitivity given above to a (suitable) base proof system yields a system complete for transitive frames, and adding the pure axiom for irreflexivity yields a system complete for irreflexive frames. These results are cumulative: adding both axioms yields a system complete for strict partial orders. Many suitable base proof systems are known: for example, see Chapter 7.3 of [6] and [5] for Hilbert-style approaches, and Seligman [15] for sequent calculi. And, once a small technical point has been observed, the tableau calculus just presented is suitable too.

The small technical point is this: the tableaux rules only take as input (and produce as output) @-prefixed formulas. This means that the pure axioms used with tableaux must be @-prefixed too, but some natural axioms (such as the trichotomy defining formula given above) are not of this form. However this is only an apparent restriction: if a pure formula ϕ defines a class of frames F, then $@_k\phi$ defines F as well, where k is any nominal not occurring in ϕ. Thus any pure axiom can be put a form suitable for tableau processing simply by prefixing a new nominal. For example

$$@_k(@_i\Diamond j \vee @_i j \vee @_j\Diamond i)$$

is a suitable tableaux axiom for trichotomy. With this observed, we have the following completeness result:

Theorem 3.1 *Let A be a finite collection of @-prefixed pure formulas. Let TA be the tableau system given above extended with the following rule: at any stage in the tableau, we are free to choose a formula from A, instantiate it with nominals occurring on some branch B, and then add the result to the end of branch B. Then TA is complete with respect to the class of frames defined by $\bigwedge_{\phi\in A}\phi$.*

Proof. The result for unary modalities is proved in [3]. The extension to arbitrary arity modalities is routine. □

Note the restriction on instantiations: we only introduce instances of the axioms which make use of 'old' nominals. In fact, this is the restriction we shall remove when we introduce node creating rules in the following section. But before doing this, let's look at the theorem just stated, and indeed at the whole idea of frame-definability using pure formulas, from a rather different perspective.

As we mentioned earlier, there are far stronger hybrid languages than the basic hybrid language, including hybrid languages in which we can bind nominals with the classical quantifiers $\forall$ and $\exists$. In what follows we call the extension of a basic hybrid language with such quantifiers a *strong hybrid language*. Precise definitions of the syntax and semantics of strong hybrid languages can be found in [8], but the reader will probably find the examples given below of these languages in action clear enough.

Strong hybrid languages give us full first-order expressive power over frames (this has been known since Arthur Prior's pioneering work on hybrid languages in the late 1960s, such as some of the papers in [14]; for a more recent discussion, see [8]). This means that strong hybrid languages are vastly more powerful than the basic languages of the present paper, and have undecidable satisfaction problems. But it spite of these differences, strong languages throw useful light on what is going on when we use pure axioms.

As we have already remarked, the following pure formula of the basic hybrid defines transitivity:

$$@_i(\Diamond\Diamond j \to \Diamond j).$$

Using a strong hybrid language, this can be re-expressed as follows:

$$\forall x \forall y @_x(\Diamond\Diamond y \to \Diamond y).$$

That is, we have substituted variables for nominals, and taken the universal closure. It should be clear that the basic hybrid formula we started with is *valid* on a frame $\mathfrak{F}$ iff the strong sentence is *true* on the frame $\mathfrak{F}$: the $\forall$ quantifier captures the effect of trying out all possible assignments to the nominals/variables. To give another example, the basic formula

$$@_i \neg \Diamond i$$

which defines irreflexivity can also be re-expressed as

$$\forall x @_x \neg \Diamond x.$$

Indeed, the process is completely general. Call a sentence of a strong hybrid language of the form $\forall x_1 \dots x_n \phi$, where ϕ does not contain any quantifiers, propositional letters, or nominals, a PUNF-sentence (this stands for *pure, universal, nominal-free sentence*). The previous examples show that transitivity and irreflexivity can be expressed by PUNF-sentences, and in fact we have the following:

Proposition 3.2

- *For any pure formula ϕ of the basic hybrid language there is a* PUNF*-sentence $\phi^\forall$ of the strong hybrid language such that for any frame $\mathfrak{F}$, ϕ is valid on $\mathfrak{F}$ iff $\phi^\forall$ is true on $\mathfrak{F}$.*
- *For any* PUNF*-sentence ϕ of the strong hybrid language there is a pure formula ϕ^B of the basic hybrid language such that for any frame $\mathfrak{F}$, ϕ is true in $\mathfrak{F}$ iff ϕ^B is valid on $\mathfrak{F}$.*

Proof. Item 1 is a generalization of the above examples: suppose we are given a pure formula $\phi(i_1, \dots, i_n)$, where $i_i, \dots, i_n$ are all the nominals in ϕ, then the required $\phi^\forall$ is $\forall x_1 \cdots \forall x_n \phi([i_1 \leftarrow x_1, \dots, i_n \leftarrow x_n])$.

As for item 2, any PUNF-sentence ϕ of the strong hybrid language must have the form $\forall x_1 \cdots \forall x_n \phi(x_1, \dots, x_n)$ where ϕ contains no quantifiers, proposition letters, or nominals. Hence the required ϕ^B is $\phi([x_1 \leftarrow i_1, \dots, x_n \leftarrow i_n])$. □

This gives us a new perspective on pure axioms. For a start it tells us that the frame classes expressible by pure axioms are simply the frame classes definable by PUNF-sentences. Moreover, it gives us an alternative way of thinking about Theorem 3.1:

Theorem 3.3 *Let S be a finite collection of* PUNF*-sentences, all of which have an @-prefixed matrix. Let TS be the tableau system given above extended with the following rule: at any stage in the tableau, we are free to choose a formula from S, perform universal instantiation on it using nominals present on branch B, and add the resulting basic hybrid formula to the end of branch B. Then TS is complete with respect to the class of frames defined by $\bigwedge_{\phi \in S} \phi$.*

Proof. Simply a reformulation of Theorem 3.1. It's worth stressing that we are only using the strong hybrid language in the background, as a way of recording extra axioms. The formulas actually used on the tableau are, just as before, basic formulas: in concrete terms this reformulation changes nothing.

We insist that the matrices of the PUNF-sentences be @-prefixed to ensure that the result of universal instantiation can be processed by the tableau. As we have remarked, this restriction does not decrease the frame-defining powers at our disposal. □

While the perspectival shift from basic to strong hybrid languages is not particularly deep, it will be useful. We are about to introduce node creating rules, which will give us complete tableau for many more frame classes. Which classes? As we shall eventually see, any frame class definable by a strong sentence of the form $\forall x_1 \ldots x_n \exists y_1 \ldots y_m \phi$ (where ϕ does not contain any quantifiers, propositional letters, or nominals). In essence, we are going to show how to move from the universal fragment of the strong hybrid language, to the universal-existential fragment.

4 Node Creating Rules for Geach Formulas

Many important frame classes cannot be captured using pure axioms. One example is the class of Church-Rosser frames, that is, frames satisfying the property

$$\forall wvu \exists t(wRv \wedge wRu \rightarrow vRt \wedge uRt).$$

Another is the class of *right-directed* frames, that is, frames satisfying the (stronger) property

$$\forall vu \exists t(vRt \wedge uRt).$$

We shall now show how to capture the logic of such frame classes in the setting of a tableau calculus. We do so in two steps. In this section we discuss the Church-Rosser property, and go on to define tableau rules capable of handling any class of frames definable by a Geach formula. In the following section we discuss right-directedness, and go on to show that we can handle any condition expressible in the (pure, nominal-free) universal-existential fragment of the strong hybrid language.

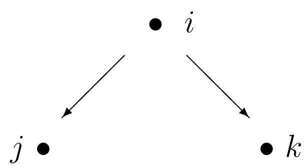
A little experimentation will quickly convince the reader that there is no pure axiom defining Church-Rosser: there is simply no way to get a handle on the convergence node (the t in the above first-order definition). However if we think more dynamically, and in particular, if we think in terms of tableau rules that create new nodes, we see that there is a very natural way of getting exactly what we want:

$$\frac{@_i\Diamond j \quad @_i\Diamond k}{\begin{array}{c} @_j\Diamond l \\ @_k\Diamond l \end{array}}$$

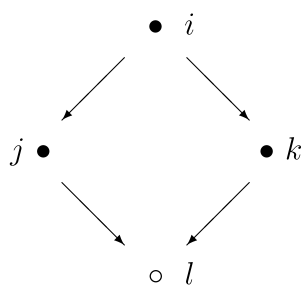
Remember our notational convention for tableau expansion rules: any nominal that occurs only below the bar is newly introduced. Therefore the above expansion rule should be read as follows:

> *Whenever $@_i\Diamond j$ and $@_i\Diamond k$ occur on a branch, we are allowed to create a new nominal l and add $@_j\Diamond l$ and $@_k\Diamond l$ to the branch.*

That is, suppose that we already have the following configuration of nodes:



Then we are allowed to introduce a brand new nominal l that acts as a name for the required convergence node:



Two remarks. First, this rule amounts to "using nominals as skolem constants": in effect we are using nominals to eliminate the existential quantifier $\exists t$ in the first-order definition given above. Second, readers familiar with completeness proofs for hybrid tableau will realize that this rule is not only sound, but complete as well. For by systematically applying this rule, we saturate any branch of a tableau with convergence points. And this guarantees that any countermodel to the input that we find will be a countermodel based on a frame with the Church-Rosser property.

Does this strategy generalise to other frame properties? In fact, it generalises straightforwardly to all frame properties that can be characterised by *Geach formulas* (see [12]). These are orthodox modal axioms of the following form (where $m, n, s, t \geq 0$, and p is an ordinary propositional letter):

$$\Diamond^m \Box^n p \to \Box^s \Diamond^t p.$$

Geach axioms cover many well-known properties, among which are transitivity ($m = 2, t = 1, n = s = 0$) and Church-Rosser ($m = n = s = t = 1$).

Now, every Geach axiom corresponds to a first-order frame property, namely the following:

$$\forall xyz \exists u(xR^m y \wedge xR^s z \to yR^n u \wedge zR^t u).$$

Here, as is customary, we have used R^n as a shorthand for a sequence of n R-transitions. In other words, the above formula is shorthand for the following formula:

$$\begin{aligned} &\forall xyza_1 \dots a_{m-1} b_1 \dots b_{s-1} \exists u c_1 \dots c_{n-1} d_1 \dots d_{t-1} \\ &\quad ((xRa_1 \wedge \dots \wedge a_{m-1}Ry) \wedge (xRb_1 \wedge \dots \wedge b_{s-1}Rz) \to \\ &\qquad (yRc_1 \wedge \dots \wedge c_{n-1}Ru) \wedge (zRd_1 \wedge \dots \wedge d_{t-1}Ru)). \end{aligned}$$

And now it is easy to define the required tableaux rules. All we need to do is 'walk along' this formula from left to right, writing down the transition relations in hybrid notation. If we do this (using i instead of x, j instead of y, k instead of z, and l instead of u) we obtain the following tableaux expansion rule:

$$\frac{@_i \Diamond a_1 \quad \dots \quad @_{a_{m-1}} \Diamond j \qquad\quad @_i \Diamond b_1 \quad \dots \quad @_{b_{s-1}} \Diamond k}{\begin{array}{c} @_j \Diamond c_1 \\ \vdots \\ @_{c_{n-1}} \Diamond l \\ @_j \Diamond d_1 \\ \vdots \\ @_{d_{t-1}} \Diamond l \end{array}}$$

Let's look at a couple of examples. If we apply this schema to the Church-Rosser property (that is, if we instantiate m, n, s and t to 1), then this gives us exactly the expansion rule we have just discussed. It's also interesting to see the rule for transitivity produced by this schema:

$$\frac{@_i \Diamond j \quad @_j \Diamond k}{@_i \Diamond k}$$

Now, this rule does not create new nodes (after all, there's no existential quantifier in the definition of transitivity to be skolemized away). Moreover, as transitivity can be characterised by a pure formula (namely $@_k(\Diamond\Diamond i \rightarrow \Diamond i)$), it doesn't offer us anything new as far as frame classes are concerned. Nonetheless, from a computational perspective, this rule is better than the rule for transitivity of the previous section, which simply dumps the defining pure formula on the tableau branch. The new rule, so to speak, sits and waits for the antecedent to be fulfilled, and then fires to build the required R-transition between i and k.

Two remarks. First, our approach to the Geach rules is similar to work by Basin, Matthews and Viganò [2] in the labelled deduction tradition. These authors don't work with hybrid logic, rather they work with an orthodox modal language and a labelling algebra interacting through a fixed interface. Nonetheless, their use of skolemization has close affinities with the rules just discussed, and their approach can handle all Geach conditions. Second, we remark that the schema for Geach formulas could be generalised to languages containing several distinct unary modalities: all that needs to be done is to add appropriate indices to the diamonds in the above rule (to indicate which binary relations interpret the various diamonds). But we shall leave this to the reader, for a broader generalization awaits us.

5 Node Creating Rules and the Universal-Existential Fragment

We now have a way of obtaining complete tableaux systems for frame properties such as the Church-Rosser property. However, the problem with the *right-directedness* property has still not been solved, since right-directedness is not equivalent to a Geach formula. In this section we will show how to handle this condition, and then go on to show that we can handle any condition definable in the (pure, nominal-free) universal-existential fragment of the strong hybrid language.

Right-directness is not definable by a Geach formula (in fact, no orthodox modal formula whatsoever defines this condition, as the class of right-directed frames is not closed under disjoint-union). There *is* a formula of the basic hybrid language that defines the class of right-directed frames, namely $@_i\Box p \rightarrow @_j\Diamond p$. Unfortunately, as this is not a pure formula, we cannot apply Theorem 3.1 to obtain completeness automatically. However, a little lateral thinking shows that it is possible to handle this frame condition naturally, using a more general kind of node creating rule.

First observe that right-directness can be defined by a very simple sentence of the *strong* hybrid language, namely

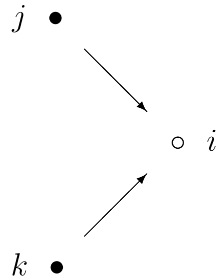
$$\forall xy\exists z(@_x\Diamond z \wedge @_y\Diamond z).$$

And now the crucial point: this strong formula provides a 'recipe' for what we should do in a tableau proof, namely this: instantiate the universally bound variables with old nominals, and the existentially bound variables with new nominals. In short, it suggests the following tableau rule:

$$\frac{(j,\, k \text{ occur on the branch})}{@_j\Diamond i \wedge @_k\Diamond i}$$

This rule provides exactly what we want. It says: suppose we have two nodes named j and k:

Then we are free to create a new node, glue it in place to the right of these nodes, and give it a name, say i.



As in the Church-Rosser example of the previous section, this new rule essentially boils down to "skolemization using nominals"; the difference is we are now thinking directly in terms of skolemizing *strong* hybrid formulas.

Of course, in one sense we have taken a step backwards. This rule is not as nice as those for Geach conditions: we are back to simply placing instances of a complex formula on the branch of a tableaux. But we gain something important in return: generality.

Call a sentence of a strong hybrid language of the form

$$\forall x_1, \cdots x_n \exists y_1 \cdots y_m \phi$$

where ϕ does not contain any quantifiers, propositional letters, or nominals, a PUENF-sentence (this stands for pure, *universal existential*, nominal-free sentence). Then for any such sentence we have the following tableaux rule:

$$\frac{(i_1, \ldots, i_n \text{ occur on the branch})}{\phi[x_1 \leftarrow i_1, \ldots, x_n \leftarrow i_n, y_1 \leftarrow k_1, \ldots, y_n \leftarrow k_m]}$$

Here we make use of the convention mentioned in Section 2: the $k_1, \ldots, k_m$ are *new, distinct*, nominals. And now we have:

Theorem 5.1 *Let S be a finite collection of* PUENF*-sentences, all of which have an @-prefixed matrix. Let TS be the tableau system given above extended* *with the following rule: at any stage in the tableau, we are free to choose a formula from S, perform the tableaux rule associated with this formula using nominals present on branch B together with the needed new nominals; we add the basic hybrid formula so obtained to the end of branch B. Then TS is complete with respect to the class of frames defined by $\bigwedge_{\phi \in S} \phi$.*

Proof. A proof is given in the full version of the paper. It uses the same technique that was used in [3] to prove Theorem 3.1 (or equivalently, Theorem 3.3). The fact that we are now instantiating PUENF-sentences rather that PUNF-sentences does not change the heart of the argument. □

Notice that, in contrast to our previous result concerning Geach formulas, Theorem 5.1 applies to modalities of arbitrary arity.

6 Conclusion

A fundamental result of hybrid logic is that it is possible to define basic proof calculi in such a way that any extension with pure axioms is automatically complete. While general, this result has limitations: it does not cover all Geach formulas, let alone conditions like right-directedness. In this paper we have shown that it straightforward to overcome these limitations. The key is to make use of node creating rules. These enable us to provide complete proof systems in the basic hybrid language that cover any frame condition definable by a PUENF-sentence. To conclude the paper we briefly note another perspective on node creating rules, and make a conjecture.

Node creating rules are not restricted to tableaux proof systems: they make perfectly good sense in Hilbert-style proof systems, where they become special rules of proof. For example, when working with a Hilbert system, the Church-Rosser tableau expansion rule takes the following form:

$$\frac{\vdash (@_i\Diamond j \wedge @_i\Diamond k \rightarrow @_j\Diamond l \wedge @_k\Diamond l) \rightarrow \theta \text{ and } l \notin \theta \text{ and } l \neq i, j, k}{\vdash \theta}$$

Likewise, the right-directedness expansion rule becomes the following Hilbert-style proof rule:

$$\frac{\vdash (@_j\Diamond i \wedge @_k\Diamond i) \rightarrow \theta \text{ and } i \notin \theta \text{ and } i \neq j, k}{\vdash \theta}$$

And in general, given any PUENF-sentence $\forall x_1, \cdots x_n \exists y_1 \cdots y_m \phi \in A$ the corresponding Hilbert-style rule is:

$$\frac{\vdash \phi[x_1 \leftarrow i_1, \ldots, x_n \leftarrow i_n, y_1 \leftarrow k_1, \ldots, y_m \leftarrow k_m] \rightarrow \theta}{\vdash \theta}$$

where $k_1, \ldots, k_m$ are distinct, unequal to $i_1, \ldots i_n$, and don't occur in θ.

Two remarks. First, proving completeness in the Hilbert system case is rather similar to the proof in the tableaux case. The crucial work takes place in an extended version of the Lindenbaum Lemma (that is the step in the completeness proof where a consistent set is transformed into a maximal consistent set; see, for example, Lemma 7.25 of [6]). Essentially we add the required new nominals when performing the inductive Lindenbaum construction; the new rules guarantee the consistency of these additions. Full details are presented in [5].

Second, the fact that node creating rules make sense in the Hilbert setting should not be a surprise, since they are essentially something familiar to modal logicians: *rules for the undefinable.* Gabbay [10] introduced the *irreflexivity rule*, an additional rule of proof for orthodox modal language that makes it possible to directly construct irreflexive models, and Venema [16] presents a far-reaching generalization to logics containing the difference operator. This is not the place to make detailed comparisons, but we make two remarks. First, because the notion of reference to nodes is primitive in hybrid logic, the hybrid rules are arguably more natural (though this is partly a matter of taste). Second, in the hybrid logical case, there is a particularly simple answer to the question "But where do such rules come from?": they reflect the skolemization possibilities offered by PUENF-formulas.

We close with a conjecture. In one sense, we know how many frame classes are covered by this result, namely precisely the frame classes definable by PUENF-formulas. But obviously it would be nice to back this up with a characterization in terms of traditional correspondence theory. We have such a result: in [4] we show that the frame conditions by PUENF-formulas are exactly the universal-existential closures of *strongly bounded* first-order formulas. This is a large class of formulas, and we conjecture it covers every Sahlqvist-definable frame class. If this could be proved it would nicely supplement the result of Goranko and Vakarelov [11], which states that in *reversive* hybrid languages (that is, languages where the set of modalities is closed under converses) every Sahlqvist formula is equivalent to a pure formula. Incidentally, note that the sets of PUENF-definable and Sahlqvist-definable frame conditions cannot be identical, since many PUENF-definable (indeed PUNF-definable) conditions such as irreflexivity, antisymmetry, and discreteness are not definable by any orthodox modal formula at all.

References

[1] C. Areces, P. Blackburn, and M. Marx. A roadmap on the complexity of hybrid logics. In J. Flum and M. Rodríguez-Artalejo, editors, *Computer Science Logic*, number 1683 in LNCS, pages 307–321. Springer, 1999. Proceedings of the 8th Annual Conference of the EACSL, Madrid, September 1999.

[2] David Basin, Seán Matthews, and Luca Viganò. Labelled propositional modal logics: Theory and practice. *Journal of Logic and Computation*, 7(6):685–717, 1997.

[3] P. Blackburn. Internalizing labelled deduction. *Journal of Logic and Computation*, 10:137–168, 2000.

[4] P. Blackburn and B. ten Cate. Correspondence theory for hybrid logic, 2002. Manuscript, INRIA Lorraine and ILLC, University of Amsterdam.

[5] P. Blackburn and B. ten Cate. Hybrid axiomatics: the propositional case, 2002. Manuscript, INRIA Lorraine and ILLC, University of Amsterdam.

[6] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[7] P. Blackburn and M. Marx. Tableaux for quantified hybrid logic. In C. Areces and M. de Rijke, editors, *Methods for Modalities 2, Workshop Proceedings, November 29-30, 2001*. ILLC Amsterdam, 2001.

[8] P. Blackburn and J. Seligman. Hybrid languages. *Journal of Logic, Language and Information*, 4(3):251–272, 1995. Special issue on decompositions of first-order logic.

[9] R. Bull. An approach to tense logic. *Theoria*, 36:282–300, 1970.

[10] D.M. Gabbay. An irreflexivity lemma with applications to axiomatizations of conditions on linear frames. In U. Mönnich, editor, *Aspects of Philosophical Logic*, pages 67–89. Reidel, Dordrecht, 1981.

[11] Valentin Goranko and Dmiter Vakarelov. Sahlqvist formulas in hybrid polyadic modal logics. *Journal of Logic and Computation*, 11(5):737–754, 2001. Special issue on hybrid logic.

[12] G. Hughes and M.J. Cresswell. *A New Introduction to Modal Logic*. Routledge, 1996.

[13] S. Passy and T. Tinchev. An essay in combinatory dynamic logic. *Information and Computation*, 93:263–332, 1991.

[14] A.N. Prior. *Papers on Time and Tense*. University of Oxford Press, 1968.

[15] J. Seligman. Internalization: The case of hybrid logics. *Journal of Logic and Computation*, 11(5):671–689, 2001. Special issue on hybrid logic.

[16] Y. Venema. Derivation rules as anti-axioms in modal logic. *Journal of Symbolic Logic*, 58:1003–1034, 1993.