Synthetic Completeness for a Terminating Seligman-Style Tableau System

Asta Halkjær From
Technical University of Denmark, Kongens Lyngby, Denmark

Abstract

Hybrid logic extends modal logic with nominals that name worlds. Seligman-style tableau systems for hybrid logic divide branches into blocks named by nominals to achieve a local proof style. We present a Seligman-style tableau system with a formalization in the proof assistant Isabelle/HOL. Our system refines an existing system to simplify formalization and we claim termination from this relationship. Existing completeness proofs that account for termination are either analytic or based on translation, but synthetic proofs have been shown to generalize to richer logics and languages. Our main result is the first synthetic completeness proof for a terminating hybrid logic tableau system. It is also the first formalized completeness proof for any hybrid logic proof system.

2012 ACM Subject Classification Theory of computation → Modal and temporal logics

Keywords and phrases Hybrid logic, Seligman-style tableau, synthetic completeness, Isabelle/HOL

Digital Object Identifier 10.4230/LIPIcs.TYPES.2020.5

Supplementary Material *Model (Isabelle/HOL formalization in the Archive of Formal Proofs (4900+ lines))*: `https://isa-afp.org/entries/Hybrid_Logic.html`
archived at `swh:1:cnt:5a830ce17c70be797b343d9078bf19c43b0f2145`

Acknowledgements We thank Patrick Blackburn, Thomas Bolander, Torben Braüner, Klaus Frovin Jørgensen and Jørgen Villadsen for discussions and Lars Hupel, Jasmin Blanchette and the anonymous reviewers for comments on the paper.

1 Introduction

Hybrid logic increases the expressiveness of modal logic by adding a special sort of propositional symbol called *nominals* to the syntax. In regular modal logic we can only reference worlds indirectly through the modalities, but nominals, that are true at exactly one world, name worlds explicitly. A nominal i gives rise to the satisfaction operator $@_i$ that states what world a formula is true "at." These features make hybrid logic well suited for applications like temporal logic [3], description logic [5] and epistemic logics for social networks [24].

There are many proof systems for classical hybrid logic [4] and we focus on tableau systems in the following. Early work relied on loop checks to ensure termination [10] but Bolander and Blackburn introduced a calculus that guarantees finite branches through local restrictions [9]. Their completeness proof is *analytic*, meaning that they reason about open branches directly. Blackburn et al. [4] introduced the Seligman-style [25] system ST with a more local proof style than previous systems. Jørgensen et al. [21] later introduced a *synthetic* completeness proof for ST and showed that it scales with extensions to the logic. The synthetic approach involves reasoning about maximal consistent sets and their properties [13, 26] and this also opens the way for other developments, notably interpolation results [1].

Blackburn et al. [4] restricted ST into the terminating ST^* but showed completeness by translation from the system by Bolander and Blackburn [9]. The synthetic completeness proof for ST relies on a symmetry in branches that neither terminating system has. We present system ST^{A}, a refinement of ST^* suitable for formalization, which is formalized in the simple type theory of Isabelle/HOL [23]. Its proof of completeness fills a gap as the first synthetic completeness proof for a terminating tableau system for hybrid logic. It is also the first standalone completeness proof for a terminating Seligman-style system and, to our knowledge, the first formalization of any proof system for hybrid logic.

The formalization provides absolute trust in the correctness of the completeness proof and serves as a companion to this paper, where the proofs can be seen in full detail.

Our system closely resembles ST^* but with restrictions that are simpler to formalize and we argue for termination based on this relationship. Formalizing termination remains future work since we want a direct proof, not one based on translation. Blanchette [6] gives an overview of efforts to formalize the metatheory of logical calculi and provers in Isabelle.

Other formalizations of hybrid logic itself exist. Doczkal and Smolka [12] formalized hybrid logic with nominals in constructive type theory using the proof assistant Coq. They gave algorithmic proofs of small model theorems and computational decidability of satisfiability, validity, and equivalence of formulas. In Isabelle/HOL, Linker [22] formalized the semantic embedding of a spatio-temporal multi-modal logic with a hybrid logic-inspired *at*-operator.

Our work is classical but hybrid logic also has a constructive variant. Braüner and de Paiva [11] defined intuitionistic hybrid logic, and a natural deduction system, and Galmiche and Salhi [19] showed its decidability via a sequent calculus. Jia and Walker [20] interpreted modal proofs as distributed programs with nominals denoting places in the network.

We formalized the synthetic completeness of ST with some of the simpler ST^* restrictions required for termination in our MSc thesis [17]. A short paper by From et al. [14] briefly described an even earlier version of the formalization and we mentioned the present completeness proof in a short presentation at Advances in Modal Logic 2020 [18].

The paper continues as follows. First, we give the syntax and semantics of basic hybrid logic (Section 2). We introduce the proof system, corresponding rule restrictions and some consequences (Section 3). Next, we show a number of properties of the system that are useful for the completeness proof (Section 4). After that, we prove completeness of the system and show how our proof relates to existing work (Section 5). We then show how ST^{A} relates to ST^* and argue for our choice of restrictions. From this relationship we claim that ST^{A} must be terminating by sketching a possible translation (Section 6). We briefly discuss some points about the formalization (Section 7) and conclude with future work (Section 8).

2 Syntax and Semantics

The well-formed formulas of the basic hybrid logic are given by the following grammar, where we use p as a propositional symbol and i, j, k, a, b for nominals.

$$\phi, \psi ::= p \mid i \mid \neg\phi \mid \phi \vee \psi \mid \Diamond\phi \mid @_i\phi$$

The $\Diamond$ operator is the usual possibility modality and $@_i$ is the aforementioned *satisfaction operator*. A formula of the form $@_i\phi$ is called a *satisfaction statement*.

We interpret the language on Kripke models $\mathfrak{M} = (W, R, V)$. The frame (W, R) consists of a non-empty set of worlds W and a binary accessibility relation R between them. V is the valuation of propositional symbols. An *assignment* g maps nominals to elements of W; if $g(i) = w$ we say that nominal i *denotes* w. Formula satisfiability is defined as follows:

$$\begin{array}{lll}
\mathfrak{M}, g, w \models p & \text{iff} & w \in V(p) \\
\mathfrak{M}, g, w \models i & \text{iff} & g(i) = w \\
\mathfrak{M}, g, w \models \neg\phi & \text{iff} & \mathfrak{M}, g, w \not\models \phi \\
\mathfrak{M}, g, w \models \phi \vee \psi & \text{iff} & \mathfrak{M}, g, w \models \phi \text{ or } \mathfrak{M}, g, w \models \psi \\
\mathfrak{M}, g, w \models \Diamond\phi & \text{iff} & \text{for some } w',\ wRw' \text{ and } \mathfrak{M}, g, w' \models \phi \\
\mathfrak{M}, g, w \models @_i\phi & \text{iff} & \mathfrak{M}, g, g(i) \models \phi
\end{array}$$

(a) Internalized. **(b)** Seligman-style.

Figure 1 Internalized and Seligman-style tableau branches.

3 Our Seligman-Style Tableau System

Our proof system of choice is tableau. In tableau we decompose an initial set of *root* formulas into a tree structure and show unsatisfiability by reaching a contradiction on each branch. This is called "closing" the branch and a branch that cannot be closed remains "open." If we can close every branch that emerges then the root formulas have a *closing tableau*.

A hybrid logic formula is true relative to a given world and our proof system must handle this. Internalized tableau systems, as depicted in Figure 1a, encode the information in every formula on the branch by working exclusively with satisfaction statements. We follow instead the Seligman style [25] adapted to tableau systems by Blackburn et al. [4]. Here, the information is attached to a group of formulas at once by dividing the branch into *blocks* as depicted in Figure 1b. The first formula on each block is ensured to be a nominal and called the *opening nominal*. It denotes the world that the formulas on the block are true at. We occasionally call a block's opening nominal its "type" and use the following shorthands:

▶ **Definition 1** (ϕ at i)**.** *If a formula ϕ occurs on a block with opening nominal i, then we say that ϕ occurs "on an i-block" or simply that ϕ occurs "at i."*

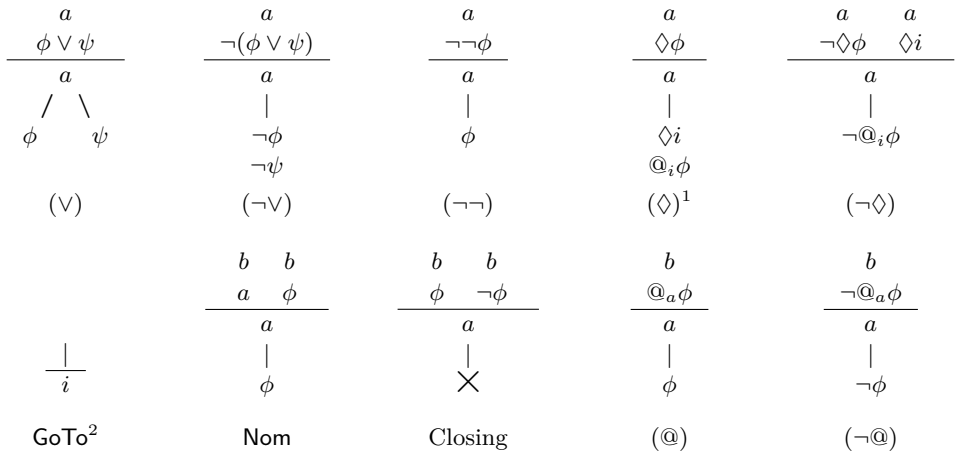
3.1 Proof System

Figure 2 gives our tableau rules. We give the rule output below the *vertical* lines and the rule input above them. The opening nominal of the latest, *current*, block is given below the horizontal line. Above each input formula we write the opening nominal of the block it occurs on. When a rule has multiple input we write these pairs side by side. Any formula on the current block may be used as input under the same restrictions on opening nominals.

▶ **Example 2.** Consider the ($\neg\neg$) rule: if $\neg\neg\phi$ occurs on an a-block and the current block is an a-block, then ϕ is a legal extension of the branch. The intuition for the Nom rule is that the current opening nominal a occurs on a b-block so nominals a and b must denote the same world and it is sound to copy ϕ from b to a. The ($\Diamond$) rule witnesses its input formula, $\Diamond\phi$, with a fresh *witnessing nominal* i by producing an accessibility formula, $\Diamond i$, and a satisfaction statement, $@_i\phi$, saying that ϕ holds at the reachable world denoted by i.

▶ Remark 3. In the internalized system, cf. Figure 1a, we may work on a formula prefixed by $@_i$ one moment and one prefixed by $@_k$ the next. The Seligman-style blocks give rise to a more local proof style by delegating this perspective switch, e.g. from i to k, to the GoTo rule that opens a new block with corresponding opening nominal.

The soundness proof for ST^{A} follows existing work [4, 14] (cf. the formalization).



[1] i is fresh and ϕ is not a nominal.
[2] i is not fresh.

Figure 2 Our Seligman-style tableau system ST^{A}.

3.2 Restrictions for Termination

Besides the side conditions, we need to impose the following four restrictions on the system to ensure that we eventually run out of applicable rules (inspired by Blackburn et al. [4]):

S1 The output of a non-GoTo rule must include a formula new to the current block type.
S2 The (◊) rule can only be applied to input $\Diamond\phi$ on an a-block if $\Diamond\phi$ is not already witnessed at a by formulas $\Diamond i$ and $@_i\phi$ for some *witnessing nominal* i.
S3 We associate *potential*, a natural number n, with each line in the tableau. GoTo must decrement the number, the other rules increment it and we may start from any amount.
S4 We parameterize the proof system by a fixed set of nominals A and impose the following:
  a. The nominal introduced by the (◊) rule is not in A.
  b. For any nominal i, Nom only applies to a formula $\phi = i$ or $\phi = \Diamond i$ when $i \in A$.

Restrictions S1 and S2 prevent us from applying the same rule to the same input repeatedly. We motivate restriction S3 by the following examples and restriction S4 in Section 3.3.

▶ **Example 4.** In Figure 3a we prove the validity of $\neg@_i\phi \vee @_i\phi$ by constructing a closing tableau for its negation. We start from potential 0 in the fourth column. Notice how regular rule applications build up potential that is then discharged to open a new block on line 5.

▶ **Example 5.** In Figure 3b we start from the unsatisfiable formula $@_i\neg i$ and potential n. Restriction S3 prevents infinite applications of GoTo and eventually forces us to make progress (or we might get stuck if no rules apply).

▶ Remark 6. The choice of a fresh opening nominal for the root block ensures that we do not close the branch because of an interplay between the formula itself and the opening nominal (imagine starting from $\neg i$ on a block with opening nominal i).

Given restrictions S3 and S4 we say that a branch has a closing tableau *with respect to* a set of allowed nominals A and potential n. We also introduce the following shorthand:

▶ **Definition 7** (Allowed ϕ)**.** *A formula ϕ is* allowed *by A if it meets condition S4b.*

0.	a		
1.	$\neg(\neg@_i\phi \vee @_i\phi)$		[0]
2.	$\neg\neg@_i\phi$	$(\neg\vee)$ 1	[1]
3.	$\neg@_i\phi$	$(\neg\vee)$ 1	[1]
4.	$@_i\phi$	$(\neg\neg)$ 2	[2]
5.	i	GoTo	[1]
6.	$\neg\phi$	$(\neg@)$ 3	[2]
7.	ϕ	$(@)$ 4	[3]
	$\times$		

(a) Building up potential.

0.	a		
1.	$@_i\neg i$		$[n]$
2.	i	GoTo	$[n-1]$
3.	i	GoTo	$[n-2]$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n+1.$	i	GoTo	[0]
$n+2.$	$\neg i$	$(@)$ 1	[1]
	$\times$		

(b) Running out of potential.

Figure 3 Two examples of potential.

3.3 Nominal Asymmetry

See Blackburn et al. [4] for why a restriction like S4 is needed. They conclude:

> We ... have to enforce some control on the "direction" we allow the copying of formulas, so that we can establish a decreasing length argument. It is OK to copy a formula true at a nominal i to a nominal j if j generated i, but not if i generated j [4].

Essentially, we need to ensure that blocks of generated nominals contain strictly smaller formulas, so that any chain of them eventually terminates. It is the $(\Diamond)$ rule that *generates* a fresh nominal i by producing the formulas $\Diamond i$ and $@_i\phi$. Only GoTo can decompose either formula into the raw nominal i. Our restriction S4a ensures $i \notin A$ so by S4b, nominal i cannot be copied to another block. Thus, unlike root nominals, the nominals generated by $(\Diamond)$ can only appear raw as opening nominals. Since Nom requires the opening nominal of the current block to appear on its own, formulas can only be copied *to* blocks with $(\Diamond)$-generated opening nominals, not *from* them. This matches the quote. It also shows how generated nominals are treated differently, causing a "nominal asymmetry."

We revisit termination in Section 6. For now, note that the *fixed* set A frees us from formalizing the *growing* set of nominals generated by $(\Diamond)$. The reader may imagine the set A to contain all root nominals, as it will in Section 5, such that these can be copied freely.

4 Properties

We briefly remark on some properties of ST^{A} that are useful for the completeness proof. We start by noting that while restriction S3 allows us to start from any amount of potential, a single unit is always sufficient to close a branch. Then we lift the S1 and S2 restrictions by showing that unrestricted versions of the proof rules are admissible. This makes it simpler to show further properties of the system, since we do not have to worry about the restrictions any longer. Finally we show a structural property.

4.1 Sufficient Potential

That a single unit is sufficient is not surprising: simply never make a detour (i.e. two applications of GoTo in a row) and the other rule applications will build up the potential as needed. Similarly, given an existing tableau, construct a more "efficient" counterpart by collapsing sequences of GoTo so only the last one remains. GoTo serves no other purpose than starting a new block so any subsequent rule applications only depend on the final GoTo. The single starting unit may, however, be needed for an initial application of the rule.

▶ **Lemma 8** (A single unit of potential). *If branch Θ closes with respect to A and potential n then Θ closes with respect to A and potential 1.*

Proof. By induction on the closing tableau for Θ (see the formalization for details). ◀

4.2 Strengthening

▶ **Lemma 9** (Strengthening). *Let Θ be a branch and Δ a set of occurrences of ϕ on i-blocks in Θ. Assume that at least one "lasting occurrence" of ϕ at i is not in Δ. If Θ closes wrt. A and potential n then so does Θ with all occurrences in Δ removed.*

Proof. By induction on the construction of the closing tableau for Θ. When an occurrence in Δ is used as rule input, use the lasting occurrence of ϕ instead to construct the tableau for the strengthened branch. No rule applications are invalidated, so the new branch closes under the same amount of potential. Similarly, we only apply rules that were applicable before, so restriction S2 cannot be violated. See the formalization for exact details. ◀

In the formalization we represent the set of occurrences as a set of indices into the branch. We state the lemma over such a set to make it work with the induction principle given by Isabelle/HOL. To lift restriction S1, fix the set of occurrences to contain only the rule output, which must occur elsewhere since S1 is violated, and apply the lemma to justify it.

4.3 Substitution

Next we show a substitution lemma. Note that substitution across a tableau can collapse formulas such that an occurrence suddenly violates restriction S1 and cannot be justified as before the substitution. This is why Lemma 9 is useful. But it also means that our substitution lemma will quantify existentially over the potential needed to close the transformed branch: we may need to start from more potential to account for the fewer rule applications. Another complication is that restriction S2 may suddenly be violated by this collapsing but, as we have also shown previously [14], collapsing witnessing nominals allows us to lift S2.

▶ **Definition 10** ($\Theta\sigma$). *Given a substitution σ, i.e. a mapping from nominals to nominals, and a branch Θ, $\Theta\sigma$ denotes the branch obtained by replacing every nominal i in Θ by $\sigma(i)$.*

Substitutions are allowed to change the type of nominals, e.g. from numbers to strings, so in the following lemma we need to ensure that it leaves enough fresh nominals available.

▶ **Lemma 11** (Substitution). *Let Θ be a branch, A be a* finite *set of allowed nominals and σ a substitution whose co-domain is at least as large as its domain. If Θ closes with respect to A then $\Theta\sigma$ closes with respect to the image of A under σ.*

Proof. By induction on the construction of the closing tableau for an arbitrary σ.

In the ($\Diamond$) case, let i be the generated witnessing nominal. After the (collapsing) substitution, the rule input may become witnessed by some nominal $\sigma(j)$, violating S2. In this case, utilize that we can pick σ in the induction hypothesis such that it maps i to $\sigma(j)$. By the side condition on ($\Diamond$), the image of A under the updated σ is the same, but now Lemma 9 justifies the rule output. The rest of the branch is unaffected since i is fresh.

If S2 is not violated, it may still be that $\sigma(i)$ is no longer fresh like i was before the substitution. Therefore, use the finiteness of both the branch and A, and the size of the co-domain of σ, to obtain a fresh nominal k. Apply the induction hypothesis at σ mapping i to k. This guarantees that the ($\Diamond$) rule applies to justify the rule output. ◀

To lift S2, collapse the involved witnessing nominals in the same way as in the proof of Lemma 11 and apply Lemma 9. The finiteness assumption on A is stronger than we need, but we forgo generalization since we work with finite sets in Section 5 anyway.

4.4 Branch Structure

The following lemma shows that we can add, contract and rearrange blocks on a branch without affecting the existence of a closing tableau. Such operations may violate both S1 and S2, but we have lifted these restrictions already, so we do not need to worry about them.

▶ **Lemma 12** (Adding, contracting and rearranging blocks). *Let Θ be a branch consisting of the set of blocks $\{B_1, \ldots, B_n\}$ and let Θ' be a branch whose blocks are a finite superset of $\{B_1, \ldots, B_n\}$. If Θ closes wrt. finite A then so does Θ'.*

Proof. By induction on the construction of the closing tableau for arbitrary Θ'. In each case we apply the induction hypothesis at Θ' extended by B, where B is the current block of the original branch. This makes the opening nominals agree on the two branches, so that the original rule applies to the new branch as well. After applying this rule, we justify the B block by Lemma 9 and the GoTo rule. Lemma 11 resolves ($\Diamond$) cases where the fresh nominal is not fresh on the new branch since we can substitute it with another fresh nominal. ◀

5 Completeness

Our completeness proof is a synthesis of two approaches, both based on showing completeness via contradiction by constructing a model for formulas on open, exhausted branches.

Bolander and Blackburn reason about the shape of such branches directly from the proof rules in their terminating, internalized calculus [9]. Jørgensen et al., on the other hand, define Hintikka sets of blocks as an abstraction of their open, exhausted branches and show model existence for formulas in such sets. They show that any set of blocks without a closing tableau can be extended to a maximal consistent set of blocks and that these are Hintikka sets [21]. Their model construction, however, assumes that all nominals are treated uniformly, which our termination restrictions prevent (cf. Section 3.3). We define Hintikka sets of blocks that characterize open branches *exhausted with respect to* a set of allowed nominals A. We then abstract the model existence result by Bolander and Blackburn, which is compatible with such branches, and apply it to our Hintikka sets. In Section 5.4 we contrast our approach with the existing work but the proof itself is self-contained.

5.1 Hintikka Sets

Figure 4 shows our definition of Hintikka sets of blocks. We reuse the "at" notation from Definition 1 and suppress "in H" for brevity. Our goal is to show a model existence result for formulas on blocks in such sets. **ProP** and **NomP** ensure consistency at the bottom by forbidding certain contradictions. The remaining requirements match the proof rules. The ones up to **Nom** ensure *downwards saturation* such that the satisfiability of a complex formula is guaranteed by conditions on its subformulas [21]. The novel condition **Nom** ensures *lateral saturation* of allowed formulas across blocks whose opening nominals denote the same world. This allows us to treat such blocks uniformly when it comes to allowed formulas.

▶ Remark 13. **Nom** replaces three requirements by Jørgensen et al. [21, (iv, v, vii)] that serve the same purpose for a smaller range of formulas.

ProP If nominal b occurs at a and prop. symbol p occurs at b then $\neg p$ does not occur at a.
NomP If nominal i occurs at a then $\neg i$ does not occur at a.
NegN If $\neg\neg\phi$ occurs at a then ϕ occurs at a.
DisP If $\phi \vee \psi$ occurs at a then either ϕ or ψ occurs at a.
DisN If $\neg(\phi \vee \psi)$ occurs at a then both $\neg\phi$ and $\neg\psi$ occur at a.
DiaP If $\Diamond\phi$ occurs at a and ϕ is not a nominal then for some i, $\Diamond i$ and $@_i\phi$ occur at a.
DiaN If $\neg\Diamond\phi$ and $\Diamond i$ both occur at a then $\neg@_i\phi$ occurs at a.
SatP If $@_a\phi$ occurs at b then ϕ occurs at a.
SatP If $\neg@_a\phi$ occurs at b then $\neg\phi$ occurs at a.
GoTo If ϕ occurs at a and i is a nominal in ϕ then some block in H has opening nominal i.
Nom If ϕ and nominal a both occur at b and ϕ is *allowed* by A then ϕ occurs at a.

Figure 4 Eleven requirements for a set of blocks H to be a Hintikka set with respect to A.

5.1.1 Equivalence

Assume for the rest of the section that H is a Hintikka set with respect to the set of allowed nominals A. We define an equivalence between nominals:

▶ **Definition 14** (Equivalence). *Nominals i, j are equivalent, $i \sim_H j$, if j occurs at i in H.*

▶ Note 15 ($\sim$ and ϕ at i). In the following we typically suppress the subscript in $\sim_H$ and likewise the fragment "in H" in sentences like "ϕ occurs at i in H".

The equivalence $i \sim j$ only implies $j \sim i$ if $i \in A$ as otherwise **Nom** does not apply: only allowed nominals are symmetric. This motivates the restriction on the following lemma:

▶ **Lemma 16** (Equivalence relation). *$\sim_H$ is an equivalence relation on the set of allowed opening nominals in H.*

Proof. *Reflexivity:* $i \sim_H i$ for any opening nominal i in H since opening nominals occur on their own block. *Symmetry:* Assume $i \sim_H j$ with $i \in A$. That is, j occurs at i in H so by **Nom**, i occurs at j in H: $j \sim_H i$. *Transitivity:* Assume $i \sim_H j$ and $j \sim_H k$ with $i, k \in A$. By symmetry, i occurs at j in H: $j \sim_H i$. Moreover, $k \in A$ occurs at j in H so by **Nom**, k occurs at i in H: $i \sim_H k$. ◀

▶ Note 17. Due to the *GoTo* Hintikka restriction, any nominal occurring in H also occurs as opening nominal, so $\sim_H$ is an equivalence relation on the allowed nominals in H.

5.1.2 Model Construction

Let $|i|_{\sim_H}$ denote the set of nominals equivalent to i with respect to H.

We make use of the following shorthand in our model construction:

▶ **Definition 18** (ϕ at a^*). *We say that ϕ occurs* at *a set of nominals $a^* = \{a_0, a_1, \ldots\}$ if it occurs at some nominal $a_k \in a^*$ and that ϕ occurs at* all *a^* if it occurs at all nominals in a^*.*

We can now define the model induced by Hintikka set H and allowed nominals A:

▶ **Definition 19** (The model $\mathfrak{M}_{H,A}$ and assignment $g_{H,A}$ induced by H and A).
Worlds *The worlds of $\mathfrak{M}_{H,A}$ are sets of equivalent nominals, written a^*, from H.*

Assignment *The assignment $g_{H,A}$ maps a nominal to the equivalence class of an equivalent, allowed nominal or a singleton set if no such nominal exists:*

$$g_{H,A}(a) = \begin{cases} |b|_{\sim H} & \exists b \in A.\ a \sim_H b \\ \{a\} & \textit{otherwise} \end{cases}$$

Reachability *From world a^* we can reach a world exactly if it is denoted by some nominal b that is reachable at a^* (as witnessed by $\Diamond b$ occurring at a^*):*

$$R_{H,A}(a^*) = \{g_{H,A}(b) \mid \exists a \in a^*.\ \Diamond b \textit{ occurs at } a \textit{ in } H\}$$

Valuation *Propositional symbol p holds at world a^* exactly if p occurs at a^* in H:*

$$V_{H,A}(a^*)(p) = \exists a \in a^*.\ p \textit{ occurs at } a \textit{ in } H$$

5.1.3 Properties of the Model

Consider first a property of the assignment:

▶ **Lemma 20** (Non-empty assignment). *The induced assignment $g_{H,A}$ is always non-empty.*

Proof. Fix an arbitrary nominal a. If $g_{H,A}(a) = \{a\}$ the thesis holds immediately. So assume there is some $b \in A$ such that $a \sim_H b$ and $g_{H,A}(a) = |b|$. $b \in |b|$ witnesses the thesis. ◀

The following lemma showcases the lateral saturation guaranteed by the **Nom** condition:

▶ **Lemma 21** (Assignment closure). *If ϕ is allowed by A and ϕ occurs at a in H then ϕ occurs at all $g_{H,A}(a)$ in H (and at least one such world exists).*

Proof. If $g_{H,A}(a) = \{a\}$ the thesis holds immediately. So assume there is some $b \in A$ where b occurs at a in H and $g_{H,A}(a) = |b|$. Then by Hintikka requirement **Nom**, ϕ occurs not only at b in H but at all $a \in |b|$ in H, proving the thesis. Lemma 20 gives the parenthetical. ◀

5.1.4 Model Existence

We can now prove model existence:

▶ **Lemma 22** (Model existence). *Let H be a Hintikka set with respect to allowed nominals A. We show two statements by mutual induction:*

- *If ϕ occurs at i in H and ϕ is allowed by A then $\mathfrak{M}_{H,A}, g_{H,A}, g_{H,A}(i) \models \phi$.*
- *If $\neg\phi$ occurs at i in H and ϕ is allowed by A then $\mathfrak{M}_{H,A}, g_{H,A}, g_{H,A}(i) \not\models \phi$.*

Proof. By induction on the structure of ϕ for an arbitrary nominal i. The proof follows the one by Bolander and Blackburn [9]. We suppress subscripts for readability.

If p at i then p at $g(i)$ by Lemma 21, which matches the valuation, so $\mathfrak{M}, g, g(i) \models p$.

If $\neg p$ at i then $\neg p$ at all $g(i)$ so by **ProP**, p does not occur at $g(i)$, so $\mathfrak{M}, g, g(i) \not\models p$.

If a at i then from the assumption $a \in A$ we have $g(i) = |a|$ and $g(a) = |a|$ and thereby $g(i) = g(a)$ so $\mathfrak{M}, g, g(i) \models a$.

If $\neg a$ at i then $\neg a$ at $g(i)$ by Lemma 21. Moreover, $a \in A$ by assumption so from Lemma 21 we have that a occurs at all $g(a)$. We thus have $\neg a$ at $g(i)$ but a at all $g(a)$ so by **NomN**, $g(i) \neq g(a)$ and therefore $\mathfrak{M}, g, g(i) \not\models a$.

If $\neg\phi$ at i then $\mathfrak{M}, g, g(i) \not\models \phi$ by the induction hypothesis so $\mathfrak{M}, g, g(i) \models \neg\phi$.

If $\neg\neg\phi$ at i then ϕ at i by **NegN** and $\mathfrak{M}, g, g(i) \not\models \neg\phi$ by the induction hypothesis.

The cases for $\phi \vee \psi$, $\neg(\phi \vee \psi)$, $@_j\phi$ and $\neg@_j\phi$ at i all follow similarly to $\neg\phi$ and $\neg\neg\phi$.

If $\Diamond j$ at i then $j \in A$ by assumption. Thus $\Diamond j$ at $g(i)$ so $g(i)\, R\, g(j)$ and $\mathfrak{M}, g, g(i) \models \Diamond j$.

If $\Diamond\phi$ at i where ϕ is not a nominal then by **DiaP** (and Lemma 21) there is some witnessing nominal k such that $\Diamond k$ and $@_k\phi$ both appear at $g(i)$. By **SatP**, ϕ then occurs at k and by the induction hypothesis at k we have $\mathfrak{M}, g, g(k) \models \phi$. From $\Diamond k$ at $g(i)$ we have $g(i)\, R\, g(k)$ so combined we get $\mathfrak{M}, g, g(i) \models \Diamond\phi$.

If $\neg\Diamond\phi$ at i then $\neg\Diamond\phi$ at $g(i)$ by Lemma 21. We need to show that all worlds reachable from $g(i)$ falsify ϕ. So assume for some arbitrary j that $\Diamond j$ occurs at some $a \in g(i)$. By **Nom**, we also have $\neg\Diamond\phi$ at a so by **DiaN** we get $\neg@_j\phi$ at a and finally by **SatN** we have $\neg\phi$ at j. The induction hypothesis at j then tells us that $\mathfrak{M}, g, g(j) \not\models \phi$ as needed. Since j was chosen arbitrarily, $\mathfrak{M}, g, g(i) \not\models \Diamond\phi$.

Each appeal to the induction hypothesis requires showing that the subformula is allowed by A but since it is a subformula this holds trivially. ◀

5.2 Maximal Consistent Sets

Our next task is to follow the classical synthetic recipe: extend a consistent set of blocks to be maximally consistent, show that such sets fulfill all Hintikka requirements and thus that formulas in them are satisfiable. Consistency and maximality are standard but wrt. A:

▶ **Definition 23** (Consistency). *The set of blocks S is consistent wrt. A if there is no finite subset $S' \subseteq S$ such that S' has a closing tableau wrt. A and any amount of potential.*

▶ **Definition 24** (Maximality). *The set of blocks S is maximal wrt. A if it is consistent wrt. A and for any block $B \notin S$ the set $S \cup \{B\}$ is inconsistent wrt. A.*

Besides maximally consistent, our constructed set will also be $\Diamond$-*saturated* [21]:

▶ **Definition 25** ($\Diamond$-Saturation). *The set of blocks S is $\Diamond$-saturated if for any ϕ at any a in S, where ϕ is not a nominal, there is a nominal i such that $@_i p$ and $\Diamond i$ both occur at a in S.*

We now construct our $\Diamond$-saturated maximally consistent set and show it is a Hintikka set:

▶ **Definition 26** (Lindenbaum-Henkin construction). *Assume an enumeration of all blocks $B_0, B_1, B_2 \ldots$ in the language. From a consistent set S_0 we build an infinite sequence of consistent sets $S_0, S_1, S_2, \ldots$ in the following way. Given S_n, construct S_{n+1} like so:*

$$S_{n+1} = \begin{cases} S_n & \text{if } S_n \cup \{B_n\} \text{ is inconsistent wrt. } A \\ S_n \cup \{B_n\} \cup \{B'\} & \text{otherwise, where } B' \text{ is a } \Diamond\text{-witness for } B_n \end{cases}$$

A $\Diamond$-witness for a block B is a block with the same opening nominal that witnesses all $\Diamond\phi$-formulas in B using fresh and disallowed nominals (when ϕ is not a nominal).

▶ **Lemma 27** (Lindenbaum-Henkin). *Let S_0 be a consistent set of blocks with respect to finite A and over a finite set of nominals. Then $\bigcup S_n$ as given by Definition 26 is a $\Diamond$-saturated maximally consistent set.*

Proof. The three-part proof follows the one by Jørgensen et al. [21].

Consistency. Proof by contradiction. Assume $\bigcup S_n$ is inconsistent. Then some finite subset $S' \subseteq \bigcup S_n$ has a closing tableau. But the sequence $S_0, S_1, S_2, \ldots$ grows with respect to $\subseteq$ so there must be an m such that $S' \subseteq S_m$. And since S_0 is consistent, it follows by induction on m that S_m is too (each $\Diamond$-witness preserves consistency due to the ($\Diamond$) rule). This contradicts the existence of an inconsistent, finite subset S'.

Maximality. Proof by contradiction. Assume that there is some block $B_m \notin \bigcup S_n$ such that $\bigcup S_n \cup \{B_m\}$ is still consistent. This block is part of the enumeration of blocks, but was not added to S_{m+1}. This can only be because $S_m \cup \{B_m\}$ is inconsistent. However, $S_m \cup \{B_m\} \subseteq \bigcup S_n \cup \{B_m\}$ contradicting the consistency of the right-hand side.

$\Diamond$-Saturation. Follows directly from the addition of $\Diamond$-witnesses. ◀

▶ **Lemma 28** (Smullyan-Fitting block lemma). *Assume S is a $\Diamond$-saturated maximal consistent set of blocks wrt. a finite set A and a finite set of nominals. Then S is a Hintikka set.*

Proof. The proof follows the one by Jørgensen et al. [21] but we have fewer cases since we have fewer Hintikka requirements. The cases are straight-forward so we only exemplify three, with the last being the typical one. The remaining cases can be found in the formalization.

Case ProP. Proof of negation. Assume that b occurs at a, p occurs at b and $\neg p$ occurs at a in S for some a, b, p. The set S is assumed to be consistent but we can construct a closing tableau from these blocks by applying the Nom rule to get $\neg p$ at b and immediately close due to the existing p at b.

Case DiaP. Follows directly from $\Diamond$-saturation.

Case Nom. Assume that both ϕ and a occur at b in S and that ϕ is allowed by A. Assume towards a contradiction that ϕ does not occur at a in S. Then by the maximality of S, we can find an inconsistent finite subset $S' \cup \{([\phi], a)\} \subseteq S \cup \{([\phi], a)\}$ where $([\phi], a)$ is an a-block that only contains ϕ. If a closing tableau exists for $S' \cup \{([\phi], a)\}$ then it also exists for the larger set $S' \cup \{([\phi], a)\} \cup \{([\phi, a], b)\}$ (Lemma 12). But now the Nom rule tells us that ϕ at a is redundant, so just $S' \cup \{([], a)\} \cup \{([\phi, a], b)\}$ is inconsistent. The GoTo rule gets us to $S' \cup \{([\phi, a], b)\}$ and this set is trivially a subset of S, contradicting its consistency. ◀

5.3 Tying It All Together

Completeness follows by constructing a model for any formula whose tableau does not close.

▶ **Theorem 29** (Completeness). *Assume that ϕ is a valid formula and a is some nominal. Let A be the set containing all nominals in ϕ. Then the branch consisting solely of $\neg\phi$ on an a-block has a closing tableau with respect to A and 1 unit of potential.*

Proof. Assume towards a contradiction that the branch does not close. Then the set $S_0 = \{([\neg\phi], a)\}$ is consistent with respect to A. We construct $\bigcup S_n$, which by Lemma 27 is a $\Diamond$-saturated maximal consistent set of blocks, so by Lemma 28 $\bigcup S_n$ is a Hintikka set.

Since $\neg\phi$ occurs at a in $\bigcup S_n$, we obtain from Lemma 22 a model that does not satisfy ϕ, namely $\mathfrak{M}_{H,A}, g_{H,A}, g_{H,A}(a) \not\models \phi$. This contradicts our validity assumption, so the branch must close. By Lemma 8 it must close from a single unit of potential. ◀

5.4 Relation to Existing Work

In this section we provide context for our induced model, Definition 19, and the corresponding Lemma 22. Readers less familiar with tableau systems for hybrid logic may skip this section. To refresh, Bolander and Blackburn give an analytic proof for a terminating, internalized calculus [9] and Jørgensen et al. give a synthetic proof for the non-terminating system ST [21].

5.4.1 Worlds

Jørgensen et al. have no restrictions on their Nom rule so they have no nominal asymmetry (cf. Section 3.3) and $\sim_H$ is an equivalence relation on all nominals. They use representatives of such equivalence classes as their worlds [21]. Since $\sim_H$ is only an equivalence relation on a subset of our nominals, we cannot use equivalence classes directly. Instead we use sets of equivalent nominals. Bolander and Blackburn use plain nominals as their worlds.

5.4.2 Assignment

Jørgensen et al. map each nominal i in H to its equivalence class $|i|_{\sim_H}$ [21]. If we artificially fix A to contain all nominals in H then $\sim_H$ becomes an equivalence relation on all nominals. Our assignment then reduces to its first clause and becomes equivalent to theirs.

Bolander and Blackburn map each nominal a to its "urfather" $u(a)$: either an equivalent "right nominal" or the nominal itself if no such nominal exists [9]. This is very similar to our assignment that maps each nominal to the equivalence class of an equivalent *allowed nominal* or the singleton set if no such nominal exists.

A *right nominal*, understood in terms of our setting, is a non-opening nominal that occurs on its own. Since there may be multiple equivalent right nominals, Bolander and Blackburn impose an ordering on them and always choose the smallest one to ensure that their assignment is well-defined [9]. Working with sets of nominals frees us from such concerns.

5.4.3 Reachability and the Bridge Rule

It is worthwhile to compare the three different reachability relations from the considered systems. By writing them in similar notation we get:

Jørgensen et al.	$\|i\| \, R_H \, \|j\|$	iff $\Diamond j$ occurs at i in H
Bolander and Blackburn	$i \, R_H \, u(j)$	iff $\Diamond j$ occurs at i in H
The present paper	$i^* \, R_H \, g_{H,A}(j)$	iff $\Diamond j$ occurs at i^* in H

If we further note that $g(j) = |j|$ for Jørgensen et al. [21] and $g(j) = u(j)$ for Bolander and Blackburn [9] we see that the relations are all defined in the same way over the assignment: *a world is reachable iff it is denoted by a nominal j such that $\Diamond j$ occurs at the current world.* Only the treatment of the worlds differ. Since Jørgensen et al. use representatives of their sets they need the following Hintikka requirement to ensure well-definedness:

> If there is an i-block in H with $\Diamond j$ on it, and a j-block in H with k on it, then there is an i-block in H with $\Diamond k$ on it [21, (vi)].

To see why, imagine that the premises hold but the conclusion does not. Then $|i|R_H|j|$ and $j \sim_H k$ but not $|i|R_H|k|$ even though $|j| = |k|$ by the second premise, so the choice of representative matters when it should not. In our setting we side-step the problem completely by having no representatives but quantifying existentially over the nominals in our worlds.

If we view the requirement as a rule, we get the known Bridge rule that produces $\Diamond k$ at i given $\Diamond j$ at i and nominal k at j. Jørgensen et al. prove the admissibility of Bridge as part of their completeness proof [21]. We include this result in the formalization (when $j \in A$) because it is interesting in its own right [4] but do not need it for completeness.

5.4.4 Valuation

Our valuation is standard but our use of sets instead of representatives slightly complicates the **ProP** Hintikka requirement, where we take equivalence of nominals into account. For Jørgensen et al. the following suffices: "if there is an i-block in H with atomic formula a on it then there is no i-block in H with $\neg a$ on it." [21].

5.4.5 Model Existence

We turn now to the model existence result, Lemma 22, inspired by Blackburn and Bolander [9].

The two nominal cases and the $\Diamond j$ case rely on the involved nominals being in A. Bolander and Blackburn work with right nominals instead of allowed nominals [9]. This gives them the positive nominal case for free, since the formula in that case is a right nominal. In the negative nominal case, however, they need to rely on a special ($\neg$) rule that upgrades a negated nominal, "$@_i\neg a$", to a right nominal "$@_a a$". They need this rule because of the nature of internalized tableau systems: the nominal i in a satisfaction statement $@_i a$ has lower status than the right nominal a. The status of nominals in our system is not defined structurally but by the set A. Thus, we make the ($\neg$) rule unnecessary by picking A carefully.

Finally, Bolander and Blackburn assume that the formula in question is not a $\Diamond j$ formula produced by the ($\Diamond$) rule. Our assumption $j \in A$ matches this, since the ($\Diamond$) rule cannot generate an allowed nominal, but we are free from keeping track of actual rule applications.

6 Relation to ST^*

Here, we relate our restrictions S1-S4 to the restrictions R1-R5 and Nom* rule in ST^* [4].

6.1 System ST^*

For reasons of space we introduce ST^* only briefly. To obtain ST^*, take the rules in Figure 2, add another rule called Name that introduces a fresh nominal to the branch and impose restrictions R1-R5 and Nom* that we explain in the following. Since the rules of ST^{A} are a subset of ST^*, it is meaningful to compare the strength of our restrictions to those of ST^*.

Blackburn et al. [4] need the Name rule since they allow the very first block to have no opening nominal. We have dispensed with this flexibility to obtain a simpler formalization.

6.2 Restrictions R1-R5

Restriction R1 states that "a formula is never added to an i-block if it already occurs in an i-block on the same branch" [4]. This formulation is more ambiguous than our S1, which states when a rule is applicable. Any rule application outlawed by R1 is also outlawed by S1:

▶ **Lemma 30** (R1 implies S1). *If R1 outlaws a rule application then so does S1.*

Proof. R1 outlaws the rule application so it must include no formulas new to the block type. Therefore, S1 outlaws it too. ◀

Restriction R2 states that "the ($\Diamond$) rule can not be applied twice to the same formula occurrence" [4]. Note that formalizing this would require keeping track of ($\Diamond$) rule applications. This is why S2 is formulated in terms of branch content instead. It is at least as strict as R2:

▶ **Lemma 31** (R2 implies S2). *If R2 outlaws an application of* ($\Diamond$) *then so does S2.*

Proof. Assume that an application of the rule ($\Diamond$) to formula $\Diamond\phi$ at a is outlawed by R2. This means that ($\Diamond$) has already been applied to $\Diamond\phi$ at a. So for some nominal i there must be formulas $@_i\phi$ and $\Diamond i$ witnessing $\Diamond\phi$ at a. Thus the application is also outlawed by S2. ◀

Restriction R3 applies to the omitted name rule so we have no equivalent of it [4].

Restriction R4 states that "the GoTo rule can not be applied twice in a row" [4]. Our counterpart is S3 that does allow repeated applications but still prevents repeating the rule ad infinitum (cf. Figure 3b). We see in Section 6.5 why this extra flexibility is desirable. For now recall the idea from Section 4.1 that any tableau with repeated applications of GoTo can be translated into one where just the final application remains. We have the following:

▶ **Lemma 32** (From S3 to R4). *A tableau satisfying S3 collapses into one that satisfies R4 where only finite sequences of GoTo are removed and all non-GoTo applications are preserved.*

Proof. By collapsing all sequences of GoTo applications into the last one (cf. Lemma 8). All such sequences are finite due to decreasing potential so "the last one" is well-defined. ◀

Finally, restriction R5 can be ignored here: it restricts the more liberal variants of rules (@) and ($\neg$@) in system ST to the versions present in ST* and ST$^{\mathrm{A}}$ [4].

6.3 Nom* and Allowed Nominals

We turn now to the Nom* rule in ST* and its relationship to our set of allowed nominals A in restriction S4. We first need the following by Blackburn et al. [4]: "A quasi-root subformula is a formula of the form ϕ, $\neg\phi$, $@_i\phi$ or $\neg@_i\phi$ where ϕ is a subformula of the root."

Their Nom* rule is then defined as follows:

> Suppose i and j are nominals, ϕ is a quasi-root subformula and $j \neq i, \phi$. If j and ϕ both occur in i-blocks on a branch Θ, then ϕ can be added to any j-block on Θ [4].

By inspecting the rules of ST* and ST$^{\mathrm{A}}$ we see that only the ($\Diamond$) rule can produce formulas that are not quasi-root subformulas [4]. As such, the only formulas that Nom* does not allow us to copy are formulas i and $\Diamond i$ where i was introduced by ($\Diamond$). This is exactly what restriction S4 enforces on our Nom rule (cf. Section 3.3). So S4 is at least as strict:

▶ **Lemma 33** (Nom implies Nom*). *Suppose that ϕ and a both occur at b in a tableau constructed under the allowed set of nominals A. If Nom can add ϕ to a then so can Nom*.*

Proof. If ϕ can be added by Nom it must be allowed by A. Thus ϕ must be a quasi-root subformula. Moreover, since adding ϕ to a does not violate S1 (or R1), $a \neq \phi$ and likewise $a \neq b$. Ultimately, Nom* can also add ϕ to a. ◀

6.4 Termination

We have covered all differences between ST* and ST$^{\mathrm{A}}$ and seen how the restrictions compare. This motivates the following unformalized theorem and proof sketch:

▶ **Theorem 34** (ST$^{\mathrm{A}}$ is terminating). *Any ST^A tableau is finite.*

Proof. Lemmas 30–33 imply that we can translate any ST$^{\mathrm{A}}$ tableau into an ST* tableau of similar size by collapsing repeated applications of GoTo (and adding an initial application of the Name rule). Since all ST* tableaux are finite [4] so must any ST$^{\mathrm{A}}$ tableau be. ◀

Blackburn et al. [4] exemplify a number of infinite branches possible in system ST and show that they are illegal in system ST*. In support of the above theorem, we note that the sequences of rule applications leading to those infinite branches are also outlawed in ST$^{\mathrm{A}}$.

(a) Possible segment on original closing tableau. **(b)** R becomes invalid causing two GoTos in a row.

Figure 5 Unjustified GoTo after applying substitution σ that unifies a and a' as well as ϕ and ϕ'.

6.5 Restricting the GoTo Rule

We should motivate our choice of S3 over R4. As Section 4 shows, we typically show lemmas of the form "if branch Θ has a closing tableau then so does $f(\Theta)$", where f is some operation like substitution or restructuring. In a proof by induction on the closing tableau under restriction R4 we need to show in each non-GoTo case that GoTo becomes applicable, since we need that assumption to discharge the GoTo case. However, the transformation may invalidate a previously valid rule application and prevent us from making this promise. Figure 5 depicts a possible case when proving the substitution lemma. Before the substitution, the application of rule R was legal, but afterwards it violates restriction R1. We can still justify the extension ϕ with the Strengthening Lemma 9 but doing so does not make GoTo applicable afterwards.

We might give a more intricate transformation that also prunes detours but that would complicate an otherwise simple idea like substitution. We could also state the lemma in weaker terms that allow for a different branch structure, but we prefer to give straight-forward lemmas and transformations. Our S3 restriction resolves the issue by dealing with detours separately. Consider Figure 5 from the perspective of potential: we need to start from more potential to close the transformed branch since we lose a rule application, but we can simply do this, so the detour becomes benign. Thus, we can give the transformation we want, we just need to existentially quantify the potential required to close the resulting branch.

7 Formalization

In general, the formalization consists of close to 5000 lines in the *intelligible semi-automated reasoning* language Isar [27] and follows the structure of the paper. It is accepted into the Archive of Formal Proofs and thus kept up to date with new versions of Isabelle/HOL.

We formalize the logic as a deep embedding into higher-order logic by specifying the syntax as a datatype and the semantics as a predicate on that datatype (alongside a model and an assignment). Types in higher-order logic are non-empty so we represent the set of worlds as a type variable *'w*. Similarly, we use *'a* for the universe of propositional symbols and *'b* for the universe of nominals. We formalize a block as a list of formulas paired with its opening nominal and a branch as a list of blocks, where lists in Isabelle/HOL are finite, ordered sequences. We use the **inductive** command to specify the proof system as ten inductive cases. The command provides a predicate $\vdash$ for whether or not a given branch closes with respect to a set A and potential n. Thus, we abstract away the concrete shape of a closing tableau and reason only about its existence. This suffices for formalizing completeness but not termination where we would need to inspect well-formed but infinite branches. However, it permits induction over the proof rules instead of the trickier coinduction.

Imagine that we formalized ST^* instead of ST^{A}. Section 6.5 motivated our choice of S3 over R4. Restriction R2 on the $(\Diamond)$ rule would require us to additionally index our predicate $\vdash$ by a list of indices, each pointing to a formula occurrence that $(\Diamond)$ cannot be applied to. When

proving lemmas by induction, we would need to make suitable assumptions about this list. Instead, our formulation S2 identifies the applicability of ($\Diamond$) from the branch content itself, which we already know. The Nom* rule considers quasi-root subformulas and would require us to remember the root segment of the tableau as we extend it, complicating induction proofs too. Our parameterization of the rules by the set A causes no such complications.

Imagine next that we adapted the completeness proof for ST^* to ST^{A}. That proof works by translation from a different system with an analytic completeness proof, which we would have to formalize as well. This could be done: Blanchette, Popescu and Traytel [7, 8] have formalized analytic completeness proofs for first-order logic in Isabelle/HOL. Instead, our *standalone* synthetic completeness proof joins a family of such proofs in Isabelle/HOL [2, 15, 16]. While possible, a similar proof for ST^* would, as described, be harder to formalize.

8 Conclusion and Future Work

We have presented a Seligman-style tableau system for hybrid logic with a formalization in Isabelle/HOL of its soundness and completeness and argued that it is terminating. The restrictions required for termination cause an asymmetry in branches that makes a previous synthetic completeness proof for hybrid logic tableau systems inapplicable. We have presented a novel proof that works in this case and described its relation to existing work. The use of plain sets instead of representatives in the model construction relieves us of some concerns about well-definedness. Our work is the first sound and complete formalized proof system for hybrid logic and the first synthetic proof for a terminating hybrid logic tableau system.

Blackburn et al. showed termination of ST^* by a translation of any branch into a terminating system and we claim termination of ST^{A} by possible translation into ST^*. We are currently working on a direct, formalized termination proof for ST^{A} through a decreasing measure argument in the style of Bolander and Blackburn [9]. This will allow code generation for a verified decision procedure based on the tableau system. We also want to explore extensions to the logic and investigate a Seligman-style system for intuitionistic hybrid logic.

References

1 Carlos Areces, Patrick Blackburn, and Maarten Marx. Hybrid logics: Characterization, interpolation and complexity. *The Journal of Symbolic Logic*, 66(3):977–1010, 2001.

2 Stefan Berghofer. First-Order Logic According to Fitting. *Archive of Formal Proofs*, 2007. Formal proof development. URL: `http://isa-afp.org/entries/FOL-Fitting.html`.

3 Patrick Blackburn. Representation, reasoning, and relational structures: A hybrid logic manifesto. *Logic Journal of the IGPL*, 8(3):339–365, 2000.

4 Patrick Blackburn, Thomas Bolander, Torben Braüner, and Klaus Frovin Jørgensen. Completeness and termination for a Seligman-style tableau system. *Journal of Logic and Computation*, 27(1):81–107, 2017.

5 Patrick Blackburn and Miroslava Tzakova. Hybridizing concept languages. *Annals of Mathematics and Artificial Intelligence*, 24(1-4):23–49, 1998.

6 Jasmin Christian Blanchette. Formalizing the metatheory of logical calculi and automatic provers in isabelle/hol (invited talk). In Assia Mahboubi and Magnus O. Myreen, editors, *Certified Programs and Proofs, CPP. Proceedings*, pages 1–13. ACM, 2019.

7 Jasmin Christian Blanchette, Andrei Popescu, and Dmitriy Traytel. Abstract completeness. *Archive of Formal Proofs*, 2014. Formal proof development. URL: `https://isa-afp.org/entries/Abstract_Completeness.html`.

8 Jasmin Christian Blanchette, Andrei Popescu, and Dmitriy Traytel. Soundness and completeness proofs by coinductive methods. *Journal of Automated Reasoning*, 58(1):149–179, 2017.

9 Thomas Bolander and Patrick Blackburn. Termination for Hybrid Tableaus. *Journal of Logic and Computation*, 17(3):517–554, 2007.

10 Thomas Bolander and Torben Braüner. Tableau-based decision procedures for hybrid logic. *Journal of Logic and Computation*, 16(6):737–763, 2006.

11 Torben Braüner and Valeria de Paiva. Intuitionistic hybrid logic. *Journal of Applied Logic*, 4(3):231–255, 2006.

12 Christian Doczkal and Gert Smolka. Constructive formalization of hybrid logic with eventualities. In Jean-Pierre Jouannaud and Zhong Shao, editors, *Certified Programs and Proofs, CPP. Proceedings*, volume 7086 of *Lecture Notes in Computer Science*, pages 5–20. Springer, 2011.

13 Melvin Fitting. *Proof Methods for Modal and Intuitionistic Logics*, volume 169. Springer Science & Business Media, 1983.

14 Asta Halkjær From, Patrick Blackburn, and Jørgen Villadsen. Formalizing a Seligman-style tableau system for hybrid logic. In Nicolas Peltier and Viorica Sofronie-Stokkermans, editors, *Automated Reasoning*, pages 474–481, Cham, 2020. Springer International Publishing.

15 Asta Halkjær From. Epistemic Logic. *Archive of Formal Proofs*, 2018. Formal proof development. URL: `http://isa-afp.org/entries/Epistemic_Logic.html`.

16 Asta Halkjær From. A sequent calculus for first-order logic. *Archive of Formal Proofs*, 2019. Formal proof development. URL: `http://isa-afp.org/entries/FOL_Seq_Calc1.html`.

17 Asta Halkjær From. Hybrid Logic. Master's thesis, Technical University of Denmark, 2020.

18 Asta Halkjær From. Hybrid logic in the Isabelle proof assistant: Benefits, challenges and the road ahead. In Nicola Olivetti and Rineke Verbrugge, editors, *Short Papers: Advances in Modal Logic (AiML) 2020*, pages 23–27, 2020.

19 Didier Galmiche and Yakoub Salhi. Sequent calculi and decidability for intuitionistic hybrid logic. *Information and Computation*, 209(12):1447–1463, 2011.

20 Limin Jia and David Walker. Modal proofs as distributed programs (extended abstract). In David A. Schmidt, editor, *Programming Languages and Systems, ESOP, Proceedings*, volume 2986 of *Lecture Notes in Computer Science*, pages 219–233. Springer, 2004.

21 Klaus Frovin Jørgensen, Patrick Blackburn, Thomas Bolander, and Torben Braüner. Synthetic completeness proofs for Seligman-style tableau systems. In *Advances in Modal Logic, Volume 11*, pages 302–321, 2016.

22 Sven Linker. Hybrid Multi-Lane Spatial Logic. *Archive of Formal Proofs*, 2017. Formal proof. URL: `http://isa-afp.org/entries/Hybrid_Multi_Lane_Spatial_Logic.html`.

23 Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL - A Proof Assistant for Higher-Order Logic*, volume 2283 of *Lecture Notes in Computer Science*. Springer, 2002.

24 Jeremy Seligman, Fenrong Liu, and Patrick Girard. Facebook and the epistemic logic of friendship. In Burkhard C. Schipper, editor, *Proceedings of the 14th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 2013)*, pages 229–238, Chennai, India, 2013.

25 Jerry Seligman. Internalization: The case of hybrid logics. *Journal of Logic and Computation*, 11(5):671–689, 2001.

26 Raymond M Smullyan. *First-Order Logic*. Dover Publications, 1995.

27 Makarius Wenzel. Isabelle/Isar – a generic framework for human-readable proof documents. *From Insight to Proof – Festschrift in Honour of Andrzej Trybulec, Studies in Logic, Grammar and Rhetoric*, 10(23):277–298, 2007.