Formalization of AMR Inference via Hybrid Logic Tableaux

Eli Goldner

July 16, 2021

Abstract

AMR and its extensions have become popular in semantic representation due to their ease of annotation by non-experts, attention to the predicative core of sentences, and abstraction away from various syntactic matter. An area where AMR and its extensions warrant improvement is formalization and suitability for inference, where it is lacking compared to other semantic representations, such as description logics, episodic logic, and discourse representation theory. This thesis presents a formalization of inference over a merging of Donatelli et al.'s (2018) AMR extension for tense and aspect and with Pustejovsky et al.'s (2019) AMR extension for quantification and scope. Inference is modeled with a merging of Hansen's (2007) tableau method for first-order hybrid logic with varying domain semantics (*FHL*) and Blackburn and Jørgensen's (2012) tableau method for basic hybrid tense logic (*BHTL*). We motivate the merging of these AMR variants, present their interpretation and inference in the combination of *FHL* and *BHTL*, which we will call *FHTL* (first-order hybrid tense logic), and demonstrate *FHTL*'s soundness, completeness, and decidability.

1 Merging Quantified Hybrid Logic and Indexical Hybrid Tense Logic

1.1 Background

1.2 First-order Hybrid Logic

Hansen (2007)
  from
  Blackburn and Marx (2002)

1.3 Basic Hybrid Tense Logic

Blackburn and Jørgensen (2012)

2 First-order Hybrid Tense Logic - Syntax and Semantics

The syntax of *FHTL* is identical to *FHL* as given in Hansen (2007) except uses of $\downarrow$ as in $\downarrow w.\varphi$ are omitted along with $\Box$ and $\Diamond$ as in $\Box\varphi$ and $\Diamond\varphi$. $\Box$ and $\Diamond$ are replaced by their semantic equivalents F and G and their temporal duals P and H are added.

Atomic formulae are the same as in *FHL*, symbols in NOM and SVAR together with first-order atomic formulae generated from the predicate symbols and equality over the terms. Thus complex formulae are generated from the atomic formulae according to the following rules:

$$\neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \to \psi \mid \exists x\varphi \mid \forall x\varphi \mid F\varphi \mid G\varphi \mid P\varphi \mid H\varphi \mid @_n\varphi$$

Since we want the domain of quantification to be indexed over the collection of nominals/times, we look to Fitting and Mendelsohn's (1998) treatment of first-order modal logic with varying domain semantics and use it to alter the *FHL* model definition to the following:

$$(T, R, D_t, I_{nom}, I_t)_{t\in T}$$

Thus with varying domain semantics a *FHTL* model is identical to the definition for a *FHL* model in that:

- (T, R) is a modal frame.

- I_{nom} is a function assigning members of T to nominals.

The differences manifest on the level of the model and interpretation. Namely, where $D = \cup_{t \in T} D_t$, (D, I_t) is a first-order model where:

- $I_t(q) \in D$ where q is a unary function symbol.
- $I_t(P) \in D^k$ where P is a k-ary predicate symbol.

Notice we've relaxed the requirement that $I_t(c) = I_{t'}(c)$ for c a constant and $t, t' \in T$, since the interpretation of the constant need not exist at both times. This permits us to distinguish between the domain of a frame and the domain of a time/world, in a way that prevents a variable x from failing to refer at a given time/world, even if it has no interpretation at that time. Intuitively this permits *FHTL* to handle interpretation of entities in natural language utterances, which while reasonable to refer to do not exist at a current time, e.g. previous and future presidents.

Free variables are handled similarly as in *FHL*. Where again $D = \cup_{t \in T} D_t$, a *FHTL* assignment is a function:

$$g{:}\mathsf{SVAR} \cup \mathsf{FVAR} \to T \cup D$$

Where state variables are sent to times/worlds and first-order variables are sent to D, the domain of the frame. Thus given a model and an assignment g, the interpretation of terms t denoted by $\bar{t}$ is defined as:

- $\overline{x} = g(x)$ for x a variable.
- $\overline{c} = I_t(c)$ for c a constant and some $t \in T$.
- For q a unary function symbol:
  - For n a nominal:
  $$\overline{@_n q} = I_{I_{nom}(n)}(q)$$
  - For n a state variable:
  $$\overline{@_n q} = I_{g(n)}(q)$$

Finally we say an assignment g' is an x-variant of g if g' and g on all variables except possiblly x. In particular, we say g' is an x-variant of g at t, a time, if g' and g on all variables except possiblly x and $g'(x) \in D_t$. We omit definitions for $\wedge$, $\rightarrow$, H, G, and $\forall$, since they can be defined in terms of the other rules. Given a model $\mathfrak{M}$, a variable assignment g, and a state s, the inductive definition of $\mathfrak{M}, s \models_g \varphi$ is:

$$\begin{aligned}
\mathfrak{M}, s \models_g P(t_1, \ldots, t_n) &\iff \langle \overline{t_1}, \ldots, \overline{t_n} \rangle \in I_s(P) \\
\mathfrak{M}, s \models_g t_i = t_j &\iff \overline{t_i} = \overline{t_j} \\
\mathfrak{M}, s \models_g n &\iff I_{nom}(n) = s, \text{for } n \text{ a nominal} \\
\mathfrak{M}, s \models_g w &\iff g(w) = s, \text{for } w \text{ a state variable} \\
\mathfrak{M}, s \models_g \neg\varphi &\iff \mathfrak{M}, s \not\models_g \varphi \\
\mathfrak{M}, s \models_g \varphi \vee \psi &\iff \mathfrak{M}, s \models_g \varphi \text{ or } \mathfrak{M}, s \models_g \psi \\
\mathfrak{M}, s \models_g \exists x \varphi &\iff \mathfrak{M}, s \models_{g'} \varphi \text{ for some } x\text{-variant } g' \text{ of } g \text{ at } s \\
\mathfrak{M}, s \models_g F\varphi &\iff \mathfrak{M}, t \models_g \varphi \text{ for some } t \in T \text{ such that } Rst \\
\mathfrak{M}, s \models_g P\varphi &\iff \mathfrak{M}, t \models_g \varphi \text{ for some } t \in T \text{ such that } Rts \\
\mathfrak{M}, s \models_g @_n\varphi &\iff \mathfrak{M}, I_{nom}(n) \models_g \varphi \text{ for } n \text{ a nominal} \\
\mathfrak{M}, s \models_g @_w\varphi &\iff \mathfrak{M}, g(w) \models_g \varphi \text{ for } w \text{ a state variable}
\end{aligned}$$

2.1 The Tableau Calculus

Following Fitting and Mendelsohn (1998) we assume for each nominal or state variable s, there is an infinite list of parameters, where parameters are free variables which are never quantified over, arranged in such a way that different nominals/state variables never share the same parameter. Informally we write p_s to indicate a parameter is associated with a nominal/state variable s.

Propositional rules:

$$\frac{@_s(\varphi \vee \psi)}{@_s\varphi \mid @_s\psi}\ (\vee) \qquad \frac{@_s\neg(\varphi \vee \psi)}{\begin{array}{c}@_s\neg\varphi\\ @_s\neg\psi\end{array}}\ (\neg\vee) \qquad \frac{@_s\neg\neg\varphi}{@_s\varphi}\ (\neg\neg)$$

Modal rules:

$$\frac{@_sF\varphi}{\begin{array}{c}@_sFa\\ @_a\varphi\end{array}}\ (F)^{ab} \qquad \frac{@_sP\varphi}{\begin{array}{c}@_sPa\\ @_a\varphi\end{array}}\ (P)^{ab} \qquad \frac{@_s\neg P\varphi \quad @_sPt}{@_t\neg\varphi}\ (\neg P) \qquad \frac{@_s\neg F\varphi \quad @_sFt}{@_t\neg\varphi}\ (\neg F)$$

Quantifier rules:

$$\frac{@_s\exists x\varphi}{@_s\varphi[s{:}p/x]}\ (\exists)^{c} \qquad \frac{@_s\neg\exists x\varphi}{@_s\neg\varphi[t/x]}\ (\neg\exists)^{d}$$

Equality rules:

$$\frac{}{@_ij{:}t = j{:}t}\ \textbf{(ref)}^{e} \qquad \frac{@_ij{:}t = k{:}s \quad @_i\varphi}{@_i\varphi[j{:}t//k{:}s]}\ \textbf{(sub)}^{f}$$

@ rules:

$$\frac{@_s@_t\varphi}{@_t\varphi}\ (@) \qquad \frac{@_s\neg@_t\varphi}{@_t\neg\varphi}\ (\neg@) \qquad \frac{@_st \quad @_s\varphi}{@_t\varphi}\ \textbf{(nom)} \qquad \frac{[i \text{ on the branch}]}{@_ii}\ \textbf{(nom ref)}$$

$$\frac{@_sPt}{@_tFs}\ (P\textbf{–trans}) \qquad \frac{@_sFt}{@_tPs}\ (F\textbf{–trans}) \qquad \frac{@_sPt \quad @_tu}{@_sPu}\ (P\textbf{–bridge}) \qquad \frac{@_sFt \quad @_tu}{@_sFu}\ (F\textbf{–bridge})$$

FHTL **term rules:**

$$\frac{@_ik_1{:}t = k_2{:}s}{@_ik_1{:}t = k_2{:}s}\ \textbf{(:1)} \qquad \frac{@_ij}{@_ki{:}t = j{:}t}\ \textbf{(:2)}^{e} \qquad \frac{}{@_ik{:}j{:}t = j{:}t}\ \textbf{(:3)}^{e}$$

$$\frac{@_iR(t_1,...,t_n)}{@_iR(i{:}t_1,...,i{:}t_n)}\ \textbf{(:fix 1)} \qquad \frac{@_i\neg R(t_1,...,t_n)}{@_i\neg R(i{:}t_1,...,i{:}t_n)}\ \textbf{(:fix 2)} \qquad \frac{@_it = s}{@_ii{:}t = i{:}s}\ \textbf{(:fix 3)} \qquad \frac{@_i\neg t = s}{@_i\neg i{:}t = i{:}s}\ \textbf{(:fix 4)}$$

$$\frac{}{@_if(t_1,...,t_n) = f(i{:}t_1,...,i{:}t_n)}\ \textbf{(:func)}^{g}$$

[a] The nominal a is new to the branch.
[b] The formula φ is not a nominal.
[c] $s{:}p$ is new to the branch.
[d] p is any ground term or parameter which exists at s.
[e] Where t is a closed term.
[f] $\varphi[j{:}t//k{:}s]$ is φ where some occurences of $j{:}t$ have been replaced by $k{:}s$.
[g] f is an n–ary function symbol and $t_1,...,t_n$ are closed terms.

2.2 Soundness and completeness

Definition 2.1 (Quasi-subformula). A formula φ is a *quasi-subformula* of a formula ψ if one of the the following is the case:

1. φ is a subformula of ψ modulo renaming of free variables and substitution of free variables in φ for grounded terms.
2. φ is of the form $\neg\chi$ where χ is a subformula of ψ modulo renaming of free variables in χ for grounded terms.

Altering the definition to allow grounded terms being substituted for free variables ensures compatibility of the following proofs with the quantifier and term rules. We say a formula $@_s\varphi$ on a *FHTL* tableau branch Θ is a *root subformula* if φ is a quasi subformula of the root formula of the tableau.

Lemma 2.1 (Subformula Property). *Where Θ is a tableau branch in the FHTL calculus, and a formula $@_s\varphi$ occurs on Θ where φ is not of the form a, Fa, or Pa for a a nominal, or $u = t$ for optionally prefixed closed terms t and u, then φ is a root subformula.*

Proof. This is verified by checking the tableau rules. □

Definition 2.2 ($\prec_\Theta$)**.** Where Θ is a tableau branch in the *FHTL* calculus, if a nominal a is introduced to the branch by application of of F or P to a premise $@_s\varphi$, we say a is *generated* by s on Θ and write $s \prec_\Theta a$. We write $\prec^*_\Theta$ to denote the reflexive and transitive closure of $\prec_\Theta$.

Definition 2.3 (N_Θ)**.** The set of nominals and state variables which occur on Θ is written N_Θ

Lemma 2.2. *Where Θ is a tableau branch in the FHTL calculus, the graph $G = (N_\Theta, \prec_\Theta)$ is a wellfounded finitely branching tree.*

Proof. Each aspect is proved below:

- *Wellfoundedness of trees in G*

  We have that if $a \prec_\Theta b$ then the first occurence of a on Θ is before the first occurence of b, thus by induction any subset of N_Θ under the relation $\prec_\Theta$ has a least element and each tree in G is wellfounded.

- *G is a tree*

  Every nominal in N_Θ can be generated by at most one other nominal, and every nominal in N_Θ must have one of the finitely many nominals in the root formula as an ancestor.

- *G is finitely branching*

  We show G is finitely branching by showing that given a nominal a, there can only be finitely many distinct nominals b such that $a \prec_\Theta b$. Each nominal b such that $a \prec_\Theta b$ is generated by applying one of F, P to a premise of the form $@_iF\varphi$ or $@_iP\varphi$ respectively, where by our restrictions, either φ is not a nominal, or the entire premise is a root subformula. Since there can only be finitely many root subformulae of the form of one of the the possible premises, where i is the prefix nominal in each case, only finitely many new nominals have been generated from i. Thus G is finitely branching.

□

Lemma 2.3. *Where Θ is a tableau branch in the FHTL calculus, Θ is infinite if and only if there exists an infinite chain of nominals and state variables $a_1 \prec_\Theta a_2 \prec_\Theta \ldots \prec_\Theta a_n \prec_\Theta \ldots$*

Proof. Since the structure of the formulae and tableau rules are not involved in the proof from Bolander and Blackburn (2009) holds here as well.

□

Definition 2.4 ($\subseteq_\Theta$)**.** Where a and b are nominals occuring on an *FHTL* tableau branch Θ, a is *included* in b with respect to Θ if for any root subformula φ, if $@_a\varphi$ occurs on Θ then $@_b\varphi$ also occurs on Θ, similarly for their negations. If a is included in b with respect to Θ, and the first occurence of b on Θ is before the first occurence of a on Θ, then we write $a \subseteq_\Theta b$.

Definition 2.5 ($\sim_\Theta$)**.** Where Θ is a *FHTL* tableau branch, define a binary relation $\sim_\Theta$ on N_Θ by $a \sim_\Theta b$ if and only if $@_ab$ occurs on Θ. Let $\sim^*_\Theta$ be reflexive, transitive, and symmetric closure of $\sim_\Theta$.

Definition 2.6 ($W, \approx$)**.** Let W be the subset of N_Θ containing any nominal a with the property that there is no nominal b such that $a \subseteq_\Theta b$. Let $\approx$ be the restriction of $\sim_\Theta$ to W.

2.3 Tableau Construction

Definition 2.7 (**Closed and open).** If a tableau branch contains a formula $@_s\varphi$ and its negation $@_s\neg\varphi$ we say the branch is *closed*. If every branch of the tableau is closed we say the tableau itself is closed. If a tableau or branch is not closed we say it is *open*.

A closed tableau is a proof of the unsatisfiability of the tableau's root formula, i.e. there is no model or assignment of variables in which it holds. The question of when a tableau indicates satisfiability of the root formula leads us to our next definition.

Definition 2.8 (**Saturation).** A tableau branch is *saturated* if no more rules can be applied to the branch in a way that satisfies their constraints. If every branch of the tableau is saturated we say the tableau is saturated.

Lemma 2.4 (**Decreasing length).** *Let Θ be a FHTL tableau branch. For any nominal a on Θ, $m_\Theta(a) \leq |root_\Theta| - d_\Theta(a)$.*

Proof. The proof is by induction on the depth of a. For the base case, where $d_\Theta(a) = 0$, we have that a a is a root nominal and consequently $m_\Theta(a) \leq |root_\Theta|$. For the inductive step let b be nominal such that $d_\Theta(b) > 0$. We need to demonstrate $m_\Theta(b) \leq |root_\Theta| - d_\Theta(b)$ assuming the inequality holds for all nominals of lower depth. If there are no root subformulae true at b on Θ then definitionally $m_\Theta(b) = -\infty$ and the inequality holds. Otherwise where φ is the earliest formula (without loss of generality) of maximal length for which $@_b\varphi$ is a root subformula on Θ, $@_b\varphi$ has been introduced by application of an *FHTL* tableau rule to a root subformula. The formula $@_b\varphi$ cannot have been introduced to the branch by an application of one of the propositional or quantifier rules since that would contradict the maximality of φ, nor can it have been introduced to the branch by application of **(sub)** or any term rules, since then $@_b\varphi$ would not be the earliest maximal formula. Assume $@_b\varphi$ has been introduced to the branch by applying **(@)** to a root subformula of the form $@_c@_b\varphi$, then b is a root nominal which contradicts the inductive assumption and **(@)** cannot have been the rule introducing $@_b\varphi$, similarly if $@_b\varphi$ is introduced by **($\neg$@)**. Assume $@_b\varphi$ is introduced by **(nom)**, then it was introduced from premises of the form $@_a\varphi$, $@_ab$ for some nominal a,

□

3 Model Checking via Tableaux

For our task of AMR inference, we are not concered with the determining the general satisfiability or validity of an AMR formula translated into *FHTL*, but rather whether it holds in the smallest model consistent with an established set of *FTHL* translations of AMR sentences. This model will necessarily be finite, since across any finite number of AMR sentences only a finite number of times and entities can be referenced. In particular, we have a case of a local model-checking problem where given formula φ, a finite *FHTL* model structure $\mathfrak{M}$, a time t in $\mathfrak{M}$, and a variable assignment g, we need to determine whether $\mathfrak{M}, t \models_g \varphi$ (Müller-Olm et al., 1999).

Consequently our use of tableaux for *FHTL* formulae will provide a decision procedure for their satisfiability within a finite model generated by some set of AMR sentences, rather than their general validity or invalidity, as is usually the case with tableaux methods. We develop an approach to using tableaux as a means of model checking for *FHTL* based on the approach in Bohn et al. (1998).

3.1 Systematic tableau construction

Definition 3.1 (*Tableau construction algorithm*). Where $@a_a\varphi$ is the formula whose validity we are deciding. We inductively define a sequence of finite tableaux $\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \ldots$ each where each element of the sequence is embedded in all of its successors. We let $\mathcal{T}_0$ be the finite tableau consisting of the formula $\neg@a_a\varphi$. Assuming the finite tableau $\mathcal{T}_n$ is defined. If possible, apply an arbitrary *FHTL* tableau rule with the following restriction.

- **(Loop check)** The rule F is not applied to a formula occurence $@_aF\varphi$ at a branch Θ if there is a nominal b such that $a \subseteq_\Theta b$, and simlarly for the rule P.

Let $\mathcal{T}_{n+1}$ be the resulting tableau.

Theorem 3.1. *The systematic tableau construction algorithm terminates.*

Proof. Suppose by contradiction the ned □

3.2 Node Annotation

The approach involves annotating each node of an open branch with the variable assignments in the model which witness the formula at the node, building inductively from the terminal nodes. If the root formula of the tableau with at least one open branch can be annotated with non-empty set of variable assignments, then it is satisfiable in the model. If a tableau is closed then the root formula $@_s\varphi$ is unsatisfiable. As a result if the root formula is of the form $@_s\neg\psi$ then this constitutes a proof of the validity of $@_s\psi$ by contradiction. We now view each node in the tableau graph as a pair $(@_s\varphi, \mathcal{V})$, of the formula at the node and the set $\mathcal{V}$ of variable assignments in our model $\mathfrak{M}$ which witness the formula. We define an annotation function $ann(@_s\varphi) = \mathcal{V}$ beginning with terminal nodes:

$$ann(@_s\varphi) = \{g \mid \mathfrak{M}, I_{nom}(s) \vDash_g \varphi\}$$

4 AMR Interpretation in *FHTL*

4.1 Examples

(1) a. Carl filled out the forms and everyone will submit them tomorrow.

b.

```
(a / and
   :op1 (s / scope
      :pred (f / fill-out-03 :ongoing - :complete + :time (b / before :op1 (n / now))
         :ARG0 (p / person
            :name (n2 / name
               :op "Carl"))
         :ARG1 (f2 / form))
      :ARG0 p
      :ARG1 f2)
   :op2 (s2 / scope
      :pred (m / submit-01 :ongoing - :complete + :time (a2 / after :op1 n)
         :ARG0 (p2 / person
            :mod (a3 / all))
         :ARG1 f2)
      :ARG0 f2
      :ARG1 p2))
```

c. Technically correct:

$@_{now}\exists x[\texttt{form}(x) \wedge P\texttt{fill-out-03}(\texttt{Carl}, x)] \wedge @_{now}\exists x[\texttt{form}(x) \wedge \forall y[\texttt{person}(y) \rightarrow F\texttt{submit-01}(y, x)]]$

d. Correct wrt plurality:

$@_{now}\forall x[\texttt{form}(x) \wedge P\texttt{fill-out-03}(\texttt{Carl}, x)] \wedge @_{now}\forall x[\texttt{form}(x) \wedge \forall y[\texttt{person}(y) \rightarrow F\texttt{submit-01}(y, x)]]$

e. Correct wrt reentrance (but not plurality) (maybe requires the passive for singular case?):

$@_{now}\exists x[\texttt{form}(x) \wedge P\texttt{fill-out-03}(\texttt{Carl}, x) \wedge \forall y[\texttt{person}(y) \rightarrow F\texttt{submit-01}(y, x)]]$

(2) a. It was impossible not to notice the car.

b.

```
(s / scope
      :pred (p / possible-01
         :ARG0 (n / notice-01 :ongoing - :complete + :time (b / before :op1 (n2 / now))
            :polarity (n3 / not)
            :ARG1 (c / car)
         :polarity (n4 / not))
      :ARG0 n4
      :ARG1 p))
```

c. Incorrect:

$@_{now}\neg F\exists x[\texttt{car}(x) \wedge \neg P\texttt{notice-01}(x)]$

d. Technically correct:

$@_{now}\neg F\exists x[\texttt{car}(x) \wedge \neg\forall y[\texttt{person}(y) \rightarrow P\texttt{notice-01}(x, y)]]$

e. Correct wrt particularity of the car:

$@_{now}\neg F\neg\forall x[\texttt{person}(y) \rightarrow P\texttt{notice-01}(\texttt{car}, y)]]$

NB: Will complete these translations in full.

4.2 Extraction Steps

With the chosen annotation, the root node can consist of either a logical connective (`and`, `or`, or `cond`) linking two AMR graphs, or a `scope` node with its following predicate and arguments.

4.3 General Extraction Algorithm

Algorithm 1: Basic transformation into *FHTL* clauses and connectives.

```
Input: AMR sentence
Output: FHTL formula
Def InterpretEntry(AMR):
    root = Root(AMR)
    now = current date/time
    if root ∈ {and, or, cond} then
        connective = filter(root, {∧, ∨, →})
        clauses = []
        for op ∈ Children(root) do
            append(clauses, InterpretClause (op))
        end
        return @_now join(connective, clauses)
    end
    return @_now InterpretClause (root)

Def InterpretClause(AMR):
    time = Time (AMR)
    nominal = Reference (time)
    tense = Tense (time)
    pred = Pred (AMR)(tense)
    Arg_0, Arg_1 = GetArgs (AMR)
    return @_nominal Apply (Arg_0, Apply (Arg_0, pred))
```

Algorithm 2: Supporting definitions.

Input: AMR sentence
Output: *FHTL* formula

```
Def Apply(pred_1, pred_2):
    return λφ.pred_1(λψ.pred_2(λγ.φ(ψ(γ))))

Def InterpretPred(UnaryPred):
    if hasMods(UnaryPred) then
        mods = [name(UnaryPred)]
        for mod ∈ Children(UnaryPred) do
            append(mods, name(mod)(x))
        end
        FinalPred = λx. join(mods, ∧)
    end
    else
        FinalPred = λx.name(UnaryPred)(x)
    end
    if Quant(UnaryPred) == :all then
        return λk.∀x[FinalPred(x) → k(x)]
    end
    else
        return λk.∃x[FinalPred(x) ∧ k(x)]
    end
```

References

Girish Shivanand Bhat. 1998. *Tableau-based approaches to model-checking*. Ph.D. thesis, North Carolina State University.

Patrick Blackburn and Klaus Frovin Jørgensen. 2012. Indexical hybrid tense logic. *Advances in Modal Logic*, 9:144–60.

Patrick Blackburn and Maarten Marx. 2002. Tableaux for quantified hybrid logic. In *Automated Reasoning with Analytic Tableaux and Related Methods*, pages 38–52, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jürgen Bohn, Werner Damm, Orna Grumberg, Hardi Hungar, and Karen Laster. 1998. First-order-ctl model checking. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 283–294. Springer.

Thomas Bolander and Patrick Blackburn. 2009. Terminating tableau calculi for hybrid logics extending k. *Electronic Notes in Theoretical Computer Science*, 231:21–39.

Lucia Donatelli, Michael Regan, William Croft, and Nathan Schneider. 2018. Annotation of tense and aspect semantics for sentential AMR. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 96–108, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Melvin Fitting. 1988. First-order modal tableaux. *Journal of Automated Reasoning*, 4(2):191–213.

Melvin Fitting and Richard L Mendelsohn. 1998. *First-Order Modal Logic*, volume 277. Springer Science & Business Media.

Jens Ulrik Hansen. 2007. A tableau system for a first-order hybrid logic. In *Proceedings of the International Workshop on Hybrid Logic (HyLo 2007)*, pages 32–40.

Markus Müller-Olm, David Schmidt, and Bernhard Steffen. 1999. Model-checking. In *International Static Analysis Symposium*, pages 330–354. Springer.

James Pustejovsky, Ken Lai, and Nianwen Xue. 2019. Modeling quantification and scope in Abstract Meaning Representations. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 28–33, Florence, Italy. Association for Computational Linguistics.