Formalization of AMR Inference via Hybrid Logic Tableaux

Eli Goldner

May 23, 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 Blackburn and Marx's tableau method for quantified hybrid logic (*QHL*) and Blackburn and Jørgensen's tableu method for basic hybrid tense logic (*BHTL*). We motivate the merging of these AMR variants, present their interpretation and inference in the combination of *QHL* and *BHTL*, which we will call *QHTL* (quantified hybrid tense logic), and demonstrate *QHTL*'s the soundess, completeness, and decidability.

1 Merging Quantified Hybrid Logic and Indexical Hybrid Tense Logic

1.1 Background

1.2 Quantified Hybrid Logic

1.3 Basic Hybrid Tense Logic

2 Quantified Hybrid Tense Logic - Syntax and Semantics

The syntax of $QHTL$ is identical to QHL except uses of $\downarrow$ as in $\downarrow w.\phi$ are omitted along with $\Box$ and $\Diamond$ as in $\Box\phi$ and $\Diamond\phi$. $\Box$ 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 QHL, 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:

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

Since we want the domain of quantification to be indexed over the collection of nominals/times, we alter the QHL model definition to a structure:

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

Identical to the definition for a QHL 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. That is, for every $t \in T$, (D_t, I_t) is a first-order model where:

- $I_t(q) \in D_t$ where q is a unary function symbol.
- $I_t(P) \subseteq^k D_t$ 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.

Free variables are handled similarly as in QHL. A $QHTL$ 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_t where t is the time assigned to the state variable by g. Thus given a model and an assignment g, the interpretation of terms t denoted by $\bar{t}$ is defined as:

- $\overline{x} = g_t(x)$ for x a variable and the relevant $t \in T$.
- $\overline{c} = I_t(c)$ for c a constant and some $t \in T$.
  - For q a unary function symbol:

    For n a nominal:

$$\overline{@_n q} = I_{I_{nom}}(n)$$

  - For n a state variable:

$$\overline{@_n q} = I_{g(n)}(q)$$

With the final adjustment of having $g^x_{d,s}$ denoting the assignment which is just like g_s except $g_s(x) = d$ for $d \in g(s)$, we can proceed with the inductive definition for satisfaction of a formula give a model $\mathfrak{M}$, a variable assignment g, and a state s. The inductive definition is:

$$\begin{array}{lll}
\mathfrak{M}, g, s \Vdash P(t_1, \ldots, t_n) & \iff & \langle \overline{t_1}, \ldots, \overline{t_n} \rangle \in I_s(P) \\
\mathfrak{M}, g, s \Vdash t_i = t_j & \iff & \overline{t_i} = \overline{t_j} \\
\mathfrak{M}, g, s \Vdash n & \iff & I_{nom}(n) = s, \text{for } n \text{ a nominal} \\
\mathfrak{M}, g, s \Vdash w & \iff & g(w) = s, \text{for } s \text{ a state variable} \\
\mathfrak{M}, g, s \Vdash \neg\phi & \iff & \mathfrak{M}, g, s \nVdash \phi \\
\mathfrak{M}, g, s \Vdash \phi \wedge \psi & \iff & \mathfrak{M}, g, s \Vdash \phi \text{ and } \mathfrak{M}, g, s \Vdash \psi \\
\mathfrak{M}, g, s \Vdash \phi \vee \psi & \iff & \mathfrak{M}, g, s \Vdash \phi \text{ or } \mathfrak{M}, g, s \Vdash \psi \\
\mathfrak{M}, g, s \Vdash \phi \to \psi & \iff & \mathfrak{M}, g, s \Vdash \phi \text{ implies } \mathfrak{M}, g, s \Vdash \psi \\
\mathfrak{M}, g, s \Vdash \exists x\phi & \iff & \mathfrak{M}, g^x_{d,s}, s \Vdash \phi \text{ for some } d \in D_s \\
\mathfrak{M}, g, s \Vdash \forall x\phi & \iff & \mathfrak{M}, g^x_{d,s}, s \Vdash \phi \text{ for all } d \in D_s \\
\mathfrak{M}, g, s \Vdash F\phi & \iff & \mathfrak{M}, g, t \Vdash \phi \text{ for some } t \in T \text{ such that } Rst \\
\mathfrak{M}, g, s \Vdash G\phi & \iff & \mathfrak{M}, g, t \Vdash \phi \text{ for all } t \in T \text{ such that } Rst \\
\mathfrak{M}, g, s \Vdash P\phi & \iff & \mathfrak{M}, g, t \Vdash \phi \text{ for some } t \in T \text{ such that } Rts \\
\mathfrak{M}, g, s \Vdash H\phi & \iff & \mathfrak{M}, g, t \Vdash \phi \text{ for all } t \in T \text{ such that } Rts \\
\mathfrak{M}, g, s \Vdash @_n\phi & \iff & \mathfrak{M}, g, I_{nom}(n) \Vdash \phi \text{ for } n \text{ a nominal} \\
\mathfrak{M}, g, s \Vdash @_w\phi & \iff & \mathfrak{M}, g, g(w) \Vdash \phi \text{ for } w \text{ a state variable}
\end{array}$$

2.1 The Tableau Calculus

Non-branching rules:

Branching rules:

Binary rules:

$$\frac{@_s\neg\phi}{\neg @_s\phi}\ [\neg] \qquad \frac{\neg @_s\neg\phi}{@_s\phi}\ [\neg\neg]$$

$$\frac{@_s @_t\phi}{@_t\phi}\ [@] \qquad \frac{\neg @_s @_t\phi}{\neg @_t\phi}\ [\neg @]$$

$$\frac{@_s F\phi}{\begin{array}{c}@_s Fa\\ @_a\phi\end{array}}\ [F] \qquad \frac{@_s P\phi}{\begin{array}{c}@_s Pa\\ @_a\phi\end{array}}\ [P]$$

$$\frac{\neg @_s G\phi}{\begin{array}{c}@_s Fa\\ \neg @_a\phi\end{array}}\ [\neg G] \qquad \frac{@_s H\phi}{\begin{array}{c}@_s Pa\\ \neg @_a\phi\end{array}}\ [\neg H]$$

$$\frac{@_s Pt}{@_t Fs}\ P\text{-trans} \qquad \frac{@_s Ft}{@_t Ps}\ F\text{-trans}$$

$$\frac{@_s\exists x\phi(x)}{@_c\phi(c)}\ [\exists] \qquad \frac{@_s\neg\forall x\phi(x)}{\neg @_c\phi(c)}\ [\neg\forall]$$

$$\frac{@_s\forall x\phi(x)}{@_c\phi(c)}\ [\forall] \qquad \frac{\neg @_s\exists x\phi(x)}{\neg @_c\phi(c)}\ [\neg\exists]$$

$$\frac{[s \text{ on the branch}]}{@_s s}\ [\text{ Ref }] \qquad \frac{@_t s}{@_s t}\ [\text{ Sym }]$$

$$\frac{}{t = t}\ [\ =\text{-Ref }] \qquad \frac{@_n m}{@_n q = @_m q}\ [\text{ DD }]$$

$$\frac{@_n(t_i = t_j)}{t_i = t_j}\ @= \qquad \frac{\neg @_n(t_i = t_j)}{\neg(t_i = t_j)}\ \neg @=$$

Figure 1: Non-Branching Rules.

$$\frac{@_s(\phi\vee\psi)}{@_s\phi \mid @_s\psi}\ \vee \qquad \frac{\neg @_s(\phi\wedge\psi)}{\neg @_s\phi \mid \neg @_s\psi}\ \neg\wedge$$

$$\frac{@_s(\phi\to\psi)}{\neg @_s\phi \mid @_s\psi}\ \to$$

$$\frac{@_s H\phi \qquad @_s Pt}{@_t\phi}\ H \qquad \frac{@_s G\phi \qquad @_s Ft}{@_t\phi}\ G$$

$$\frac{@_s Pt \qquad @_t u}{@_t Pu}\ P\text{-bridge} \qquad \frac{@_s Ft \qquad @_t u}{@_t Fu}\ F\text{-bridge}$$

$$\frac{@_s t \qquad @_s\phi}{@_t\phi}\ \text{Nom} \qquad \frac{@_s t \qquad @_t\phi}{@_s\phi}\ \text{Nom}^{-1}$$

$$\frac{@_s t \qquad @_t r}{@_s r}\ \text{Trans}$$

2.2 Soundness

The proof of the soundness of the tableau method for QHTL is adapted from the proof of soundness of the tableau method for $\mathcal{H}(@)$ given in Blackburn (2000).

We can observe from the tableau rules that every formula in a tableau is of the form $@_s\phi$ or $\neg @_s\phi$. We call formulae of these forms *satisfaction statements*. Give a set of satisfaction statements Σ and a tableau rule R we develop the notion of Σ^+ as an expansion of Σ by R as follows based on the different cases for R:

1. If R is *not* a branching rule, and R takes a single formula as input, and Σ^+ is the set obtained by adding to Σ the formulae yielded by applying R to $\sigma \in \Sigma$, then we say Σ^+ is the result of expanding Σ by R.

2. If R is a binary rule, and Σ^+ is the set obtained by adding to Σ the formulae yielded by applying R to $\sigma_1, \sigma_2, \sigma_2 \in \Sigma$, then we say Σ^+ is the result of expanding Σ by R.

3. If R is a branching rule, and Σ^+ is the set obtained by adding to Σ the formulae yielded by one of the possible outcomes of applying R to $\sigma_1 \in \Sigma$, then we say Σ^+ is the result of expanding Σ by R.

4. If a nominal s belongs to some formula in Σ, then $\Sigma^+ = \Sigma \cup @a_s s$ is the result of expanding Σ by Ref.

5. If a nominal s belongs to some formula in Σ, then $\Sigma^+ = \Sigma \cup t = t$ is the result of expanding Σ by $=$-Ref.

Definition 2.1 (Satisfiable by label)**.** Suppose Σ is a set of satisfaction statements and $\mathfrak{M} = (T, R, D_t, I_{nom}, I_t)_{t\in T}$ is a standard *QHTL* model. We say Σ is *satisfied by label* in $\mathfrak{M}$ under a *QHTL* assignment g if and and only if for all formulae in Σ:

1. If $@_s\phi \in \Sigma$ then $\mathfrak{M}, g, I_{nom}(s) \Vdash \phi$

2. If $\neg @_s\phi \in \Sigma$ then $\mathfrak{M}, g, I_{nom}(s) \nVdash \phi$

We say Σ is *satisfiable by label* if and only if there is a standard *QHTL* model and assignment in which it is satisfied by label.

Theorem 2.1 (Soundess)**.** *If Σ is a set of satisfaction statements which is satisfiable by label, then for any tableau rule R, at least one of the sets obtainable by expanding Σ by R is satisfiable by label.*

(*Proof*). We prove soundness by induction on the tableau rules, with particular attention to "existential" rules as they are called in Blackburn (2000) which introduce a new parameter on the branch. In all cases discussed below let $\mathfrak{M} = (T, R, D_t, I_{nom}, I_t)_{t\in T}$ and g be the *QHTL* model and assignment in which Σ is satisfiable by label.

- *Non-branching Rules*

  We will take the $\wedge$ rule as an example. Beginning from $@_s\phi \wedge \psi$ we have:

  $$\mathfrak{M}, g, I_{nom}(s) \Vdash \phi \wedge \psi$$

  and consequentially

  $$\mathfrak{M}, g, I_{nom}(s) \Vdash \phi \iff @_s\phi$$
  $$\mathfrak{M}, g, I_{nom}(s) \Vdash \psi \iff @_s\psi$$

  Similarly for their negations if at least one of $@_s\phi\wedge\psi$ are not satisfied in $\mathfrak{M}$ under g. Thus the results of the application of the $\wedge$ rule, $@_s\phi$ and $@_s\psi$ are satisfiable in $\mathfrak{M}$ under g and the expansion of Σ by $\wedge$ is satisfiable by label. The proof for other non-branching rules are analogous.

- *Binary Rules* We will take the H rule as an example. Beginning with $@_s H\phi$ and $@_s Pt$ we have from the former:

$$\mathfrak{M}, g, t \Vdash \phi \text{ for all } t \in T \text{ such that } RtI_{nom}(s)$$

and from the latter:

$$\mathfrak{M}, g, t' \Vdash t \text{ for some } t' \in T \text{ such that } RtI_{nom}(s)$$

and consequentially since Rts

$$\mathfrak{M}, g, I_{nom}(t) \Vdash \phi \iff @_t\phi$$

Similarly for their negations if at least one of $@_s H\phi$ and $@_s Pt$ are not satisfied in $\mathfrak{M}$ under g. Thus the result of application of the H rule, $@_t\phi$ is satisfiable in $\mathfrak{M}$ under g and the expansion of Σ by H is satisfiable by label. The

proof for other binary rules is analogous.

- *Branching Rules* We will take the $\vee$ rule as an example. Beginning from $@_s\phi \vee \psi$ if it's satisfied we have:

$$\mathfrak{M}, g, I_{nom}(s) \Vdash \phi \vee \psi$$

and consequentially at least one of

$$\mathfrak{M}, g, I_{nom}(s) \Vdash \phi \iff @_s\phi$$

or

$$\mathfrak{M}, g, I_{nom}(s) \Vdash \psi \iff @_s\psi$$

Similarly for their negations if $@_s\phi \vee \psi$ is not satisfied in $\mathfrak{M}$ under g. Thus at least one of the results of the application of the $\vee$ rule, $@_s\phi$ or $@_s\psi$ or their negations are satisfiable in $\mathfrak{M}$ under g and the expansion of Σ by $\vee$ is satisfiable by label. The proof for other branching rules are analogous.

- *"Existential" Rules* We will take the $\forall$ rule as an example. Beginning from $@_s\forall x\phi(x)$ if it's satisfied we have (where $s' = I_{nom}(s)$):

$$\mathfrak{M}, g^x_{d,s'}, s' \Vdash \phi \text{ for all } d \in D_{s'} \iff @_s\phi(t')$$

Similarly for its negation if $@_s\forall x\phi(x)$ is not satisfied in $\mathfrak{M}$ under g. In accordance with the constraints for the rule we can select t' to be any grounded term on the branch which is also a member of $D_{I_{nom}(s)}$ Thus the result of the application of the $\forall$ rule, or its negations are satisfiable in $\mathfrak{M}$ under g and the expansion of Σ by $\forall$ is satisfiable by label. The proof for $\neg\exists$ is analogous.

- *Ref rules* $t = t$ is a tautology, invariant of model or assignment. For $@_s s$, we begin with having s is the branch, as result we certainly have

$$\mathfrak{M}, g, I_{nom}(s) \Vdash I_{nom}(s) \iff @_s s$$

Thus the expansion of Σ by $= -\mathsf{Ref}$ or Ref is satisfiable by label.

$\square$

2.3 Decidability

The proof of the tableau construction algoritheorem's termination is adapted from the proof given in Bolander and Braüner (2006) for the termination of the tableau construction algoritheorem for $\mathcal{H}(@)$ as described in Blackburn (2000) except extended with the universal modality.

Lemma 2.2 (Quasi-subformula property)**.**

Definition 2.2.

Theorem 2.3.

$$
\begin{aligned}
AT_x(p) &:= Px\\
AT_x(n) &:= x = n\\
AT_x(w) &:= x = w\\
AT_x(\neg\phi) &:= \langle \lambda x.\neg AT_x(\phi)\rangle(x)\\
AT_x(\phi \wedge \psi) &:= \langle \lambda x.AT_x(\phi) \wedge AT_x(\psi)\rangle(x)\\
AT_x(G\phi) &:= \langle \lambda x.\forall y(Rxy \to AT_y(\phi))\rangle(x)\\
AT_x(H\phi) &:= \langle \lambda x.\forall y(Ryx \to AT_y(\phi))\rangle(x)\\
AT_x(@_n\phi) &:= \langle \lambda x.\forall x(x = n \to AT_x(\phi))\rangle(x)\\
AT_x(P(t_1,\dots,t_k)) &:= P'(x,t_1,\dots,t_k)\\
AT_x(t_i = t_j) &:= \langle \lambda x.t_i = t_j\rangle(x)\\
AT_x(\forall v\phi) &:= \langle \lambda x.\forall v AT_x(\phi)\rangle(x)
\end{aligned}
$$

Figure 2: TEST

$$
\begin{aligned}
AT_x^-(Px) &:= p\\
AT_x^-(x = n) &:= n\\
AT_x^-(x = w) &:= w\\
AT_x^-(\langle \lambda x.\neg\phi\rangle(x)) &:= \neg AT_x^-(\phi)\\
AT_x^-(\langle \lambda x.\neg\phi \wedge \psi\rangle(x)) &:= AT_x^-(\phi) \wedge AT_x^-(\psi)\\
AT_x^-(\langle \lambda x.\forall y(Rxy \to \phi)\rangle(x)) &:= GAT_y^-(\phi)\\
AT_x^-(\langle \lambda x.\forall y(Ryx \to \phi)\rangle(x)) &:= HAT_y^-(\phi)\\
AT_x^-(\langle \lambda x.\forall x(x = n \to \phi)\rangle(x)) &:= @_n AT_x^-(\phi)\\
AT_x^-(P'(x,t_1,\dots,t_k)) &:= P(t_1,\dots,t_k)\\
AT_x^-(\langle \lambda x.t_i = t_j\rangle(x)) &:= t_i = t_j\\
AT_x^-(\langle \lambda x.\forall v\phi\rangle(x)) &:= \forall v AT_x^-(\phi)
\end{aligned}
$$

Figure 3: TEST

$$
\begin{aligned}
P(t)^* &:= @_t p\\
(t = u)^* &:= @_t u\\
(Rst)^* &:= @_s Ft\\
(\langle \lambda x.\phi\rangle(t))^* &:= @_t AT_x^-(\langle \lambda x.\phi\rangle(x))\\
(\langle \lambda y.\phi\rangle(t))^* &:= @_t AT_y^-(\langle \lambda y.\phi\rangle(y))\\
(AT_x(\phi)[t/x])^* &:= @_t\phi\\
(\neg AT_x(\phi)[t/x])^* &:= \neg @_t\phi\\
P'(s,t_1,\dots,t_k)^* &:= @_s P(t_1,\dots,t_k)\\
(t_i = t_j)^* &:= t_i t_j
\end{aligned}
$$

Figure 4: TEST

Corollary 1.

Definition 2.3.

Theorem 2.4.

Definition 2.4.

Proposition 1.

Definition 2.5.

Definition 2.6.

Theorem 2.5.

References

Patrick Blackburn. 2000. Internalizing labelled deduction. *Journal of Logic and Computation*, 10(1):137–168.

Thomas Bolander and Torben Braüner. 2006. Tableau-based Decision Procedures for Hybrid Logic. *Journal of Logic and Computation*, 16(6):737–763.

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.

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.