Formalization of AMR Inference via Hybrid Logic Tableaux

Eli Goldner

May 28, 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 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 *QHL* and *BHTL*, which we will call *QHTL* (quantified hybrid tense logic), and demonstrate *QHTL*'s soundess, completeness, and decidability.

1 Introduction

2 Related Work

We draw from a number of areas which motivate this approach, namely designing semantic representations for inferentiability, the history and goals of AMR and its different annotations, and hybrid logic with its variants and their accompanying tableaux methods for proof.

2.1 Inference in Semantic Representation

Semantic representation is the task of representing meaning at the sentential and potentially the discourse levels of language in a formally specifiable way.

2.2 Discourse Representation Theory

Discourse representation theory was developed in 1981 by Hans Kamp (Kamp and Reyle, 1993) and a simlar theory independently in 1982 by Irene Heim under the name of *File Change Semantics* (FCS) (Heim, 1982).

Discourse representation theory (DRT) uses discourse representation structures (DRS) to represent a listeners understanding of the discourse content as it develops over time.

A DRS has two main components:

- A set of *referents*, representing entities which are in the discourse.
- A set of *conditions*, representing information about the discourse referents.

For instance, the single sentence *The farmer saw the rancher.* admits the DRS:

(1) [x, y : farmer(x), rancher(y), saw(x,y)]

The conditions in 1 are *simple*, that is they do not take the form of DRSs or constructions of DRSs, an example of a DRS with complex conditionals, would be for the sentence *If a farmer owns a dog, they feed it.*

(2) $[_1 x$: farmer(x),

$[_2 y$: dog(y), owns(x, y) $\rightarrow$ $[_3 v, w$: feeds(v, w)]]]

2.2.1 Segmented Discourse Representation Theory

2.3 Description Logics

More expressive than propositional logic, less expressive than first-order logic. Examples: OWL-DL (Sirin et al., 2007), (Cimiano et al., 2014).

A DL models *concepts*, *roles*, and *individuals*, and relationships among them .

DL relates roles and concepts via *axioms*, the key modeling concept of DL, in contrast to frame spacifications in AI which declare and completely define classes (Grau et al., 2008).

First DL-based knowledge representation system was KL-ONE (Brachman and Schmolze, 1985).

Other 1980s DL systems – *substructural subsumption algorithms*, lower expressivity but polynomial time reasoning (Van Harmelen et al., 2008, Chapter 3).

Introduction of tableaux based algorithms in 90s allowed greater efficiency on problems in more expressive DL.

Modern DL (Sirin et al., 2007) (Tsarkov and Horrocks, 2006) and RacerPro (Haarslev et al., 2012) (from Racer (Haarslev and Möller, 2001))

Advantages: Comparatively expressive but still decidable, good for domain specific knowledge.

Disadvantages: Not expressive enough for semantic representation of language, designed to be a knowledge representation (Schubert, 2015a).

2.4 Situation Semantics and Related

2.4.1 Situation Semantics

2.4.2 Type Theory with Records

2.4.3 PTT

2.4.4 KoS

2.5 Episodic Logic

Episodic Logic is a Montague-style logical form based semantic representation and knowledge representation, with relative strength in semantic expressivity and inferetiability in comparison to other semantic representation, making it better comparatively better suited for deep NLU (Schubert, 2015b). Episodic Logic allows for generalized quantifiers, lambda abstraction, reification and modification of sentences and predicates, intensional predicates, unreliable generalizations, and explicit situational variables (Schubert and Hwang, 2000). Episodic Logic with its inference engine EPILOG provide a way to capture a relatively comprehensive range of semantic phenomena compared to other SR/KRs, in a way which affords inference about semantic data at a comparable efficiency with automated inference engines for first order logic (Schubert, 2015b). It also has an associated knowledge base KNEXT which is capable of parsing sentences into factoids, generalizing them (through a process called quantificational sharpening), and making certain judgements about whether a generalized factoid is redudant or inconsistent with anything established in the knowledge base.

2.6 Semantic Features in AMR and Possbility of Inference

Separating argument structure in AMR from logical structure, enables translation from AMR to DRT (Bos, 2020)

AMR expressivity without recurrent variables (and with no more than one universal quantifier per sentence) are in the decidable two-variable fragment of first-order logic (Bos, 2016)

Extension of sentential AMR to incorporate a coarse grained treatment of tense and aspect (Donatelli et al., 2018)

Continuation based semantics for translating AMR into first-order logic in a way that preserves projection phenomena such as quantification, negation, bound variables, and donkey anaphora, which better affords inference than other first-order logic semantics for AMR (Lai et al., 2020).

2.7 Hybrid Logic and Our Chosen Semantic Features

3 Merging AMR Annotations

3.1 AMR Annotated for Tense and Aspect

3.2 AMR Annotated for Scope and Quantification

4 Merging Quantified Hybrid Logic and Indexical Hybrid Tense Logic

4.1 Background

4.2 Quantified Hybrid Logic

4.3 Basic Hybrid Tense Logic

5 Quantified Hybrid Tense Logic - Syntax and Semantics

The syntax of *QHTL* is identical to *QHL* as given in Blackburn and Marx (2002) 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:

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

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

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

Thus with varying domain semantics a *QHTL* model is 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. 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 *QHTL* 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 *QHL*. Where again $D = \cup_{t\in T} D_t$, a *QHTL* assignment is a function:

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

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 $g'(x) \in D_t$. Given a model $\mathfrak{M}$, a variable assignment g, and a state s, the inductive definition is:

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

5.1 The Tableau Calculus

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

Figure 1: Negation rules

For Nom we have the constraint that if the premise $@_t\phi$ are of the form $@_t Xc$ where $X \in \{F, P, \neg G, \neg H\}$ and c is a nominal or state variable, then $@_t\phi$ is a root subformula. Similarly for Nom^{-1} and the premise $@_s\phi$.

In all rules in **??**, the nominal a is new to the branch. We have the additional constraint that if ϕ in the premise is a nominal or state variable, then the premise must be a root subformula in order for the rule to be applicable.

Following Fitting and Mendelsohn (1998) we assume for each nominal 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 never share the same parameter. Informally we write p_s to indicate a parameter is associated with a nominal s.

$$\frac{@_s(\phi \wedge \psi)}{\begin{array}{c}@_s\phi\\ @_s\psi\end{array}}\,[\wedge] \qquad \frac{\neg @_s(\phi \vee \psi)}{\begin{array}{c}\neg @_s\phi\\ \neg @_s\psi\end{array}}\,[\neg\vee] \qquad \frac{\neg @_s(\phi \to \psi)}{\begin{array}{c}@_s\phi\\ \neg @_s\psi\end{array}}\,[\neg \to]$$

Figure 2: Conjunctive rules.

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

Figure 3: Disjunctive rules.

We also introduce the notion of a grounded term. A grounded term is either a first-order constant, a parameter, or a grounded definite description, i.e. a term of the form $@_n q$ for n a nominal and q a unary function symbol.

In the existential rules **??**, p_s is a parameter associated with the nominal s, with the constraint that it is new to the branch. Since parameters are never quantified over, p_s is free in $\phi[p_s/x]$.

In the universal rules **??** t is a grounded term on the branch which exists at D_s.

5.2 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 $\Sigma^+ = \Sigma \cup \{@_s s\}$ is the result of expanding Σ by Ref.

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

Definition 5.1 (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 $\mathfrak{M}, g, I_{nom}(s) \Vdash \phi$

$$\frac{@_sH\phi \qquad @_sPt}{@_t\phi}\,[H] \qquad\qquad \frac{@_sG\phi \qquad @_sFt}{@_t\phi}\,[G]$$

Figure 6: G and H rules

2. If $\neg @_s \phi \in \Sigma$ then $\mathfrak{M}, g, I_{nom}(s) \not\Vdash \phi$

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 5.1 (Soundness). *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 rules which introduce nominals new to the branch, namely $\{F, P, \neg G, \neg H\}$ **??**, and rules which introduce new parameters to the branch, namely the universal rules **??** and existential rules **??**. In all cases discussed below let

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

be *QHTL* model and g the 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 proofs

  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

  proofs 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 proofs for other branching rules are analogous.

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

$$\mathfrak{M}, g', s' \Vdash \phi \text{ for every } x\text{-variant of } g \text{ at } s$$

  That is for every c in $D_{I_{nom}(s)}$, $\phi[t/x]$ is satisfied in $\mathfrak{M}$ under g, similarly for $\neg\phi[t/x]$ 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 proofs for $\exists$, $\neg\exists$, and $\neg\forall$ are analogous.

- *Rules Introducing a Nominal to the Branch*

  We will take the F rule as am example. Beginning from $@_s F\phi$, if it's satisfied we have:

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

  Let a denote a nominal such that $RI_{nom}(s)I_{nom}(a)$ as above. As a result we have:

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

$$\mathfrak{M}, g, t \Vdash a \text{ for some } t \in T \text{ such that } RI_{nom}(s)t \iff$$

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

  and

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

  Similarly for their negations if $@_s F\phi$ is not satisfied in $\mathfrak{M}$ under g.

- *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.

  Using this we have demonstrated that the results of applicatition of a tableau rule to one or more premises reflect the validity or non-validity of the premises. $\square$

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

Figure 10: *QHTL* Formulae to First-order Formulae Translation.

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

Figure 11: First-order Formulae to *QHTL* Formulae Translation.

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

Figure 12: Translation of First-order Tableau Literals to *QHTL* Literals.

5.3 Decidability

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

Definition 5.2. When a formula $@_s\phi$ occurs in a tableau branch Θ we will write $@_s\phi \in \Theta$, and say ϕ is true at s on Θ or s makes ϕ true on Θ.

Definition 5.3. Given a tableau branch Θ and a nominal or state variable s the *set of true formulae* at s on Θ, is written $T^\Theta(s)$ and defined as follows:

$$T^\Theta(s) = \{\phi \mid @_s\phi \in \Theta\}$$

Definition 5.4. 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.

2. ϕ is of the form $\neg\chi$ where χ is a subformula ψ

Lemma 5.2 (Quasi-subformula Property). *Let $\mathcal{T}$ be a tableau with the formula $@_s\phi$ as root. For any formula $@_t\psi$ occuring on $\mathcal{T}$, ψ is a quasi-subformula of ϕ.*

(*Proof*) This is verified by induction on the tableau rules.

Definition 5.5.

Theorem 5.3.

Corollary 1.

Definition 5.6.

Theorem 5.4.

Definition 5.7.

Proposition 1.

Definition 5.8.

Definition 5.9.

Theorem 5.5.

6 AMR Interpretation in Hybrid Logic

6.1 Examples

(3) a. Carl submitted the forms and everyone will sign up again 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 p2
      :ARG1 f2))
```

c. It was impossible not to notice the car.
d.

```
(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))
```

NB: Will complete these translations in full.

6.2 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.

6.3 General Extraction Algorithm

7 Future Work

7.1 $\downarrow$ and Quantification over Nominals

Main points, at the cost of undecidability with adding $\downarrow$ some additional things can be done, and at the cost of the integration of generalized quantifiers you can ostensibly handle even things like habitual aspect.

7.2 AMR Reentrancy and Non-Temoral Nominals

There are some difficulties with maintaining the usual notion of possible worlds being maximal with this idea, but there seems to be a direct sympathy between the predicative core of an AMR sentence and in general reentrancy of the nodes with the idea of a nominal as a "point of view" rather than the "name" of a world. Maybe things like epistemic logic could be helpful here.

7.3 Automated Inference and HTab

HTab (Hoffmann and Areces, 2009) provides an implementation of $\mathcal{H}(@, \mathsf{A})$, which does not natively provide a way to reason with P, H, or first-order quantification. The effort required in making the needed changes to handle these remains to be determined.

7.4 The Future of AMR and Parsing for Semantic Features

To what extent can current AMR parsers extract the needed semantic features to make full use of automated inference? Of UMR, Dialogue-AMR, and the AMR annotation variants we've used, which logistically has the best outlook?

8 Conclusion

References

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

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.

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

Johan Bos. 2016. Squib: Expressive power of Abstract Meaning Representations. *Computational Linguistics*, 42(3):527–535.

Johan Bos. 2020. Separating argument structure from logical structure in AMR. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 13–20, Barcelona Spain (online). Association for Computational Linguistics.

Ronald J. Brachman and James G. Schmolze. 1985. An overview of the kl-one knowledge representation system*. *Cognitive Science*, 9(2):171–216.

Philipp Cimiano, Christina Unger, and John McCrae. 2014. *Ontology-Based Interpretation of Natural Language*. Morgan & Claypool.

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

Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter Patel-Schneider, and Ulrike Sattler. 2008. Owl 2: The next step for owl. *Journal of Web Semantics*, 6(4):309–322. Semantic Web Challenge 2006/2007.

V. Haarslev and R. Möller. 2001. Racer system description. In *International Joint Conference on Automated Reasoning, IJCAR'2001, June 18-23, Siena, Italy*, pages 701–705. Springer-Verlag.

Volker Haarslev, Kay Hidde, Ralf Möller, and Michael Wessel. 2012. The racerpro knowledge representation and reasoning system. *Semantic Web Journal*, 3(3):267–277.

Irene Heim. 1982. *The Semantics of Definite and Indefinite Noun Phrases*. Ph.D. thesis, UMass Amherst.

Guillaume Hoffmann and Carlos Areces. 2009. Htab: a terminating tableaux system for hybrid logic. *Electronic Notes in Theoretical Computer Science*, 231:3–19. Proceedings of the 5th Workshop on Methods for Modalities (M4M5 2007).

Hans Kamp and Uwe Reyle. 1993. From discourse to logic.

Kenneth Lai, Lucia Donatelli, and James Pustejovsky. 2020. A continuation semantics for Abstract Meaning Representation. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 1–12, Barcelona Spain (online). 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.

Lenhart K. Schubert. 2015a. Semantic representation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 4132–4138. AAAI Press.

Lenhart K. Schubert. 2015b. What kinds of knowledge are needed for genuine understanding? In *IJCAI 2015 Workshop on Cognitive Knowledge Acquisition and Applications (Cognitum 2015)*.

Lenhart K. Schubert and Chung Hee Hwang. 2000. *Episodic Logic Meets Little Red Riding Hood: A Comprehensive Natural Representation for Language Understanding*, page 111–174. MIT Press, Cambridge, MA, USA.

Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. 2007. Pellet: A practical owl-dl reasoner. *Journal of Web Semantics*, 5(2):51–53. Software Engineering and the Semantic Web.

Dmitry Tsarkov and Ian Horrocks. 2006. Fact++ description logic reasoner: System description. In *Automated Reasoning*, pages 292–297, Berlin, Heidelberg. Springer Berlin Heidelberg.

Frank Van Harmelen, Vladimir Lifschitz, and Bruce Porter. 2008. *Handbook of knowledge representation*. Elsevier.