Formalizing AMR Inference via Hybrid Logic Tableaux

CL Masters Thesis Defense

Eli Goldner

July 30, 2021

Introduction

- Semantic representation:
  - Capture meaning of natural language content.
  - Designed for manipulation via software.
- Abstract Meaning Representation (AMR):
  - Graph-based (DAG), nodes are *concepts*, edges are *relations*.
  - Built on predicative core of a sentence.
  - Ignores syntactic differences between equivalent sentences.
  - PropBank framesets are used for concepts (entities, events, properties, states).

Introduction

(Basic) AMR is reductionistic. This is awesome for:

- Annotation (esp. by non-experts).
- Semantic parsing (smaller target space).

This is not awesome for:

- Representing and recovering fine-grained meaning.
- Automating reasoning/inference.

Introduction

- The trade-off between ease of generation/use and rich expressivity/inferentiability is at least as old as computing.
- AMR has made a choice that works well in data-driven NLP.
- However AMR can bridge this gap:
- AMR already does this in a modular way with extensions.
- Some of these extensions give afford interpretation in first-order logic:
    - Automated inference for logics is a rich area with lots of tools.
    - This is where we come in.

Motivation

"Why do we need formal methods? Can't state-of-the-art language models do this already?"

Short answer: Not really, and even if they could:

- Statistically driven techniques are unecessarily expensive for formal inference.
- Increasing need for ability to guarantee/verify properties of software:
  - Does the software give us the right *type* of result for an input?
  - Bias in NLP.
- Machine learning (by itself) does not lend itself well to this.

Approach

- Combine two AMR extensions for richer interpretation:
  - Scope and quantification (Pustejovsky et al., 2019)
  - Tense and aspect (Donatelli et al., 2018)
- Interpret these extended AMR into a logic that handles quantification and tense.
- Develop tableau methods for this logic:
  - General method for proving/disproving sentences in the logic.
  - Restricted method for checking if sentence holds in some model.

AMR with Scope and Quantification

- Disambiguates scope.
- Annotates central predicate and its arguments.
- Clearest path for AMR $\rightarrow$ standard first-order predicate logic.

AMR with Tense and Aspect

- Standard AMR structure.
- Central predicate annotated for:
  - Aspect.
  - Event time.
  - Reference time.

Combined Extensions

- Assume each AMR has information from both extensions.
- Attach tense and aspect information to central predicate node.
- Extract a tense-sensitive FOPL representation (details later).

Modal Logic

- Propositional logic lets us form statements like $p \land (q \lor \neg r)$.
- Modal propositional logic extends propositional logic with an operator $\Diamond$, read as "possible". i.e. it is not possible that p and $\neg p$ are the case would be:

$$\neg\Diamond(p \land \neg p)$$

- (More) formal meaning of $\Diamond$: There is a *possible world* where p is true, and this possible world is *accesible* from the current one.
- The problem: the "current world" is an implicit notion dependent on context. Is there something more expressive?

Hybrid Logic

- Idea: take propositional modal logic, and add an operator @, that lets us know which world we're referring to.
- p or r is possible at world i: $@_i \Diamond (p \vee r)$
- In the above proposition i is called a nominal since it *names* some/is true at exactly one world.
- Everything true at the nominal j is true at the nominal i (they name the same world): $@_i j$

Hybrid Logic Variants

- Hybrid tense logic:
    - Two tense modalities: $\langle F \rangle$ and $\langle P \rangle$
    - World j is in the past of world i: $@_i Pj$
- Quantified hybrid logic:
    - Hybrid logic with first-order quantifiers, relation, and function symbols.
    - At n (now) there is a person for whom it is possible to own a car:

$$@_n(\exists x)(Person(x) \wedge (\exists y)(Car(y) \wedge \Diamond Afford(x, y)))$$

Hybrid Logic Variants

A problem

$$@_n(\exists x)(Person(x) \wedge (\exists y)(Car(y) \wedge \Diamond Afford(x, y)))$$

What's the domain of quantification?

- Option 1: The domain is the same at every world: *There is someone (out of all people all people at all worlds) who can afford some car (out of all cars at all worlds).*
- Option 2: Each world has a different domain: *There is someone (out of everyone in this world) who can afford some car (out of all cars in this world).*

The first option in quantified hybrid (or modal) logic is called possibilist quantification, the second is called actualist. When worlds are interpreted as times, the former is called eternalist quantification, and the latter is callled presentist. We will use presentist quantification.

First-Order Hybrid Tense Logic

First-order Hybrid Tense Logic (*FHTL*) is:

- Quantified hybrid logic, with tense modalities F and P instead of $\Diamond$.
- Presentist quantification:
  - Each world has its own domain.
  - Quantification is domain sensitive.

First-Order Hybrid Tense Logic

In logic models are interpretations of symbols and constants in the language. An *FHTL* model $\mathfrak{M}$ is a tuple

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

Where:

- T is a set of times/worlds.
- $\mathcal{R}$ is the binary accesibility relation over times.
- D_t is the domain of a time t
- I_{nom} assigns nominals to worlds.
- I_t interprets the value of terms at a time t.

The satisfiability of a formula with free variables depends on a variable assignment function. A formula that does not depend on variable assignment (every variable is bound by a quantifier) is called a *sentence*.

First-Order Hybrid Tense Logic

- Two main tasks for a sentence s:
    - Theorem proving – is s true regardless of the model?
    - Model checking – is s true in some model $\mathfrak{M}$?
- For both we use versions of the tableau method.

Tableau Method

- Tableau for a formula is a tree structure.
- A tableau caclulus for a logic breaks down and transforms formulae
- Rules of the calculus use semantics of connectives/quantifiers/modifiers.
- *FHTL* tableau based on *QHL* tableau modified for tense rules.

Tableau Method

- A tableau branch is a subtree of the main tableau tree.
- A tableau *branches* for rules involving disjunctions.
- A branch is *closed* if it contains both a formula $@_s\phi$ and its negation $@_s\neg\phi$. Otherwise it is open.
- A branch is *saturated* if no rules of the calculus can be applied without adding a redundant formula on the branch.

Tableau Example Rules

$$\frac{@_s F\phi}{@_s Fa} \; \mathbf{(}F\mathbf{)}$$ [1][2]

$$@_a\phi$$

$$\frac{@_s(\exists x)\phi}{@_s\phi[s{:}p/x]} \; \mathbf{(\exists)}$$ [3]

$$\frac{@_s @_t \phi}{@_t\phi} \; \mathbf{(@)}$$

$$\frac{@_i j{:}t = k{:}s \qquad @_i\phi}{@_i\phi[j{:}t//k{:}s]} \; \mathbf{(sub)}$$ [4]

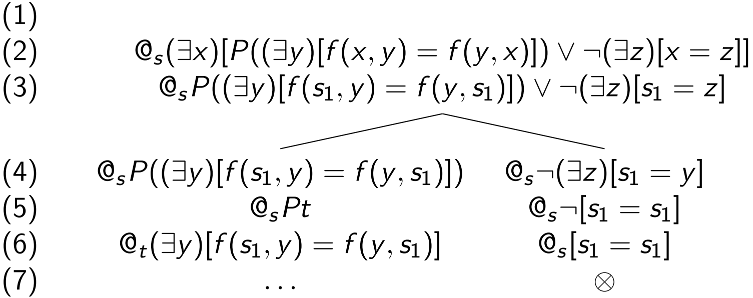
[1] The nominal a is new to the branch.

[2] The formula ϕ is not a nominal.

[3] $s{:}p$ is new to the branch.

[4] $\phi[j{:}t//k{:}s]$ is ϕ where some occurences of $j{:}t$ have been replaced by $k{:}s$.

FHTL Tableau Example



(1)

(2) $@_s(\exists x)[P((\exists y)[f(x,y) = f(y,x)]) \vee \neg(\exists z)[x = z]]$

(3) $@_s P((\exists y)[f(s_1,y) = f(y,s_1)]) \vee \neg(\exists z)[s_1 = z]$

(4)	$@_s P((\exists y)[f(s_1,y) = f(y,s_1)])$	$@_s \neg(\exists z)[s_1 = y]$
(5)	$@_s Pt$	$@_s \neg[s_1 = s_1]$
(6)	$@_t(\exists y)[f(s_1,y) = f(y,s_1)]$	$@_s[s_1 = s_1]$
(7)	$\ldots$	$\otimes$

Model Checking Example

- *Every computer will be located at a desk.*
- AMR with quantification and tense:

```
(s / scope
      :pred (b / be-located-at-91 :ongoing -
                                  :complete +
                                  :time (a / after
                                           :op1 (n / now))
         :ARG0 (c / computer)
         :ARG1 (d / desk
            :quant (e / every)))
      :ARG0 d
      :ARG1 c)
```

- *FHTL* translation:
  $@_{now}(\forall y)[\text{desk}(y) \rightarrow (\exists x)[\text{computer}(x) \wedge F(\text{be-located-at-91}(x, y))]]$

Model Checking Example

Define a small *FHTL* model $\mathfrak{M} = (T, \mathcal{R}, (D_t)_{t \in T}, I_{nom}, (I_t)_{t \in T})$ where:

$$T = \{yesterday, now, tomorrow\}$$

$$\mathcal{R} = \{(yesterday, now), (now, tomorrow), (yesterday, tomorrow)\}$$

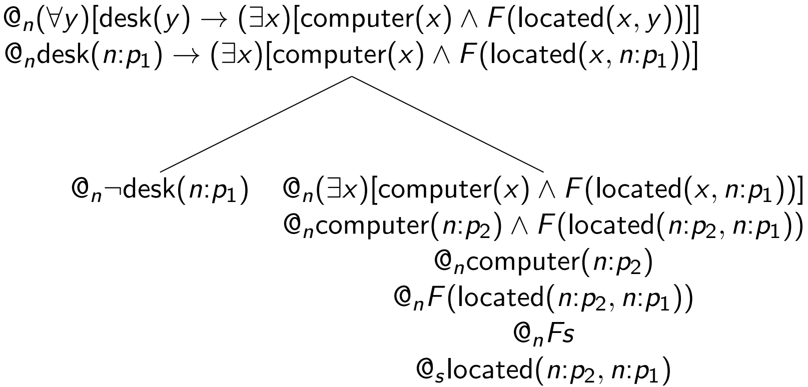
$$I_{nom} = \{(y, yesterday), (n, now), (t, tomorrow)\}$$

$$D_{yesterday} = \{computer_1, desk_1\}$$

$$D_{now} = \{computer_1, computer_2, desk_1, desk_2, desk_3\}$$

$$D_{tomorrow} = \{computer_1, computer_2, desk_1, desk_2\}$$

Model Checking Example



$@_n(\forall y)[\mathrm{desk}(y) \rightarrow (\exists x)[\mathrm{computer}(x) \wedge F(\mathrm{located}(x, y))]]$

$@_n\mathrm{desk}(n{:}p_1) \rightarrow (\exists x)[\mathrm{computer}(x) \wedge F(\mathrm{located}(x, n{:}p_1))]$

$@_n\neg\mathrm{desk}(n{:}p_1)$

$@_n(\exists x)[\mathrm{computer}(x) \wedge F(\mathrm{located}(x, n{:}p_1))]$

$@_n\mathrm{computer}(n{:}p_2) \wedge F(\mathrm{located}(n{:}p_2, n{:}p_1))$

$@_n\mathrm{computer}(n{:}p_2)$

$@_nF(\mathrm{located}(n{:}p_2, n{:}p_1))$

$@_nFs$

$@_s\mathrm{located}(n{:}p_2, n{:}p_1)$

Where we assign $s = t = \textit{tomorrow}$, $n{:}p_2 = \textit{computer}_1$ and $n{:}p_1 = \textit{desk}_2$, or $n{:}p_2 = \textit{computer}_2$ and $n{:}p_1 = \textit{desk}_1$ we see that $\mathfrak{M}$ satisfies the *FHTL* sentence.

Extraction

Interpretation

References I

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.