Formalizing AMR Inference via Hybrid Logic Tableaux
CL Masters Thesis Defense

Eli Goldner

July 28, 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.

▶ Hybrid systems and explainable AI.

Approach

AMR with Tense and Aspect

AMR with Scope

Modal Logic

▶ Propositional logic lets us form statements like $p \wedge (q \vee \neg r)$.

▶ 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:

$$\neg \Diamond (p \wedge \neg p)$$

▶ (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: $@_i \Diamond (p \lor r)$
- ▶ In the above proposition i is called a nominal since it *names*/uniquely picks out some world.
- ▶ Everything true at the world j is true at the world i (they name the same world): $@_i j$

Hybrid Logic Variants

First-Order Hybrid Tense Logic

FHTL Tableau Example

(1)
(2) $\qquad @_s(\exists x)[P((\exists y)[f(x,y) = f(y,x)]) \lor \neg(\exists z)[x = z]]$
(3) $\qquad @_s P((\exists y)[f(s_1,y) = f(y,s_1)]) \lor \neg(\exists z)[s_1 = z]$

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

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:
$@_{now}(\forall y)[\text{desk}(y) \to (\exists x)[\text{computer}(x) \land F(\text{be-located-at-91}(x, y))]]$

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:

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

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

$$I_{nom} = \{(y, yesterday), (n, now), (t, tomorrow)\}$$
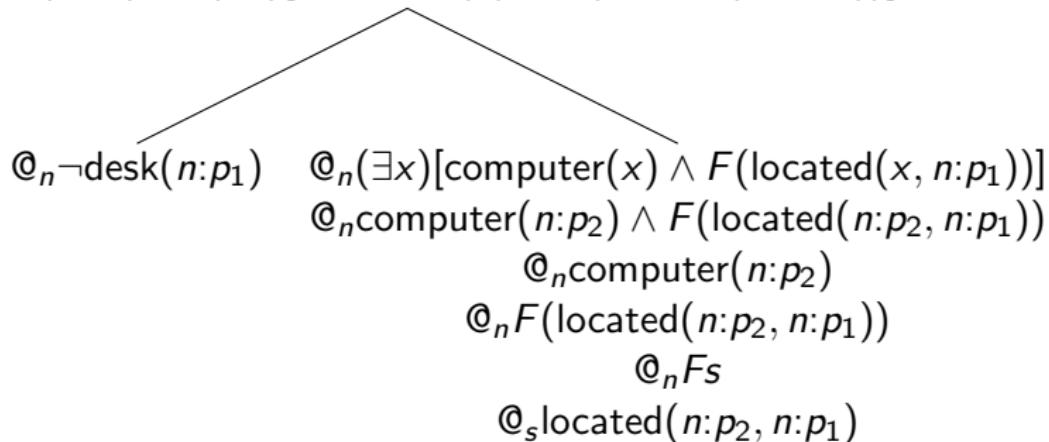
$$D_{yesterday} = \{computer_1, desk_1\}$$

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

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

Model Checking Example

$$@_n(\forall y)[\text{desk}(y) \to (\exists x)[\text{computer}(x) \land F(\text{located}(x, y))]]$$
$$@_n\text{desk}(n{:}p_1) \to (\exists x)[\text{computer}(x) \land F(\text{located}(x, n{:}p_1))]$$

$$@_n\neg\text{desk}(n{:}p_1) \qquad @_n(\exists x)[\text{computer}(x) \land F(\text{located}(x, n{:}p_1))]$$
$$@_n\text{computer}(n{:}p_2) \land F(\text{located}(n{:}p_2, n{:}p_1))$$
$$@_n\text{computer}(n{:}p_2)$$
$$@_nF(\text{located}(n{:}p_2, n{:}p_1))$$
$$@_nFs$$
$$@_s\text{located}(n{:}p_2, n{:}p_1)$$

Where we assign $s = t = tomorrow$, $n{:}p_2 = computer_1$ and $n{:}p_1 = desk_2$, or $n{:}p_2 = computer_2$ and $n{:}p_1 = desk_1$ we see that $\mathfrak{M}$ satisfies the *FHTL* sentence.

Extraction

Interpretation

References I