Intensionalizing Abstract Meaning Representations: Non-Veridicality and Scope

Gregor Williamson
Department of Computer Science
Emory University
Atlanta, GA 30322, USA
gregor.jude.williamson@emory.edu

Patrick Elliott
Department of Linguistics and Philosophy
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
pdell@mit.edu

Yuxin Ji
Department of Quantitative Theory and Methods
Emory University
Atlanta, GA 30322, USA
jessica.ji@emory.edu

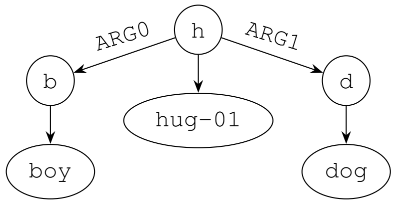
Abstract

Abstract Meaning Representation (AMR) is a graphical meaning representation language designed to represent propositional information about argument structure. However, at present it is unable to satisfyingly represent non-veridical intensional contexts, often licensing inappropriate inferences. In this paper, we show how to resolve the problem of non-veridicality without appealing to layered graphs through a mapping from AMRs into Simply-Typed Lambda Calculus (STLC). At least for some cases, this requires the introduction of a new role `:content` which functions as an intensional operator. The translation proposed is inspired by the formal linguistics literature on the event semantics of attitude reports. Next, we address the interaction of quantifier scope and intensional operators in so-called de re/de dicto ambiguities. We adopt a scope node from the literature and provide an explicit multidimensional semantics utilizing Cooper storage which allows us to derive the de re and de dicto scope readings as well as intermediate scope readings which prove difficult for accounts without a scope node.

1 Introduction

Abstract Meaning Representation (AMR) is a graphical meaning representation in which graphs are rooted, directed, and acyclic (Banarescu et al., 2013). Non-terminal nodes are assigned variable IDs, terminal nodes are sense concepts (e.g., `believe-01`, `boy`, etc.) or constants (e.g., the polarity attribute −, cardinals, names, etc.), and labelled edges represent semantic relations between nodes. The inventory of AMR disambiguated predicate senses (e.g., `believe-01`) are based on PropBank argument structure frames (Bonial et al., 2015; Palmer et al., 2005; Bonial et al., 2014). The graph in (1-b) is an AMR for the sentence in (1-a). More commonly, however, AMRs are represented in Penman notation (Matthiessen and Bateman, 1991) as in (1-c) or occasionally as a conjunction of logical triples (1-d).

(1) a. *The boy hugged the dog.*

b. 

c.
```
(h / hug-01
   :ARG0 (b / boy)
   :ARG1 (d / dog))
```

d. INSTANCE(h, hug-01)
∧ INSTANCE(b, boy) ∧ ARG0(h, b)
∧ INSTANCE(d, dog) ∧ ARG1(h, d)

The main strength of AMR is its ability to represent argument structure, as these logical triples translate naturally into a rudimentary neo-Davidsonian event semantics (Davidson, 1967; Parsons, 1990), with every INSTANCE relation split into an existential quantifier and a one-place predicate.

(2) $\exists x(\textbf{hug-01}(x) \wedge \exists y(\textbf{boy}(y) \wedge ARG0(x)(y) \wedge \exists z(\textbf{dog}(z) \wedge ARG1(x)(z))))$

Recent developments have seen the expressive power of AMR improved both in terms of its

Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop, pages 160–169
November, 2021.

graphic representation as well as its translation into logical forms. For instance, AMR graphical representations have been enriched to represent Tense and Aspect (Donatelli et al., 2018, 2019; Van Gysel et al., 2021), quantifier scope (Pustejovsky et al., 2019; Van Gysel et al., 2021), semantic number (Stabler, 2017), and speech acts (Bonial et al., 2020), while translations into first and higher-order logics have been proposed as a means of capturing coreference (Artzi et al., 2015) and quantifier scope (Bos, 2016, 2020; Stabler, 2017; Lai et al., 2020).

Despite these advances in theoretical work, AMR encounters issues when it comes to the semantics of intensional contexts. For instance, Crouch and Kalouli (2018) note that an AMR like (3-b), represented as a conjunction of logical triples, would permit an inference to *'The girl is sick'* by conjunction elimination.

(3) a. *The boy believes that the girl is sick.*

b.
```
(b / believe-01
   :ARG0 (b2 / boy)
   :ARG1 (s / sick-05
      :ARG1 (g / girl)))
```

An even more striking consequence of this non-veridical problem is demonstrated by the following examples.[1]

(4) a. *The boy believes he is sick.*

b.
```
(b / believe-01
   :ARG0 (b2 / boy)
   :ARG1 (s / sick-05
      :ARG1 b2))
```

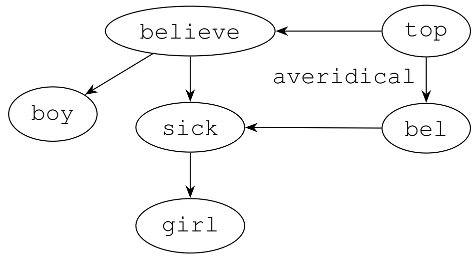
c.
```
(b / believe-01
   :ARG0 (b2 / boy
      :ARG1-of (s / sick-05))
   :ARG1 s))
```

At first glance, it might appear that the predicate `sick-05` is in a non-veridical context in (4-b) and a veridical context in (4-c). This is because the grammatical subject position of the verb *believe* is a veridical environment, while its sentential complement is not. However, these graphs are in fact logically equivalent because for any AMR relation R, $\mathtt{R}(x)(y) \Leftrightarrow \mathtt{R\text{-}of}(y)(x)$. Consequently, (4-b) and (4-c) depict the same conjunction of logical triples once inverse relations are normalized.

[1]These examples are based on examples in the *freakshow* section of the AMR guidelines, available at: `https://github.com/amrisi/amr-guidelines/blob/master/amr.md#amr-freak-show`

Crouch and Kalouli (2018) provide a solution to the problem of non-veridicality in a graph-based representation by making use of a Graphic Knowledge Representation (GKR) (Kalouli and Crouch, 2018), a layered graph in which identifiers, or names, are assigned to sub-graphs (Carroll et al., 2005). A GKR separates conceptual structure (i.e., predicate-argument structure) and contextual structure into two sub-graphs. The graph in (5) is a simplified GKR of (3-a). The lower context `bel` represents the intensional context of the boy's belief and is non-veridical (or averidical) with respect to the upper context `top`. Consequently, the GKR does not permit the inference from *'The boy believes the girl is sick'* to *'The girl is sick.'*

(5)



Another problem for AMR's representation of attitude reports such as those containing the verb *'believe'* is that PropBank argument structures often do not distinguish between propositional and non-propositional arguments. For instance, the PropBank argument structure for `believe-01` assigns both nominal and clausal arguments the same argument role, `:ARG1`.[2] Despite this, the PropBank frame includes a note that *'believe'* can have both a theme and a propositional argument simultaneously (e.g., *'Mary believed John that he didn't eat the last piece of pie'*).[3]

Given this lack of distinction between different types of arguments, it is not sufficient to lexically specify that the `:ARG1` of `believe-01` is always a non-veridical environment. While this might work for (3-b), it would also mean that we could not infer the existence of a girl from (6).

(6) a. *The boy believed the girl.*

b.
```
(b / believe-01
   :ARG0 (b2 / boy)
   :ARG1 (g / girl))
```

[2]PropBank frames are available at: `https://github.com/propbank/propbank-frames/`

[3]The annotator concludes "we could add an `:ARG2` [...] and use it only when `:ARG1` is already present, but that makes me sad, so let's just not mess with it until we actually see such an instance."

In what follows, we propose the introduction of an intensional relation `:content` responsible for introducing propositional arguments. Replacing the `:ARG1` of `believe-01` in (4) with the new `:content` role ensures firstly that (4-b) contains a non-veridical environment and secondly that (4-c) is not a representation for a coherent natural language sentence. Crucially, the addition of `:content` and the translation function proposed for AMRs into logical forms offers a satisfying representation for intensional contexts without the need for additional graph structure. Finally, we show how our logical forms interact with scope taking elements to derive attested interpretations of attitude reports with quantifier phrases (QPs).

2 Extensional Semantics for AMRs

We start by defining a simple translation for basic AMRs (without intensional operators or quantifiers) into the Simply-Typed Lambda Calculus (STLC). Following Bos (2016) we define the syntax of AMRs recursively. A simplex AMR is a constant `c`, variable `x`, or an instance assignment `(x / P)`. Complex graphs are defined recursively as one or more subgraphs $\{\mathtt{A}_1, \ldots, \mathtt{A}_n\}$ connected to an instance assignment by n relations.

(7) $\mathtt{A} :=$

a. $\mathtt{c}$

b. $\mathtt{x}$

c. $\mathtt{(x\,/\,P)}$

d. $\mathtt{(x\,/\,P\ :R_1\,A_1 \ldots :R_n\,A_n)}$

Our semantics for AMRs is a departure from that of Bos (2016) and Lai et al. (2020). For now, we assume that the interpretation function $[\![.]\!]$ compositionally maps AMRs to a simple first-order calculus embedded in the STLC.[4] AMR constants and variables are mapped to STLC constants of type e, AMR predicates are mapped to STLC constants of type $e \rightarrow t$, and AMR roles are mapped to STLC constants of type $e \rightarrow e \rightarrow t$.[5]

(8) a. $[\![\mathtt{c}]\!] = \mathbf{c}$ $\qquad e$

b. $[\![\mathtt{x}]\!] = x$ $\qquad e$

Simple instance assignments are mapped to functional applications:

(9) **Instance assignment**

$[\![\mathtt{(x\,/\,P)}]\!] = P(x)$ $\qquad t$

To state a semantics for complex AMRs, we first state a semantics for a *role assignment* in (10), consisting of a role and an embedded AMR, via pattern matching on the embedded AMR. A sequence of role assignments $\rho_1 \ldots \rho_n$ is interpreted via iterated conjunction (11), and finally a complex AMR is interpreted by saturating a role sequence with the main variable of the AMR (12).

(10) **Role assignment**

a. $[\![\mathtt{:R\,y}]\!] = \lambda x \,.\, R(x)(y)$

b. $[\![\mathtt{:R\ (y\ /\ \ldots\ )}]\!]$
$= \lambda x \,.\, R(x)(y) \wedge [\![\mathtt{(y\ /\ \ldots\ )}]\!]$

(11) **Role sequence**

$[\![\rho_1 \ldots \rho_n]\!] = \lambda x \,.\, [\![\rho_1]\!]\,(x) \wedge \ldots \wedge [\![\rho_n]\!]\,(x)$

(12) **Complex AMR**

$[\![\mathtt{(x\,/\,P}\ \rho_1 \ldots \rho_n\mathtt{)}]\!]$
$= P(x) \wedge [\![\rho_1 \ldots \rho_n]\!]\,(x)$

Basic AMRs are thereby translated into simple conjunctive first-order formulae.

(13) *The boy admires himself.*

a.
```
(a / admire-01
   :ARG0 (b / boy)
   :ARG1 b))
```

b. $\textbf{admire-01}(a) \wedge ARG0(a)(b)$
$\wedge\, \textbf{boy}(b) \wedge ARG1(a)(b)$

Finally, we declare an operation `close` (version 1) which, applied to a STLC expression φ, introduces an existential quantifier which unselectively binds variables in the set $\textbf{Free}(\varphi)$.

(14) **Close (version 1)**

$\mathtt{close}(\varphi) = \exists x_1 \ldots x_n(\varphi)$
where $\{x_1, \ldots, x_n\} = \textbf{Free}(\varphi)$

Crucially for what follows, $\textbf{Free}(\varphi)$ does not correspond to the classical notion of free variables in φ, but rather should be defined to ensure that recurrent variables are not bound by `close` prior to instance assignment. We define **Free** as follows:[6]

(15) a. $\textbf{Free}\,(\mathtt{c}) = \emptyset$

b. $\textbf{Free}\,(\mathtt{x}) = \emptyset$

c. $\textbf{Free}\,(\mathtt{(x\,/\,P)}) = \{x\}$

d. $\textbf{Free}\,(\mathtt{(x\,/\,P\ :R_1A_1 \ldots R_nA_n)})$
$= \{x\} \cup \textbf{Free}\,(\mathtt{A}_1) \cup \ldots \cup \textbf{Free}\,(\mathtt{A}_n)$

[4]The STLC is widely used in analytical work on natural language semantics, and has a well-understood proof-theory and model-theory (Carpenter, 1998).

[5]e is the basic type of *individuals*, and t is the basic type of *truth-values*. The constructor for functional types $a \rightarrow b$ is right-associative.

[6]We are grateful to an anonymous reviewer for pressing us on this point.

Applying `close` to the example above returns an existential statement.

(16) $\exists a, b.\textbf{admire-01}(a) \wedge ARG0(a)(b) \wedge \textbf{boy}(b) \wedge ARG1(a)(b)$

Putting re-entrant nodes to one side, it is harmless to defer existential binding of free variables, since $\psi \wedge \exists x(\varphi) \Leftrightarrow \exists x(\psi \wedge \varphi)$ (if $x \notin \textbf{Free}(\psi)$). This does away with the need to use continuation-passing style (c.f., Bos 2016; Lai et al. 2020). Cases involving re-entrant nodes such as *the dog scratched itself* are handled straightforwardly via matching variables.

3 An Intensional Semantics

Next, we systematically intensionalize the interpretation in a standard way by replacing the propositional type t with $s \to t$ (the type of a function from *worlds* to truth values; Gallin 1975). Our existing interpretation procedure for basic AMRs remains largely intact, although we tweak the definitions of conjunction and existential quantification in order to accommodate the presence of additional world arguments.

(17) a. $\varphi \wedge_w \psi := \lambda w. \varphi(w) \wedge \psi(w)$
b. $\exists_w x(\varphi) := \lambda w. \exists x(\varphi(w))$

To provide a semantics for intensional operators such as attitude predicates in AMR, we adopt a variant of Kratzer's (2006) Davidsonian event semantics for attitude verbs which has undergone a number of refinements (e.g., Moulton, 2009; Elliott, 2016, 2020). More specifically, we propose a translation of AMRs of attitude reports into logical forms in which attitude events are associated with propositional content via a dedicated modal operator **cont**. In order to achieve this, we increase the AMR inventory of semantic roles with a `:content` role which is interpreted as a relation of type $e \to (s \to t) \to (s \to t)$.

(18) $[\![$`:content A`$]\!] = \lambda x\,.\,\textbf{cont}(x)(\texttt{close}([\![\texttt{A}]\!]))$

To see how this resolves the problem of non-veridicality consider again the two AMRs in (4), the first of which is repeated in (19-a) with the `:ARG1` role changed to `:content`. The underlined argument shows that the world of evaluation for the translation of the `:content` argument is shifted to w_2, and thus we cannot infer that the boy is sick in w as desired.

(19) a.
```
(b / believe-01
   :ARG0 (b2 / boy)
   :content (s / sick-05
      :ARG1 b2))
```
b. $\texttt{close}([\![\,(19\text{-a})\,]\!])$
$= \lambda w\,.\,\exists b, b_2(\textbf{believe-01}(b)(w) \wedge \textbf{boy}(b_2)(w) \wedge ARG0(b)(b_2)(w) \wedge \textbf{cont}(b)(\underline{\lambda w_2\,.\,\exists s(\textbf{sick-01}(s)(w_2) \wedge ARG1(s)(b_2)(w_2))})(w))$

Next, consider the same modification for (4-c) repeated in (20), modulo `:ARG1`$\Rightarrow$`:content`.

(20)
```
(b / believe-01
   :ARG0 (b2 / boy
      :ARG1-of (s / sick-05))
   :content s))
```

The semantic rule for `:content` is only well-defined if the embedded AMR is an *instance assignment*, due to the type of **cont**. It follows that the interpretation of (20), in which `:content` embeds a recurrent variable, is undefined. For this reason, AMRs in which `:content` embeds a constant or variable should be avoided by annotators. The issue here is that reentrant nodes are used to model a diverse range of linguistic phenomena (Szubert et al., 2020), but are inappropriate for modelling anaphora to a proposition, such as *'The boy who is sick believes it'*.[7]

4 Content and Quantifier Scope: de re and de dicto Readings

Another property of intensional operators is their ability to interact scopally with other operators such as quantifier phrases (QPs) (see Keshet and Schwarz 2019 for a recent overview). For example, the sentence in (21) could be paraphrased as in (21-a) which can be analyzed as an existential QP *'a violin'* taking scope over the attitude predicate *'hope'*, in which case the restrictor argument of the QP is evaluated in the actual world (i.e., there is an actual violin that the boy wants). In contrast, the reading in (21-b) can be analyzed as the existential being within the scope of the attitude (i.e., the boy hopes to be a violin-owner, but is not necessarily concerned about owning any particular violin).

[7]It should be noted that the AMR in (4-c) in which `:ARG1` is not replaced with `:content` does have an interpretation which can be paraphrased as: *'the boy who is in a state s of being sick, believes the state s'*. However, it is not clear that this corresponds to any coherent natural language sentence.

(21) *The boy hopes to buy a violin.*

a. There is a violin the boy hopes to buy. *De re*

b. The boy hopes to be a violin-owner. *De dicto*

To capture these readings, we develop a semantics for AMRs enriched with additional graph structure for modelling scope (Pustejovsky et al., 2019) based on the mechanism of Cooper storage (Cooper, 1983; Kobele, 2018).

4.1 Scope Semantics

Before discussing the scope interaction of quantifiers and **cont**, let us first develop a semantics for scope in non-intensional contexts. Following Pustejovsky et al. (2019) and Van Gysel et al. (2021) we make use of a `scope` node, which has a predicative argument representing the core argument structure and reentrant variables to represent the order of quantifier scope. For example, in (22-b), the QP of `:ARG0` scopes over that of `:ARG1`.

(22) a. A computer is on every desk.

b.
```
(s / scope
   :pred (b / be-located-at-91
      :ARG0 (c / computer)
      :ARG1 (d / desk
         :quant every))
   :ARG0 d
   :ARG1 c)
```

Next, we define an explicit interpretation of scope nodes using Cooper storage (Cooper, 1983; Kobele, 2018). A *store* is an *assignment* of variables to STLC expressions of type $(e \to t) \to t$.[8]

(23) $\{(x, \mathbf{every}(\mathbf{boy})), (y, \mathbf{some}(\mathbf{girl})\}$

Instead of simply mapping AMRs to expressions of STLC, we map them to a pair consisting of a store s and an ordinary semantic value (i.e., an STLC expression). We assume the following notational conventions.

(24) a. $[\![\mathtt{A}]\!] = s \cdot \varphi$

b. $[\![\mathtt{A}]\!]_s = s$

c. $[\![\mathtt{A}]\!]_o = \varphi$

In this system, AMRs like those we have considered so far update the store vacuously (25-a) while retaining their ordinary semantic value (25-b).

(25) **Instance assignment (revised)**

a. $[\![\mathtt{(d\,/\,dance\text{-}01)}]\!]_s = \emptyset$

b. $[\![\mathtt{(d\,/\,dance\text{-}01)}]\!]_o = \mathbf{dance\text{-}01}(d)$

In order to illustrate how `:quant` is interpreted, we begin by stating a semantics for an instance assignment decorated with a single `:quant` role.[9] We assume that AMR determiner constants such as `every` are mapped to STLC constants of type $(e \to t) \to (e \to t) \to t$, such as the quantificational determiner **every** (e.g., Barwise and Cooper, 1981). The store is updated with a generalized quantifier constructed from the determiner and the property in the instance assignment (26-a). The ordinary semantic value, on the other hand, is the truth constant $\top$, the addition of which is redundant in a string of conjunctions (26-b).

(26) **Quantifier storage**

a. $[\![\mathtt{(x\,/\,P\;:quant\;D)}]\!]_s$
$= \{(x, \mathbf{D}(P))\}$

b. $[\![\mathtt{(x\,/\,P\;:quant\;D)}]\!]_o = \top$

On this semantics, basic AMRs have a non-trivial ordinary value but perform a vacuous store update. Conversely, QPs perform a non-trivial store update but have a redundant ordinary semantic value. Crucially, however, QPs still contribute one or more variables to the ordinary value of a role assignment via pattern matching, as in (27) below.

We revise our translation function to ensure that the store gets passed up during the derivation.

(27) **Role assignment (revised)**

$[\![\mathtt{:R\;(y\,/\,\ldots)}]\!]$
$= [\![\mathtt{(y\,/\,\ldots)}]\!]_s$
$\cdot\, \lambda x\,.\, R(x)(y) \wedge [\![\mathtt{(y\,/\,\ldots)}]\!]_o$

(28) **Role sequences (revised)**

$[\![\rho_1 \ldots \rho_n]\!]$
$= [\![\rho_1]\!]_s \cup \ldots \cup [\![\rho_n]\!]_s$
$\cdot\, \lambda x\,.\, [\![\rho_1]\!]_o\,(x) \wedge \ldots \wedge [\![\rho_n]\!]_o\,(x)$

(29) **Complex AMR (revised)**

$[\![\mathtt{(x\,/\,P}\;\rho_1 \ldots \rho_n\mathtt{)}]\!]$
$= [\![\rho_1 \ldots \rho_n]\!]_s$
$\cdot\, P(x) \wedge [\![\rho_1 \ldots \rho_n]\!]_o\,(x)$

In combination with (26-a), this ensures that the store of a complex AMR will contain the index-quantifier pairs added to the store by its subgraphs, as shown in the following example.

[8]We leave world variables implicit in section 4.1 for the sake of readability.

[9]We generalize this rule to the more complex case involving `:quant` and a sequence of role assignments, as well as cases of nested quantification in appendix A.

(30) *Every boy danced.*

a.
```
(d / dance
    :ARG0 (b / boy
        :quant every))
```

b. $[\![$ (30-a) $]\!] =$
$\{(b, \mathbf{every}(\mathbf{boy}))\}$
$\cdot\, \mathbf{dance}(d) \wedge \mathit{ARG0}(d)(b)$

In order to retrieve the quantifier from the store, we declare an operation $\texttt{pop}_x$ which, given a variable x, store s, and logical form φ retrieves the expression in s paired with x, and applies it to $\lambda x\,.\,\varphi$. We write s_x for the expression in s paired with x.

(31) **Pop**
$\texttt{pop}_x(s, \varphi) = s - \{(x, s_x)\} \cdot s_x(\lambda x\,.\,\varphi)$

We also restate our `close` operation which now existentially binds any free variables which are not associated with an index in the store.

(32) **Close (version 2)**
$\texttt{close}(s, \varphi) = s \cdot \exists x_1 \ldots x_n(\varphi)$
$\{x_1, \ldots, x_n\}$
$= \mathbf{Free}(\varphi) - \{v \mid (v, *) \in s\}$

As mentioned above, scope nodes are decorated with roles embedding reentrant nodes indicating scope-takers (`:ARG`n), and a role to indicate the scope site (`:pred`). We state the semantics for a complex AMR headed by a scope node syncategorematically: a scope node with arguments $x_1,\ldots,x_n$ induces evaluation of the quantifiers stored at $x_1,\ldots,x_n$.[10,11]

(33) **Interpreting scope nodes**

$$\left[\!\!\left[\begin{array}{l}\texttt{(s / scope}\\ \quad\texttt{:ARG0 x}_1 \ \ldots \ \texttt{:ARG}n_{-1}\ \texttt{x}_n\\ \quad\texttt{:pred A)}\end{array}\right]\!\!\right]$$
$= \texttt{pop}(x_1)$
$(\ldots(\texttt{pop}(x_n)$
$(\texttt{close}([\![\texttt{A}]\!])))\ldots)$

Consider again the example *'every boy danced'*, but now with a scope node.

(34) a.
```
(s / scope
    :ARG0 b
    :pred (d / dance-01
        :ARG0 (b / boy
            :quant every)))
```

b. $[\![$ (34-a) $]\!] =$
$\emptyset \cdot \mathbf{every}(\mathbf{boy})$
$(\lambda b\,.\,\exists d(\mathbf{dance\text{-}01}(d)$
$\wedge\, \mathit{ARG0}(d)(b)))$

4.2 Deriving the de re and de dicto readings

Now we reintroduce world variables to see how this interpretation function can translate both the de re and de dicto readings of (21) above, starting with the de re reading. In (35-a), the variable `v` is the `:ARG0` of the scope node and consequently the QP *a violin* takes scope over the attitude verb.

(35) a.
```
(s / scope
    :ARG0 v
    :pred (h / hope-01
        :ARG0 (b / boy)
        :content (b2 / buy-01
            :ARG0 b
            :ARG1 (v / violin
                :quant a)))))
```

b. $[\![$ (35-a) $]\!] = \emptyset\,\cdot$
$\lambda w\,.\,\exists v, b, h(\mathbf{violin}(v)(w)$
$\wedge\, \mathbf{hope\text{-}01}(h)(w)$
$\wedge\, \mathbf{boy}(b)(w)$
$\wedge\, \mathit{ARG0}(h)(b)(w)$
$\wedge\, \mathbf{cont}(h)$
$(\lambda w_2\,.\,\exists b_2(\mathbf{buy\text{-}01}(b_2)(w_2)$
$\wedge\, \mathit{ARG0}(b_2)(b)(w_2)$
$\wedge\, \mathit{ARG1}(b_2)(v)(w_2)))(w))$

Here, there is a specific violin in the world of evaluation w which the boy hopes to buy.

For the de dicto reading, we do not need to do anything special. However, to close off the interpretation we can either embed the entire AMR under a scope node or use the **close** operation to bind any free variables.

(36) a.
```
(h / hope-01
    :ARG0 (b / boy)
    :content (b2 / buy-01
        :ARG0 b
        :ARG1 (v / violin)))
```

b. $\mathbf{close}([\![$(36-a)$]\!]) = \emptyset\,\cdot$
$\lambda w\,.\,\exists h, b(\mathbf{hope\text{-}01}(h)(w)$
$\wedge\, \mathbf{boy}(b)(w)$
$\wedge\, \mathit{ARG0}(h)(b)(w)$
$\wedge\, \mathbf{cont}(h)$
$(\lambda w_2\,.\,\exists b_2, v(\mathbf{buy\text{-}01}(b_2)(w_2)$
$\wedge\, \mathbf{violin}(v)(w_2)$
$\wedge\, \mathit{ARG0}(b_2)(b)(w_2)$
$\wedge\, \mathit{ARG1}(b_2)(v)(w_2)))(w))$

[10] $\texttt{x}_n$ is associated with `:ARG`n_{-1} because the indexing of `:ARG`s starts at zero.

[11] Note that this property of scope nodes ensures that event quantification takes narrow scope with respect to other operators in the sentence (Champollion, 2015).

Here, the existential is within the scope of the lambda operator λw_2 and the restrictor argument $\mathbf{violin}$ is evaluated in the boy's hope worlds w_2.

4.3 Intermediate de dicto reading

Although Pustejovsky et al. (2019) frame their proposal as "embed[ding an AMR] under a scope graph", our implementation also permits the embedding of a scope node within an AMR. Doing so allows us to derive intermediate scope readings. Consider the following example.

(37) *The boy thinks the girl hopes to buy a violin.*
   a. The boy thinks there is a violin that the girl hopes to buy.

In this intermediate reading, the intensional object might not be a violin in the actual world, nor in the girl's desire worlds, rather it is a violin in the boys belief worlds. Stabler (2017) notes that accounts such as Bos (2016), and later Lai et al. (2020), cannot capture these sorts of intermediate scope readings since their projective semantics always derives widest scope of QPs. However, this reading can be captured on the present account straightforwardly from the following structure.[12]

(38)
```
(t / think-01
   :ARG0 (b / boy)
   :content (s / scope
      :ARG0 v
      :pred (h / hope-01
         :ARG0 (g / girl)
         :content (b / buy-01
            :ARG0 g
            :ARG1 (v / violin
               :quant a)))))
```

4.4 Problematic Scope

Besides the de re and de dicto readings, Fodor (1970) considers two further readings in which the restrictor argument of the QP is interpreted separately from the scopal position of the quantifier.

(39) *The boy hopes to buy a violin.*
   a. There are things which are violins and the boy hopes to buy one of them.
   Non-specific de re
   b. #There is a thing the boy hopes to buy which he believes is a violin.
   Specific de dicto

[12] Van Gysel et al. (2021) adopt the scope node approach in combination with a variant of Lai et al.'s semantics. Such an approach should also derive intermediate scope readings.

Although there is a general consensus in the literature that (39-b) is not a possible reading of (39), the reading in (39-a) is possible. At present, we cannot derive the reading in (39-a). While it is possible to enrich AMRs further to accommodate such interpretations, it remains to be seen whether such an effort is worthwhile. Although such interpretations are attested and theoretically significant, they are not common, and accounting for such interpretations would likely involve enriching the graphical representation of AMRs in ways which would make them far less tractable for annotators and parsers. We leave it to future research to determine whether we can accommodate this third reading without unintentionally complicating the AMRs in undesirable ways.

5 Discussion

We have proposed to extend the expressive power of AMR in two respects. Firstly, we have enriched the graphical form of AMRs by increasing the inventory of AMR roles with the role `:content`. We did so in order to distinguish between intensional and non-intensional arguments of modal operators such as attitude predicates. Secondly, we provided a translation of AMRs into logical forms which allowed us to solve the problem of non-veridicality as well as complex scope interactions between quantifiers and intenstional operators.

The addition of `:content` has ramifications for AMR annotation as well as backwards compatibility. We opted to avoid proposing the addition of new numbered argument roles for predicate like `believe-01` since this would involve a wide-scale revision of AMR's frameset as well as complicating any effort to convert between existing corpora and an enriched corpus, as this would involve a many-to-many mapping. Instead, we proposed a semantically motivated intensional relation `:content` which will reduce the complexity of any conversion effort, requiring only a many-to-one mapping.

When designing meaning representations, there is inevitably a trade-off between how adequate the representation is (e.g., how much semantic information is present) and how tractable the representations are for large scale annotation projects. Although we believe that `:content` should not be any more difficult to annotate than non-core roles in AMR, it is likely that annotators may struggle to resolve and represent quantifier scope and de re/de dicto ambiguities. Thankfully, in addition to being representationally adequate, the scope node approach adopted from Pustejovsky et al. (2019) and Van Gysel et al. (2021) is both intuitive and transparent. Future research could aim to gauge the level inter-annotator agreement when representing such phenomena in a small scale annotation task.

Beyond AMR, the enriched graphical language Uniform Meaning Representation (UMR) (Van Gysel et al., 2021) also utilizes scope nodes to capture quantifier scope relations. The `:content` role and translation function proposed here can be adopted wholesale for UMR, and thus will prove useful in future annotation projects for this more expressive annotation scheme.

Finally, the `:content` role and its intensional translation may also facilitate downstream NLP tasks. Specifically, different attitude predicates trigger different lexical inferences regarding the truth of their complement depending on whether they are factive (e.g., `know-01`), counterfactive (e.g., `pretend-01`) or non-veridical (e.g., `believe-01`). A number of rich resources exist in this domain including the MegaVeridicality data sets (White and Rawlins, 2018; White et al., 2018) which contain factuality judgments for a comprehensive list of English verbs that embed finite clauses as well as a variety of predicates which embed non-finite clauses. Resources such as these may be deployed alongside AMR for NLI, since the logical forms which our interpretation function produces represent the scope of the attitude verb, unlike in a flat list of logical triples.

A python script for translating AMR into STLC is available at https://github.com/emorynlp/Intensionalizing-AMR, and we are currently working on developing a script to convert intensional uses of numbered arguments into the new `:content` role which we also plan to make publicly available.

6 Conclusion

Abstract Meaning Representations (AMRs) are unable to represent non-veridical environments in a semantically satisfying way. When an AMR is translated into a conjunctions of logical triples, it permits spurious inferences to the truth of any of its subgraphs via conjunction elimination. We proposed to rectify this through the introduction of a novel AMR role `:content`, before providing an intensional interpretation for AMRs which correctly invalidates such inferences. We then showed how such a semantics can be combined with a means of modelling scope to derive de re and de dicto readings of natural language sentences with intensional operators and quantifiers. We concluded that the inclusion of a scope node (Pustejovsky et al., 2019; Van Gysel et al., 2021) is necessary in order to capture intermediate scope readings, and we provided a translation function from AMRs into STLC permitting the derivation of complex interactions of natural language quantifiers with intensional operators. This work is part of the concerted effort to increase the expressive power of AMRs while also maintaining tractable representations ensuring that large scale annotation projects can be performed with minimal instruction.

References

Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. 2015. Broad-coverage CCG semantic parsing with AMR. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1699–1710, Lisbon, Portugal. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

John Barwise and Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4(2):159–219.

Claire Bonial, Julia Bonn, Kathryn Conger, Jena D. Hwang, and Martha Palmer. 2014. PropBank: Semantics of new predicate types. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3013–3019, Reykjavik, Iceland. European Language Resources Association (ELRA).

Claire Bonial, Julia Bonn, Kathryn Conger, Jena D. Hwang, Martha Palmer, and Nicholas Reese. 2015. English PropBank Annotation Guidelines.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-AMR: Abstract Meaning Representation for dialogue. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 684–695, Marseille, France. European Language Resources Association.

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.

Bob Carpenter. 1998. *Type-Logical Semantics*. Language, Speech, and Communication. MIT Press, Cambridge, Mass.

Jeremy J Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler. 2005. Named graphs, provenance and trust. In *Proceedings of the 14th international conference on World Wide Web*, pages 613–622.

Lucas Champollion. 2015. The interaction of compositional semantics and event semantics. *Linguistics and Philosophy*, 38(1):31–66.

Robin Cooper. 1983. *Quantification and Syntactic Theory*. Studies in Linguistics and Philosophy. Springer Netherlands.

Richard Crouch and Aikaterini-Lida Kalouli. 2018. Named graphs for semantic representation. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 113–118, New Orleans, Louisiana. Association for Computational Linguistics.

Donald Davidson. 1967. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*, pages 81–95. University of Pittsburgh Press.

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.

Lucia Donatelli, Nathan Schneider, William Croft, and Michael Regan. 2019. Tense and aspect semantics for sentential AMR. *Proceedings of the Society for Computation in Linguistics*, 2(1):346–348.

Patrick D Elliott. 2016. Explaining dps vs. cps without syntax. *Proceedings of CLS*, 52:171–185.

Patrick D. Elliott. 2020. A plea for equality: Remarks on Moltmann's semantics for clausal embedding. *Theoretical Linguistics*, 46(3-4):241–251.

Janet Fodor. 1970. *The linguistic description of opaque contexts*. MIT.

Daniel Gallin. 1975. *Intensional and Higher-Order Modal Logic*. Number 19 in North-Holland Mathematics Studies. North-Holland Publ. Co. [u.a.], Amsterdam.

Aikaterini-Lida Kalouli and Richard Crouch. 2018. Gkr: the graphical knowledge representation for semantic parsing. In *Workshop on Computational Semantics beyond Events and Roles (SemBEaR 2018)*, pages 27–37.

William R. Keller. 1988. Nested Cooper Storage: The Proper Treatment of Quantification in Ordinary Noun Phrases. In U. Reyle and C. Rohrer, editors, *Natural Language Parsing and Linguistic Theories*, Studies in Linguistics and Philosophy, pages 432–447. Springer Netherlands, Dordrecht.

Ezra Keshet and Florian Schwarz. 2019. *De Re/De Dicto*. In Jeanette Gundel and Barbara Abbott, editors, *The Oxford Handbook of Reference*. Oxford University Press.

Gregory M Kobele. 2018. The cooper storage idiom. *Journal of Logic, Language and Information*, 27(2):95–131.

Angelika Kratzer. 2006. Decomposing attitude verbs. *Talk given at the Hebrew University of Jerusalem*.

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.

Christian M. I. M. Matthiessen and John A Bateman. 1991. *Text generation and systemic-functional linguistics: experiences from English and Japanese*. Pinter.

Keir Moulton. 2009. *Natural selection and the syntax of clausal complementation*. University of Massachusetts Amherst.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Terence Parsons. 1990. *Events in the Semantics of English: A Study in Subatomic Semantics*. MIT Press.

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.

Edward Stabler. 2017. Reforming AMR. *International Conference on Formal Grammar*, pages 72–87.

Ida Szubert, Marco Damonte, Shay B. Cohen, and Mark Steedman. 2020. The Role of Reentrancies in Abstract Meaning Representation Parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2198–2207, Online. Association for Computational Linguistics.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, et al. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, pages 1–18.

Aaron Steven White and Kyle Rawlins. 2018. The role of veridicality and factivity in clause selection. In *Proceedings of the 48th Annual Meeting of the North East Linguistic Society*, pages 221–234.

Aaron Steven White, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2018. Lexicosyntactic inference in neural models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4717–4724, Brussels, Belgium. Association for Computational Linguistics.

A Cooper storage and nested quantification

In the main body, we state a semantics for instance assignments decorated with `:quant`. Here, we state a more general rule for interpreting an instance assignment decorated with `:quant` and an additional sequence of role assignments $\rho_1 \ldots \rho_2$. This is necessary for interpreting AMRs arising from sentences involving nested quantifiers, such as (40-a), which may correpond to a `scope`-enriched AMR such as (40-b), in which a `:quant` role co-occurs with an embedded AMR, itself decorated with a `:quant` role.

(40) a. *Every class with a certain two professors is difficult.*

b.
```
(s / scope
   :ARG0 p
   :ARG1 c
   :pred (d / difficult
      :domain (c / class
         :quant every
         :prep-with (p / prof.
            :quant 2))))
```

The rule for interpreting complex AMRs with a `:quant` role and an additional role sequence is given below. The idea is that stores associated with embedded AMRs are passed up, and the determiner introduce by `:quant` takes scope over the role sequence.

(41) $$[\![\,(\texttt{x / P :quant D}\ \rho_1 \ldots \rho_n)\,]\!]_s = [\![\rho_1 \ldots \rho_n]\!]_s \cup \{(x, \mathbf{D}(\lambda x\,.\,P(x) \wedge [\![\rho_1 \ldots \rho_n]\!]_o\,(x))\}$$

(42) $$[\![\,(\texttt{x / P :quant D}\ \rho_1 \ldots \rho_n)\,]\!]_o = \top$$

This more sophisticated interpretation rule allows us to interpret AMRs such as that in (40-b). The derivation involves storing a quantificational expression containing a free variable, which comes to be bound once the scope node is resolved. In (43) we provide the interpretation of the AMR embedded under `:pred` in (40-b). A full derivation is left as an exercise to the reader.

(43) $$\left\{\begin{array}{c}(p, \mathbf{two}(\mathbf{professor}),\\ (c, \mathbf{every}(\lambda x.\, \mathbf{class}(x) \wedge \mathbf{with}(x)(p)))\end{array}\right\} \cdot \mathbf{difficult}(d) \wedge \mathbf{domain}(d)(c)$$

As a final note, care has to be taken to ensure that the quantifiers are evaluated in a certain order in cases involving nested quantification. Concretely, if the values of `:ARG0` and `:ARG1` are flipped, in the AMR in (40-b), then the free variable in the stored universal quantifier will remain free post-evaluation, which does not correspond to an attested interpretation of (40-a). Besides placing an implausible cognitive load on annotators, this is a known deficiency of the rudimentary Cooper storage system adopted here (Keller, 1988). More sophisticated approaches to Cooper storage which avoid this issue (Kobele, 2018) could be adapted for the purpose of interpreting AMRs; we leave this to future work.