

A Continuation Semantics for Abstract Meaning Representation

Kenneth Lai¹, Lucia Donatelli², James Pustejovsky¹

¹Department of Computer Science
Brandeis University, USA

²Department of Language Science and Technology
Saarland University, Germany

{klai12, jamesp}@brandeis.edu
donatelli@coli.uni-saarland.de

Abstract

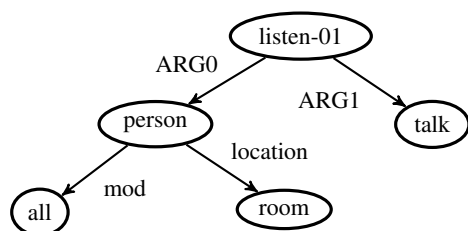
Abstract Meaning Representation (AMR) is a simple, expressive semantic framework whose emphasis on predicate-argument structure is effective for many tasks. Nevertheless, AMR lacks a systematic treatment of projection phenomena, making its translation into logical form problematic. We present a translation function from AMR to first order logic using continuation semantics, which allows us to capture the semantic context of an expression in the form of an argument. This is a natural extension of AMR’s original design principles, allowing us to easily model basic projection phenomena such as quantification and negation as well as complex phenomena such as bound variables and donkey anaphora.

1 Introduction

Abstract Meaning Representation (AMR) is a general-purpose meaning representation that has become popular for its simple structure, ease of annotation and available corpora, and overall expressiveness (Banarescu et al., 2013; Knight et al., 2019). Specifically, AMR focuses on representing the predicative core of a sentence as an intuitive representation of the semantics of a sentence, advantageous for parsing and matching algorithms. As an example, the AMR for the English sentence “Everyone in the room listened to a talk.” is given in example (1), in three equivalent formats.

(1) a. Everyone in the room listened to a talk.

b.



c. (l / listen-01

```
:ARG0 (p / person
      :mod (a / all)
      :location (r / room))
:ARG1 (t / talk))
```

d. $\text{instance}(l, \text{listen-01}) \wedge \text{instance}(p, \text{person}) \wedge \text{instance}(a, \text{all}) \wedge \text{instance}(r, \text{room}) \wedge$
 $\text{instance}(t, \text{talk}) \wedge \text{ARG0}(l, p) \wedge \text{mod}(p, a) \wedge \text{location}(p, r) \wedge \text{ARG1}(l, t)$

For the sentence in (1a), the predicate and its arguments are represented as nodes in the AMR graph in (1b), while the edges represent the relations between the predicate and each of its arguments. PENMAN notation in (1c) provides a more readable version of the AMR (Matthiessen and Bateman, 1991).

By design, AMR emphasizes argument structure over logical structure, distinguishing it from several other meaning representations based on formal semantic frameworks (Kamp and Reyle, 2013; Copestake et al., 2005). Although AMRs can be represented as conjunctions of logical triples, as seen in (1d),

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: <http://creativecommons.org/licenses/by/4.0/>.

such representations cannot be used directly for drawing valid inferences such as textual entailment or contradiction. Quantifiers, for example, are not given any special semantics, being represented as simple modifiers. Furthermore, AMR does not represent quantifier scope in any way, leaving open the question in (1) of how many talks were listened to in the room.

Several proposals have been put forth to provide AMRs with more meaningful logical forms, driven both by parsing and theoretical concerns. Artzi et al. (2015) present a CCG-based grammar induction technique for AMR parsing; a learned joint model integrates classical lambda calculus and underspecification to improve search for compositional and non-compositional elements of AMR, respectively. Semantically speaking, each AMR variable gets its own lambda term (scoped as low as possible) and each AMR role becomes a binary predicate that is applied to those variables. More strictly theoretical work also provides systematic mappings from AMR into logic. Bos (2016) defines a syntax of AMR and provides a recursive translation function into first-order logic (FOL) that is able to handle scope phenomena. In a similar vein, Stabler (2017) adds tense and number information into AMR leaves and makes explicit quantificational determiners as a basis for a mapping to a higher order, dynamic logic. Finally, Pustejovsky et al. (2019) add scope itself to AMR graph structure, representing it relationally in the form of a scope node that attaches to the predicative core when necessary.

Adding to the discussion about the expressive capacity of AMR, Crouch and Kalouli (2018) argue that any purely graphical representation of meaning is unable to capture basic natural language semantics such as Boolean operators. Instead, the authors propose layering graphs based on the Resource Description Framework (RDF) (Schreiber and Raimond, 2014) and named graphs (Carroll et al., 2005) such that the interaction between different layers can capture Booleans (such as negation and disjunction), modals and irrealis contexts, distributivity and quantifier scope, co-reference, and sense selection (Kalouli and Crouch, 2018). However, in this work, we will show how we can give AMR a semantics expressive enough to handle at least some of these phenomena, namely quantification and negation, without needing to introduce additional graph structure.

Here, we build on previous work and present a translation function from AMR to FOL using continuations. A continuation of an expression encodes surrounding contextual information relevant to its interpretation. More specifically, the *continuation hypothesis* assumes that some natural language expressions denote functions that take their own semantic context as an argument (Barker, 2002; Barker and Shan, 2014). In the present discussion, continuations have interesting consequences for the representations associated with the predicative core in an AMR graph structure. Namely, the continuation effectively allows the graph to be rooted at the predicate level, while treating the continuation as an associated argument to the relation associated with that predicate. Thus, our methods allow us to work with standard AMRs without adding features or modifying the graphs, yet still permit us to draw valid inferences.

We take as our starting point the work presented in Bos (2016), as it is the most similar to ours in translating AMR to FOL without significantly altering the AMR feature inventory. We consider the semantics of basic AMRs and show how Bos (2016)’s translations can be naturally formulated in terms of continuations (Section 2). We then demonstrate how continuations allow us to give a simpler, more transparent semantics that can handle scope phenomena (Section 3). We go on to discuss our treatment of universal quantification (Section 4) and negation (Section 5) before examining how our continuation semantics deals with “donkey sentences” (Section 6). Finally we discuss implications of our work and conclude (Sections 7 and 8).

2 A Basic Continuation Semantics

Continuations capture important contextual information surrounding an expression useful for disambiguating the expression’s interpretation. In the same way that the generalized quantifier interpretation of an entity is the set of properties true of that entity, the continuation of an expression is the set of *contexts* that, when combined with that expression, result in a true sentence. Alternatively, in terms of the characteristic function of that set of contexts, the continuation of a word or phrase is the entire future of the computation of that expression, packaged as a function over the expression itself (Barker and Shan,

2014; Reynolds, 1993).

With the goal of integrating continuations into our methodology, we begin in the same way Bos (2016) does, by considering the semantics of basic AMRs. Basic AMRs are comprised of constants c (used in, e.g., proper names) and instance assignments (x/P) where a variable x is declared to be an instance of a predicate P . Instance assignments may also have out-going roles $R_i A_i$, where the instance denoted by the AMR A_i fills the role R_i of its parent. We provide a semantics for these basic AMRs below in Definition 1:

Definition 1 (Semantics of Basic AMRs)

$$\begin{aligned} \llbracket c \rrbracket &= \lambda\phi.\phi(c) \\ \llbracket (x/P) \rrbracket &= \lambda\phi.\exists x.P(x) \wedge \phi(x) \\ \llbracket (x/P : R_1 A_1 \dots : R_n A_n) \rrbracket &= \lambda\phi.\exists x.P(x) \wedge \llbracket A_1 \rrbracket(\lambda y.R_1(x, y)) \wedge \dots \wedge \llbracket A_n \rrbracket(\lambda y.R_n(x, y)) \wedge \phi(x) \end{aligned}$$

The above definition is nearly identical to that given by Bos (2016) for basic AMRs, and also, in turn, very similar to that of Artzi et al. (2015). In fact, our definition can be seen as a notational variant (namely, an eta-conversion) of Bos’ definition: where our translations have the general functional form $\llbracket A \rrbracket = \lambda\phi.f(\phi)$, Bos’ translations have $\llbracket A, \phi \rrbracket = f(\phi)$. Bos describes ϕ as a “ λ -expression for roles”; the purpose of ϕ is to “delay the translation of roles” until we have evaluated the meaning of the instance to which we are assigning the role. But it is also possible to consider ϕ from another perspective: in a formula such as $\phi(c)$, ϕ contains contextual information, namely, about the role c plays in relation to its parent node. In other words, ϕ is a continuation. In Definition 1, we thus define the meanings of AMRs directly, as functions of their continuations. We note that the first two meanings in Definition 1 are those of generalized quantifiers (Montague, 1973; Barwise and Cooper, 1981), or continuized noun phrases (NPs) (Barker, 2002), for “c” and “some P”, respectively.

A very simple example is given below. An AMR for the sentence “A dog barked” is given in example (2). The calculation of the FOL formula from the AMR is then shown in (3).

(2) a. A dog barked.

b. (b / bark-01
:ARG0 (d / dog))

$$\begin{aligned} (3) \llbracket \text{a dog barked} \rrbracket &= \lambda\phi.\exists b.\text{bark-01}(b) \wedge \llbracket (d/\text{dog}) \rrbracket(\lambda y.\text{ARG0}(b, y)) \wedge \phi(b) \\ &= \lambda\phi.\exists b.\text{bark-01}(b) \wedge (\lambda\psi.\exists d.\text{dog}(d) \wedge \psi(d))(\lambda y.\text{ARG0}(b, y)) \wedge \phi(b) \\ &= \lambda\phi.\exists b.\text{bark-01}(b) \wedge \exists d(\text{dog}(d) \wedge (\lambda y.\text{ARG0}(b, y))(d)) \wedge \phi(b) \\ &= \lambda\phi.\exists b.\text{bark-01}(b) \wedge \exists d(\text{dog}(d) \wedge \text{ARG0}(b, d)) \wedge \phi(b) \\ &\rightsquigarrow \exists b.\text{bark-01}(b) \wedge \exists d(\text{dog}(d) \wedge \text{ARG0}(b, d)) \end{aligned}$$

3 Unifying Assertive and Projective Semantics

Next, we consider the case of projection phenomena, which take wide scope over their parent predicates. Such constructions are very common in natural language; Bos (2016) notes that proper names, appositive expressions, definite descriptions, and possessive constructions all fall into this category, while Stabler (2017), following Champollion (2015) and Landman (1996), points out that arguments of an event in general should scope over the event predicate by default.

As an example, the sentence in (4a), “A dog scratched itself”, contains a reflexive pronoun. In its AMR in (4b), $(d \setminus \text{dog})$ must project over $(s / \text{scratch-01})$, in order for its quantifier to bind the re-entrant $:\text{ARG1 } d$ (following Bos (2016), we use the backslash “\” to mark projection phenomena).

(4) a. A dog scratched itself.

b. (s / scratch-01
:ARG0 (d \ dog)
:ARG1 d)

Bos (2016) accounts for projection phenomena by computing a “projective semantics” and an “assertive semantics” for each sentence. We will not give the full derivations here, but in the above example, the meaning contributed by the projective semantics is $\lambda p.\exists d.\text{dog}(d) \wedge p$, and that from the assertive semantics is $\exists s.\text{scratch-01}(s) \wedge \text{ARG0}(s, d) \wedge \text{ARG1}(s, d)$. The projective meaning is then applied to the assertive meaning, resulting in $\exists d.\text{dog}(d) \wedge \exists s.\text{scratch-01}(s) \wedge \text{ARG0}(s, d) \wedge \text{ARG1}(s, d)$.

One way to look at the above procedure is to note that the projective semantics translates the projective concepts (here, $(d \setminus \text{dog})$), while the assertive semantics translates everything else. Then, applying the projective meaning to the assertive meaning ensures that the projective concepts take scope over the rest of the sentence. An alternative viewpoint notes that the projective meaning $\lambda p.\exists d.\text{dog}(d) \wedge p$ is an expression that, when applied to the assertive meaning p , yields the meaning of the sentence. This is extremely similar to how an expression is applied to its continuation to yield a sentence meaning in a continuation semantics. In fact, we can write a single translation function that makes this connection explicit, and dispenses with the need for separate projective and assertive semantics.

To do so, we revisit the meaning $\llbracket (x/P : R_1 A_1 \dots : R_n A_n) \rrbracket$. Instead of giving a direct translation as in Definition 1, we decompose it into two parts: an out-going role $\llbracket :R_1 A_1 \rrbracket$ and the rest of the AMR $\llbracket (x/P : R_i A_i) \rrbracket$, where $:R_i A_i$ is a (possibly empty) list of the rest of the out-going roles. From Definition 1, we find that the part of the meaning corresponding to $:R_1 A_1$ is $\llbracket A_1 \rrbracket(\lambda y.R_1(x, y))$, i.e., the meaning $\llbracket A_1 \rrbracket$ being applied to the continuation $\lambda y.R_1(x, y)$. Now we can consider the rest of the AMR, in particular, its continuation. If we take the continuation to represent the rest of the sentence, then this also has two parts: the out-going role meaning $\llbracket A_1 \rrbracket(\lambda y.R_1(x, y))$, and $\phi(x)$, the original continuation. We can conjoin them, and bind the x with a λ , to get $\lambda x.\llbracket A_1 \rrbracket(\lambda y.R_1(x, y)) \wedge \phi(x)$. Then we can apply the meaning of the rest of the AMR, namely $\llbracket (x/P : R_i A_i) \rrbracket$, to this continuation (with a couple of α -conversions to avoid name collisions), to get a recursive semantics for an AMR with out-going roles:

$$\llbracket (x/P : R_1 A_1 : R_i A_i) \rrbracket = \lambda \phi.\llbracket (x/P : R_i A_i) \rrbracket(\lambda m.\llbracket A_1 \rrbracket(\lambda n.R_1(m, n)) \wedge \phi(m))$$

In a continuation semantics, scope and application order are inextricably linked. As such, to make an argument scope over its predicate, we can reverse the order of the role and the rest of the AMR. Now $\llbracket (x/P : R_i A_i) \rrbracket$ only gets the λ -expression for R_1 and the original continuation $\phi(m)$, while the projective concept $\llbracket (y \setminus Q \dots) \rrbracket$ is fed the rest of the sentence as its continuation:

$$\llbracket (x/P : R_1(y \setminus Q \dots) : R_i A_i) \rrbracket = \lambda \phi.\llbracket (y \setminus Q \dots) \rrbracket(\lambda n.\llbracket (x/P : R_i A_i) \rrbracket(\lambda m.R_1(m, n) \wedge \phi(m)))$$

We can apply the two rules above recursively until $:R_i A_i$ becomes empty, i.e., we run out of arguments. Then we can use our previous rule $\llbracket (x/P) \rrbracket = \lambda \phi.\exists x.P(x) \wedge \phi(x)$, as the base case of our recursion. We also adopt the rule $\llbracket x \rrbracket = \lambda \phi.\phi(x)$ to handle re-entrant nodes, such as d in example (4b). Finally, it is important to note that the application order depends solely on whether the argument A_1 is a projective concept $(y \setminus Q \dots)$ or not, and not at all on whether the parent node (x/P) is projective or not. We can use a vertical bar “|” to denote that a rule applies regardless of whether a concept is projective or not. We summarize our work thus far in Definition 2:

Definition 2 (Semantics of AMRs with Projection)

$$\begin{aligned} \llbracket c \rrbracket &= \lambda \phi.\phi(c) \\ \llbracket x \rrbracket &= \lambda \phi.\phi(x) \\ \llbracket (x|P : R_1(y \setminus Q \dots) : R_i A_i) \rrbracket &= \lambda \phi.\llbracket (y \setminus Q \dots) \rrbracket(\lambda n.\llbracket (x|P : R_i A_i) \rrbracket(\lambda m.R_1(m, n) \wedge \phi(m))) \\ \llbracket (x|P : R_1 A_1 : R_i A_i) \rrbracket &= \lambda \phi.\llbracket (x|P : R_i A_i) \rrbracket(\lambda m.\llbracket A_1 \rrbracket(\lambda n.R_1(m, n)) \wedge \phi(m)) \\ \llbracket (x|P) \rrbracket &= \lambda \phi.\exists x.P(x) \wedge \phi(x) \end{aligned}$$

As an example, we calculate the meaning of the sentence in (4a), “A dog scratched itself” below. We already know that the meaning of “a dog” is $\lambda \phi.\exists d.\text{dog}(d) \wedge \phi(d)$, so in (5) we focus on the meaning of “scratched itself”:

$$\begin{aligned}
(5) \llbracket \text{scratched itself} \rrbracket &= \lambda\phi. \llbracket \text{scratched} \rrbracket (\lambda m. \llbracket \text{itself} \rrbracket (\lambda n. \text{ARG1}(m, n)) \wedge \phi(m)) \\
&= \lambda\phi. \llbracket \text{scratched} \rrbracket (\lambda m. (\lambda\psi. \psi(d)) (\lambda n. \text{ARG1}(m, n)) \wedge \phi(m)) \\
&= \lambda\phi. \llbracket \text{scratched} \rrbracket (\lambda m. (\lambda n. \text{ARG1}(m, n))(d) \wedge \phi(m)) \\
&= \lambda\phi. \llbracket \text{scratched} \rrbracket (\lambda m. \text{ARG1}(m, d) \wedge \phi(m)) \\
&= \lambda\phi. (\lambda\psi. \exists s. \text{scratch-01}(s) \wedge \psi(s)) (\lambda m. \text{ARG1}(m, d) \wedge \phi(m)) \\
&= \lambda\phi. \exists s. \text{scratch-01}(s) \wedge (\lambda m. \text{ARG1}(m, d) \wedge \phi(m))(s) \\
&= \lambda\phi. \exists s. \text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge \phi(s)
\end{aligned}$$

Then in (6) we combine the meanings of “a dog” and “scratched itself”. Note that as an intermediate step, we apply $\lambda\psi. \exists d. \text{dog}(d) \wedge \psi(d)$ to $\lambda n. \exists s. \text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge \text{ARG0}(s, n) \wedge \phi(s)$, in a very similar manner to how Bos (2016) applies the projective meaning of a sentence to the assertive meaning.

$$\begin{aligned}
(6) \llbracket \text{a dog scratched itself} \rrbracket &= \lambda\phi. \llbracket \text{a dog} \rrbracket (\lambda n. \llbracket \text{scratched itself} \rrbracket (\lambda m. \text{ARG0}(m, n) \wedge \phi(m))) \\
&= \lambda\phi. \llbracket \text{a dog} \rrbracket (\lambda n. (\lambda\psi. \exists s. \text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge \psi(s)) (\lambda m. \text{ARG0}(m, n) \wedge \phi(m))) \\
&= \lambda\phi. \llbracket \text{a dog} \rrbracket (\lambda n. \exists s. \text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge (\lambda m. \text{ARG0}(m, n) \wedge \phi(m))(s)) \\
&= \lambda\phi. \llbracket \text{a dog} \rrbracket (\lambda n. \exists s. \text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge \text{ARG0}(s, n) \wedge \phi(s)) \\
&= \lambda\phi. (\lambda\psi. \exists d. \text{dog}(d) \wedge \psi(d)) (\lambda n. \exists s. \text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge \text{ARG0}(s, n) \wedge \phi(s)) \\
&= \lambda\phi. \exists d. \text{dog}(d) \wedge (\lambda n. \exists s. \text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge \text{ARG0}(s, n) \wedge \phi(s))(d) \\
&= \lambda\phi. \exists d. \text{dog}(d) \wedge \exists s. \text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge \text{ARG0}(s, d) \wedge \phi(s) \\
&\rightsquigarrow \exists d. \text{dog}(d) \wedge \exists s. \text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge \text{ARG0}(s, d)
\end{aligned}$$

4 Universal Quantification

As previously mentioned, quantifiers in AMR are treated as simple modifiers of their head concepts. In a logical form like (1d), all variables are assumed to be existentially quantified. Nevertheless, it is possible to give AMR a proper semantics of quantification. Stabler (2017), for example, is able to translate a number of different kinds of quantifiers into a higher-order logic. In this paper, to stay within the setting of FOL, we will focus on the universal quantifier.

One way of introducing universal quantification into a semantics for AMR, as noted by Bos (2016), is to include a rule such as $\llbracket (x/P : \text{quant } \forall) \rrbracket = \lambda\phi. \forall x. P(x) \rightarrow \phi(x)$, where “:quant \forall ” denotes any expression that can be interpreted as a universal quantifier. One can verify that, for a simple sentence like “Every dog scratched itself”, combining the meaning of “every dog”, i.e., $\lambda\psi. \forall d. \text{dog}(d) \rightarrow \psi(d)$, and “scratched itself”, from (5), one arrives at $\forall d. \text{dog}(d) \rightarrow \exists s. \text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge \text{ARG0}(s, d)$, the correct meaning of the sentence.

However, for more complex NPs, including those that are modified by relative clauses, or even simply by a pre-nominal adjective, the above rule will not provide the right semantics. Consider the sentence “Every brown dog scratched itself” in (7) below:

$$\begin{aligned}
(7) \text{ a. Every brown dog scratched itself.} \\
\text{ b. } (s / \text{scratch-01} \\
&\quad : \text{ARG0 } (d \setminus \text{dog} \\
&\quad \quad : \text{quant } (e / \text{every}) \\
&\quad \quad : \text{mod } (b / \text{brown})) \\
&\quad : \text{ARG1 } d)
\end{aligned}$$

Again, we will not give the full derivation here, but the meaning we get if we use the above rule is $\forall d.\text{dog}(d) \rightarrow \exists b(\text{brown}(b) \wedge \text{mod}(d, b)) \wedge \exists s.\text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge \text{ARG0}(s, d)$, i.e., “every dog is brown and scratched itself”. The problem arises from the fact that in the formula $\lambda\phi.\forall x.P(x) \rightarrow \phi(x)$, only the predicate P is allowed to appear in the restriction; everything else is forced into the continuation ϕ , which is interpreted in the nuclear scope.

Our solution to this problem is inspired by Barker and Shan (2008)’s treatment of the universal quantifier, and is also similar to the dynamic meaning of “every” given by de Groote (2006). We will use the below rule for universal quantification:

$$\llbracket (x|P : \text{quant } \forall : R_i A_i) \rrbracket = \lambda\phi.\neg\llbracket (x|P : R_i A_i) \rrbracket(\lambda m.\neg\phi(m))$$

To understand how this rule works, note that the meaning $\llbracket (x|P : R_i A_i) \rrbracket$ will have a logical form like $\lambda\psi.\exists x.P(x) \wedge \dots \wedge \psi(x)$. Applying this meaning to the continuation $\lambda m.\neg\phi(m)$ eventually results in $\exists x.P(x) \wedge \dots \wedge \neg\phi(x)$, which we then negate to get $\neg\exists x.P(x) \wedge \dots \wedge \neg\phi(x)$. Using the logical identity $\neg\exists x.\phi(x) \wedge \neg\psi(x) \equiv \forall x.\phi(x) \rightarrow \psi(x)$, we finally get the meaning $\lambda\phi.\forall x.(P(x) \wedge \dots) \rightarrow \phi(x)$, with the “...” representing material introduced by an adjective or in a relative clause, correctly interpreted in the restriction of the quantifier. This can be seen in the derivation of the meaning of “every brown dog” below in (8):

$$\begin{aligned} (8) \llbracket \text{every brown dog} \rrbracket &= \lambda\phi.\neg\llbracket \text{brown dog} \rrbracket(\lambda p.\neg\phi(p)) \\ &= \lambda\phi.\neg(\lambda\psi.\llbracket \text{dog} \rrbracket(\lambda m.\llbracket \text{brown} \rrbracket(\lambda n.\text{mod}(m, n)) \wedge \psi(m)))(\lambda p.\neg\phi(p)) \\ &= \lambda\phi.\neg\llbracket \text{dog} \rrbracket(\lambda m.\llbracket \text{brown} \rrbracket(\lambda n.\text{mod}(m, n)) \wedge (\lambda p.\neg\phi(p))(m)) \\ &= \lambda\phi.\neg\llbracket \text{dog} \rrbracket(\lambda m.\llbracket \text{brown} \rrbracket(\lambda n.\text{mod}(m, n)) \wedge \neg\phi(m)) \\ &\dots \\ &= \lambda\phi.\neg\llbracket \text{dog} \rrbracket(\lambda m.\exists b(\text{brown}(b) \wedge \text{mod}(m, b)) \wedge \neg\phi(m)) \\ &\dots \\ &= \lambda\phi.\neg\exists d.\text{dog}(d) \wedge \exists b(\text{brown}(b) \wedge \text{mod}(d, b)) \wedge \neg\phi(d) \\ &\equiv \lambda\phi.\forall d.(\text{dog}(d) \wedge \exists b(\text{brown}(b) \wedge \text{mod}(d, b))) \rightarrow \phi(d) \end{aligned}$$

We can then combine this meaning of “every brown dog” with the meaning of “scratched itself” from (5) below in (9) to get the correct meaning of the sentence:

$$\begin{aligned} (9) \llbracket \text{every brown dog scratched itself} \rrbracket &= \lambda\phi.\llbracket \text{every brown dog} \rrbracket(\lambda n.\llbracket \text{scratched itself} \rrbracket(\lambda m.\text{ARG0}(m, n) \wedge \phi(m))) \\ &\dots \\ &= \lambda\phi.\llbracket \text{every brown dog} \rrbracket(\lambda n.\exists s.\text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge \text{ARG0}(s, n) \wedge \phi(s)) \\ &= \lambda\phi.(\lambda\psi.\forall d.(\text{dog}(d) \wedge \exists b(\text{brown}(b) \wedge \text{mod}(d, b))) \rightarrow \psi(d)) \\ &\quad (\lambda n.\exists s.\text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge \text{ARG0}(s, n) \wedge \phi(s)) \\ &= \lambda\phi.\forall d.(\text{dog}(d) \wedge \exists b(\text{brown}(b) \wedge \text{mod}(d, b))) \\ &\quad \rightarrow (\lambda n.\exists s.\text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge \text{ARG0}(s, n) \wedge \phi(s))(d) \\ &= \lambda\phi.\forall d.(\text{dog}(d) \wedge \exists b(\text{brown}(b) \wedge \text{mod}(d, b))) \\ &\quad \rightarrow \exists s.\text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge \text{ARG0}(s, d) \wedge \phi(s) \\ &\rightsquigarrow \forall d.(\text{dog}(d) \wedge \exists b(\text{brown}(b) \wedge \text{mod}(d, b))) \rightarrow \exists s.\text{scratch-01}(s) \wedge \text{ARG1}(s, d) \wedge \text{ARG0}(s, d) \end{aligned}$$

We present our translation function thus far in Definition 3 below.

Definition 3 (Semantics of AMRs with Projection and Universal Quantification)

$$\begin{aligned}
\llbracket c \rrbracket &= \lambda\phi.\phi(c) \\
\llbracket x \rrbracket &= \lambda\phi.\phi(x) \\
\llbracket (x|P : \text{quant } \forall : R_i A_i) \rrbracket &= \lambda\phi.\neg\llbracket (x|P : R_i A_i) \rrbracket(\lambda m.\neg\phi(m)) \\
\llbracket (x|P : R_1(y \setminus Q \dots) : R_i A_i) \rrbracket &= \lambda\phi.\llbracket (y \setminus Q \dots) \rrbracket(\lambda n.\llbracket (x|P : R_i A_i) \rrbracket(\lambda m.R_1(m, n) \wedge \phi(m))) \\
\llbracket (x|P : R_1 A_1 : R_i A_i) \rrbracket &= \lambda\phi.\llbracket (x|P : R_i A_i) \rrbracket(\lambda m.\llbracket A_1 \rrbracket(\lambda n.R_1(m, n)) \wedge \phi(m)) \\
\llbracket (x|P) \rrbracket &= \lambda\phi.\exists x.P(x) \wedge \phi(x)
\end{aligned}$$

Note that, like other continuation-passing rules, the order in which the rules apply is critical to deriving the correct interpretation. If we interpret the universal quantifier after “brown”, rather than before, then “brown” will become part of the continuation of “every dog”, and we get the previous incorrect meaning. We will discuss ways to specify the application order in Section 7, but for now, we will process nodes in linear order, i.e., the order they are written in the AMR.

5 Negation

Negation is central to sentence-level meaning, as it directly impacts the assignment of truth values to propositions. In AMR, negation is represented as a predicate rather than a scope operator: a fixed polarity relation is given between the negated concept and negation constant “-”. As a result, AMR makes wrong predictions for inferences based on negated sentences; for example, it will allow the inference “a dog barked” from “a dog did not bark.” Both Bos (2016) and Stabler (2017) reformulate AMR semantics to allow polarity to act on propositions instead of concepts; here, we expand on the work of Bos (2016) and show the need for two kinds of negation to allow for distinct readings in ambiguous contexts.

Bos (2016) proposes a translation of AMR to FOL in which negation attaches to the quantifier that binds its concept. For example, the AMRs in (10a) and (11a) receive the respective interpretations in (10b) and (11b):

<p>(10) a. (m / meow-01 :ARG0 (d / dog) :polarity -)</p> <p>b. $\neg\exists m.\text{meow-01}(m) \wedge \exists d.\text{dog}(d) \wedge \text{ARG0}(m, d)$ <i>“No dog meowed.”</i></p>	<p>(11) a. (m / meow-01 :ARG0 (d / dog :polarity -))</p> <p>b. $\exists m.\text{meow-01}(m) \wedge \neg\exists d.\text{dog}(d) \wedge \text{ARG0}(m, d)$ <i>“It was not a dog that meowed.”/ “A non-dog meowed.”</i></p>
--	---

Note that a logical form such as $\exists m.\text{meow-01}(m) \wedge \exists d.\neg\text{dog}(d) \wedge \text{ARG0}(m, d)$, with the negation attached directly to the predicate, rather than the quantifier, can also give rise to the interpretation in (11b). In fact, assuming that the meowing event has exactly one ARG0, the two forms are equivalent. It may then seem that any formula with predicate negation can be given an equivalent formula with quantifier negation, and that therefore quantifier negation is sufficient for the semantics of AMRs.

However, there are meanings that cannot be represented with negation only attaching to quantifiers, particularly involving the interaction of negation and universal quantification. For example, a meaning such as “Every non-dog meowed”, i.e., $\forall d.\neg\text{dog}(d) \rightarrow \exists m.\text{meow-01}(m) \wedge \text{ARG0}(m, d)$, cannot be generated from any AMR using the rules in Bos (2016). For this reason, we introduce the following two rules for negation:

$$\begin{aligned}
\llbracket (x|P : \text{polarity } (n \setminus -) : R_i A_i) \rrbracket &= \lambda\phi.\neg\llbracket (x|P : R_i A_i) \rrbracket(\phi) \\
\llbracket (x|P : \text{polarity } (n / -) : R_i A_i) \rrbracket &= \lambda\phi.\llbracket (x|\neg P : R_i A_i) \rrbracket(\phi)
\end{aligned}$$

We adopt the same backslash/forward slash syntax as for projective and non-projective concepts, here simply meaning “wide-scope” (i.e., negation attaching to the quantifier) and “narrow-scope” (i.e., negation attaching to the predicate), respectively. An example AMR for the sentence “Every non-dog meowed”, with narrow-scope negation, is shown in (12), and the derivation of its meaning is given in (13).

(12) a. Every non-dog meowed.

b. (m / meow-01
 :ARG0 (d \ dog
 :quant (e / every)
 :polarity (n / -)))

(13) \llbracket every non-dog meowed \rrbracket

= $\lambda\phi.\llbracket$ every non-dog $\rrbracket(\lambda n.\llbracket$ meowed $\rrbracket(\lambda m.\text{ARG0}(m, n) \wedge \phi(m)))$
 ...
 = $\lambda\phi.\llbracket$ every non-dog $\rrbracket(\lambda n.\exists m.\text{meow-01}(m) \wedge \text{ARG0}(m, n) \wedge \phi(m))$
 ...
 = $\lambda\phi.\neg\llbracket$ non-dog $\rrbracket(\lambda n.\neg\exists m.\text{meow-01}(m) \wedge \text{ARG0}(m, n) \wedge \phi(m))$
 = $\lambda\phi.\neg(\lambda\psi.\llbracket$ neg-dog $\rrbracket(\psi))(\lambda p.\neg\exists m.\text{meow-01}(m) \wedge \text{ARG0}(m, p) \wedge \phi(m))$
 = $\lambda\phi.\neg\llbracket$ neg-dog $\rrbracket(\lambda p.\neg\exists m.\text{meow-01}(m) \wedge \text{ARG0}(m, p) \wedge \phi(m))$
 ...
 = $\lambda\phi.\neg\exists d.\neg\text{dog}(d) \wedge \neg\exists m.\text{meow-01}(m) \wedge \text{ARG0}(m, d) \wedge \phi(m)$
 $\rightsquigarrow \forall d.\neg\text{dog}(d) \rightarrow \exists m.\text{meow-01}(m) \wedge \text{ARG0}(m, d)$

One can verify that reversing the direction of the slash, i.e., changing the polarity to $(n \setminus -)$, results in the meaning $\neg\forall d.\text{dog}(d) \rightarrow \exists m.\text{meow-01}(m) \wedge \text{ARG0}(m, d)$, i.e., “Not every dog meowed”. Definition 4 shows the final version of our translation function:

Definition 4 (Semantics of AMRs with Projection, Universal Quantification, and Negation)

$\llbracket c \rrbracket = \lambda\phi.\phi(c)$
 $\llbracket x \rrbracket = \lambda\phi.\phi(x)$
 $\llbracket (x|P \text{ :polarity } (n \setminus -) :R_i A_i) \rrbracket = \lambda\phi.\neg\llbracket (x|P :R_i A_i) \rrbracket(\phi)$
 $\llbracket (x|P \text{ :polarity } (n / -) :R_i A_i) \rrbracket = \lambda\phi.\llbracket (x|\neg P :R_i A_i) \rrbracket(\phi)$
 $\llbracket (x|P \text{ :quant } \forall :R_i A_i) \rrbracket = \lambda\phi.\neg\llbracket (x|P :R_i A_i) \rrbracket(\lambda m.\neg\phi(m))$
 $\llbracket (x|P :R_1(y \setminus Q \dots) :R_i A_i) \rrbracket = \lambda\phi.\llbracket (y \setminus Q \dots) \rrbracket(\lambda n.\llbracket (x|P :R_i A_i) \rrbracket(\lambda m.R_1(m, n) \wedge \phi(m)))$
 $\llbracket (x|P :R_1 A_1 :R_i A_i) \rrbracket = \lambda\phi.\llbracket (x|P :R_i A_i) \rrbracket(\lambda m.\llbracket A_1 \rrbracket(\lambda n.R_1(m, n)) \wedge \phi(m))$
 $\llbracket (x|P) \rrbracket = \lambda\phi.\exists x.P(x) \wedge \phi(x)$

6 AMR for Donkey Sentences

“Donkey sentences” (Geach, 1962) are known for raising interesting issues regarding the interaction between quantification and anaphora. In the sentence “Every farmer who owns a donkey loves it” in (14a) below, “a donkey” must be able to take scope over the anaphor “it”. Furthermore, “a donkey” must also be interpreted as being universally quantified, i.e., every farmer loves every donkey they own. An AMR for this sentence is shown in (14b). In this section, we will show how our continuation semantics can give this AMR an appropriate meaning.

(14) a. Every farmer who owns a donkey loves it.

b. (1 / love-01
 :ARG0 (f \ farmer
 :quant (e / every)
 :ARG0-of (o \ own-01
 :ARG1 (d \ donkey)))
 :ARG1 d)

Before we begin our translation from the AMR to FOL, we note that the concepts ($f \setminus \text{farmer}$), ($o \setminus \text{own-01}$), and ($d \setminus \text{donkey}$) are all projective. Certainly we want ($d \setminus \text{donkey}$) to take wide scope. Following Champollion (2015), we have, in general, the arguments of an event scoping over the event, hence the wide scope for ($f \setminus \text{farmer}$). As for ($o \setminus \text{own-01}$), there is evidence to suggest that events in the restriction should be able to project over the rest of the sentence; e.g., in “Everyone who dined last night got sick afterward”, the dining event should take wide scope, in order to be accessible to “afterward”.

We now derive the meaning of the subject NP “every farmer who owns a donkey” below in (15). Using the translation rules we defined earlier, we eventually get the meaning $\lambda\phi.\neg\exists d.\text{donkey}(d) \wedge \dots$. While it may seem odd for the meaning of “every farmer...” to begin with “there does not exist a donkey...”, we can use a few logical identities to transform the meaning into a more intuitive form. First, just as in example (8), we can use the identity $\neg\exists x.\phi(x) \wedge \neg\psi(x) \equiv \forall x.\phi(x) \rightarrow \psi(x)$ to rewrite the meaning in terms of universal quantifiers and material conditionals. Then, we can take advantage of the facts that $\phi \rightarrow \forall x.\psi \equiv \forall x(\phi \rightarrow \psi)$ (if x is not free in ϕ) and $\phi \rightarrow (\psi \rightarrow \chi) \equiv (\phi \wedge \psi) \rightarrow \chi$ to move all the universal quantifiers to the beginning (i.e., write the formula in prenex normal form), and move all the terms except $\phi(f)$ into the antecedent of a single conditional. At that point, we can rearrange the quantifiers and the conjuncts to get the more natural-looking meaning at the end of (15).

$$\begin{aligned}
(15) \llbracket \text{every farmer who owns a donkey} \rrbracket &= \lambda\phi.\neg\llbracket \text{farmer who owns a donkey} \rrbracket(\lambda m.\neg\phi(m)) \\
&\dots \\
&= \lambda\phi.\neg\llbracket \text{owns a donkey} \rrbracket(\lambda n.\exists f.\text{farmer}(f) \wedge \text{ARG0-of}(f, n) \wedge \neg\phi(f)) \\
&\dots \\
&= \lambda\phi.\neg\llbracket \text{a donkey} \rrbracket(\lambda n.\exists o.\text{own-01}(o) \wedge \text{ARG1}(o, n) \wedge \exists f.\text{farmer}(f) \wedge \text{ARG0-of}(f, o) \wedge \neg\phi(f)) \\
&\dots \\
&= \lambda\phi.\neg\exists d.\text{donkey}(d) \wedge \exists o.\text{own-01}(o) \wedge \text{ARG1}(o, d) \wedge \exists f.\text{farmer}(f) \wedge \text{ARG0-of}(f, o) \wedge \neg\phi(f) \\
&\dots \\
&\equiv \lambda\phi.\forall d.\text{donkey}(d) \rightarrow \forall o.(\text{own-01}(o) \wedge \text{ARG1}(o, d)) \rightarrow \forall f.(\text{farmer}(f) \wedge \text{ARG0-of}(f, o)) \rightarrow \phi(f) \\
&\dots \\
&\equiv \lambda\phi.\forall f.\forall d.\forall o.(\text{farmer}(f) \wedge \text{donkey}(d) \wedge \text{own-01}(o) \wedge \text{ARG0-of}(f, o) \wedge \text{ARG1}(o, d)) \rightarrow \phi(f)
\end{aligned}$$

One can verify that the meaning of “loves it” is $\lambda\phi.\exists l.\text{love-01}(l) \wedge \text{ARG1}(l, d) \wedge \phi(l)$, similar to the meaning of “scratched itself” in (5). Then we can combine the two meanings to get the meaning of the entire sentence in (16):

$$\begin{aligned}
(16) \llbracket \text{every farmer who owns a donkey loves it} \rrbracket &= \lambda\phi.\llbracket \text{every farmer who owns a donkey} \rrbracket(\lambda n.\llbracket \text{loves it} \rrbracket(\lambda m.\text{ARG0}(m, n) \wedge \phi(m))) \\
&\dots \\
&\rightsquigarrow \forall f.\forall d.\forall o.(\text{farmer}(f) \wedge \text{donkey}(d) \wedge \text{own-01}(o) \wedge \text{ARG0-of}(f, o) \wedge \text{ARG1}(o, d)) \\
&\quad \rightarrow \exists l.\text{love-01}(l) \wedge \text{ARG1}(l, d) \wedge \text{ARG0}(l, f)
\end{aligned}$$

7 Discussion

Continuations are a natural solution to the challenge of translating AMR to FOL: by treating the continuation of an expression as an associated argument to the relation associated with that predicate, we maintain AMR’s focus on the predicative core while still allowing valid inferences from projection phenomena to fall out. Our method avoids the pitfalls of underspecification, allowing us to prioritize the most plausible interpretation of a scope ambiguity yet also to capture less common interpretations when necessary (Bos and Abzianidze, 2019). Our method also does not modify standard AMR nodes, leaves, or edges, which allows us to utilize existing AMR corpora. Formalizing basic projection phenomena in AMR paves the way for more comprehensive meaning representation, allowing simple translation of complex phenomena such as negative raising that evidence speaker belief and intent.

Another interesting advantage of the continuation-passing style semantics introduced here for sentence or utterance level expressions, is the natural way it can be extended to model how AMRs have recently

been used in human-robot interaction dialogues (Bonial et al., 2020). The model presented here can easily adopt the dynamic semantics of “discourse moves as continuations”, as introduced by de Groot (2001), and extended by Asher and Pogodalla (2010).

As previously mentioned, in a continuation semantics, the order of application determines the relative scope of a predicate and each of its arguments. In this paper, we take the out-going roles of a predicate to be ordered, and the order of application to be the order the arguments are written in the AMR. In contrast, Pustejovsky et al. (2019) attach an optional scope node to the predicate, that explicitly marks the arguments of a predicate with a relative scope ordering.

Ease of annotation is considered one of the major advantages of AMR compared to other meaning representations (Banarescu et al., 2013; Knight et al., 2019). It is possible that introducing a new backslash notation for projective concepts and wide-scope negation may increase annotators’ cognitive load. We believe, though, that this effect can be mitigated by specifying default interpretations of projection and negation where possible. For example, Stabler (2017) notes that arguments of an event should be projective in general, and Champollion (2015) comments that negation should take wide scope over event quantifiers by default. In addition to providing guidance to annotators, such defaults can also be used on existing corpora, another area where AMR excels. We plan to conduct a corpus study to evaluate the feasibility of using these defaults to generate logical forms from AMRs in the future.

Although our semantics is able to handle many different kinds of scope phenomena, much more work is needed to capture the plethora of meanings possible in natural language. Extending the translation we present here to logical connectives, conditionals, modality, non-declarative sentences that depend on notions of common ground, and other types of quantifiers (e.g. “most”) are all possible next steps. In recent years, there have been a number of proposals to extend AMR to handle definiteness (Stabler, 2017), tense and aspect (Donatelli et al., 2018), and discourse relations (O’Gorman et al., 2018). We look forward to seeing how these proposals and others can be integrated with our semantics.

8 Conclusion

In this paper, we presented a continuation semantics for AMR, building off previous work in translating AMRs to logical forms. We showed that our semantics is powerful enough to handle a wide variety of scope phenomena, including quantification, negation, bound variables, and donkey anaphora. Code for this paper, combining a PENMAN parser based on Goodman (2020), with a computational implementation of our translation function, is available at <https://github.com/klai12/amr2fol>.

Acknowledgements

We would like to thank the reviewers for their helpful comments. This work is supported by the IIS Division of National Science Foundation via Award No. 1763926 entitled “Building a Uniform Meaning Representation for Natural Language Processing”. All views expressed in this paper are those of the authors and do not necessarily represent the view of the National Science Foundation.

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, September. Association for Computational Linguistics.
- Nicholas Asher and Sylvain Pogodalla. 2010. SDRT and continuation semantics. In *JSAI International Symposium on Artificial Intelligence*, pages 3–15. Springer.
- 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.
- Chris Barker and Chung-chieh Shan. 2008. Donkey anaphora is in-scope binding. *Semantics and Pragmatics*, 1:1–46.

- Chris Barker and Chung-chieh Shan. 2014. *Continuations and Natural Language*, volume 53. Oxford Studies in Theoretical Linguistics.
- Chris Barker. 2002. Continuations and the nature of quantification. *Natural Language Semantics*, 10(3):211–242.
- Jon Barwise and Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4(2):159–219.
- 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, May. European Language Resources Association.
- Johan Bos and Lasha Abzianidze. 2019. Thirty musts for meaning banking. In *Proceedings of the First International Workshop on Designing Meaning Representations*.
- Johan Bos. 2016. Expressive power of abstract meaning representations. *Computational Linguistics*, 42(3):527–535.
- Jeremy J. Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler. 2005. Named graphs. *Journal of Web Semantics*, 3(4):247–267.
- Lucas Champollion. 2015. The interaction of compositional semantics and event semantics. *Linguistics and Philosophy*, 38(1):31–66.
- Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2-3):281–332.
- 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.
- Philippe de Groote. 2001. Type raising, continuations, and classical logic. In *Proceedings of the thirteenth Amsterdam Colloquium*, pages 97–101.
- Philippe de Groote. 2006. Towards a montagovian account of dynamics. In *Semantics and Linguistic Theory*, volume 16, pages 1–16.
- 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, August. Association for Computational Linguistics.
- Peter Thomas Geach. 1962. *Reference and generality: An examination of some medieval and modern theories*. Cornell University Press.
- Michael Wayne Goodman. 2020. Penman: An open-source library and tool for AMR graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 312–319, Online, July. Association for Computational Linguistics.
- 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.
- Hans Kamp and Uwe Reyle. 2013. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*, volume 42. Springer Science & Business Media.
- Kevin Knight, Lauren Baranescu, Claire Bonial, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, and Nathan Schneifer. 2019. Abstract meaning representation (AMR) annotation release 1.2.6. *Web download*.
- Fred Landman. 1996. Plurality. In Shalom Lappin, editor, *The Handbook of Contemporary Semantic Theory*, pages 425–457. Oxford University Press, Oxford, UK.
- Christian Matthiessen and John Bateman. 1991. *Text Generation and Systemic-Functional Linguistics : Experiences from English and Japanese*. Pinter, London.
- Richard Montague. 1973. The proper treatment of quantification in ordinary english. In *Approaches to Natural Language*, pages 221–242. Springer.

- Tim O’Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018. AMR beyond the sentence: the multi-sentence AMR corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3693–3702, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.
- James Pustejovsky, Nianwen Xue, and Kenneth Lai. 2019. Modeling quantification and scope in abstract meaning representations. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 28–33.
- John C. Reynolds. 1993. The discoveries of continuations. *Lisp and Symbolic Computation*, 6(3-4):233–247.
- Guus Schreiber and Yves Raimond. 2014. RDF 1.1 primer.
- Edward Stabler. 2017. Reforming AMR. In *International Conference on Formal Grammar*, pages 72–87. Springer.