Squib

Expressive Power of Abstract Meaning Representations

Johan Bos*
University of Groningen

The syntax of abstract meaning representations (AMRs) can be defined recursively, and a systematic translation to first-order logic (FOL) can be specified, including a proper treatment of negation. AMRs without recurrent variables are in the decidable two-variable fragment of FOL. The current definition of AMRs has limited expressive power for universal quantification (up to one universal quantifier per sentence). A simple extension of the AMR syntax and translation to FOL provides the means to represent projection and scope phenomena.

1. Introduction

There is a new kid on the semantics block: AMR. Abstract meaning representation, based on the PENMAN notation and introduced in Langkilde and Knight (1998), has suddenly gained in popularity in computational linguistics (Artzi, Lee, and Zettlemoyer 2015; Chen 2015; Peng, Song, and Gildea 2015; Pust et al. 2015; Sawai, Shindo, and Matsumoto 2015; Wang, Xue, and Pradhan 2015; Werling, Angeli, and Manning 2015). There are several reasons for this trend. First of all, the simple tree structure of AMRs, showing the connections between concepts and events, make them easy to read. Second, because AMRs can simply be expressed as directed acyclic graphs, machine-generated output can be evaluated in a standard way by computing precision and recall on triples of gold-standard AMRs (Cai and Knight 2013). And, third, AMRs are arguably easier to produce manually than traditional formal meaning representations, and, as a result, there are now corpora with gold-standard AMRs available (Banarescu et al. 2015).

AMRs do not resemble classical meaning representations and seem substantially different from well-understood formalisms such as discourse representation structures (DRSs) (Kamp and Reyle 1993) or minimal recursion structures (MRSs) (Copestake et al. 2005). AMR has an unorthodox way of expressing negation, no explicit means to deal with universal quantification, and no model theory. This raises the question: What do they actually *mean*? This is an important question, because the ultimate goal of producing meaning representations is to be able to use them for inference tasks and

---

* University of Groningen, the Netherlands. E-mail: `johan.bos@rug.nl`.

build upon automated deduction methods (Blackburn and Bos 2005). Hence, the goal of this squib is to find answers to the following questions:

1. Are AMRs any different from traditional meaning representations?
2. Can we provide a model-theoretic semantics for AMRs?
3. Is there a systematic translation to first-order logic?
4. Is universal quantification expressible in AMRs?
5. Does the AMR language need to be extended?

To answer these questions, we will formulate a systematic translation function from AMR into classical logic. First, we will consider the simplest of AMRs. Then, we will add negation and investigate the role of recurrent variables. Finally, an extension to AMR to cover scope phenomena will be proposed. Throughout, some basic knowledge of first-order logic (FOL) and the λ-calculus is assumed.

2. Basic AMRs

In this section we will take a closer look at AMRs without polarity and recurrent variables. We will provide a definition of its syntax, discuss viewing AMRs as triples, its unique ability to display information structure by inversing roles, and finally provide a model-theoretic definition. Figure 1 shows example AMRs represented as trees (tense is ignored in the AMR analyses throughout this article).

Every AMR has a unique root, displayed as the top node in the tree. What we further see here are variables (e, x, y, etc.), events and concepts (`moan-01`, `child`, etc.), constants ("Ms Ribble"), and roles (`ARG0`, etc.). The slash indicates an instance: `x/child` means that x is an instance of the concept `child`. The colon is a punctuation symbol to display roles. The brackets indicate which role belongs to which concept, and, as line breaks are optional, AMRs can also be given in linear format, for example, (`e/read-01 :ARG0 (x/girl) :ARG1 (y/book)`) for *a girl read a book*.

A remarkable property of AMRs is the ability to invert roles. Role inversion swaps the arguments of a relation: R(x,y) ≡ R-of(y,x). Role inversion will lead (for polarity-free AMRs) to AMRs that are semantically equivalent, but are structured differently. For the given example we can apply role inversion in two ways, as Figure 2 shows.

The first AMR in Figure 2 puts the focus on *girl*, paraphrased as *a girl read a book*. In the second AMR we have placed *book* in the foreground, corresponding to *a book that was read by a girl*. Role inversion has its limitations: we cannot "pull out" both *book* and *girl*, because we would be left with an ill-formed AMR equipped with two roots.

Although AMRs are usually visualized as tree structures, they can also be viewed as directed acyclic graphs with a unique root (with vertices denoting variables and edges denoting roles and instances). As a consequence, AMRs can be transformed into sets of triples, which is convenient for evaluation purposes (Cai and Knight 2013). For semantic

```
(e / moan-01              (e / give-01
   :ARG0 (x / child))        :ARG0 (x / person :named "Ms Ribble")
                             :ARG2 (y / child)
                             :ARG1 (z / envelope))
```

Figure 1
AMRs for *The children moaned* (left) and *Ms Ribble handed out envelopes to the children* (right).

```
(x / girl                           (y / book
    :ARG0-of (e / read-01                :ARG1-of (e / read-01
                :ARG1 (y / book)))                     :ARG0 (x / girl)
```

Figure 2
AMR's feature to display information structure by role inversion.

interpretation, the tree structure is handier, because for operators like negation we need to be able to assign scope.

In order to give a semantic, model-theoretic interpretation, we could do this in a fairly straightforward way by converting concepts and events to one-place predicates, the roles to two-place predicates, and by existentially quantifying over all variables introduced by concepts and events. But this would not allow us to include scopal operators, such as negation, quantification, and projection in a systematic way.

What we will do instead is provide a formal definition of the syntax of AMRs and then define a recursive translation function from AMR to FOL. This function bears strong similarities with the conversion from AMR to λ-calculus presented in Artzi, Lee, and Zettlemoyer (2015). We will use the following notational conventions: A_i to denote AMRs, x for variables, c for constants, P for properties, and R for roles.

Definition 1 (Syntax of Basic AMRs)
$A ::= c \mid (x/P) \mid (x/P :R_1A_1 \ldots :R_nA_n)$

This definition says that constants, instance assignments, and instance assignments decorated with "out-going" roles are all AMRs. This is perhaps slightly counter-intuitive, because semantic objects of different types are placed in the same equation. The translation function that we will define next will clarify this issue, because it will translate all AMR-constructs into something of a propositional type. The best way to conceive this is by viewing an AMR c as the proposition "there exists an entity denoted by the constant c" and an AMR (x/P) as "there exists an x with property P," and so on.

To deal with correct assignment of scope (relevant when we introduce polarity in AMRs), we delay the translation of roles by converting them first into λ-expressions abstracting over role players. For instance, if the already-translated x is connected to the not-yet-translated AMR A via a role R, we get the expression $\lambda y.R(x,y)$. The resulting recursive translation function maps an AMR paired with a λ-expression for roles to an FOL formula (ϕ denotes a λ-expression for roles):

Definition 2 (Semantics of Basic AMRs)
$\|c,\phi\| = \phi(c)$
$\|(x/P),\phi\| = \exists x(P(x) \land \phi(x))$
$\|(x/P :R_1A_1 \ldots :R_nA_n),\phi\| = \exists x(P(x) \land \|A_1,\lambda y.R_1(x,y)\| \land \ldots \land \|A_n,\lambda y.R_n(x,y)\| \land \phi(x))$

When translating a concept that is related to other concepts, we do not know what semantic material these concepts will introduce, to which we can bind the roles that connect them. The translation function deals with this by postponing this decision with the help of λ-bound formulas representing roles. However, when we start translating a fresh AMR, we start with the root node. The root is not connected to other concepts by out-going roles, so we need to give it a "dummy" formula: $\lambda x.\top$. Here $\top$ stands for a

formula that is always true (so for any formula ϕ, $(\phi \wedge \top)$ is logically equivalent to ϕ). Figure 3 shows an example derivation.

The resulting structure is a closed formula (all its variables are bound), since the translation ensures that no free occurrences of variables can appear. Interestingly, basic AMRs are equivalent to the controlled DRT-fragment introduced by Bos (2010). As the latter are in the two-variable fragment of FOL, so are basic AMRs. Recall that first-order logic is undecidable. The two-variable fragment, however, is a decidable first-order language where formulas have maximally two differently named variables, no function symbols, but possibly do have equality. It has the finite model property, which means that if a formula of this fragment is satisfiable, it is satisfiable in a finite model (Mortimer 1975).

3. AMRs with Polarity

Negation is expressed in AMRs with a polarity relation. The relation is between the concept that is negated and the constant "–". Needless to say this is a special kind of relation: It cannot be inverted, and it needs a distinctive treatment during semantic interpretation, because it is propositions that are negated, not concepts. One could see the polarity relation as a flag, indicating that an AMR is negated.

In order to accommodate polarity in AMR, we first need to extend its syntax, and then add a clause in the translation function. This is done in the following two definitions (from here on we abbreviate :R_1A_1 … :R_nA_n as :R_iA_i for convenience):

Definition 3 (Syntax of AMRs with Polarity)
A ::= c | (x/P) | (x/P :R_iA_i) | (x/P :R_iA_i :polarity–)

Definition 4 (Semantics of AMRs with Polarity)
$\|c,\phi\| = \phi(c)$
$\|(x/P),\phi\| = \exists x(P(x) \wedge \phi(x))$
$\|(x/P :R_iA_i),\phi\| = \exists x(P(x) \wedge \|A_i,\lambda y.R_i(x,y)\| \wedge \phi(x))$
$\|(x/P :R_iA_i \ :polarity–),\phi\| = \neg\exists x(P(x) \wedge \|A_i,\lambda y.R_i(x,y)\| \wedge \phi(x))$

This translation function maps the polar AMR given in Figure 4 to the following formula: $\neg\exists e(\text{giggle-01}(e) \wedge \exists x(\text{boy}(x) \wedge \text{ARG0}(e,x)))$. As can be seen here, negation outscopes all other existential quantifiers because the polarity is assigned to the root concept of the AMR. If we assign the negative polarity to x instead, the AMR would be translated as $\exists e(\text{giggle-01}(e) \wedge \neg\exists x(\text{boy}(x) \wedge \text{ARG0}(e,x)))$. This formula could be true in a situation where there was someone giggling, but not a boy: *It was not a boy who was giggling*. As we observed in the previous section, promoting concepts by inversing

$$
\begin{array}{rcl}
\|(e \text{ / shout-01 :ARG0 } (x \text{ / teacher})), \lambda u.\top\| & = & [2.c] \\
\exists e(\text{shout-01}(e) \wedge \|(x/\text{teacher}), \lambda y.\text{ARG0}(e,y)\| \wedge \lambda u.\top(e)) & = & [2.b] \\
\exists e(\text{shout-01}(e) \wedge \exists x(\text{teacher}(x) \wedge \lambda y.\text{ARG0}(e,y)(x)) \wedge \lambda u.\top(e)) & = & [\beta\text{-conv}] \\
\exists e(\text{shout-01}(e) \wedge \exists x(\text{teacher}(x) \wedge \text{ARG0}(e,x)) \wedge \top) & = & [\top\text{-elim}] \\
\exists e(\text{shout-01}(e) \wedge \exists x(\text{teacher}(x) \wedge \text{ARG0}(e,x))) & &
\end{array}
$$

Figure 3
Derivation for *the teacher shouted*.

```
(e / giggle-01              (x / language
   : polarity -                 :domain-of (a / appropriate
   : ARG0 (x / boy))                       : polarity - ))
```

Figure 4
AMRs for *the boy didn't giggle*, and *the language was inappropriate*, expressing negation.

roles has no impact on the meaning of basic AMRs. However, pulling out concepts has potential meaning-changing consequences for AMRs with polarity.

4. AMRs with Recurrent Variables and Projection Phenomena

Recurrent variables in AMRs serve to cope with co-reference phenomena such as bound pronouns and control constructions. Two examples are shown in Figure 5.

The translation provided in the previous section yields unwanted results: In both cases free variables appear, whereas they ought to be bound by the existential quantifier introduced by Mr Krupp and George. This is caused by the scope of the quantifiers, which are restricted to the branch in the AMR tree in which they are introduced. Hence, if the recurrent variable appears on a sister branch, it cannot be bound by its quantifier. To deal with this problem, we propose a "projection mechanism" that gives wide scope to linguistic phenomena that require scope.

Projection phenomena, such as proper names, appositive expressions, definite descriptions, and possessive constructions, are known to project out of negation and take wide scope (Van der Sandt 1992; Kamp and Reyle 1993; Potts 2005). Projective material needs to be distinguished from meaning that is introduced in situ. We therefore extend the syntax of AMR with a new construction: $(x \backslash P)$, that signals that instance x of concept P belongs to the class of projection phenomena.

Definition 5 (AMR with Polarity and Projection)
$$A ::= c \mid x \mid (x/P) \mid (x/P :R_iA_i) \mid (x/P :R_iA_i \text{ :polarity}-)$$
$$\mid (x \backslash P) \mid (x \backslash P :R_iA_i) \mid (x \backslash P :R_iA_i \text{ :polarity}-)$$

We note in passing that we have now officially left the two-variable fragment, because it is possible to construct an AMR that requires three differently named variables. Although we are still working in a fragment of first-order logic, it is unclear whether this fragment has any interesting complexity properties.

We split the semantic interpretation procedure in two parts: a translation function for assertive semantics $\|.,.\|^{\downarrow}$ and one for projective semantics $\|.\|^{\uparrow}$. The projective semantics builds a function that is then applied to the result of the assertive semantics:

Definition 6 (Semantics of AMRs with Polarity and Projection)
$$\|A\| = \|A\|^{\uparrow}(\|A, \lambda x. \top\|^{\downarrow})$$

```
(e / dry-01                 (w / want-01
   :ARG0 (x / person            :ARG0 (g / person :named "George")
          :named "Mr Krupp")    :ARG1 (p / play-01
   :ARG1 x)                             : ARG0 g))
```

Figure 5
AMRs with recurrent variables, *Mr Krupp dried himself* (left) and *George wanted to play* (right).

The assertive semantics is essentially the same as given in Definition 3, but now it includes the new constructs for projection phenomena. As can be seen from Definition 7, all concepts that need to be projected are ignored in the translation function for the assertive semantics of an AMR:

Definition 7 (Assertive Semantics of AMRs)

$\|c,\phi\|^{\downarrow} = \phi(c)$
$\|x,\phi\|^{\downarrow} = \phi(x)$
$\|(x/P),\phi\|^{\downarrow} = \exists x(P(x)\wedge\phi(x))$
$\|(x/P :R_i A_i),\phi\|^{\downarrow} = \exists x(P(x) \wedge \|A_i,\lambda y.R_i(x,y)\|^{\downarrow} \wedge\phi(x))$
$\|(x/P :R_1 A_1 :\text{polarity}-),\phi\|^{\downarrow} = \neg\exists x(P(x)\wedge \|A_i,\lambda y.R_i(x,y)\|^{\downarrow}\wedge\phi(x))$
$\|(x\backslash P),\phi\|^{\downarrow} = \phi(x)$
$\|(x\backslash P :R_i A_i),\phi\|^{\downarrow} = \phi(x)$
$\|(x\backslash P :R_i A_i :\text{polarity}-),\phi\|^{\downarrow} = \phi(x)$

The definition of the projective semantics is presented as a function from AMRs to λ-expressions of type $\langle t, t\rangle$. It ignores all assertive material in an AMR, but when it is given a projective concept, it calls upon the assertive translation function.

Definition 8 (Projective Semantics of AMRs)

$\|c\|^{\uparrow} = \lambda p.p$
$\|x\|^{\uparrow} = \lambda p.p$
$\|(x/P)\|^{\uparrow} = \lambda p.p$
$\|(x/P :R_i A_i)\|^{\uparrow} = \lambda p.\|A_1\|^{\uparrow}(\|A_2\|^{\uparrow}(\dots\|A_n\|^{\uparrow}(p)))$
$\|(x/P :R_i A_i :\text{polarity}-)\|^{\uparrow} = \lambda p.\|A_1\|^{\uparrow}(\|A_2\|^{\uparrow}(\dots\|A_n\|^{\uparrow}(p)))$
$\|(x\backslash P)\|^{\uparrow} = \lambda p.\|(x/P),\lambda x.p\|^{\downarrow}$
$\|(x\backslash P :R_i A_i)\|^{\uparrow} = \lambda p.\|(x/P :R_i A_i),\lambda x.p\|^{\downarrow}$
$\|(x\backslash P :R_i A_i :\text{polarity}-)\|^{\uparrow} = \lambda p.\|(x/P :R_i A_i :\text{polarity}-),\lambda x.p\|^{\downarrow}$

We will now illustrate the translation by giving it the example AMR for *Mr Krupp dried himself*. We first need to compute the assertive semantics with the help of $\|.,.\|^{\downarrow}$, as shown in Figure 6. We then compute the projective material with the aid of $\|.\|^{\uparrow}$, as Figure 7 illustrates. Finally, we put together the projective and assertive semantics, a process that binds the free variables introduced by the assertive semantics by quantifiers generated by the projective semantics (Figure 8).

Because this technique works well for projection phenomena, a natural question to ask is how universal quantifiers would fit in this picture. In the next section, therefore, we will investigate the possibilities of universal quantification in AMRs.

$\|A,\lambda u.\top\|^{\downarrow}=\|(e/\text{dry-01} :\text{ARG0} (x\backslash\text{person} :\text{named "Mr Krupp"}) :\text{ARG1} x), \lambda u.\top\|^{\downarrow} = [7.4]$
$\exists e(\text{dry-01}(e)\wedge\|(x\backslash\text{person} :\text{named "Mr Krupp"}),\lambda y.\text{ARG0}(e,y)\|^{\downarrow}\wedge\|x,\lambda y.\text{ARG1}(e,y)\|^{\downarrow}\lambda u.\top) = [7.6]$
$\exists e(\text{dry-01}(e)\wedge\lambda y.\text{ARG0}(e,y)(x)\wedge\|x.\lambda y.\text{ARG1}(e,y)\|^{\downarrow}\wedge\lambda.u.\top) = [7.2]$
$\exists e(\text{dry-01}(e)\wedge\lambda y.\text{ARG0}(e,y)(x)\wedge\lambda y.\text{ARG1}(e,y)(x)\wedge\lambda u.\top) = [\beta\text{-conv}]$
$\exists e(\text{dry-01}(e)\wedge\text{ARG0}(e,x)\wedge\text{ARG1}(e,x)\wedge\top) = [\top\text{-elim}]$
$\exists e(\text{dry-01}(e)\wedge\text{ARG0}(e,x)\wedge\text{ARG1}(e,x))$

Figure 6
Derivation of the assertive semantics for *Mr Krupp dried himself*.

$\|A\|^{\uparrow}=\|(e/dry\text{-}01\ :\!ARG0\ (x\backslash person\ :\!named\ ''Mr\ Krupp'')\ :\!ARG1\ x)\|^{\uparrow}$ = [8.4]

$\lambda p.\|(x\backslash person\ :\!named\ ''Mr\ Krupp'')\|^{\uparrow}(\|x\|^{\uparrow}(p))$ = [8.6]

$\lambda p.\lambda p'.\|(x/person\ :\!named\ ''Mr\ Krupp''),\lambda x.p'\|^{\downarrow}(\|x\|^{\uparrow}(p))$ = [8.2]

$\lambda p.\lambda p'.\|(x/person\ :\!named\ ''Mr\ Krupp''),\lambda x.p'\|^{\downarrow}(\lambda p''.p''(p))$ = [7.4]

$\lambda p.\lambda p'.\exists x(person(x)\wedge\|''Mr\ Krupp'',\lambda y.named(x,y)\|^{\downarrow}\wedge\lambda x.p'(x))(\lambda p''.p''(p))$ = [β-conv]

$\lambda p.\lambda p'.\exists x(person(x)\wedge\|''Mr\ Krupp'',\lambda y.named(x,y)\|^{\downarrow}\wedge\lambda x.p'(x))(p)$ = [β-conv]

$\lambda p.\lambda p'.\exists x(person(x)\wedge\lambda y.named(x,y)(''Mr\ Krupp'')\wedge p')(p)$ = [β-conv]

$\lambda p.\exists x(person(x)\wedge named(x,''Mr\ Krupp'')\wedge p)$

Figure 7
Derivation of the projective semantics for *Mr Krupp dried himself*.

5. Universal Quantification in AMRs

According to Banarescu et al. (2015), AMRs cannot express universal quantification. This is not exactly true. The technique of pulling out concepts in combination with multiple use of the polarity flag enables one to correctly represent universally quantified sentences such as *every boy whistled*. This can be seen on the left in Figure 9. The basic AMR shown here translates to $\neg\exists x(boy(x)\wedge\neg\exists e(whistle\text{-}01(e)\wedge ARG0(e,x)))$, and this is logically equivalent to $\forall x(boy(x)\rightarrow\exists e(whistle\text{-}01(e)\wedge ARG0(e,x)))$, resembling the classic FOL formula for *every boy whistled*.

It is important to point out the limitations given to us by the one-rooted nature of AMRs: Not more than one universal quantifier can be handled in this way. Therefore, it is *impossible* to provide a proper AMR representing a sentence with a transitive verb and two universally quantified noun phrases, as in *every dog chased every cat* (unless logical conjunction is applied in the root of the AMR). A general solution to this limitation is to extend the AMR language to cope with universal quantification. A proposal can be seen on the right-hand side in Figure 9. The quantification relation flags the variable to which it connects so that it needs a universal quantifier. Definition 7 can then be extended with clauses like $\|(x/P\ :\!quant\ \forall),\phi\|^{\downarrow} = \forall x(P(x)\rightarrow\phi(x))$. In this way, quantifier scope ambiguities are underspecified and different readings can be obtained by changing the order of applications in Clauses 4 and 5 of Definition 8.

6. Conclusion

Are AMRs any different from traditional meaning representations? Yes. What AMRs have in common with DRSs and MRSs is that logical conjunction is implicit (apart from those cases where it is triggered by coordination phenomena). The expressive power of AMRs is also lower (Bender et al. 2015). Scope is not explicitly present in AMRs, and therefore the way negation is represented in AMRs is substantially different from the ways this is done in DRSs or MRSs. Unlike DRSs or MRSs, AMRs have the capacity to display some aspects of information structure (the process of role inversion can change the scope of concepts), although, as soon as negation is present, this can lead to changes in meaning.

$\|A\| = \|A\|^{\uparrow}(\|A,\lambda u.\top\|^{\downarrow})$ = [6]

$\lambda p.\exists x(person(x)\wedge named(x,''Mr\ Krupp'')\wedge p)(\exists e(dry(e)\wedge ARG0(e,x)\wedge ARG1(e,x)))$ = [β-conv]

$\exists x(person(x)\wedge named(x,''Mr\ Krupp'')\wedge\exists e(dry(e)\wedge ARG0(e,x)\wedge ARG1(e,x)))$

Figure 8
Combining projective and assertive semantics for *Mr Krupp dried himself*.

```
(x / boy :polarity -                   (e / whistle-01
   :ARG0-of (e /whistle-01               :ARG0 (x \ boy
             :polarity - ))                     :quant A)
```

Figure 9
Basic AMR with universal quantification (left) and extended AMR (right) for *every boy whistled*.

Can we give a model-theoretic semantics for AMRs, and is there a systematic translation to first-order logic? Yes and yes. We have shown that basic AMRs can be systematically translated into first-order logic, and this gives the AMR language an indirect model-theoretic interpretation. As a matter of fact, AMRs without recurrent variables can be translated into a decidable fragment of first-order logic. This includes the polarity flag used in AMRs to express negation. The translation that we provided is purely declarative and can be easily implemented, for example as a Prolog program.

Is universal quantification expressible in AMRs? Yes. Universal quantification *can* be represented in AMRs, by multiple use of the polarity relation. However, at most one (wide-scope) universal quantifier can be used in standard AMRs.

Does the AMR language need to be extended? Yes. In order to deal with projection phenomena and quantifier scope the AMR language needs to be extended. We have sketched an initial proposal of how this extension might look. There is still work to be done though. Hence, an appropriate way to finish this article is to use this quote from the AMR manual (Banarescu et al. 2015): "AMR 1.2 looks forward to AMR 2.0!"

References
Artzi, Yoav, 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.

Banarescu, Laura, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2015. Abstract Meaning Representation (AMR) 1.2.2 Specification. github.com/ amrisi/amr-guidelines/blob/master/ amr.md.

Bender, Emily M., Dan Flickinger, Stephan Oepen, Woodley Packard, and Ann Copestake. 2015. Layers of interpretation: On grammar and compositionality. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 239–249, London.

Blackburn, Patrick and Johan Bos. 2005. *Representation and Inference for Natural Language. A First Course in Computational Semantics*. CSLI, Stanford, CA.

Bos, Johan. 2010. Economical discourse representation theory. In *CNL 2009 Workshop*, pages 121–134, Marettimo Island.

Cai, Shu and Kevin Knight. 2013. Smatch: An evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia.

Chen, Wei-Te. 2015. Learning to map dependency parses to abstract meaning representations. In *Proceedings of the ACL-IJCNLP 2015 Student Research Workshop*, pages 41–46, Beijing.

Copestake, Ann, Dan Flickinger, Ivan Sag, and Carl Pollard. 2005. Minimal recursion semantics: An introduction. *Journal of Research on Language and Computation*, 3(2–3):281–332.

Kamp, Hans and Uwe Reyle. 1993. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht.

Langkilde, Irene and Kevin Knight. 1998. Generation that exploits corpus-based

statistical knowledge. In *Proceedings of the 36th Annual Meeting of the ACL and 17th International Conference on Computational Linguistics*, pages 704–710, Montreal.

Mortimer, Michael. 1975. On languages with two variables. *Mathematical Logic Quarterly*, 21(1):135–140.

Peng, Xiaochang, Linfeng Song, and Daniel Gildea. 2015. A synchronous hyperedge replacement grammar based approach for AMR parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 32–41, Beijing.

Potts, Christopher. 2005. *The Logic of Conventional Implicatures*. Oxford Studies in Theoretical Linguistics. Oxford University Press, Oxford.

Pust, Michael, Ulf Hermjakob, Kevin Knight, Daniel Marcu, and Jonathan May. 2015. Parsing English into abstract meaning representation using syntax-based machine translation. In *Proceedings of the EMNLP 2015*, pages 1143–1154, Lisbon.

Sawai, Yuichiro, Hiroyuki Shindo, and Yuji Matsumoto. 2015. Semantic structure analysis of noun phrases using abstract meaning representation. In *Proceedings of the 53rd Annual Meeting of the ACL (Volume 2: Short Papers)*, pages 851–856, Beijing.

Van der Sandt, Rob A. 1992. Presupposition projection as anaphora resolution. *Journal of Semantics*, 9:333–377.

Wang, Chuan, Nianwen Xue, and Sameer Pradhan. 2015. Boosting transition-based AMR parsing with refined actions and auxiliary analyzers. In *Proceedings of the 53rd Annual Meeting of the ACL (Volume 2: Short Papers)*, pages 857–862, Beijing.

Werling, Keenon, Gabor Angeli, and Christopher D. Manning. 2015. Robust subgraph generation improves abstract meaning representation parsing. In *Proceedings of the 53rd Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 982–991, Beijing.