Reforming AMR

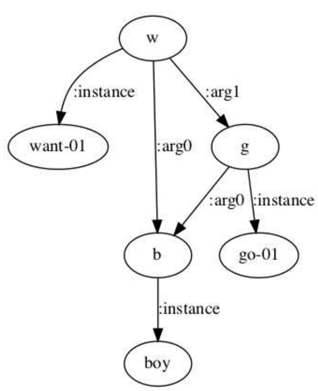
Edward Stabler

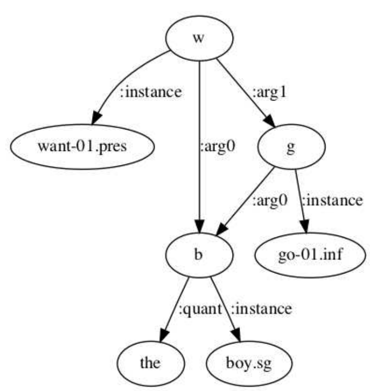
University of California, Los Angeles, California, and
Nuance Communications, Sunnyvale, California
edward.stabler@nuance.com

Abstract. Many recent proposals aim to simplify semantic representations, and Abstract Meaning Representation (AMR) comes from this tradition, but it is nevertheless quite expressive. Bos 2016 proposes a slightly reformed AMR for translation to first order logic. This paper proposes a different augmentation of AMR that is more easily provided, and a slightly different mapping to higher order and dynamic logic. The proposed augmentation can be, at least in most cases, easily computed from standard 'unreformed' AMR corpora. The mapping from this augmented AMR to logical representation is a finite state multi bottom up tree transduction.

With a variety of scientific and engineering motivations, a number of recent studies have aimed to simplify semantic representations in various ways that include reducing recursion, reducing the number of modes of composition, and leaving some ambiguities unresolved [2,14,36,40]. Abstract Meaning Representation (AMR) [4,5,27] falls into this tradition; it was initially designed to be easily learnable by automated translation systems [31]. An ongoing effort uses AMR to annotate large corpora [27] for engineering applications and other quantitative studies of the important question: which constructions are used to mean which things in which contexts?

In [4], the AMR on the left below is proposed to represent the meaning of *The boy wants to go*. The AMR on the right is slightly augmented with syntactic features in the leaves and a :quant arc for the determiner *the*, as discussed below.





In the graph on the left, above, the 3 leaves (i.e., the nodes with no outgoing arcs) are disambiguated lexical concepts, while the 3 internal nodes are variables, written with a single letter optionally followed by a number. The 6 arcs are labeled with roles written with an initial colon. Intuitively, the graph on the left could be read: "there is an :instance of wanting, w, with an :arg0 (subject) role filled by an instance b of a boy, and with an :arg1 (theme) role filled by an :instance g of going, which in turn has an :arg0 role again filled by b." In this version of AMR, the arc labels and the relevant senses of the verbs *want* and *go* are taken from OntoNotes [41] and PropBank [8].

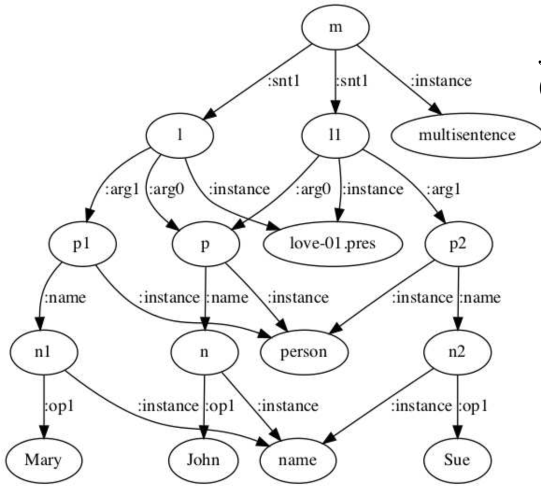
So AMR indicates verb senses, coreference, and some aspects of argument and modifier structure. In some discourse contexts, the prepositional phrase modifier in *The money was stolen by the bank* is an agent and not a locative. And in some contexts, a spoken or casually written *let's meet at last call* could intend the prepositional phrase to specify not a time but a location, one of the many bars named 'Last Call'. In a large corpus, some of the relevant cues for such distinctions can be studied, cues to which normal human speakers are obviously exquisitely attuned. But standard AMR does not indicate tense, plurality, quantification, or scope. For some engineering applications, tense, plurality and quantification may not matter, but for other applications it is obviously important to get the whole message of an utterance approximately right. Following preliminary work by Bos and others [9,3,6], this paper argues that with a very minor reform that does not significantly affect the syntactic complexity of AMR notation, it can become considerably more sophisticated. The reform is indicated in the augmented AMR (AAMR) shown above on the right: we add tense to leaves corresponding to verbs, grammatical number to leaves corresponding to nouns, and for each noun concept, if that noun has an associated article (*the, a*) or quantifier (*every*, *some*, *most*, *exactly 5*, *between 10 and 20*, ...), those go into the :quant role.

1 AMR triples and trees

Each AMR is a connected, directed, arc-labeled graph with a designated 'root' node, where (a) at most one node has no incoming arcs, and when there is such a node, it is the designated 'root', (b) any node with an outgoing arc is a variable, (c) the arc-labels are roles (chosen from a small, finite set), (d) every internal node has a unique outgoing arc with the :instance role, and (e) every leaf is a lexical concept, where a lexical concept is a sense-disambiguated stem possibly with tense, number, gender information. We can represent any AMR by a pair (rootNode, arcs) where arcs is a set of (sourceNode,role,targetNode) triples. So for example the AAMR displayed on the right above is:

(w, { (w,:instance,want-01.pres), (w,:arg0,b), (w,:arg1,g),
(b,:instance,boy.sg), (b,:quant,the),
(g,:instance,go-01.sg), (g,:arg0,b) })

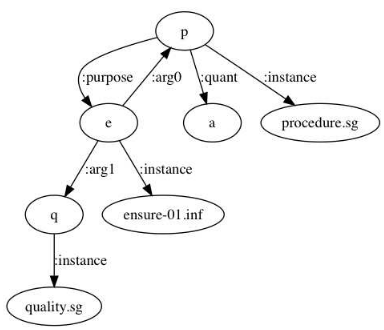
Note that in this AAMR, one and the same :instance of a boy b plays a role in two predications, because *want* is a 'control verb' [29]. English obviously has many ways to indicate intended coreference. When we construct AAMRs for conjunctions of sentences and discourses, it is common to have many long chains of coreferring argument expressions. The following is a very simple case, showing again a notation on the right that is discussed below.




```
John loves Mary. John loves Sue.
(m / multisentence
    :snt1 (l / love-01.pres
        :arg0 (p / person
            :name (n / name
                :op1 John))
        :arg1 (p1 / person
            :name (n1 / name
                :op1 Mary)))
    :snt2 (l1 / love-01.pres
        :arg0 p
        :arg1 (p2 / person
            :name (n2 / name
                :op1 Sue))))
```

In this structure we see that the named entities are classified by lexical concepts (like *person*) that do not correspond to explicit items in the sentence, as explained in [5]. This classification of named entities will not be a focus in this paper.

[4, §2] says AMRs are rooted and acyclic, but [5] provides a non-rooted, cyclic structure for *a procedure to ensure quality*, which we also allow, augmented:

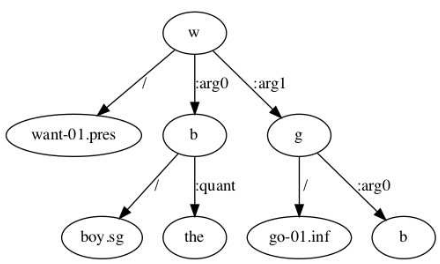



```
(p / procedure.sg
    :purpose (e / ensure-01.inf
        :arg0 p
        :arg1 (q / quality.sg))
    :quant a)
```

[4, §2] uses a node labeling function, which we do not need for AMRs, but only for the AMR tree representations introduced in this section, as discussed above. And in addition to lexical concepts, [5] uses some special constants, like '-' on a :polarity arc to indicate negation. But '-' can at least sometimes be regarded as the representation the lexical 'not'; see §§2-3 below.

Letting the K-width of a directed graph be the maximum number of distinct simple paths (possibly overlapping) between any two nodes [20], it is easy to see that there is no finite bound on the K-width of AMR structures of English. For example, even just with arbitrarily long conjunctions of the form *John loves Mary and John loves Sue and...*, in which the love relation can be asserted between any pair of named individuals in any finite set, we can make K-width arbitrarily high. Notice that even as K-width grows quite high, discourses with this kind of structure (perhaps adding some modifiers) can be not only intelligible but exciting in some contexts. Ignoring the direction of the arcs, AMR structures for English sentences also have unbounded treewidth [15,1]; any number of nodes in these love relations can be connected to each other and involved in any number of cycles.[1] AMRs are not simple graphs. But for some calculations, we can split the AMR nodes with more than one predecessor to make a tree, putting all descendants of the split node under only one of its copies, and calculate on the tree without ever needing to check the identities among those split nodes. All the calculations described in this paper have that property.

Turning to the tree representations for AMRs already shown above, these can be derived from AMRs by the introduction of a node labeling function which is the identity function for every node with $p \leq 1$ predecessors, but for any node n with $p > 1$ predecessors in the graph, the tree has p different nodes $\mathrm{n}_0, \ldots, \mathrm{n}_{p-1}$ all labeled n, with all descendants of the original node attached under only one of the copies n_i in the tree. As we saw above, because AMRs can be cyclic, it can happen that no node lacks a predecessor, but when that happens the 'designated root' is chosen as the root of the corresponding tree. Applying these ideas to the AAMR on the first page of this paper, and abbreviating :instance with / (as is standard in the literature), we obtain a tree that can be drawn or pretty-printed:




```
(w / want-01.pres
    :arg0 (b / boy.sg
        :quant the)
    :arg1 (g / go-01.inf
        :arg0 b))
```

Additional examples of the pretty-printed tree format have already been provided for the love and purpose-clause examples above.

In order to have a unique tree representation for each AMR, let's initially adopt the convention that (i) subtrees are ordered left-to-right with the :instance subtree (often labeled /) first and otherwise in standard alphanumeric order, and

[1] A tree decomposition of undirected graph g=(V,R) is a tree t=(U,S) where (i) $\bigcup_{\mathrm{u}\in\mathrm{U}} \mathrm{u} = \mathrm{V}$, (ii) if an arc in g connects v_i and v_j, then some $\mathrm{u} \in \mathrm{U}$ contains both v_i and v_j, and (iii) if some node v of g is in two nodes $\mathrm{u}_i, \mathrm{u}_j$ of U, then v is in every node on the path between u_i and u_j. The treewidth of a decomposition is $\max_{\mathrm{u}\in\mathrm{U}}|\mathrm{u}| - 1$. The treewidth of g is the minimum treewidth over all decompositions of g. Many problems have complexities that increase with treewidth [20,7], and Courcelle's theorem relates treewidth to MSO definability [15]. Computing treewidth is NP-complete, but code for computing treewidth of small graphs is available at [1].

(ii) for any set of nodes with the same label, all must be leaves except possibly the one that occurs first in the preorder traversal. (i) is revised in §3 below.

2 Raising negation and first order quantifiers

The intuitive reading of AMRs suggested in the first paragraph of this paper binds the variables of an AMR with existential operators. But that will not work when negation is attached under the verb. as in some dependency grammars. [9] proposes a translation from AMRs with negation to first order logic (FOL), raising the negative polarity to take wide scope. Assuming *The boy didn't giggle* gets the AMR on the left, it is mapped to the FOL translation on the right:

```
(g / giggle-01
   :arg0 (b / boy)
   :polarity - )
```

$$\neg\exists g(\text{giggle-01.past}(g) \wedge \exists b(\text{boy.sg}(b) \wedge \text{:arg0}(g, b)))$$

But that FOL means something like *No boy giggled*, while the sentence *The boy didn't giggle* is clearly talking about some particular boy. For this, Bos proposes a reform of AMR which marks constituents with a backwards :instance slash whenever they should be raised to take scope over negation. In the translation to FOL, a concept :instance introduced with the backward slash takes scope over the negation:

```
(g / giggle-01
   :arg0 (b \ boy)
   :polarity - )
```

$$\exists b(\text{boy}(b) \wedge \neg\exists g(\text{giggle-01}(g) \wedge \text{:arg0}(g, b)))$$

Note that the FOL representation here actually means roughly: *some boy doesn't giggle*, still not quite what we want.

Extending this backslash idea to universal quantification, Bos proposes that AMRs be augmented to indicate *every* as shown in this example, with a backslash to give it wide scope:

```
(g / giggle-01
   :arg0 (b \ boy
             :quant A ))
```

$$\forall b(\text{boy}(b) \rightarrow \exists g(\text{giggle-01.past}(g) \wedge \text{:arg0}(g, b)))$$

In summary, Bos's proposal is roughly this: (1) The event structure associated with each verb is existentially closed; (2) negation scopes over the existential closure of the event that the :polarity role is attached to; (3) arguments marked by reversing their :instance slash are raised to take scope over the whole structure; and (4) alternative permutations of the quantifiers with reversed slashes are generated nondeterministically.

These steps head in the right direction, but observe these four points:

- If articles, grammatical number and tense are not indicated, then the AMR assigned to Bos's example *The boy didn't giggle* is also assigned to *boys don't giggle* and *the boys will not giggle* and *a boy won't giggle*. These sentences

mean different things in respects that could be important for some applications of AMR annotation, and it is easy to add a little more to the annotation to distinguish them, as proposed in the introduction to this paper.
- Bos's backslash proposal (3,4) provides a way to allow arguments to scope over the existential event quantifier, but, as noted for example by Champollion [12] and Landman [30], it is not the exception but the rule that this should happen. For various theoretical and empirical reasons, it is more natural to take a sentence like *John kissed every girl* to assert possibly many different kissing events, one for each girl, rather than a single event that includes possibly many different girls at many different times and places. And Bos points out that among the phrases that regularly outscope the event quantifier and negation are not only quantified phrases like *every boy* but also "proper names, appositive expressions, definite descriptions, and possessive constructions." This could be built into the translation, rather than requiring every instance to be marked. This idea is developed in §3.
- When AMRs are used to represent conjunctions of sentences and discourses, proposals (3,4) will raise quantifiers over the whole conjunction or discourse, and permuting them will generate spurious scope interactions. Scope inversions are usually clause- and sentence-bound, and even with those restrictions, the number of alternative scopings can grow quickly. Scope locality restrictions are imposed in §3, and we don't generate alternatives.
- As noted above, since the definite article is ignored in current AMR, the FOL translation for *the boy didn't giggle* actually means something like *some boy didn't giggle*. Articles have not only quantificational force but also a discourse role that affects utterance meaning. AAMRs, augmented with quantifiers and articles as suggested in the introduction, allow a translation to logic that can respect the force of these elements. This idea is briefly developed in §5.

3 Towards a proper treatment of quantifiers

The survey in [38, §§13-16] shows that among the many things not expressible by any iterations of first order quantifiers are the following, using As, Bs for determiner phrase denotations, R for properties/relations, and R-er for comparatives: *most As R, more than one third of As R, 80% of As R, more As than Bs R, an even number of As R, different As R different Bs, As are usually R-er than Bs, an infinite number of As R.* A longer, wider-ranging list is provided in [26]. And the list gets much longer if we include quantifiers whose representation in FOL with identity is possible but not feasible: it is an old point that FOL with identity is not a good language to express propositions like *More than seven billion people need clean water*. FOL is not appropriate for linguistic semantics. And even for a wide range of common engineering applications, operators for cardinalities and cardinal comparisons are common enough that logics like OWL, carefully designed for engineering goals, provide them [24,7].

So we begin by proposing a translation from AAMR to higher order logic (HOL) in which quantifiers like *every* and *some* are treated like *most, seven* *billion, more than 1/3 of,....* Let's assume that these quantifiers apply to pairs of properties to yield truth values. (We revise this assumption about the higher order type of binary quantifiers in section §5.) For example, taking a simple variant of Bos's example, we will map the AAMR for *Most boys giggled* on the left to this HOL on the right:[2]

```
(g / giggle-01.pres
   :arg0 (b / boy.pl                most(boy.pl,λb∃g(giggle-01.pres(g)∧:arg0(g,b)))
            :quant: most ))
```

That is, roughly, most boys have the property that there is an event in which they giggle. Note that we scope *most* over the event existential by default, rather than needing a backslash to trigger that effect.

Adding negation to get *Most boys do not giggle*, there are actually two readings: either the 'surface order' (i) most boys are such that it's not the case that they giggle, or the 'inverted order' (ii) it's not the case that most boys are such that they giggle. This is a case where the two readings are not so easy to disassociate. But if we assume that *most* means more that half, precisely, then in a situation where exactly half the boys giggle, then (i) is false, but (ii) is true. As is often the case in structures with interacting scopes, I think the surface order reading is the most natural here:

```
(g / giggle-01.pres
   :arg0 (b / boy.pl
            :quant most )         most(boy.pl,λb¬∃g(giggle-01.pres(g)∧:arg0(g,b)))
   :polarity - )
```

Rather than generating the non-surface order scopal orders for AAMRs, we will generate just the basic surface order reading. If needed, alternative orders can be computed from the basic order either with an algorithm that raises the quantifiers to get all possible alternative orders, or by using some kind of heuristic method to generate only scopes that are most likely in some sense, as discussed in §4.

In sum, while Bos computes all the raised quantifiers for the whole structure together, and then applies them to the representation of the unraised structure, we will only raise quantifier arguments to scope over the AAMR that they are arguments of, to generate surface-order scope only. We can see already that this calculation of HOL from the tree representation of AAMR is quite simple; it can be done by a simple finite state mechanism, a deterministic multi bottom up tree transducer (mbutt) [19,16]. To set the stage for that, first we revise our earlier convention about the linear order in tree representations of AAMRs: instead of putting subtrees into alphanumeric order, we will order them according to the linear order of their corresponding elements in the input string.[3] Second, we will

[2] HOL with generalized quantifiers is introduced, for example, in Carpenter's [11, §3]. Like Carpenter, we write $\forall x\phi$ for every$(\lambda x.\phi)$, and $\exists x\phi$ similarly.

[3] The correspondence between AAMR subgraphs and elements of the input string is sometimes given by hand-specified alignments, and there are a number of proposals

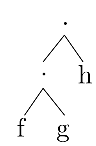
modify the tree representation so that the roles are not treated as arc labels but as node labels. This allows AMRs to have a standard term representation. So our previous example actually is given to the transducer as the tree on the left:

```
g(:instance(giggle-01.pres),
   :arg0(b(:instance(boy.pl),
             :quant(most))),          most(boy.pl,λb¬∃g(giggle-01.pres(g)∧:arg0(g,b)))
   :polarity(-))
```

Intuitively, a bottom up tree transducer differs from a bottom up tree acceptor in that the states may have a subtree, their output. An mbutt differs from a simple tree transducer in allowing states to have more than one subtree, so that up to k subtrees can be carried up the tree. What needs to happen in our transformation is that the subtrees corresponding to negation and to each argument of each lexical concept need to be lifted to scope over each verbal concept. We will define the transducer so that the order of the quantifier subtrees is preserved as they raise, so that the transducer can place those elements in the order that corresponds to the string surface order.

Slightly more formally, a bottom up tree acceptor $\mathrm{A} = (Q, F, \Sigma, \delta)$ where Q is a finite set of states, $F \subseteq Q$ is the set of final states, Σ is a ranked alphabet, and δ is a finite set of transition rules of the form $f(q_1, \ldots, q_n) \to q$, mapping a tuple of states to a state. Leaves are 0-ary, and hence treated by rules of the form $f \to q$, mapping a leaf to a state. The acceptor is deterministic iff no two rules have the same left side. To get a transducer we let the states be 1-ary, with rules of the form $f(q_1(t_1), \ldots, q_n(t_n)) \to q(t)$, where t may contain $t_1, \ldots, t_n$ as subtrees, together with any fixed structure. If a rule has at most one occurrence of each t_i in t, it is linear, and the transducer is linear iff all its transitions are. And we extend this to mbutts by letting the states have arity up to k for some k. See [18,19,16].

Formalizing and implementing the transduction we need for any corpus is straightforward, but slightly tedious since many elements are rearranged in the HOL. Stepping through the simple example above, *most boys do not giggle*, will make clear what the task requires. Since HOL allows higher order functions, the expressions denoting functions can be complex, and so expressions like (f(g))(h) are well formed – with some operator precedence conventions this is written more simply as fgh. To represent such expressions as trees, we will make the applications explicit, using a period as the application operator, representing (f(g))(h) with the tree .(.(f,g),h):



about how to compute them when hand-specifications are not available [17,13]. Note that ":polarity -" will be aligned with the negation in the input string. In the example above with 'most' and 'not', the surface order and the alphanumeric order coincide.

Since the translation from (f(g))(h) to .(.(f,g),h) is easy and the former is easier to read, in the following discussion we use the former notation, sometimes omitting parentheses when no confusion results. And sometimes, $\lambda x.\mathrm{p}(x)$ is reduced to p.

We assume that the set of basic vocabulary of concepts w is fixed and given. For every concept w, we have a rule mapping that concept to a 1-ary state q_c that has the concept as an argument:

$$\mathrm{w} \rightarrow q_c(\mathrm{w})$$

That is, for the leaves of the AAMR just above for *most boys do not giggle*, we have the rules:

giggle-01.pres $\rightarrow q_c$(giggle-01.pres)
boy.pl $\rightarrow q_c$(boy.pl)
most $\rightarrow q_c$(most).
- $\rightarrow q_c$(-)

Now moving up the tree, we have the :instance role dominating the state q_c(giggle-01.pres). For these :instance roles, we have the rule

$$:\mathrm{instance}(q_c(t)) \rightarrow q_i(t).$$

So at that node, we now have the state q_i(giggle-01.pres). Climbing up from - , we have the :polarity role, a special case handled by the rule

$$:\mathrm{polarity}(q_c(\text{-})) \rightarrow q_{-}.$$

Climbing up from boy.pl use the instance rule just above to get q_i(boy.pl). Climbing from most, we have the :quant role and use the rule:

$$:\mathrm{quant}(q_c(t)) \rightarrow q_q(t).$$

Now we get to the more interesting steps. The AAMR dominated by b is processed with this rule:

$$\mathrm{b}(q_i(t_0), q_q(t_1)) \rightarrow q_{qa}(t_1, \mathrm{b}, t_0).$$

So this last step yields q_{qa}(most,b,boy.pl). Climbing from that last step, the :arg0 role is assigned:

$$:\mathrm{arg0}(q_{qa}(t_0, t_1, t_2)) \rightarrow q_{rcq}(:\mathrm{arg0}, t_0, t_1, t_2).$$

At this point we have q_{rcq}(:arg0,most,b,boy.pl). Now all the needed parts are available for the last step:

$$\mathrm{g}(q_i(t_0), q_{-}, q_{rcq}(t_1, t_2, t_3, t_4)) \rightarrow q_a(t_2(t_4, \lambda t_3 \neg\exists \mathrm{g}(t_0(\mathrm{g}) \wedge t_1(\mathrm{g}, t_3)))).$$

This rule yields the desired q_a(most(boy.pl, λb¬∃g(giggle-01.pres(g)∧:arg0(g, b)))).

Clearly, this kind of deterministic bottom up assembly of HOL is going to be possible as long as the number of subtrees needed at each step is finite and the number of different things we need to do with those subtrees – controlled by the states – is finite. AMR does not impose a fixed bound on the number of argument roles a predicate can have, but for any fixed AMR corpus, there will be a maximum. And we carry quantifiers only up to the verbal sub-AMR that introduces them – a kind of locality restriction on scope interaction – so we only ever have a finite number. So clearly, within the finite bounds needed for any corpus, an mbutt mapping AAMR to the surface-order scope HOL can be defined. This finite state transduction is ϵ-free and deterministic, but not linear because of variable copying.

4 Bounded quantifier raising

The calculation described in the previous section has nice properties, but it leaves quantifiers in their low 'surface order' positions. If we provide a fixed finite bound on the number of quantifiers requiring non-surface scope, a nondeterministic mbutt can be defined that permutes the quantifiers to get the alternative scopes. We sketch an extension to the transducer of the previous section with this effect, but a similar alternative approach could define the quantifier-raising transducer independently, and compose it with the previous transducer when alternative scopes are desired.

Hobbs&Shieber [23] observe that the following sentence with 3 quantifiers has not 3!=6 but only 5 alternative scopes:

Every representative of some company saw most samples.

The missing reading is the one where *every representative* takes widest scope, and *most samples* scopes over *some company*. That scope order is excluded because *representative* is relational and the *of*-phrase names one of its arguments.[4] Consider first the HOL surface scope of these simpler sentences:

Every representative saw most samples
surface scope: $\text{every}(\text{representative}, \lambda y.\text{most}(\text{sample}, \lambda x.\text{saw}(x, y)))$

Every representative of some company laughed
surface scope: $\text{every}(\lambda x.\text{some}(\text{company}, \lambda y.\text{representative}(x, y)), \text{laughed})$

Note that in the former example, *most* is in the second argument (sometimes called the 'scope') of *every*, while in the latter example, *some* is in the first argument (the 'restriction') of *every*. Returning now to the sentence considered by Hobbs&Shieber, it is clear that we cannot put *most* between *every* and *some*,

[4] The discussion in Hobbs&Shieber has an error that does not affect their main point. Their example sentence is not talking about things that are both representatives and also of-some-company – that doesn't quite make sense intuitively, and in fact gets the wrong entailments; see e.g. [37]. Rather, *representative* is relational and *of some company* specifies one of its arguments. We rephrase the Hobbs&Shieber argument here without that mistake. In the LDC AMR corpus [27], *representative* is treated relationally as it should be, as denoting an :arg0-of the predicate represent-01, where :arg0 is the representer and :arg1 is the thing represented.

since *some* must be in the first argument of *every*, and *most* needs to be in the second argument of *every*. Hobbs&Shieber mention a number of quantifier-scoping proposals that derived ill-formed structures for the missing reading, and they also note that while reducing 5 to 4 may seem unimportant, the same consideration reduces the possibilities for the following sentence from 5!=120 to 42:

> Some representative of every department in most companies saw
> a few samples of each product.

The ill-formed structures are avoided by any approach that respects the binary type of these quantifiers and only raises them, never lowering a quantifier into the argument of another quantifier.

It is now easy to see that a nondeterministic mbutt can do this quantifier raising if there is a fixed, finite bound on the number of quantifiers to raise. For the previous 2 simple examples, consider what is required to derive these non-surface scopes:

Every representative saw most samples
inverse scope: most(sample, λy.every(representative, λx.saw(x, y)))

Every representative of some company laughed
inverse scope: some(company, λy.every(λx.representative(x, y), laughed))

The former inverse LF means something like: *most samples are in the set of things that every representative saw*, and the latter inverse LF means roughly: *some company is such that every one of its representatives laughed*. To derive the former LF, instead of using the rule placing *most* and its restriction into surface scope position, these are lifted up to take wide scope over the whole formula. In the latter case, instead of using the rule placing *some* and its restriction into surface position, we lift them to wide scope. Comparing the surface and inverse representations for each of these examples, notice that just finitely many subtrees are lifted in each case – and they are, in fact, subtrees already computed by the transduction from AMR to surface order HOL. Clearly, with a finite bound on the number of moving quantifiers, we can define rules to place the moving quantifiers in all possible orders.

The quantifier raising steps are nondeterministic but it is ϵ-free and linear; as we see in the examples above, the quantifier raising step rearranges subtrees from the surface order, and never needs to delete or produce multiple copies of those subtrees. The Hobbs&Shieber algorithm achieves the same effect, but one advantage of a linear mbutt representation is that linear mbutts are closed with respect to composition [19,16]. For example, we can compose an mbutt that accepts any particular input HOL with the quantifier raising transduction to get a representation of the whole set of alternative scopes for the input. Hobbs&Shieber point out that many preferences can be incorporated relatively easily by controlling the nondeterministic choices, and at least some of these appear to be within the range of weighted mbutts [33] – a topic for future work.

There is an enormous and still growing literature on quantifier scoping patterns and preferences. In a recent summary review, Szabolcsi [39] discusses some preferences that were not mentioned in the paper by Hobbs & Shieber more than 20 years earlier, but points out that the main advance is the recognition that many different things are interacting to produce effects that are sometimes misleadingly lumped together as scope interactions. We split out one of the important factors in §5.

5 Dynamics and accommodation

We conclude with a brief sketch of a possible approach to the definite article and some of the many things related to that.[5] As noted at the beginning of this paper, when an entity is the argument of more than one predicate, roles from the respective predicates point to a single representation of that entity. This can happen in control constructions, with pronouns, with proper names, and with descriptions. Computing the coreference relations as a typical speaker would is clearly a hard problem [35], so as in the previous section, we consider how to factor this hard problem away from other aspects of AAMR calculation.

In a sense, the coreference in the first example of the paper is the easiest to recognize: it is unambiguously determined by the grammar of English control constructions. Repetition of names in a discourse usually, but not always, signals coreference, as in our second example. Recognizing intended coreference in descriptions and pronouns is obviously harder:

John giggled. He's delighted.
John giggled. The boy is very happy.

Though the details remain controversial, there are clearly common elements in these coreference determinations, whether they involve control verbs, names, pronouns or descriptions. Just as *The present king of France is bald* presupposes that the subject describes something, so *He is bald* presupposes that there is something salient and male available to refer to. To handle this in discourse requires some notion of context that extends across sentence boundaries. This can seem impossible because context includes all kinds of things and grows without bound, but Montague famously pointed out that it is not necessary to consider contexts in their full complexity: we need only track what is essential for understanding the particular discourse [34]. Furthermore, we have the fact that humans can not only do this with little or no conscious effort, but can predict fairly well how others will do it in a given context, as any careful writer knows.

There is a large literature about coreference resolution and tracking entities in discourse, but here I will sketch the bare outlines of a preliminary framework

[5] Here we focus on the definite article, sketching briefly the fundamental change to a dynamic perspective. But indefinite articles are even trickier and complicate the picture of how scope works, impeding progress until it was recognized that they require special treatment. As discussed for example in Kratzer [28] and references cited there, they are unlike quantifiers like *every* or *three*, unlike referential expressions formed with *the*, and not adequately handled by the discourse closure proposed by Heim [22] and DRT [25]. See e.g. [10].

proposed by Lebedeva [32], based on de Groote [21]. Where the static theory has propositions of type o, this framework gives them the type $\gamma \to (\gamma \to o) \to o$, a map from contexts γ to continuations $\gamma \to o$ to propositions o. This 'dynamization of types' can be propagated throughout the type hierarchy. The logical constants are also dynamized so that, for example, conjunction applies the update of the first proposition and then the update of the second. And the terms are similarly dynamized, with a special treatment reserved for pronouns and the definite article. The selection of an antecedent for a pronoun can be modeled by assuming the existence of a a dynamic selection function $\widetilde{\mathsf{sel}}$ which takes the context to return a dynamic entity. Let $\widetilde{\text{it}}$ represent the dynamized term that represents a function that selects a referent from context. Similarly let $\widetilde{\text{the}}$ represent a function that combines with a dynamic property to return an entity from the discourse context. When appropriate entities are not present, these functions can raise an exception that triggers an accommodation, e.g. the listener could just add a relevant entity. This is a big picture with many parts, but here I just want to make a small suggestion.

Consider again the simple example *The boy giggled.* In AAMR, the past tense and the definite article are indicated, and so now the natural proposal for its HOL representation is this:

```
(g / giggle-01.past
   :arg0 (b / boy.pl
            :quant: the ))
```

$$\widetilde{\text{the}}(\text{boy.pl}, \lambda b \exists g(\text{giggle-01.pres}(g) \wedge \text{:arg0}(g,b)))$$

Obviously, adding this treatment of the definite article has no effect on the complexity of the AAMR to HOL mapping. This approach simply marks this sentence as unlike the dynamized representation of *Some boy giggled* in a relevant respect. The operator $\widetilde{\text{the}}$ signals that an appropriate boy should be selected from context, or if that's not possible, some kind of accommodation should be triggered. Lebedeva proposes that not only definite descriptions, but names and pronouns should be treated in an analogous fashion.

Recognizing that the calculation of the selection function $\widetilde{\mathsf{sel}}$ is often challenging, notice that we could first calculate AAMR without the coreference links, without evaluating $\widetilde{\mathsf{sel}}$. In the example 2-sentence discourse given in §1, that would mean having two instances of individuals named *John*. Computing AMRs without selection is obviously going to be easier in many cases. Instead of an HOL translation of the AAMR given for the small dialog in §1, with the two occurrences linked, we could instead have only two dynamic terms. Elements that look for antecedents in the context can be marked in the translation but left unresolved. Coreference resolution, application of selection function, can then be done in a separate step and evaluated separately to see whether it assumes the same coreference links as a human speaker probably would in the same context.

6 Reforming the corpus and other future work

Why not translate directly from the strings in context to the higher order dynamic logic, dispensing with the AAMR? **First**, some practical points: AMR already annotates fairly large corpora, and it is being developed largely by volunteers and academics who disagree about what counts as an appropriate annotation for linguistic meaning. For this reason, a shared effort necessarily focuses on fundamental points of agreement. And for this same reason, it is relevant that the revisions proposed here are minor. AAMR adds some additional lexical information to AMR leaves, and attaches quantifiers to the nominal concepts they are associated with. In an aligned and parsed AMR corpus, the generation of the corresponding AAMR can be done largely mechanically. There is also a **second**, more elusive but possibly more important motivation for AMR, noted by many of the projects aiming to simplify semantic representations. For both practical and scientific reasons, it can be useful to have a semantic formalism that makes it easy to express those things that are most commonly expressed in natural language, without requiring a precision in interpretation that is completely unlike anything humans do. AMR is designed to fit the language, composing verb frames with classified argument and modifier roles, without settling the sometimes complex issues about quantifier scope, etc.

AAMR specifies a number of things that standard parse trees do not explicitly provide: verb senses, a semantic classification of arguments and modifiers, and intended coreference relations. This paper shows that fairly sophisticated logical representations with higher order argument structure can also be computed from these structures. If the computation of coreference is left to a later step, AAMR calculation should be quite feasible when context makes argument and modifier roles clear. And if determining the intended scope of quantifiers is left to a later step, the translation from AAMR into HOL is easy to define and compute. We have shown that when scope is local and scope determination is postponed, the translation to HOL is finite state and deterministic. Additional lexical information in the leaves sets the stage for treating other semantically important matters, some of which may similarly allow some of the relatively tractable parts of meaning representation to be factored away from the harder and less understood parts. In at least many cases, this approach will produce satisfactory results. In future work we hope to contribute to the production of an AAMR corpus with transducers that map it to reasonable HOL.

Acknowledgments: Many thanks to the anonymous reviewers for their valuable suggestions.

References

1. Adler, I., Dell, H., Husfeldt, T., Larisch, L., Salfelder, F.: The parameterized algorithms and computational experiments challenge – Track A: Treewidth (2016), https://pacechallenge.wordpress.com/pace-2016/track-a-treewidth/
2. Allen, J.F., Swift, M., de Beaumont, W.: Deep semantic analysis of text. In: ACL, SIGSEM Symposium on Semantics in Systems for Text Processing (STEP) (2008)
3. Artzi, Y., Lee, K., Zettlemoyer, L.: Broad-coverage CCG semantic parsing with AMR. In: 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1699–1710 (2015)
4. Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., Schneider, N.: Abstract meaning representation for sembanking. In: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse. pp. 178–186 (2013)
5. Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., Schneider, N.: Abstract meaning representation 1.2.2 specification (2015), https://github.com/amrisi/amr-guidelines/blob/master/amr.md, 18 Sept 2015
6. Bender, E.M., Flickinger, D., Oepen, S., Packard, W., Copestake, A.: Layers of interpretation: On grammar and compositionality. In: 11th International Conference on Computational Semantics. pp. 239–249 (2015)
7. Bienvenu, M., Kikot, S., Podolskii, V.: Tree-like queries in OWL 2 QL: Succinctness and complexity results (2015), https://arxiv.org/abs/1406.3047
8. Bonial, C., Babko-Malaya, O., Choi, J.D., Hwang, J., Palmer, M.: http://clear.colorado.edu/compsem/documents/propbank_guidelines.pdf (2010)
9. Bos, J.: Expressive power of abstract meaning representations. Computational Linguistics 42(3), 527–535 (2016)
10. Brasoveanu, A., Farkas, D.F.: How indefinites choose their scope. Linguistics and Philosophy 34, 1–55 (2011)
11. Carpenter, B.: Type-Logical Semantics. MIT Press, Cambridge, MA (1997)
12. Champollion, L.: The interaction of compositional semantics and event semantics. Linguistics and Philosophy 38(1), 31–66 (2015)
13. Chen, W.T., Palmer, M.: Unsupervised AMR-dependency parse alignment. In: 15th Conference of the European Chapter of the Association for Computational Linguistics (2017)
14. Copestake, A., Flickinger, D., Pollard, C., Sag, I.: Minimal recursion semantics: An introduction. Research on Language and Computation 3, 281–332 (2005)
15. Courcelle, B., Engelfriet, J.: Graph Structure and Monadic Second-Order Logic. Cambridge University Press, NY (2012)
16. Engelfriet, J., Lilin, E., Maletti, A.: Composition and decomposition of extended multi bottom-up tree transducers. Acta Informatica 46(8), 561–590 (2009)
17. Flanigan, J., Thomson, S., Carbonell, J., Dyer, C., Smith, N.A.: A discriminative graph-based parser for the abstract meaning representation. In: Annual Meeting of the Association for Computational Linguistics (2014)
18. Fülöp, Z., Kühnemann, A., Vogler, H.: A bottom-up characterization of deterministic top-down tree transducers with regular look-ahead. Information Processing Letters 91, 57–67 (2004)
19. Fülöp, Z., Kühnemann, A., Vogler, H.: Linear deterministic multi bottom-up tree transducers. Theoretical Computer Science 347, 276–287 (2005)

20. Ganian, R., Hliněný, P., Kneis, J., Langer, A., Obdržálek, J., Rossmanith, P.: Digraph width measures in parameterized algorithmics. Discrete Applied Mathematics 168, 88–107 (2014)
21. de Groote, P.: Towards a Montagovian account of dynamics. In: Semantics and Linguistic Theory 16 (2006)
22. Heim, I.: The Semantics of Definite and Indefinite Noun Phrases. Ph.D. thesis, University of Massachusetts, Amherst (1982)
23. Hobbs, J.R., Shieber, S.M.: An algorithm for generating quantifier scopings. Computational Linguistics 13, 47–63 (1987)
24. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: The making of a web ontology language. Web Semantics: Science, Services and Agents on the World Wide Web 1, 7–26 (2003)
25. Kamp, H.: A theory of truth and semantic representation. In: Groenendijk, G., Janssen, T., Stokhof, M. (eds.) Formal Methods in the Study of Language. Mathematisch Centrum, Amsterdam (1981)
26. Keenan, E.L.: Further beyond the Frege boundary. In: van der Does, J., van Eijck, J. (eds.) Quantifiers, Logic, and Language. CSLI Publications, Amsterdam (1996)
27. Knight, K., et al: Abstract meaning representation (AMR) annotation release 1.0 (2014), https://catalog.ldc.upenn.edu/LDC2014T12. See also http://amr.isi.edu/download.html
28. Kratzer, A.: Scope or pseudoscope? Are there wide-scope indefinites? In: Rothstein, S. (ed.) Events and Grammar. Springer, Dordrecht (1998)
29. Landau, I.: Control in Generative Grammar: A Research Companion. Cambridge University Press, NY (2013)
30. Landman, F.: Plurality. In: Lappin, S. (ed.) Handbook of Contemporary Semantic Theory, pp. 425–457. Oxford University Press, Oxford (1996)
31. Langekilde, I., Knight, K.: Generation that exploits corpus-based statistical knowledge. In: 36th Annual Meeting of the ACL. pp. 704–710 (1998)
32. Lebedeva, E.: Expression de la dynamique du discours a l'aide de continuations. Ph.D. thesis, Université de Lorraine (2012)
33. Maletti, A.: The power of weighted regularity-preserving multi bottom-up tree transducers. International Journal of Foundations of Computer Science 26(7), 293–305 (2015)
34. Montague, R.: Pragmatics. In: Thomason, R.H. (ed.) Formal Philosophy: Selected papers of Richard Montague. Yale University Press, 1974, New Haven (1968)
35. Morgenstern, L., Davis, E., Ortiz, C.L.: Planning, executing, and evaluating the Winograd schema challenge. AI Magazine 37(1), 50–54 (2016)
36. Muskens, R.: Order-independence and underspecification. In: Kamp, H., Partee, B. (eds.) Context-dependence in the Analysis of Linguistic Meaning, pp. 239–254. Elsevier (2004)
37. Partee, B.H., Borschev, V.: Genitives, relational nouns, and argument-modifier ambiguity. In: Lang, E., Maienborn, C., Fabricius-Hansen, C. (eds.) Modifying Adjuncts, pp. 67–112. Mouton de Gruyter, Berlin (2003)
38. Peters, P.S., Westerståhl, D.: Quantifiers in Language and Logic. Oxford University Press, Oxford (2006)
39. Szabolcsi, A.: Quantification. Cambridge University Press, NY (2010)
40. van Lambalgen, M., Hamm, F.: The Proper Treatment of Events. Blackwell, Oxford (2005)
41. Weischedel, R., Pradhan, S., Ramshaw, L., Kaufman, J., Franchini, M., El-Bachouti, M.: https://catalog.ldc.upenn.edu/docs/ldc2013t19/ontonotes-release-5.0.pdf (2015)