A Structured Representation for Noun Phrases and Anaphora

Syed S. Ali
Department of Computer Science
State University of New York at Buffalo
226 Bell Hall
Buffalo, NY 14260
syali@cs.buffalo.edu

Abstract

I present a computationally-based representation for indefinite noun phrases and anaphora that models their use in natural language. To this end, three goals for knowledge representation for natural language processing: natural form, conceptual completeness, and structure sharing are described. In addressing these goals, an augmentation to the representation of variables (corresponding to indefinite noun phrases or anaphora) so that variables are not atomic terms is suggested. This leads to an extended, more "natural" representation. It is shown how this representation resolves some representational difficulties with sentences with nonlinear quantifier scoping, in particular, donkey sentences.

1 Introduction

The intent of the work in this paper is to present a computationally-based representation for indefinite noun phrases and anaphora that models their use in natural language. To this end, I believe the following natural-language-specific goals must be addressed. First, the mapping from natural language sentences into the representation language should be as direct as possible, and the representation should reflect the structure of the natural language sentence it purports to represent. I call this the "natural form" constraint. This difficulty is particularly evident for rule-type sentences, such as *small dogs bite harder than big dogs,* where its first-order predicate logic representation takes the form of an implication whose antecedent constraints specify what kind of dog bite harder than another type. This representation, as a logical rule, contrasts with the predicate-argument structure of the original sentence, as below:

$$\forall x, y((\text{small}(x) \wedge \text{dog}(x) \wedge \text{large}(y) \wedge \text{dog}(y)) \Rightarrow \text{bites-harder}(x, y)) \quad (1)$$

By comparison, the representation of *Fido bites harder than Rover* is more consistent with the structure of the original sentence,

$$\text{bites-harder(Fido, Rover)} \quad (2)$$

This is so, despite the intuitive observation that the two sentences have nearly identical syntactic structure, and similar meaning.

Second, the subunits of the representation should be conceptually complete in the sense that any component of the representation of a sentence should have a meaningful interpretation independent of the interpretation of the entire sentence representation. For example, for the representation of the sentence as in (1) above, we might ask what is the meaning of x or y? Presumably, some thing in the world, or a set denoting the non-empty universe. Note that the original sentence mentions only dogs. I suggest that a better translation might be:

$$\text{bites-harder}(\forall x \text{ such that small dog}(x), \forall y \text{ such that large dog}(y))$$

where the variables, x and y, have their own internal structure that reflects their conceptualization. Note that I am suggesting something stronger than just restricted quantification (simple type constraints can certainly be felicitously represented using restricted quantifiers). Complex internalized constraints (that is, other than simple type) and internalized quantifier structures characterize this approach to the representation of variables. Thus the representation of the sentence: *Every small dog that is owned by a bad-tempered person bites harder than a large dog* should reflect the structure of the representation of (2).

Third, a high degree of structure sharing should be possible, as multi-sentence connected discourse often uses reduced forms of previously used terms in subsequent reference to those terms. This corresponds to the use of pronouns and some forms of ellipsis in

(a) $\forall x$ (farmer(x) $\Rightarrow$ $\exists y$ (donkey(y) & owns(x, y) & beats(x, y)))
(b) $\forall x$ (farmer(x) $\Rightarrow$ (($\exists y$ donkey(y) & owns(x, y)) $\Rightarrow$ beats(x, y)))
(c) $\forall x$ $\forall y$ ((farmer(x) & donkey(y) & owns(x, y)) $\Rightarrow$ beats(x, y))

Figure 1: Three FOPL Representations of the Donkey Sentence

discourse. An example of this phenomena is the representation of intersentential pronominal reference to scoped terms, e. g.,

> Every apartment had *a dishwasher*. In some of them *it* had just been installed.
> Every chess set comes with *a spare pawn*. *It* is taped to the top of the box.

(examples from [Heim, 1990]). The structures that are being shared in these sentences are the variables corresponding to the italicized noun phrases. Logical representations can only model this "sharing" by combining multiple sentences of natural language into one sentence of logic. This method is unnatural for at least two reasons. First, when several sentences must be combined into one sentence the resulting logical sentence, as a conjunction of several potentially disparate sentences, is overly complex. Second, this approach is counter-intuitive in that a language user can re-articulate the original sentences that he/she represents. This argues for some form of separate representations of the original sentences and their associated noun phrases. The problem with logic in this task is that logic requires the complete specification of a variable, corresponding to a noun phrase, and its constraints in the scope of some quantifier. This difficulty is not restricted to noun phrases, indeed it is frequently the case that entire subclauses of sentences are referred to using reduced forms such as "too" e. g.,

> John *went to the party*. Mary did, *too*.

Generation of such reduced forms (and other constructions such as locative expressions [Haller and Ali, 1990]) requires a knowledge representation formalism that models this sort of reference, minimally by structure sharing.

Finally, any computational theory must incorporate knowledge-structuring mechanisms, such as subsumption and inheritance of the sort supported in frame-based and semantic network based systems [Brachman, 1979]. A taxonomy provides "links" that relate more general concepts to more specific concepts. This allows information about more specific concepts to be associated with their most general concept, and information filters down to more specific concepts in the taxonomy via inheritance. More general concepts in such a taxonomy *subsume* more specific concepts with the subsumee inheriting information from its subsumers. For atomic concepts, subsumption relations between concepts is specified by the links of the taxonomy. A clear example of subsumption in natural language is the use of descriptions such as *person that has children* subsuming *person that has a son*. If one were told: *People that have children are happy,* then it follows that *People that have a son are happy*. The intuitive idea is that more general descriptions should subsume more specific descriptions of the same sort, which in turn inherit attributes from their more general subsumers.

The so-called *donkey sentences* [Geach, 1962] illustrate the utility of these goals. These are sentences that pronominally refer to quantified variables in closed subclauses, for example *Every farmer who owns a donkey beats it* where the noun phrase *a donkey* is a variable inside the scope of a universally quantified variable (*every farmer*) and is referred to pronominally outside the scope of the existentially quantified donkey. Figure 1 has some attempts to represent the above sentence in FOPL.

Representation (a) says that *every* farmer owns a donkey that he beats, which is clearly more than the original sentence intends. Representation (b) is a better attempt, since it captures the notion that we are considering only farmers who own donkeys; however, it contains a free variable. Representation (c) fails to capture the sense of the original sentence in that it quantifies over *all* farmers and donkeys, rather than just farmers that own donkeys. To see this, consider the case of the farmer that owns two donkeys and beats only one of them. Clearly, the donkey sentence can apply to this case, but interpretation (c) does not.

Summarizing, I have presented some characteristics of natural language that a knowledge representation and reasoning system should support. In the remainder of this paper I will present an alternative representation for simple unstructured variables which

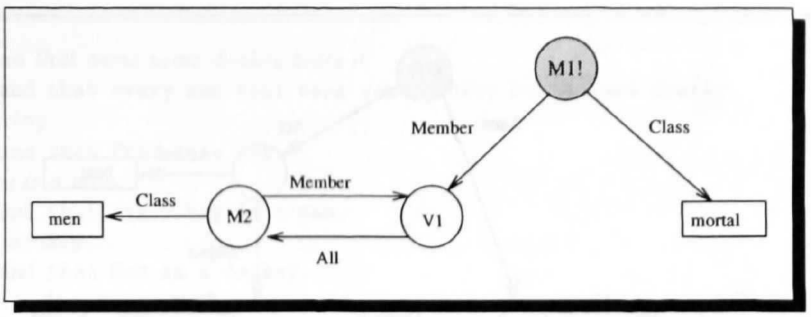

Figure 2: Structured Variable Representation of *All men are mortal.*

involves according variables potentially complex internal structure and show how it addresses the main goals of this paper. This involves providing a syntax and semantics for structured variables. The syntax and semantics of the associated logic is specified by a complete definition of a propositional semantic network representation formalism (an augmentation of [Shapiro, 1991]) and is fully described in [Ali, 1993b]. The implemented system is called ANALOG (A NAtural LOGic). Because of space limitations a full specification of the system cannot be presented here. In particular, the subsumption procedure is described in [Ali, 1993a]. However, I illustrate the utility of the formalism, in addressing the goals of a natural logic, with a natural language demonstration using the donkey sentence.

2 Structured Variables

I am attempting to represent variables as a "bundle of constraints and a binding structure (quantifier). I term these bundles "structured variables" because variables, in this scheme, are non-atomic terms. The implemented language of representation is a propositional semantic network representation system called ANALOG (A NAtural LOGic) which is a descendant of SNePS [Shapiro, 1979,Shapiro and Rapaport, 1987]. An example of a structured variable (the node labelled `V1`) is given in Figure 2. Scoping of existential structured variables (with respect to universal structured variables) is expressed by the **depends** arcs to universal structured variable nodes (an example is the node labelled `V2` in Figure 3). The complete semantics of structured variables is augmented (by the addition of arbitrary individuals) semantic theory based on [Shapiro, 1979, Shapiro and Rapaport, 1987, Fine, 1985a, Fine, 1985b] and described in [Ali, 1993a].

3 Advantages of the Representation

I suggest that the representation of numerous types of quantifying expressions, using structured variables, is more "natural" than first-order-based logics, because the mapping of natural language sentences is direct. An example is given in Figure 2.[1] Note that the shaded node, labelled `M1!`, corresponds to the believed proposition that *all men are mortal.* `V1` is the structured variable corresponding to *all men.* The `member-class` case frame is the representation for the proposition that an object is a member of a class. This representation is more natural in that the top-level proposition is one of class membership, rather than a rule-like if-then proposition. Further the structure of *any* proposition about class membership (e. g., a ground proposition such as *Bill is a man*) would be represented similarly.

The representation of structured variables suggested here can represent most first-order quantifying expressions directly. Also, we can represent general quantifying expressions directly (although their semantics need to be detailed) as with generalized quantifiers [Barwise and Cooper, 1981]. In general, there is a direct mapping from natural language quantifying expressions into structured variable representations, as structured variables correspond directly to noun phrases with restrictive relative clause complements.

In typical logics, terms in one formula are not referenced in other formulas. In general, re-using a term

[1] a node may be labeled by a "name" (e.g., `BILL`, `M1`, `V1`) as a useful (but extra-theoretic) way to refer to the node. This naming of a node is of the form `M`n, where n is some integer. A "!" is appended to the name to show that the proposition represented by the node is believed to be true. However, the "!" does not affect the identity of the node, nor the proposition it represents. Similarly, variable nodes naming is `V`n and base nodes naming is `B`n where n is some integer.

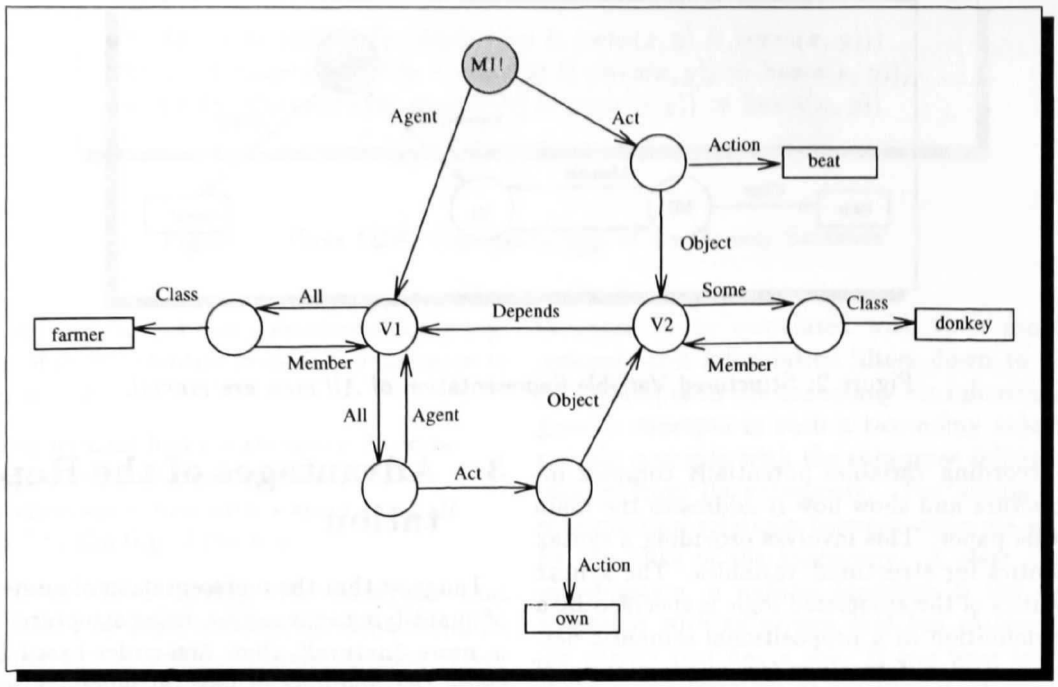

Figure 3: Representation for the Donkey Sentence: *Every farmer that owns a donkey beats it.*

involves re-writing the term in the new formula. Ideally, I would like to re-use exactly the same terms in different formulas, and I would like these terms to be closed formulas, i.e., *complete*, so that they may be meaningfully re-used and shared by multiple formulas. Terms with this property are called *conceptually complete*. With structured variables all terms are conceptually complete, because variables have all their constraints and quantifier in their structure. This is also an issue in the representation of multisentential dialog, where intersentential reference to sententially scoped objects frequently occurs (as described in section 1).

3.1 Donkey Sentences

With structured variables there are no open terms and it is possible for a sentence to reuse constituent terms anywhere. Thus the ill-formed representation (in FOPL) of (b) (of Figure 1) is possible and is the preferred representation. It is shown in Figure 3. The shaded node, labelled `M1!`, denotes the proposition *Every farmer who owns a donkey beats it* and `V1` and `V2` are *every farmer that beats a donkey he owns* and *a beaten donkey that is owned by any farmer*, respectively. This representation is also more natural because the top-level proposition is one of an act of owning, thus it is similar to any other sentence about owning such as *Fred owns a donkey*. This is the representation that is built and generated from when the donkey sentence is processed.

4 A Demonstration

ANALOG includes a generalized augmented transition network (GATN) natural language parser and generation component linked up to the knowledge base (based on [Shapiro, 1982]). A GATN grammar specifies the translation/generation of sentences involving complex noun phrases into/from ANALOG structured variable representations.

An apparent advantage of the use of structured variables lies in the representation and generation of complex noun phrases that involve restrictive relative clause complements. The restriction set of a structured variable typically consists of a type constraint along with property constraints (adjectives) and other more complex constraints (restrictive relative clause complements). Parsing and generation of these noun phrases is simplified because structured variables collect all relevant restrictions on a variable into one unit, a structured variable. The parser parses the user's sentence and builds an ANALOG representation for the user input. The resulting representation is then passed to the generation component which generates the output response (perhaps, with the canned phrase `I understand that`). If con-

```
: Every man that owns some donkey beats it
I understand that every man that owns some donkey beats some donkey.      (1)
: Fred is a boy
I understand that Fred is a boy.
: Every boy is a man
I understand that every boy is a man.
: Doc is a donkey
I understand that Doc is a donkey.
: Dumbo is a donkey
I understand that Dumbo is a donkey.
: Fred owns Doc
I understand that Fred owns Doc.
: Fred owns Dumbo
I understand that Fred owns Dumbo.
: Does Fred beat Doc                                                      (2)
I don't know.
: Does any man beat some donkey
Yes, every man that owns some donkey beats some donkey.                   (3)
: Every man that owns any donkey beats it
I understand that every man that owns any donkey beats every donkey.      (4)
: Does Fred beat Doc
Yes, Fred beats Doc.                                                      (5)
: Does any man beat some donkey
Yes, every man that owns some donkey beats some donkey.                   (6)
: Does any man beat any donkey
Yes, Fred beats Doc and every man that owns any donkey beats every
donkey and every man that owns some donkey beats some donkey.             (7)
: Who beats a donkey
Fred beats Doc and every man that owns any donkey beats every donkey
and every man that owns some donkey beats some donkey.                    (8)
```

Figure 4: Dialog about the Donkey Sentence.

straints on variables corresponding to the complex noun phrases were represented using FOPL then it would be difficult to generate natural language noun phrases corresponding to these variables. This is because the constraints on variables would, likely, be well separated from the variables in the antecedents of rules involving these variables. This is not the case in a structured variable representation.

Since this representational formalism is grounded in an inherently nonlinear notation (semantic networks), the representation of tree-like quantifier scopings is straightforward. In addition, the structure of ANALOG's semantic network representation allows structure-sharing (and, indeed, may require it) to a high degree. A goal was the representation of coreference in multisentential dialog by structure sharing. A good example of both of these points is the donkey sentence, where a scoped constituent of a noun phrase (the *donkey* in *Every farmer who owns a donkey beats it*) in used in the main clause of the sentence.

Figure 4 illustrates a (slightly edited) dialog involving questions about the donkey sentence. User input is emphasized and some system output (CPU timings, etc) has been removed. Noun phrases are uniformly represented using structured variables. Parsing and generation of these noun phrases is simplified because structured variables collect all restrictions on a variable into one unit, a structured variable. The parser parses the user's sentence and builds an ANALOG representation for the user input. The resulting representation is then passed to the generation component which generates the output response (perhaps, with the canned phrase `I understand that`). If constraints on variables corresponding to the complex noun phrases were represented using FOPL then it would be difficult to generate natural language noun phrases corresponding to these variables. This is because the constraints

on variables would, likely, be well separated from the variables in the antecedents of rules involving these variables. This is not the case in a structured variable representation.

In Figure 4 the system reiterates its understanding of the donkey sentence with sentence (1). Note that the **some donkey** referred to in the subclause and main clause are the identical existential structured variable, the system cannot, as yet, generate pronouns. The system's response with sentence (2) indicated that the representation of the donkey sentence is not that of Figure 1(c), since it would have answered yes had that been the case. The system's response with sentence (3) indicates that the representation of the donkey sentence is not that of Figure 1(a), since the system just reiterates the rule rather than stating that Fred beats some donkey. The system is initially unable to determine whether Fred beats Doc or Dumbo. This is because the initial rule (**every man that owns some donkey beats it**) is satisfied in a model where only one of the donkeys is being beaten. After the system is told that *all* such donkeys are beaten (sentence (4)), it does determine that Fred beats Doc. Note that, as in natural language, the answers to many questions are often rules themselves (e. g., **Who beats a donkey** has as one answer **Every man that owns some donkey**). This is possible because the natural form and completeness constraints require that the representations for rules have the same structure as ground propositions. Thus, by subsumption the answer to a question is any proposition subsumed by the question (since the question and answer will have the same structure). This is illustrated in responses (6-8).

5 Summary

I presented a computationally-based representation for indefinite noun phrase and anaphora that models their use in natural language I have described part of a KRR system that incorporates this representation for variables which correspond to indefinite noun phrases or anaphora. ANALOG is a propositional semantic-network-based knowledge represen tation and reasoning system that supports many aspects of natural language use, in particular, the representation and generation of complex noun phrases and sentences (natural form), the representation of various types of quantified variable scoping (conceptual completeness), description subsumption, and a high degree of structure sharing. I have presented an example of natural language dialog involving the donkey sentence that illustrates some of the utility of this formalism for natural language processing.

References

[Ali, 1993a] Syed S. Ali. *A "Natural Logic" for Natural Language Processing and Knowledge Representation.* PhD thesis, State University of New York at Buffalo, Computer Science, 1993. Forthcoming.

[Ali, 1993b] Syed S. Ali. Natural Language Processing Using Propositional Semantic Networks. *Minds and Machines*, 1993. Special Issue on Knowledge Representation for Natural Language Processing, (to appear).

[Barwise and Cooper, 1981] Jon Barwise and Robin Cooper. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4:159–219, 1981.

[Brachman, 1979] Ronald J. Brachman. On the Epistemological Status of Semantic Networks. In N. V. Findler, editor, *Associative Networks: Representation and Use of Knowledge in Computers.* Academic Press, New York, 1979.

[Fine, 1985a] Kit Fine. Natural deduction and arbitrary objects. *Journal of Philosophical Logic*, 14:57–107, 1985.

[Fine, 1985b] Kit Fine. *Reasoning with Arbitrary Objects.* Basil Blackwell, Oxford, 1985.

[Geach, 1962] Peter Thomas Geach. *Reference and Generality.* Cornell University Press, Ithaca, New York, 1962.

[Haller and Ali, 1990] Susan M. Haller and Syed S. Ali. Using focus for generating felicitous locative expressions. In *Proceedings of the Third International Conference on Industrial and Engineering Applications of Artificial Intelligence*, pages 472–477. ACM, 1990.

[Heim, 1990] Irene Heim. Discourse Representation Theory, 1990. Tutorial material from ACL-90.

[Shapiro and Rapaport, 1987] Stuart C. Shapiro and William J. Rapaport. SNePS Considered as a Fully Intensional Propositional Semantic Network. *Proceedings of the 5th National Conference on Artificial Intelligence*, 1:278–283, 1987.

[Shapiro, 1979] Stuart C. Shapiro. The SNePS Semantic Network Processing System. In N. V. Findler, editor, *Associative Networks: Representation and Use of Knowledge by Computers*, pages 179–203. Academic Press, New York, 1979.

[Shapiro, 1982] S. C. Shapiro. Generalized augmented transition network grammars for generation from semantic networks. *The American Journal of Computational Linguistics*, 8(1):12–25, 1982.

[Shapiro, 1991] Stuart C. Shapiro. Cables, paths, and "subconscious" reasoning in propositional semantic networks. In John F. Sowa, editor, *Principles of Semantic Networks*, pages 137–156. Morgan Kaufmann, 1991.