CONCEPTUAL DEPENDENCY AND MONTAGUE GRAMMAR: A STEP TOWARD CONCILIATION

Mark A. Jones
David S. Warren*

State University of New York at Stony Brook
Long Island, New York 11794

ABSTRACT

In attempting to establish a common basis from which the approaches and results can be compared, we have taken a conciliatory attitude toward natural language research in the conceptual dependency (CD) paradigm and Montague Grammar (MG) formalism. Although these two approaches may seem to be strange bedfellows indeed with often noticeably different perspectives, we have observed many commonalities. We begin with a brief description of the problem view and ontology of each and then create a formulation of CD as logic. We then give "conceptual" MG translations for the words in an example sentence which we use in approximating a word-based parsing style. Finally, we make some suggestions regarding further extensions of logic to introduce higher level representations.

I INTRODUCTION

In the past decade a series of "process models" have been developed that attempt to capture various aspects of natural language understanding. We refer primarily to models having some form of underlying conceptual representation such as Schank's conceptual dependency (CD) notation [1] and, possibly, higher level knowledge structures such as scripts [2] (or frames [3]), plans [4], MOPs [5], TAUs [6], etc. that allow inferences to be made. Since we shall focus primarily on the systems of Schank's group we will refer to these models as CD models. The language understanding mechanisms in CD models have been explained by example, by English prose, and by the publication of micro versions (programs) [7], but not yet in any truly formal way that would facilitate comparison to other approaches and evaluation of alternative representational choices.

On the other hand, one of the most formally elaborated systems for natural language description is Montague Grammar (MG) [8,9]. MG is a logic system based on the typed lambda calculus that is capable of expressing modality, tense, intension, extension, etc. It provides for a particularly extensive treatment of reference (quantification, possible worlds, etc.).

The emphases and goals of the CD and MG research have not generally coincided and a direct comparison is difficult. The CD systems have attempted to model cognitive processes and have focused on contextual (story) understanding, summarization and question answering tasks. MG has been a research tool for language philosophers and linguists, has not been computationally applied ([10] is an exception), and has been directed primarily at declarative sentences in isolation. The remainder of this paper will attempt to formally characterize aspects of the CD systems by extending the MG framework to accommodate the objects and processes used in CD systems for contextual understanding. It is our hope that we can make the reference strategies and representational choices in CD systems perspicuous, and also extend the MG formalism to include a wider range of phenomena.

II ONTOLOGY OF CD AND MG

The basic CD ontology views the world in terms of (a) picture producers (PPs), which correspond to real world entities, (b) real world events (occurrences of acts), (c) states, and (d) temporal and causal relations that may exist between pairs of events or states. Schank maintains that a relatively small number of distinct types of primitive acts, states and relations combine in a variety of ways to represent simple physical events and their interactions. Scripts, plans, goals, themes, MOPs, TAUs, etc. have been proposed as useful ways to represent general knowledge about particular configurations of CD objects for the purpose of inference and disambiguation.

A PP is either explicitly introduced in a noun phrase (e.g., John, a man, the boy in the blue coat) or is implicitly introduced via reference to a higher level representation such as a script in which it participates (or can be inferred to participate). In sentence (1), the policeman is implicitly introduced in the first clause and only by that introduction can the pronominal reference "he" in the second clause be understood. Similarly, in sentence (2), the use of the definite referent "the" is not odd despite the fact that a policeman has not yet been explicitly introduced.

(1) I was stopped yesterday for speeding, but he didn't give me a ticket.

(2) I was stopped yesterday for speeding, but the policeman didn't give me a ticket.

A conceptual analysis of a natural language

---

* This material is based upon work supported by the National Science Foundation under grant #IST-8010834.

expression requires that the underlying acts, states and relations be identified. "John killed Mary" is analyzed as "John did some (unspecified) action which resulted (by result causation) in Mary undergoing a state change from alive to dead" [1, p.50]. The action may be specified or inferred from other phrases added to the sentence or the context in which the sentence appears.

The ontology of MG is based on the set of truth values (true, false), the set of entities (PPs of CD), and the set of indices consisting of possible worlds and points of time. A function space is constructed inductively from these basic sets. These include sets of entities, sets of sets of entities, etc. For example, common nouns are treated as denoting sets of entities. For most of our discussion here we will consider only extensional aspects of MG and so the set of possible worlds does not enter.

An important aspect of MG is the close coupling of syntax and semantics. A Montague Grammar consists of an inductive definition of the set of meaningful phrases of English. The model-theoretic interpretation of each phrase is defined recursively over its syntactic derivation. For perspicuity, this interpretation is defined by means of a translation into a typed lambda calculus for which a model-theoretic semantics has previously been provided. So even though most of the mechanics of manipulating phrases and meanings in MG look like syntactic operations of lambda formulas (or LISP-like code), we should bear in mind that the real semantic objects are such things as sets of entities and their properties and not the lambda formulas themselves.

One contribution of MG is the unified treatment of quantified NPs (e.g., every man, a woman) and proper nouns (e.g., John). This is achieved by considering all noun phrases as referring to sets of properties of entities. So the meaning of the proper name "John" is taken to be the set of properties of the entity john; in symbols $\lambda P\,(P(\mathrm{john}))$. (Some intuition can be gained by considering this as a LISP function that takes a predicate as argument and returns the result of applying that predicate to the atom JOHN: (LAMBDA (P) (P JOHN)).)

Intransitive verbs are translated to sets of entities, i.e., simple predicates. E.g., the intransitive verb "walk" has as its meaning a set of entities: those that walk; in the lambda calculus this is referred to by a predicate walk'. (In LISP a predicate (LAMBDA (X) (WALK X)), or just WALK.) The meaning of a sentence is obtained by applying the function that is the meaning of the subject to the predicate that is the meaning of the verb phrase. Thus "John walks" has the meaning

$[\lambda P\,(P(\mathrm{john}))]\,(\mathrm{walk}')$ which λ-reduces to $\mathrm{walk}'(\mathrm{john})$

The advantage of this added complexity is that it also handles quantified noun phrases. For example, "a woman" translates to the set of properties that are true of some woman:

$$\lambda P\,(\exists x\,(\mathrm{woman}'(x) \wedge P(x))).$$

The sentence rule given above generates the meaning of "A woman walks" as

$$[\lambda P\,(\exists x\,(\mathrm{woman}'(x) \wedge P(x)))]\,(\mathrm{walk}') \overset{\lambda}{\Rightarrow} \exists x\,(\mathrm{woman}'(x) \wedge \mathrm{walk}'(x))$$

Similarly, "every woman walks" is

$$[\lambda P\,(\forall x\,(\mathrm{woman}'(x) \rightarrow P(x)))]\,(\mathrm{walk}') \overset{\lambda}{\Rightarrow} \forall x\,(\mathrm{woman}'(x) \rightarrow \mathrm{walk}'(x))$$

Another contribution of MG is its handling of pronominalization and coreference. This is done by introducing syntactic variables as NPs and then substituting a normal NP for the first occurrence of a particular variable in a phrase and appropriate pronouns for subsequent occurrences of the same variable. For example, we can first generate the sentence "y walks and y talks", where y is a syntactic NP variable. Then we substitute the NP "a man" and obtain "a man walks and he talks". The corresponding semantic rule is to λ-abstract the variable substituted for over the sentence translation and then to apply to that result the translation of the substituted NP. (This asserts that the NP substituted in has the property defined by the sentence with respect to the variable substituted for.) For example, for the sentence above, we get

$$[\lambda P\,(\exists x(\mathrm{man}'(x) \wedge P(x)))]\,(\lambda y\,(\mathrm{walk}'(y) \wedge \mathrm{talk}'(y))) \overset{\lambda}{\Rightarrow} \exists x\,(\mathrm{man}'(x) \wedge [\lambda y\,(\mathrm{walk}'(y) \wedge \mathrm{talk}'(y))]\,(x)) \overset{\lambda}{\Rightarrow} \exists x\,(\mathrm{man}'(x) \wedge \mathrm{walk}'(x) \wedge \mathrm{talk}'(x))$$

This substitution mechanism provides a very powerful and flexible way to bind various occurrences of a variable to an entity or, in CD terms, to fill many slots with the same referent.

III A CONCILIATION

Now that we have introduced the two approaches we move to an example that shows how they might overlap. We formulate CD as logic in order to facilitate the comparison (cf. [11]). We will assume a sorted and typed lambda calculus in which there is an IS-A hierarchy of entities which distinguishes among and within PPs, acts, and states. The notation x_i: #<class> is used to indicate that the variable x_i takes values from the sort, #<class>. Consider the conceptual case frame for "John walked to a store":

```
(PTRANS (ACTOR HUM0)
        (OBJECT HUM0)
        (FROM NIL)
        (TO PHYSOBJ0)
        (INST (MOVE (ACTOR HUM0)
                    (OBJECT (BODYPART (TYPE
                       (FEET))))
                    (TIME TIME0)))
        (TIME TIME0)
```

where HUM0 is: (CLASS (#PERSON))
(FIRSTNAME (JOHN))
(GENDER (MASC))

TIME0 is: (CLASS (#TIME))
(BEFORE (*NOW*)

PHYSOBJ0 is: (CLASS (#STORE))

We propose the following logical representation of this case frame:

$$\begin{array}{l}
\exists x_1{:}\#\text{EVENT}\quad \exists x_2{:}\#\text{PERSON}\quad \exists x_3{:}\#\text{PHYSOBJ}\quad \exists x_4{:}\#\text{STORE}\\
\exists x_5{:}\#\text{EVENT}\quad \exists x_6{:}\#\text{TIME}\\
\quad [\text{PTRANS}(x_1) \wedge \text{ACTOR}(x_1,x_2) \wedge \text{OBJECT}(x_1,x_2) \wedge\\
\qquad \text{FROM}(x_1,x_3) \wedge \text{TO}(x_1,x_4) \wedge \text{INST}(x_1,x_5) \wedge\\
\qquad \text{MOVE}(x_6) \wedge \text{ACTOR}(x_5,x_2) \wedge \text{OBJECT}(x_5,\text{FEET}(x_2)) \wedge\\
\qquad \text{TIME}(x_5,x_6) \wedge \text{TIME}(x_1,x_6) \wedge \text{FIRSTNAME}(x_2,\text{JOHN}) \wedge\\
\qquad \text{GENDER}(x_2,\text{MASC}) \wedge \text{BEFORE}(x_6,{*}\text{NOW}{*})]
\end{array}$$

We can abbreviate this formula, to appear more like the CD case frames, as:

$$\begin{array}{l}
\exists x_1{:}\#\text{EVENT}\quad \exists x_2{:}\#\text{PERSON}\quad \exists x_3{:}\#\text{PHYSOBJ}\quad \exists x_4{:}\#\text{STORE}\\
\exists x_5{:}\#\text{EVENT}\quad \exists x_6{:}\#\text{TIME}\\
\quad [(\text{PTRANS}\ x_1\ (\text{ACTOR}\ x_2)\\
\qquad\qquad\qquad (\text{OBJECT}\ x_2)\\
\qquad\qquad\qquad (\text{FROM}\ x_3)\\
\qquad\qquad\qquad (\text{TO}\ x_4)\\
\qquad\qquad\qquad (\text{INST}\ (\text{MOVE}\ x_5\ (\text{ACTOR}\ x_2)\\
\qquad\qquad\qquad\qquad\qquad\qquad (\text{OBJECT}\ (\text{FEET}\ x_2))\\
\qquad\qquad\qquad\qquad\qquad\qquad (\text{TIME}\ x_6)))\\
\qquad\qquad\qquad (\text{TIME}\ x_6))\\
\quad \wedge(\#\text{PERSON}\ x_2\ (\text{FIRSTNAME}\ (\text{JOHN}))\\
\qquad\qquad\qquad (\text{GENDER}\ (\text{MASC})))\\
\quad \wedge(\#\text{TIME}\ x_6\ (\text{BEFORE}\ ({*}\text{NOW}{*})))\\
\quad \wedge(\#\text{STORE}\ x_4)]
\end{array}$$

The table below gives translations for the words that occur in our example sentence, 'John walked to a store'. For each word, the table gives a syntactic rule that describes how it combines with related expressions, and a corresponding semantic rule that shows how their meanings combine.

'word'	Translation [[word]]	Syntactic Rule	Semantic Rule
John	λP (∃y:#PERSON P(y) ∧ (#PERSON y (FIRSTNAME (JOHN)) (GENDER (MASC))))	'John'	[['John']]
store	#STORE	'store'	[['store']]
a	λQ λP (∃y:Q P(y) ∧ Q(y))	'a' β:common noun	[['a']] ([[β]])
-ed	λP (∃y:#TIME P(y) ∧ (#TIME y (BEFORE (*NOW*))))	α:verb '-ed'	[['-ed']] (λz[[α]])
walk*	∃x_1:#EVENT ∃x_5:#EVENT (PTRANS x_1 (ACTOR y_2) (OBJECT y_2) (FROM y_3) (TO y_4) (INST (MOVE x_5 (ACTOR y_2) (OBJECT (FEET y_2)) (TIME y_6))) (TIME y_6))	The PP-expressions y_2:#ANI, y_3:#PHYSOBJ, y_4:#PHYSOBJ, and y_6:#TIME appear in syntactically appropriate sequences with 'walk', e.g., y_2 appears before walk, y_4 may be marked with 'to', etc.	For the first PP-expression y_i, [[y_i]] (λy_i [['walk']]) = w_1. w_{j+1} is formed by taking another PP, y_k, & forming [[y_k]](λy_k (w_j)) for j=1,2,...

---

*This is a slight simplification -- to be precise we would add syntactic variables in an analogous way to Montague's use of he_i.

The tree in Figure 1 shows how the word translations are combined using the semantic rules given in the table to form the meaning of the sentence. For perspicuity, we have chosen bound variable names (e.g., x_6) to correspond to the names in the CD case frame of the example.

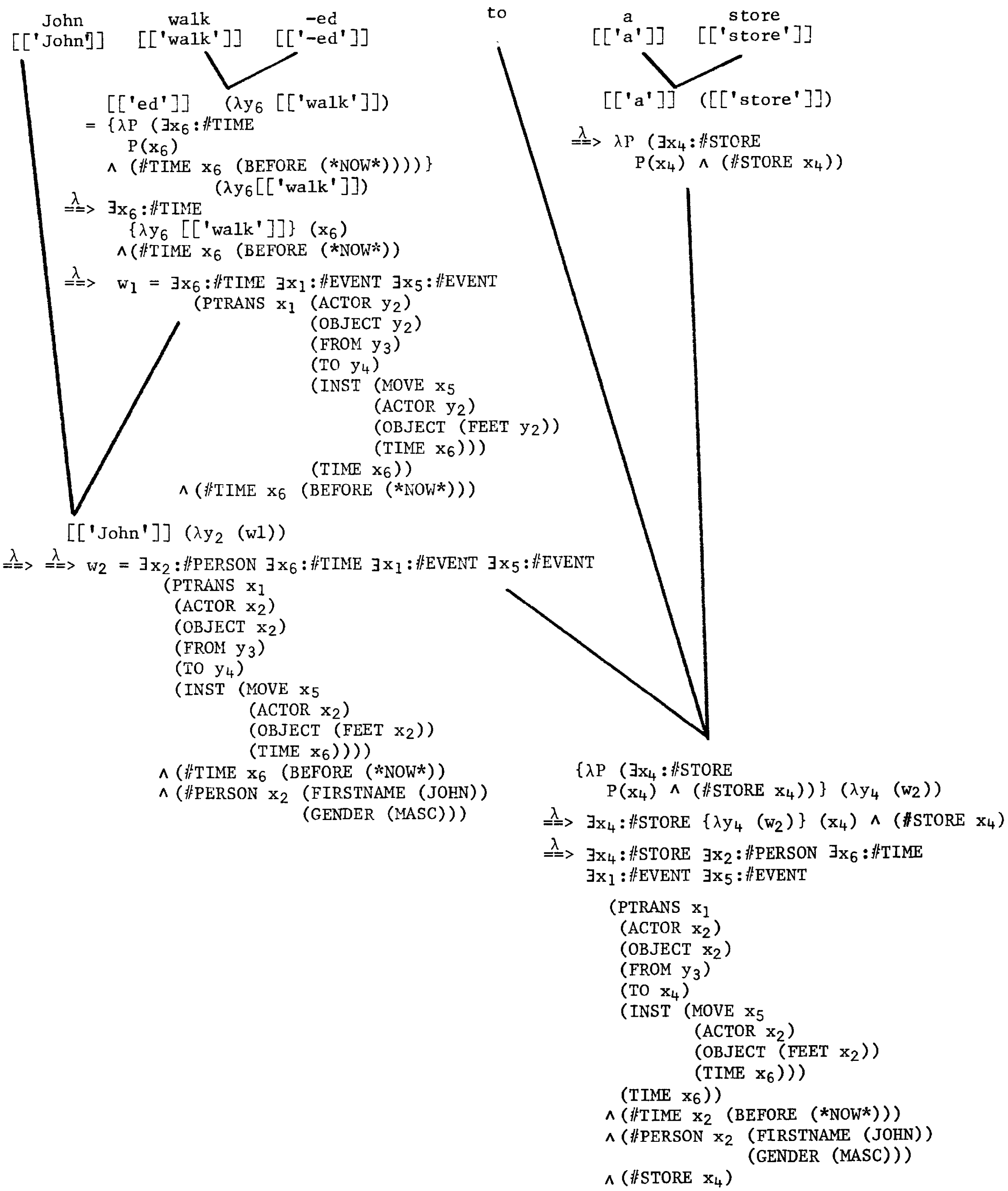

Figure 1: Parse Tree Example

The tree in Figure 1 reproduces a particular order of applying the rules which corresponds to a particular demon firing sequence in a CD parser. Variables which remain free in the final semantic representation, e.g. x_3, are assumed to be existentially quantified.

An important area for future investigation is the logical analysis of inference processes. Low level inference processes are quite naturally handled in logic. For example, the inference rule below states, "if x_2 EXPELs x_3 from x_4 to x_5, then x_3 was previously INGESTed".

$$\begin{aligned}&\forall x_1 x_2 x_3 x_4 x_5 x_6\ (\text{EXPEL } x_1 \text{ ACTOR } x_2 \text{ OBJECT } x_3 \text{ FROM } x_4 \text{ TO } x_5 \text{ TIME } x_6)\\ &\quad\rightarrow \exists x_7 x_8 x_9 x_{10}\ ((\text{INGEST } x_7 \text{ ACTOR } x_2 \text{ OBJECT } x_3 \text{ FROM } x_8 \text{ TO } x_9 \text{ TIME } x_{10})\\ &\qquad \wedge (\#\text{TIME } x_{10}\ (\text{BEFORE } x_6)))\end{aligned}$$

Higher level inference processes such as script application may be viewed as generalized configurations of acts and states with quantified script variables (associated PPs that participate in the script). By matching the patterns that arise in processing text with a generalized script configuration, references such as the policeman in sentences (1) and (2) can be computed. The instantiation of the "speeding" script in the first clause introduces the existentially quantified policeman that the referent in the second clause requires.

IV CONCLUSION

We have presented here a first step towards a conciliation of CD and MG. This hesitant step was taken at the expense of some simplifications in both approaches; we admittedly have not included important aspects of each. We believe, however, that we have found a common base that will allow the further interaction and development of each theory. The stage is set for more extensive communication in which the ideas important to each approach can be evaluated in terms of the other, and in which each can incorporate the other's successes.

REFERENCES

[1] Schank, R.C. *Conceptual Information Processing*. North-Holland, Amsterdam, 1975.

[2] Cullingford, R.E. Script Application: Computer Understanding of Newspaper Stories, Ph.D. Dissertation, Research Report #116, Computer Science Department, Yale University, 1978.

[3] Charniak, E. Ms. Malaprop, A Language Comprehension Program, IJCAI-5, Cambridge, Mass., 1977.

[4] Wilensky, R. Understanding Goal-Based Stories, Ph.D. Dissertation, Research Report #140, Computer Science Department, Yale University, 1978.

[5] Schank, R.C. Reminding and Memory Organization: An Introduction to MOPs, Research Report #170, Computer Science Department, Yale University, 1979.

[6] Lehnert, W. et al. BORIS -- An Experiment in In-Depth Understanding of Narratives, Research Report #188, Computer Science Department, Yale University, 1981

[7] Schank, R.C. and Riesbeck, C.K. (eds) *Inside Computer Understanding*, Lawrence Erlbaum Associates, New Jersey, 1981.

[8] Montague, R. *Formal Philosophy: Selected Papers of Richard Montague*, edited and with an introduction by Richard Thomason, Yale University Press, New Haven, 1974.

[9] Dowty, D.R., Wall, R.E., and Peters, S. *Introduction to Montague Semantics*, D. Reidel Publishing Co., Dordrecht, Holland, 1981.

[10] Friedman, J. and Warren, D.S. A Parsing Method for Montague Grammars, *Linguistics and Philosophy*, 2, 1978, 347-372.

[11] Davidson, D. Truth and Meaning, *Synthese* 17, 1967, pp. 304-323.