American Journal of Computational Linguistics *Microfiche 29 : 46*

INTERPRETATION & INTEGRATION OF SENTENCES INTO A C-NET [01]

TH. R. HOFMANN

Groupe d'Etudes pour la Traduction Automatique
Université Scientifique et Médicale de Grenoble

ABSTRACT

A detailed proposal for the representation & manipulation of C-nets (suitable for computer programming) in interpreting pronominal reference. It is shown how this theory accounts for the disambiguation of pronominal reference, & the determination of focus & comment, more completely than any existing semantic or syntactic theory.

The theory of C-nets appears to be the most adequate linguistic theory for semantic analysis of content [my TREES]. We explore here the possibility of automating this analysis to aid in automatic translation. Translation involves analysis of content, without which it can only be a matching of lexical & syntactic structures between languages. Such matching has been shown inadequate by many researchers. Besides being necessary for automatic translation, an automated analysis of content is necessary for other tasks such as constructing general question-answering systems, voice-writers, automatic indexing & abstracting, propaganda measurement & explication, fallacy finding, & others.

In addition to being useful, & in the final analysis, necessary for practical problems in computer understanding or decoding of human language, this theory [see my DESCRIPTIONS] is shown to allow disambiguation of pronouns which are left as ambiguous by contemporary theories of semantics or syntax, but which in fact are not ambiguous. As will be noted but not explored, this theory also allows the consistent determination of the focus of a sentence from its context, i.e. not using position or prosodic features.

The theory of C-nets is in rapid evolution[02]. The version described here is chosen partly- because it matches the deep structures obtained by the present English grammar of the TAUM project at the Université de Montréal [TAUM]. In that system, a sentence is reduced to its deep structure by a Q-system[03] grammar. This deep structure would then be converted into a D-net, a temporary C-net, which is then interpreted lexically (i.e. lexical items are replaced by their concepts) & integrated into an overall C-net. This C-net represents the integrated meaning of that & previous sentences. In this theory, successful integration models the comprehension of the sentence, as described in my [C&C]. The interpretation & integration of D-nets are described here, with a demonstration of how later sentences are disambiguated in terms of the comprehension of the earlier ones.

In continuations (a) & (b) below, the syntactic structures are not significantly different. Yet both pronouns she refer unambiguously to different people; to Mary in (a) & to Susie in (b).

Mary told John that Susie was coming,
(a) but she said it softly.
(b) but she didn't arrive for an hour.

Any non-integrative semantic theory is forced to claim that these she's both refer to the same person, or that they are both ambiguous. Either alternative is wrong; these pronouns refer unambiguously to different people. A sentence often picks up meaning from its linguistic context. The same sentence may express different things in different contexts. The mechanism by which this can happen is explained below.

The only way to determine the correct referents of these pronouns is to take account of the relationships expressed by the verbs told, said, coming & arrive. An adequate way of doing this is to build a C-net for the 1st sentence & then integrate the continuations into it, as outlined in my [INTEGRATIVE].

A C-net is a directed, labelled graph with ordered arcs. We can represent a C-net by a list of items, 1 for each node. Each item then consists of: an index number for the node, a label which indicates its meaning, & an ordered list of the indices of the nodes it dominates. The particular index numbers assigned to the nodes are

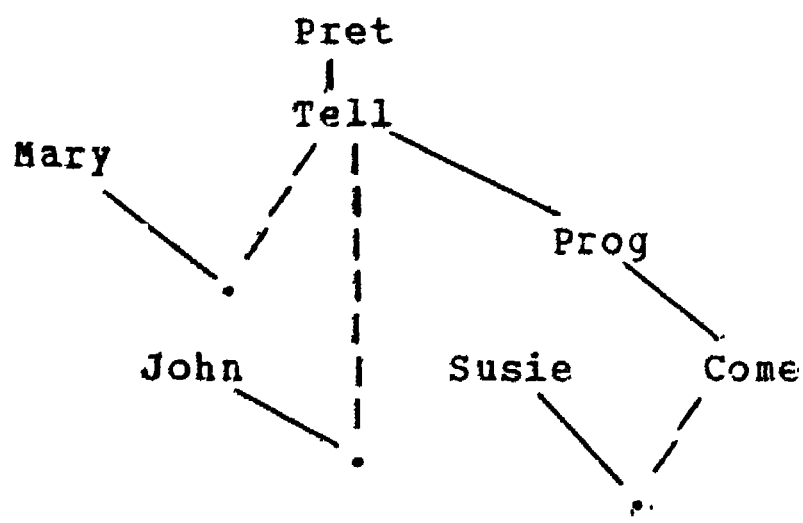
irrelevant, except that "1" & "2" are reserved to designate the speaker & addressee, respectively. Indeed, the index numbers are not part of the C-net. They are needed only to make a linear representation of a C-net, such as is needed for computer manipulation. Also it is customary to capitalize labels of nodes (i.e. semantic atoms), & for simplicity, we take Mary(x) as the meaning of Mary. Thus, a sentence;

Mary told John that Susie was coming.

has a C-net:

3	Mary(4)	8	Come(9)
4	P.	9	P.
5	Tell(4,6,7)	10	Susie(9)
6	P.	11	John(6)
7	Prog(8)	12	Pret(5) [05]

This list of nodes with connections to dominated nodes represents the graph below. In this & following graphs, domination is represented downward as in a dependency tree.



The "referential point" ("P." in the list notation &

"." in the graph notation) is roughly equivalent to the "individual variable" in the predicate calculus & the "referential index" in recent transformational studies. In denotational interpretation, points stand in correspondence with (i.e. they refer to) portions of the universe of interpretation. In the synatx of C-nets, they do not dominate any other predicate (node label).

The above representation[06] is subjected to lexical interpretation rules. These rules explicate concepts (the C-nets of lexical items, see my [C&C]) in terms of their semantic components. They replace single lexical items in a C-net by a small network of unanalysable predicates (semantic atoms). These rules operate successively on a C-net. Each lexical interpretation rule replaces the node(s) at the left by the set of nodes at the right. i, j, k & m are variables ranging over node numbers. A value is assigned to each variable during the operation of a rule, &·n-1 is defined as the highest node number in the C-net when the rule applies. All new nodes (n, n+1, &c) which are not dominated by that which dominates i (the node being replaced) are presuppositions[07].

Lexical Interpretation Rules

$$i\ \ulcorner \text{Susie} \urcorner (j) \\ \llcorner \text{Mary} \lrcorner \qquad \longrightarrow \begin{array}{ll} \ulcorner i & \ulcorner \text{Susie} \urcorner (j) \\ | & \llcorner \text{Mary} \lrcorner \\ | \ n & \text{Hum}(j) \\ \llcorner n+1 & \varphi(j) \end{array}$$

$$\ulcorner \text{Harry} \urcorner (j) \\ \llcorner \text{John} \lrcorner \qquad \longrightarrow \begin{array}{ll} \ulcorner i & \ulcorner \text{Harry} \urcorner (j) \\ | & \llcorner \text{John} \lrcorner \\ | \ n & \text{Hum}(j) \\ \llcorner n+1 & \delta(j) \end{array}$$

$$i\ \text{Tell}(j,k,m) \qquad \longrightarrow \begin{array}{ll} \ulcorner i & \partial(k,n)^{os} \\ | \ n & \text{Say}(j,m) \\ | n+1 & \text{Anim}(j) \\ \llcorner n+2 & \text{Hum}(j) \end{array}$$

$$i\ \text{Come}(j) \qquad \longrightarrow \begin{array}{ll} \ulcorner i & \beta(n) \\ | \ n & \partial(n+1,j) \\ | n+1 & \text{P.} \\ | n+2 & \text{Place}(n+1) \\ \llcorner n+3 & \partial(n+1,\ 1) \end{array}$$

The effect of the 1st 2 rules is to add the features Hum(x) & φ(x) or δ(x) onto the points dominated by the proper names, Susie, Mary, Harry, John. They express the linguistic fact that these names are human names (unlike e.g. Fido), & are used for females & males respectively. In the 3rd rule, the atoms Hum(x) & Anim(x) are included in the meaning of Tell(x,y,z) to restrict collocational possibilities. They thus describe selectional restrictions, & will sometimes disambiguate an otherwise ambiguous sentence.[09] Because the predicates on any point must be non-contradictory, the use of a non-human subject for tell Tell(j,k,l) causes a contradiction around node j. The predicate Place(x) occurs in the 4th rule for the same reason.

There is also a general set of rules by which

duplicate nodes are removed;

Duplicate-Reduction Rule Schema

```
i Pα()     ┐ --->   i Pα()
j Pα()     ┘       & j := i

i Pα(k)    ┐ --->   i Pα(k)
j Pα(k)    ┘       & j := i

i Pα(k,l)  ┐ --->   i Pα(k,l)
j Pα(k,l)  ┘       & j := i

              &c
```

where j := i effects the replacement of all occurrences of j in the C-net by i, & Pα is a variable ranging over node labels.

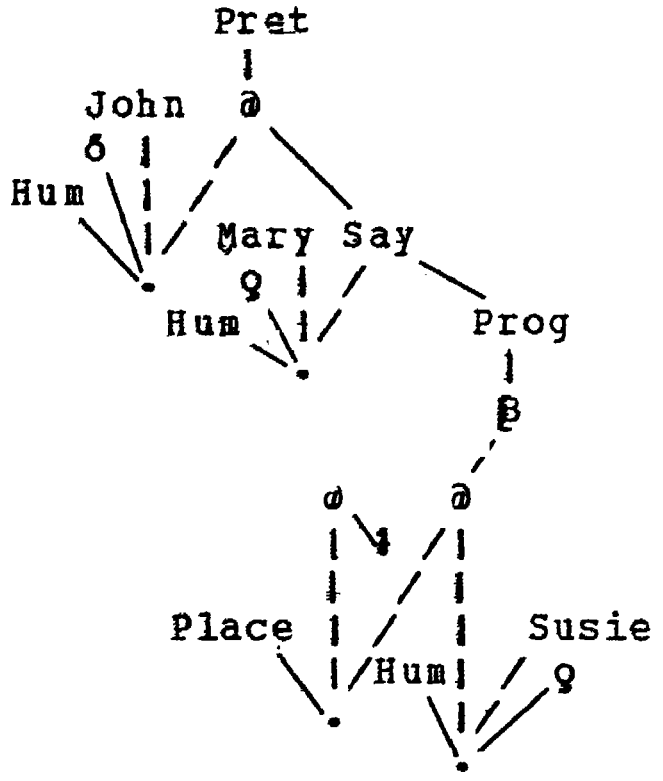
Most generally, the lexical interpretation rules may be applied in any order & as many times as needed; then this duplicate-reduction rule removes any duplicate nodes introduced by the lexical rules. To our present knowledge, however, it appears that the lexical rules can be constrained to operate only once on a C-net (at all applicable places simulataneously) & this constraint will likely require some ordering between the rules[10]. My working assumptions are that lexical rules can be ordered so that each can apply only once (everywhere) & that the duplicate-reduction rule operates last.

By this process, the 1st graph above for Mary told John that Susie was coming is converted into its final form below. Since there are no previous sentences, there is no integration to be done. This D-net is therefore also the C-

net for the discourse up to this poi

3	Mary (4)	14	Q (4)
4	P.	15	Say (4, 7)
5	∂ (6., 15)	16	∂ (17, 9)
6	P.	17	P.
7	Prog (8)	18	Place (17)
8	B (16)	19	∂ (17, 1) [11]
9	P.	20	Hum (9)
10	Susie (9)	21	Q (9)
11	John (6)	22	Hum (6)
12	Pret (5)	23	δ (6)
13	Hum (4)		

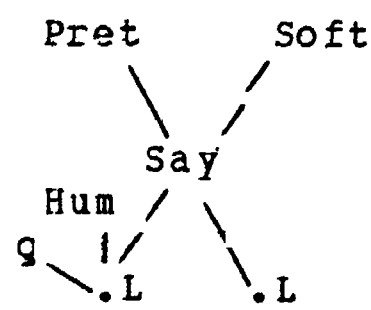
or the graph;



The 1st continuation, (a): <u>but she said it softly</u> yields the D-net below after the lexical interpretation rules & the duplication reduction rule have applied. A D-net is analogous to a deep structure. It represents the meaning of a single sentence prior to integration, i.e, in isolation from the text it is a part of. It is integrated into the existing C-net to form a new C-net.

24	Soft(25)	28	Hum(27)
25	Say(27,26)	29	ϙ(27)
26	P. L	30	Pret(25)
27	P. L		

or;



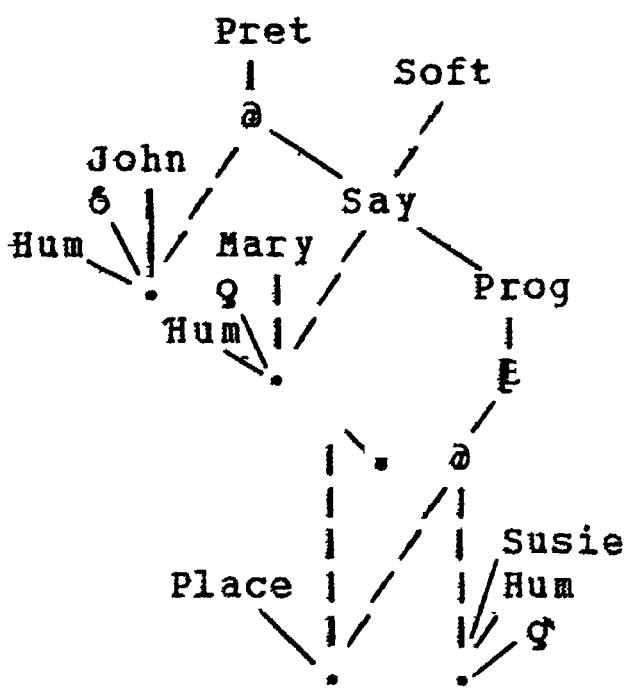
The points in a D-net may be annotated by an "L" for definite reference. This "L" is peculiar to D-nets, & indicates that the addressee should be able to find something it refers to, either in the C-net already existing, or in the context. When integrated, the "L" disappears because the L-point is either identified with a point in the C-net, or it refers to some extra-cognitive structure. This "L" derives from definite expressions such as the, this, he, &c.

This D-net is integrated in the only possible way, namely with

30	:=	12	
27	:=	4	(i.e. she = Mary)
26	:=	7	(i.e. it = that Susie is coming)
28	:=	13	
29	:=	14	
25	:=	15	(i.e. say ≡ tell)
31	:=	16	

With the duplicates removed, the only addition is 24 Soft(25)[12]. These remaining additions thus obtained necessarily include the focus (comment) & new

presuppositions. The "meaning" of this continuation in a loose sense (what information does it transmit which is not already known) is simply the predicate Soft(15). All the rest of the sentence is redundant in the sense that it does not contain anything new. It was necessary, however, to allow the listener to determine what is soft.



The referents of the pronouns are determined as a by-product of the integration. Because this continuation cannot be integrated in any other way, she must refer to Mary, a feminine antecedent which is not the closest one.

There are several possible strategies for integrating a D-net, D, which results from a continuation sentence into the C-net, C, resulting from accretions from all the previous sentences. Basically, some of the nodes in D are found in C, & the D-net is traced out in C. In general, the D-net will contain some nodes not in C, & certain types[13] of the nodes

in D may be missing in the C-net. The 1st D nodes to be sought in C should have a high information content to make this procedure more effective & for this reason we have chosen to start with the highest nodes in D: those nodes in D which are not dominated by any other nodes. In the example above, nodes 29, 28, 30, 24, & 31 are all without domination, but 29, 28, & 31 are all directly predicating on points, which leaves 24 & 30 as highest.. Nodes with these labels, Soft(▼) & Pret(▼), are sought in C. An equivalent for 24 is not found, but 30 matches 21. Then, following domination lines downward from 30 & 21, everything matches except for 5 @(6,15). @(x,▼) is one of the nodes which can be skipped in integration[14]. If the highest nodes do not yield an integration, the next highest are used until integration is possible. Integration is accomplished by a set of node equivalences such as explained above. Once these are effected, the duplicate reduction rule will remove all the nodes deriving from D which were already in C. The only nodes of D remaining are those which were not already in C.

The pronoun she in this continuation is referentially ambiguous by any method of analysis which handles sentences in isolation. It is rendered unambiguous, but wrongly so, if it is assumed to refer to the nearest preceding feminine antecedent. No English speaker can mistake that its referent is Mary; in reality this she is not ambiguous. But the only way of obtaining the correct referent for it is to utilize

the information contained in the verbs <u>say</u> & <u>tell</u>. This cannot be done simply by the similarity in the meanings of these verbs. In a different continuation,

> she wouldn't be able to tell him anything until arriving,

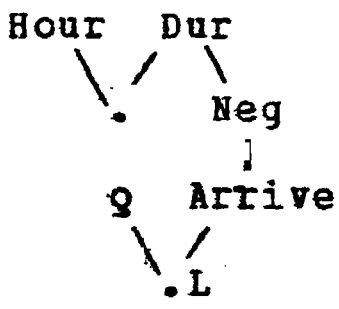
the <u>she</u> no longer refers to Mary but to Susie, even though the main verb of the continuation is identical to the verb of the initial sentence.

In that continuation, or in the less complex one (b):

> she didn't arrive for an hour,

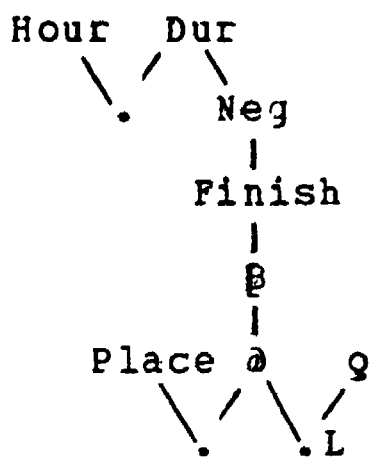
the pronoun refers to Susie, & not to Mary, primarily because of the verb <u>arrive</u> This continuation results in a C-net;

24	Hour(25)	28	Arrive(29)
25	P.	29	P. L
26	During(25,27)	30	ϙ(29)
27	Neg(28)		



This is converted into:

24	Hour(25)	27	Neg(28)	31	ϐ(32)
25	P.	28	Finish(31)	32	ϑ(33,29)
26	Dur(25,27)	29	P. L	33	P.
		30	ϙ(29)	34	Place(33)

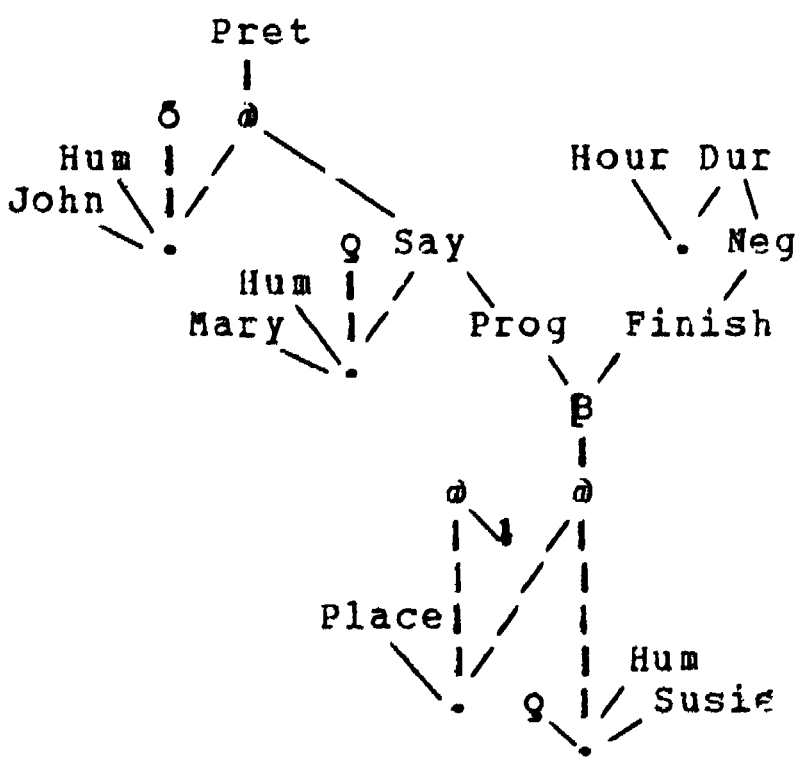


by the lexical conversion rules

```
                                  ┌ i   Finish(n)
                                  | n   β(n+1)
i Arrive(j)          ---->        |n+1  ∂(n+2,j)
                                  |n+2  P.
                                  └n+3  Place(n+2)

i ∂(k,m)      ┐ ---->  [same as above]
m Arrive(j)   ┘        & n+2 := k
```

It is integrated unambiguously into the C-net as:



with the equations,

29 := 9 (i.e. she = Susie)
33 := 17
30 := 21
31 := 8 (i.e. arfive = come)
32 := 16
34 := 18

When the duplicates are removed, the addition is that Susie's coming was not finished for an hour, the italicized portion being the comment.

In the more complex continuation mentioned above, she wouldn't be able to tell him anything until arriving, the she refers to Susie, even though the subject & the 1st verb significant for disambiguation point to Mary as the referent. This desirable result follows from the fact that the subject of arrive is the same point as the subject of tell. With the integration strategy used here, both Tell(x,y,▼) & Arrive(x) are on the same level in the D-net, but the sub-network dominated by Arrive(x) is a perfect match, while the sub-network around Tell(x,y,▼) has no match at all (because of Tell's different 3rd actant). Hence the she refers unambiguously to Susie. Interesting enough, if we insert the word more, she wouldn't be able to tell him anything more until she came, the 1st she refers to Mary, & the 2nd to Susie. The cause of this is that the interpretation of more inserts an extra Tell(x,y,▼) into the D-net, which will match the Tell(x,y,▼) in the C-net from the 1st sentence, but the only arrival remains Susie's.

We have given a detailed description of the process by which different continuations of the same sentence have their subject pronouns interpreted as a function of their main verbs. The integration process, which places interpretations on pronouns, also interprets definite occurrences of unmodified & non-specific nouns as repetitions of previously mentioned more specific (or more modified) nouns, fills in deleted nominals on the basis of prior context & interprets generic or unmodified verbs as repetitions of more specific verbs previously mentioned. It must also handle verb phrase deletion, occurrences of do so & gapping. The theory of which this is based is described in my [C&C] & [DESCRIPTIONS & its relation to other semantic theories is discussed in my [TREES], [REPRESENTATION], & in Paillet [PROBLEMES].

The essential contribution of this paper has been a formal description of the process of integration. That process is central to any integrative semantic theory, of which the C-net theory is only 1 (see. my [APPROACH]). The formalization of integration presented here is undoubtedly wrong in some aspects, & requires further research for improvement & verification. As stated here, it is apparently adequate for most cases of pronouns (but see note 11, & excluding deictic uses of pronouns & the pariphrastic it).

The integration of the D-net derived from a sentence is a model of the user's / comprehension of the sentence,

without, of course, modelling his evaluation of the truth of the sentence or the motive behind its use. These are usually dependent on the universe of interpretation & observation of the speaker. Integration provides a possible next area for research toward fully automated translation because it promises to provide a comprehensive description of the content of the paragraph & the contribution each sentenc makes to that content.

Presently conceived as an adjunct to an automated translation system, (it provides full information as to deletions, anaphora of definite articles & pronouns, & interpretations of words), the C-net could provide the total input into a target language rhetoric. This system will make "versions", translations wherein the content, but not necessarily the words, syntactic constructions or even the order of exposition are preserved.

As opposed to a translation, a "version" cannot be made with present theory, because that reqires an adequate theory of rhetoric; how the material for a sentence is selected out of a complex C-net, what constraints there are for selection of topics, comments, focuses, &c. Much of this is unknown at present, though the next several years may show a great expansion of our knowledge. (See I. Bellert for a direct attack on this problem.)

Ordinary translations is, however, a matching of

syntactic & lexical structures as far as possible, without modifying the meaning. C-nets provide a means to do this. With an adequate representation of meaning, syntactic & lexical matching can be done. The result can then be tested for changes in meaning by building a C-net from the target language expression for comparison with the original C-net. Modification of the target expression can then be made to make the input & output C-nets match to any desired degree of accuracy. A simpler means for good translation is found in my [TRANSLATION].

- - - N O T E S - - -

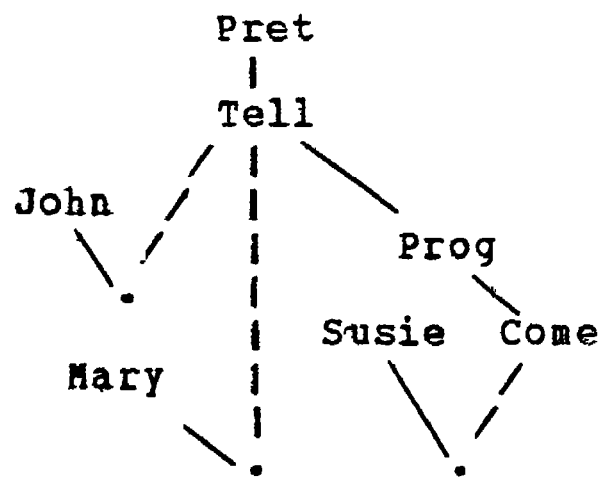
01 This is an extended version of an article "Interpretation & Integration of Sentences into a C-network" which was written at Groupe de recherche sur la traduction automatique, Université de Montréal in the summer of 1971 & appeared in Kittredge [ETUDES]. The terminology & notation has also been revised to be consistent with more recent work on C-nets.

02 Compare, e.g. my [TREES] & [JUDGING].

03 The Q-system is a high level programming language for string manipulation. See A. Colmerauer, 'Les systèmes-Q ou 1 formalisme pour analyser & synthétiser des phrases sur l'ordinateur', in TAUM '71 (1971) Groupe de recherche sur la traduction auotmatique, Université de Montréal.

05 Prog(▼) & Pret(▼) are abbreviations for the meanings of the English formatives for the progressive aspect, & the preterite or past tense.

06 In contrast to this, John told Mary that Susie was coming has a C-net,



The difference in the list notation is that node 5 is Tell(6,4,7) instead of Tell(4,6,7).

07 See discussion in my [JUDGING].

08 @(x,y) stands for an abstract locative atom of meaning which is not realized exactly by any word in English. French à is closer to @(x,y). Recently discovered evidence leads to the belief that Tell(i,j,k) has been incorrectly analysed here, & that this @(x,▼) does not occur at all. A better analysis is "to cause him to come to know it by saying it".

```
                                    ⌜ i   ==>(n,n+1)
i Tell(j,k,m)          --->         | n   Say(j,m)
                                    |n+1  B(n+2)
                                    ⌞n+2  Know(k,n)
```

The point of this paper, & the demonstrations remain unchanged under this modification, although most of the C-nets are revised.

09 The TAUM syntax presently attempts this sort of disambiguation by checking the appropriate NP for having the necessary features, as proposed in Chomsky [ASPECTS]. McCawley [POLE] has argued (p. 132f) that this cannot be done generally by syntactic means, because the critical words may be indefinitely distant, c.f.:

I will eat whatever Maria bakes.
*I will eat whatever Maria believes.
I will eat whatever Maria believes Tom to have baked.
*I will eat whatever Maria believes Tom to have dreamed.

10 That is, that the rules can be ordered extrinsically & applied in sequence; after 1 rule has been applied throughout the network, it cannot be reapplied. This will likely require the linguistician to arrange rules in an extrensic order as some rules need the output of others in an extrensic order to work.

11 The point '1' introduced from the analysis of come could now be re-interpreted as the subject (4) or indirect object (6) of Say(4,7). It is not done here for clarity of exposition. With the revision mentioned in note 08, there is a simple generalization to identify the points which '1' may be re-interpreted as. There is a list of predicates (including say, know &c) whose 1st actants may replace a '1' dominated by their 2nd actants.

12 Note that the information by which but contrasts with and is lost in integration, consistent with the principle of integrative semantics that only the information transmitted, not its order nor the speaker's reaction to it, is represented. Conjunctions or successive sentences are simply integrated together: but indicates some surprise or perhaps inappropriateness. If it turns out desirable or even necessary to include such emotive information, semantic atoms for such may be defined & will be predicated on the element which is focussed, by a slightly more sophisticated system. Here, it would result in an additional node Surprise(24). This would allow the reconstruction of the 2-clause syntactic form & the use of but from the integrated network.

[13] It is not yet known which elements may be missing from a D-net without blocking an integration. I assume that there is a finite list of such elements, including the performative atoms marking the type of illocutionary act (statement, question, &c).

[14] i ∂(j,k) can be substituted for by i=j during integration. See note [13].

- - - REFERENCES - - -

Kittredge, R. (Ed)
[ETUDES] Etudes en Linguistique Appliquée (1971) Groupe de recherche sur la traduction automatique, Université de Montréal.

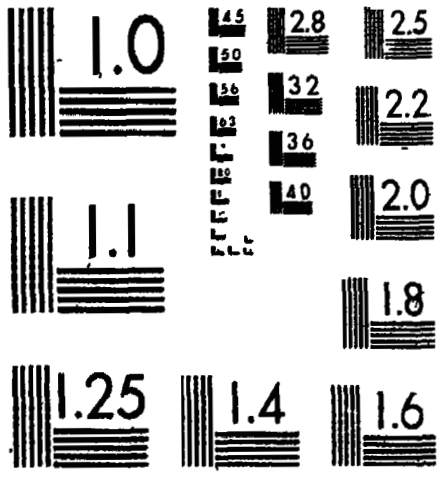
Chomsky, N.
[ASPECTS] Aspects of the theory of syntax (1965) Cambridge (M.I.T. Press).

Hofmann, Th. R.
[APPROACH] 'An integrative semantic approach to intersentential phenomena' (1975) Proceedings of the 1st National Congress on the Application of Computers & Mathematical Models to Linguistics, Sofia.
[C&C] 'Comprehension & concepts in semantics' (1972) unpublished.
[INTEGRATIVE] 'Integrative semantics' (1973) Cahiers Linguistiques 2: 19-38.
[JUDGING] 'Verbs of judging: an exercise in further semantic description' (1973) Cahiers linguistiques d'Ottawa 3: 59-67.
[REPRESENTATION] 'Semantic representations & the theory of language' (1973) Language Sciences 28: 22-24.
[TRANSLATION] 'C-nets in translation of natural language' in Kittredge [ETUDES].
[TREES] 'Meaning doesn't grow on trees' (1973) Language Sciences 26: 1-4.

McCawley, J.
[ROLE] 'The role of semantics in grammar' (1968) in Bach & Harms (Ed) Universals in linguistic theory, New York (Holt).

Paillet, J-P.
[PREREQUIS] 'Prérequis pour l'analyse sémantique' (1973) Cahiers Linguistiques 2: 1-18.
[PROBLEMES] 'Problèmes de notation pour la sémantique' (1974) Langages 32: 27-69.

END



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A