

COPING WITH DYNAMIC SYNTACTIC STRATEGIES: AN EXPERIMENTAL ENVIRONMENT FOR AN EXPERIMENTAL PARSER

Oliviero Stock

I.P. - Consiglio Nazionale delle Ricerche
Via dei Monti Tiburtini 509
00157 Roma, Italy

ABSTRACT

An environment built around WEDNESDAY 2, a chart based parser is introduced. The environment is in particular oriented towards exploring dynamic aspects of parsing. It includes a number of specialized tools that consent an easy, graphics-based interaction with the parser. It is shown in particular how a combination of the characteristics of the parser (based on the lexicon and on dynamic unification) and of the environment allow a nonspecialized user to explore heuristics that may alter the basic control of the system. In this way, for instance, a psycholinguist may explore ideas on human parsing strategies, or a "language engineer" may find useful heuristics for parsing within a particular application.

1. Introduction

Computer-based environments for the linguist are conceived as sophisticated workbenches, built on AI workstations around a specific parser, where the linguist can try out his/her ideas about a grammar for a certain natural language. In doing so, he/she can take advantage of rich and easy-to-use graphic interfaces that "know" about linguistics. Of course behind all this lies the idea that cooperation with linguists will provide better results in NLP. To substantiate this assumption it may be recalled that some of the most interesting recent ideas on syntax have been developed by means of joint contributions from linguists and computational linguists. Lexical-Functional Grammar [Bresnan & Kaplan 1982], GPSG [Gazdar 1981], Functional Grammar [Kay 1979], DCG [Pereira & Warren 1980], TAG [Joshi & Levy 1982] are some of these ideas.

Instances of the tools introduced above are the LFG environment, which was probably the first of its kind, an

environment built by Ron Kaplan for Lexical-Functional Grammars, DPATR, built by Lauri Karttunen and conceived as an environment that would suit linguists of a number of different schools all committed to a view of parsing as a process that makes use of an unification algorithm.

We have developed an environment with a somewhat different purpose. Besides a number of tools for entering data in graphic mode and inspecting resulting structures, it provides a means for experimenting with strategies in the course of the parsing process. We think that this can be a valuable tool for gaining insight in the cognitive aspects of language processing as well as for tailoring the behaviour of the processor when used with a particular (sub)language.

In this way an attempt can be made to answer basic questions when following a nondeterministic approach: what heuristics to apply when facing a certain choice point, what to do when facing a failure point, i.e. which of the pending processes to activate, taking account of information resulting from the failure?

Of course this kind of environment makes sense only because the parser it works on has some characteristics that make it a psychologically interesting realization.

2. Motivation of the parser

We shall classify psychologically motivated parsers in three main categories. First: those that embody a strong claim on the specification of the general control structure of the human parsing mechanism. The authors usually consider the level of basic control of the system as the level they are simulating and are not concerned with more particular heuristics. An instance of this class of parsers is Marcus's parser [Marcus 1979], based on the claim that, basically, parsing is a deterministic process: only sentences that we perceive as "surprising" (the so called garden paths) actually imply backtracking.

Connectionist parsers are also instances of this category. The second category refers to general linguistic performance notions such as the "Lexical Preference Principle" and the "Final Argument Principle" [Fodor, Bresnan and Kaplan 1982]. It includes theories of processing like the one expressed by Wanner and Maratsos for ATNs in the mid Seventies. In this category the arguments are at the level of general structural preference analysis. A third category tends to consider at every moment of the parsing process, the full complexity of the data and the hypothesized partial internal representation of the sentence, including, at least in principle, interaction with knowledge of the world, aspects of memory, and particular task-oriented behaviour.

Worth mentioning here is Church and Patil's [1982] work which attempts to put order in the chaos of complexity and "computational load".

Our parser lies between the second and the third of the above categories. The parser is seen as a nondeterministic apparatus that disambiguates and gives a "shallow" interpretation and an incremental functional representation of each processed fragment of the sentence. The state of the parser is supposed to be cognitively meaningful at every moment of the process.

Furthermore, in particular, we are concerned with aspects of flexible word ordering. This phenomenon is specially relevant in Italian, where, for declarative sentences, Subject-Verb-Object is only the most likely order - the other five permutations of Subject Verb and Object may occur as well. We shall briefly describe the parser and its environment and, by way of example, illustrate its behaviour in analyzing "oscillating" sentences, i.e. sentences in which one first perceives a fragment in one way, then changes one's mind and takes it in a different way, then, as further input comes in, going back to the previous pattern (and possibly continuing like this till the end of the sentence).

3. The parser

WEDNESDAY 2 [Stock 1986] is a parser based on linguistic knowledge distributed fundamentally through the lexicon. A word reading includes:

- a semantic representation of the word, in the form of a semantic net shred;
- static syntactic information, including the category, features, indication of linguistic functions that are bound to particular nodes in the net. One particular

specification is the Main node, head of the syntactic constituent the word occurs in;

- dynamic syntactic information, including impulses to connect pieces of semantic information, guided by syntactic constraints. Impulses look for "fillers" on a given search space (usually a substring). They have alternatives, (for instance the word TELL has an impulse to merge its object node with the "main" of either an NP or a subordinate clause). An alternative includes: a contextual condition of applicability, a category, features, marking, side effects (through which, for example, coreference between subject of a subordinate clause and a function of the main clause can be indicated). Impulses may also be directed to a different search space than the normal one (see below);

- measures of likelihood. These are measures that are used for deriving an overall measure of likelihood of a partial analysis. Measures are included for the likelihood of that particular reading of the word and for aspects attached to an impulse: a) for one particular alternative b) for the relative position the filler c) for the overall necessity of finding a filler.

- a characterization of idioms involving that word. (For a description of the part of the parser that deals with the interpretation of flexible idioms see [Stock 1987]).

The only other data are in the form of simple (non augmented) transition networks that only provide restrictions on search spaces where impulses can look for fillers. In more traditional words it deals with the distribution of constituents. A distinguishing symbol, \$EXP, indicates that only the occurrence of something expected by preceding words (i.e. for which an impulse was set up) will allow the transition.

The parser is based on of the idea of chart parsing [Kay 1980, Kaplan 1973] [see Stock 1986]. What is relevant here is the fact that "edges" correspond to search spaces. Edges are complex data structures provided with a rich amount of information including a semantic interpretation of the fragment, syntactic data, pending impulses, an overall measure of likelihood, etc. Data on an edge are "unified" dynamically as indicated below

An agenda is provided which includes four kinds of tasks: *lexical tasks*, *traversal tasks*, *insertion tasks*, *virtual tasks*. A lexical task specifies a possible reading of a word to be inserted in the chart. A traversal task specifies an active edge and an inactive edge that can extend it. An insertion task specifies a nondeterministic unification act, and a virtual task involves extension of an edge to include an inactive edge far away in the string (used for long distance dependencies).

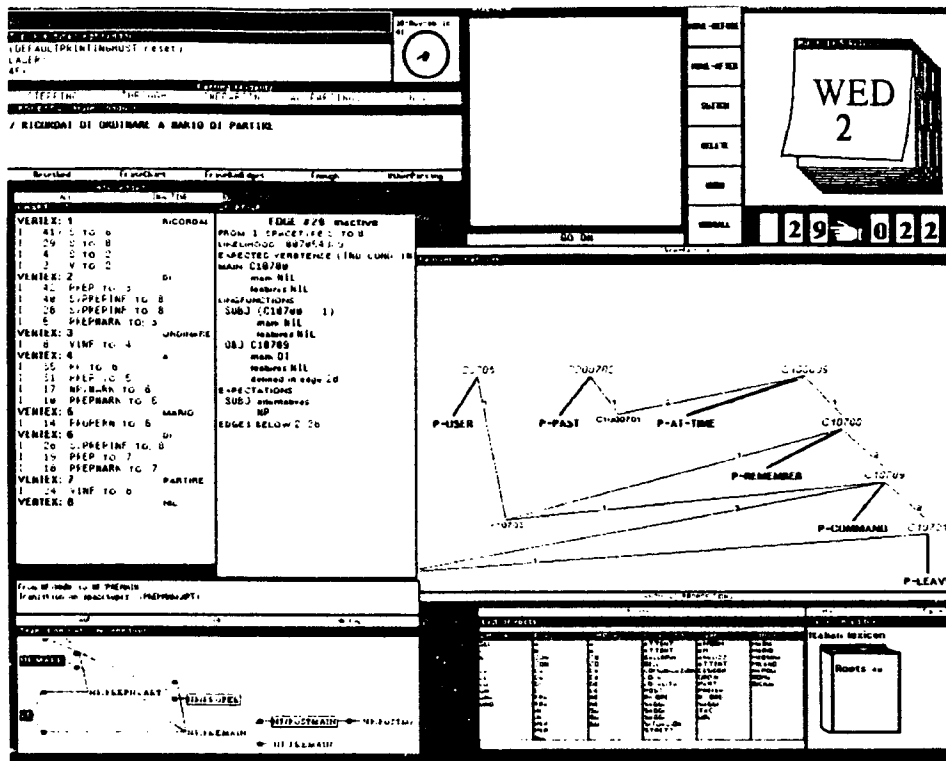


Fig. 1

The parser works asymmetrically with respect to the "arrival" of the Main node: before the Main node arrives, an extension of an edge has almost no effect. On the arrival of the Main, all the present impulses are "unleashed" and must find satisfaction. If all this does not happen then the new edge supposedly to be added to the chart is not added: the situation is recognized as a failure. After the arrival of the Main, each new head must find an impulse to merge with, and each incoming impulse must find satisfaction. Again, if all this does not happen, the new edge will not be added to the chart..

4. Overview of the environment

WEDNESDAY 2 and its environment are implemented on a Xerox Lisp Machine. The environment is composed of a series of specialized tools, each one based on one or more windows (fig 1).

Using a mouse the user selects a desired behaviour from menus attached to the windows. We have the following windows:

- the main WEDNESDAY 2 window, in which the sentence is entered. Menus attached to this window specify different modalities (including "through" and "stepping", "all parsings" or "one parsing") and a number of facilities;
- a window where one can view, enter and modify transition networks graphically (fig. 2).
- a window where one can view, enter and modify the lexicon. As a word entry is a complex object for WEDNESDAY 2, entering a new word can be greatly facilitated by a set of subwindows, each specialized in one aspect of the word, "knowing" how it may be and facilitating editing. The lexicon is a lexicon of roots: a morphological analyzer and a lexicon manager are integrated in the system. Let us briefly describe this point. A lexicalist theory such as ours requires that a large quantity of information be included in the lexicon. This information has different origins: some comes from the root and some from the affixes. All the information must be put into a coherent data structure, through a particularly constrained unification based process.

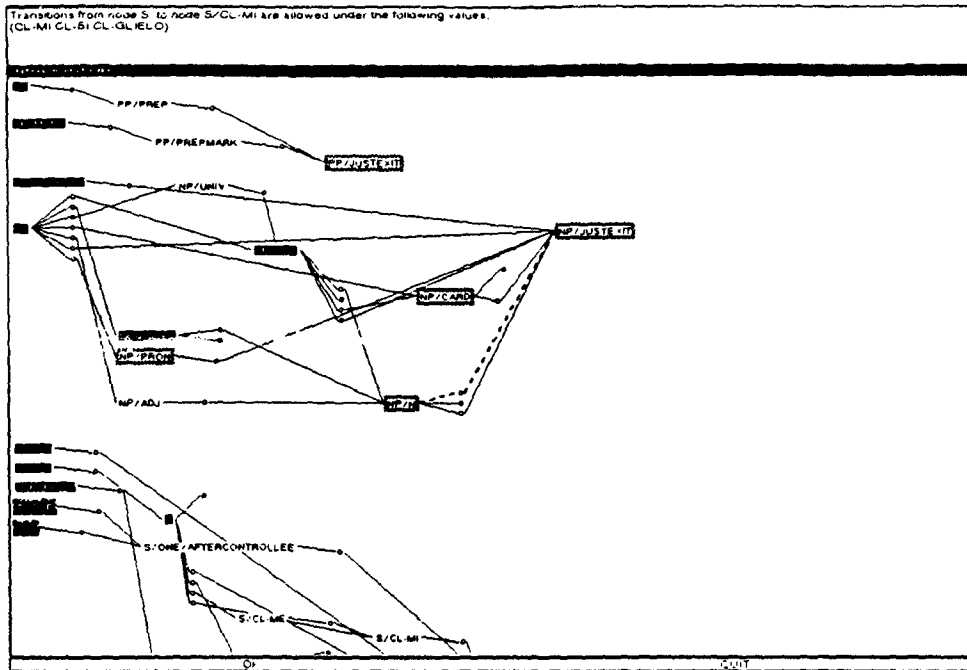


Fig. 2

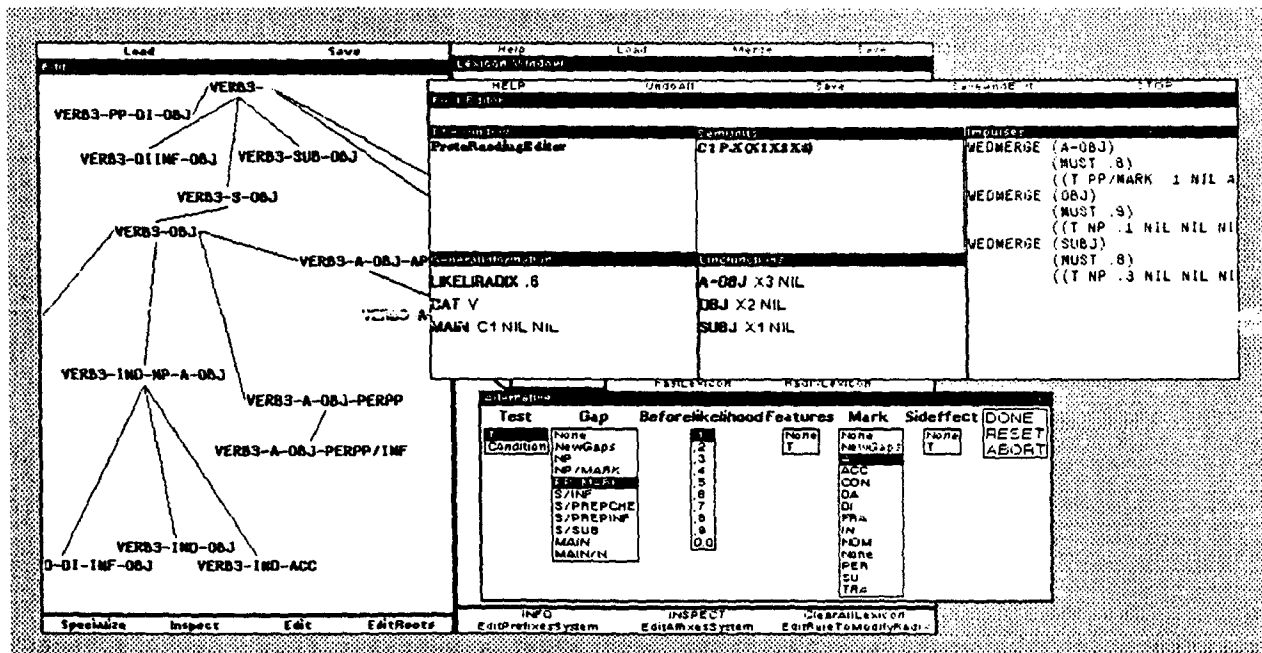


Fig. 3

Furthermore we must emphasize the fact that, just as in LFG, phenomena such as passivization are treated in

the lexicon (the Subject and Object functions and the related impulses attached to the active form are

rearranged). This is something that the morphological analyzer must deal with. The internal behaviour of the morphological analyzer is beyond the scope of the present paper. We shall instead briefly discuss the lexicon manager, the role of which will be emphasized here.

The lexicon manager deals with the complex process of entering data, maintaining, and preprocessing the lexicon. One notable aspect is that we have arranged the lexicon on a hierarchical basis according to inheritance, so that properties of a particular word can be inherited from a word class and a word class can inherit aspects from another class. One consequence of this is that we can introduce a graphic aspect (fig 3) and the user can browse through the lattice (the lexicon appears as a tree of classes where one has specialized editors at each level). What is even more relevant is the fact that one can factorize knowledge that is in the lexicon, so that if one particular phenomenon needs to be treated differently, the change of information is immediate for the words concerned. Of course this means also that there is a space gain: the same information needs not to be duplicated: complete word data are reconstructed when required.

There is also a modality by which one can enter the syntactic aspects of a word through examples, *a la TEAM* [Grosz 1984]. The results are less precise, but may be useful in a more application-oriented use of the environment.

- a window showing the present configuration of the chart;

- a window that permits zooming into one edge, showing several aspects of the edge, including: its structural aspect, its likelihood, the functional aspect, the specification of unrealized impulses etc.

- a window displaying in graphic form the semantic interpretation of an edge as a semantic net, or, if one prefers so (this is usually the case when the net is too complex) in logic format;

- a window where one can manipulate the agenda (fig 4). Attached to this window we have a menu including a set of functionalities that the tasks included in the agenda to be manipulated: MOVE BEFORE, MOVE AFTER, DELETE, SWITCH, UNDO etc. One just points to the two particular tasks one wishes to operate on with the mouse and then to the menu entry. In this way the desired effect is obtained. The effect corresponds to applying a different scheduling function: the tasks will be picked up in the order here prescribed by hand. This tool, when the parser is in the "stepping" modality,

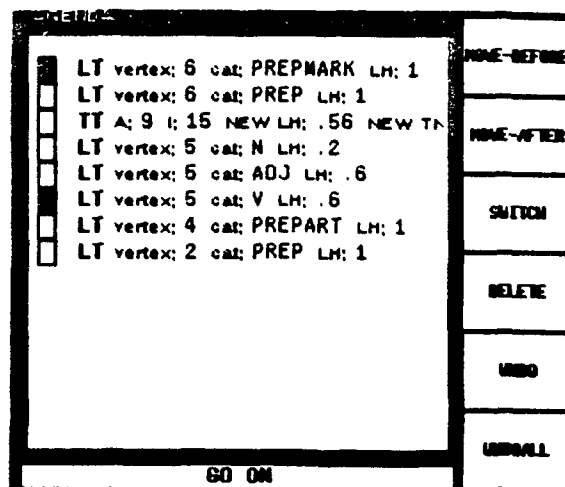


Fig. 4

provides a very easy way of altering the default behaviour of the system and of trying out new strategies. This modality of scheduling by hand is complemented by a series of counters that provide control over the penetrance of these strategies. (The penetrance of a nondeterministic algorithm is the ratio between the steps that lead to the solution and the steps that are carried out as a whole in trying to obtain the solution. Of course this measure is included between 0 and 1.)

Dynamically, one tries to find sensible strategies, by interacting with the agenda. When, after experimenting formalizable heuristics have been tried out, they can be introduced permanently into the system through a given specialized function. This is the only place where some knowledge of LISP and of the internal structure of WEDNESAY 2 is required.

5. An example of exploration: oscillating sentences

We shall now briefly discuss a processing example that we have been able to understand using the environment mentioned above. The following example is a good instance of flexibility and parsing problems present in Italian:

a Napoli preferisco Roma a Milano.

The complete sentence reads "while in Naples I prefer Rome to Milan". The problem arises during the parsing process with the fact that the "to" argument of "prefer" in Italian may occur before the verb, and the locative preposition "in" is *a*, the same word as the marking preposition corresponding to "to".

The reader/hearer first takes *a Napoli* as an adverbial location, then, as the verb *preferisco* is perceived, *a Napoli* is clearly reinterpreted as an argument of the verb, (with a sense of surprise). As the sentence proceeds after the object *Roma*, the new word *a* causes things to change again and we go back with a sense of surprise to the first hypothesis.

The following things should be noted: - when this second reconsideration takes place, we feel the surprise, but this does not cause us to reconsider the sentence, we only go back adding more to an hypothesis that we were already working at; -the surprise seems to be caused not by a heavy computational load, but by a sudden readjustment of the weights of the hypotheses. In a sense it is a matter of memory, rather than computation.

We have been in a position to get WEDNESDAY 2 to perform naturally in such situations, taking advantage of the environment. The following simple heuristics were found: a) try solutions that satisfy the impulses (if there are alternatives consider likelihoods); b) maintain viscosity (prefer the path you are already following); and c) follow the alternative that yields the edge with the greatest likelihood, among edges of comparable lengths.

The likelihood of an edge depends on: 1) the likelihood of the "included" edges; 2) the level of obligatoriness of the filled impulses; 3) the likelihood of a particular relative position of an argument in the string; 4) the likelihood of that transition in the network, given the previous transition.

The critical points in the sentence are the following (note that we distinguish between a PP and a "marked NP" possible argument of a verb, where the preposition has no semantics associated:

i) At the beginning: only the PP edge is expanded, (not the one including a "marked NP", because of static preference for the former expressed in the lexicon and in the transition network.

ii) After the verb is detected: on the one hand there is an edge that, if extended, would not satisfy an obligatory impulse, on the other hand, one that would possibly satisfy one. The "marked NP" alternative is chosen because of a) of the above heuristics.

iii) After the object *Roma*: when the preposition *a* comes in, the edge that may extend the sentence with a PP on the one hand, and on the other hand a cycling active edge that is a promising satisfaction for an impulse are compared. Since this relative position of the argument is so favourable for the particular verb *preferisco* (.9 to .1 for this position compared to the antecedent one), the

parser proceeds with the alternative view, taking *a Napoli*, as a modifier. So it goes on, after reentering that working hypothesis. The object is already there, analyzed for the other reading and does not need to be reanalyzed. So *a Milano* is taken as the filler for the impulse and the analysis is concluded properly.

It should be noted that the Final Argument Principle [Fodor, Kaplan and Bresnan 1982] does not work with the flexibility characteristic of Italian. (The principle would cause the reading "I prefer [Rome [in Milan]] to Naples" to be chosen at point iii) above).

Conclusions

We have introduced an environment built around WEDNESDAY 2, a nondeterministic parser, oriented towards experimenting with dynamic strategies. The combination of interesting theories and such tools realizes both meanings of the word "experimental": 1) something that implements new ideas in a prototype; 2) something built for the sake of making experiments. We think that this approach, possibly integrated with experiments in psycholinguistics, can help increase our understanding of parsing.

Acknowledgements

Federico Cecconi's help in the graphic aspects and lexicon management has been precious.

References

- Church, K. & Patil, R. Coping with syntactic ambiguity or how to put the block in the box on the table. *American Journal of Computational Linguistics*, 8; 139-149 (1982)
- Ferrari, G. & Stock, O. Strategy selection for an ATN syntactic parser. *Proceedings of the 18th Meeting of the Association for Computational Linguistics*, Philadelphia (1980)
- Ford, M., Bresnan, J. & Kaplan, R. A competence based theory of syntactic closure. In Bresnan, J., Ed. *The Mental Representation of Grammatical Relations*. The MIT Press, Cambridge, (1982)
- Gazdar, G. Phrase structure grammar. In Jacobson and Pullman (Eds.), *The Nature of Syntactic Representation*. Dordrecht: Reidel (1981)

Grosz, B. TEAM, a transportable natural language interface system. In *Proceedings of the Conference on Applied Natural Language Processing*, Santa Monica (1983)

Joshi, A., & Levy, L. Phrase structure trees bear more fruits than you would have thought. *American Journal of Computational Linguistics*, 8; 1-11 (1982)

Kaplan, R. A general syntactic processor. In Rustin, R. (Ed.), *Natural Language Processing*. Englewood Cliffs, N.J.: Prentice-Hall (1973)

Kaplan, R. & Bresnan, J. Lexical-Functional Grammar: a formal system for grammatical representation. In Bresnan, J., Ed. *The Mental Representation of Grammatical Relations*. The MIT Press, Cambridge, 173-281 (1982)

Kay, M. Algorithm Schemata and Data Structures in Syntactic Processing. Xerox, Palo Alto Research Center (October 1980)

Kay, M. Functional Grammar. In *Proceedings of the 5th Meeting of the Berkeley Linguistic Society*, Berkeley, 142-158 (1979)

Marcus, M. An overview of a theory of syntactic recognition for natural language. (AI memo 531). Cambridge, Mass: Artificial Intelligence Laboratory, (1979)

Pereira, F. & Warren, D., Definite Clause Grammars for language analysis. A survey of the formalism and a comparison with Augmented Transition Networks. *Artificial Intelligence*, 13; 231-278 (1980)

Small, S. Word expert parsing: a theory of distributed word-based natural language understanding. (Technical Report TR-954 NSG-7253). Maryland: University of Maryland (1980)

Stock, O. Dynamic Unification in Lexically Based Parsing. In *Proceedings of the Seventh European Conference on Artificial Intelligence*. Brighton, 212-221 (1986)

Stock, O. Putting Idioms into a Lexicon Based Parser's Head. To appear in *Proceedings of the 25th Meeting of the Association for Computational Linguistics*. Stanford, Ca. [1987]

Thompson, H.S. Chart parsing and rule schemata in GPSG. In *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics*. Alexandria, Va. (1981)