Incremental Parsing and Reason Maintenance

Mats Wirén
Department of Computer and Information Science
Linköping University
S-581 83 Linköping, Sweden
mgw@ida.liu.se

Abstract

The purpose of this paper is to compare different ways of adopting reason-maintenance techniques in incremental parsing (and interpretation). A reason-maintenance system supports incremental formation and revision of beliefs. By viewing the construction of partial analyses of a text as analogous to forming beliefs about the meanings of its parts, a relation between parsing and reason maintenance can be conceived. In line with this, reason maintenance can be used for realizing a strong notion of incremental parsing, allowing for *revisions* of previous analyses. Moreover, an assumption-based reason-maintenance system (ATMS) can be used to support efficient *comparisons* of (competing) interpretations. The paper argues for an approach which is an extension of chart parsing, but which also can be seen as a system consisting of an inference engine (the parser proper) coupled with a simplified ATMS.

1 Background and Introduction

This paper focuses on the problem of incremental parsing (and to some extent interpretation); in particular, how reason-maintenance techniques can be used to achieve a strong notion of incrementality allowing for piecemeal construction, revision, and comparison of partial analyses.

Human language understanding is apparently incremental in the sense of proceeding in a piecemeal fashion, (ideally) carried out in small, gradual steps as each word is encountered (Tyler and Marslen-Wilson 1977, Marslen-Wilson and Tyler 1980). Work on incremental parsing and interpretation is typically motivated by a desire to model, or mimic, (aspects of) this behaviour, for example, Bobrow and Webber (1980), Ades and Steedman (1982), Mellish (1985), Pulman (1985), and Haddock (1987, 1988, 1990).

However, there are also clear-cut computational reasons for trying to attain incrementality. Sparked off by the rapid development of increasingly powerful, distributed computer hardware, a paradigm of "immediate computation" is gaining popularity in interactive applications like WYSIWYG word processing, spreadsheet programs, and programming-language editors (Reps and Teitelbaum 1984, 1987). It is interesting to consider similar systems applied to interactive natural-language processing. The point is that incrementality is a prerequisite of the reactive, real-time-based behaviour of such systems. Furthermore, systems that mix for example deictic and natural-language input require that linguistic status be given to sentence fragments, thus demanding incremental analysis (Kobsa et al. 1986).

One body of work which appears to be useful in incremental parsing and interpretation is reason (or truth) maintenance.[1] A reason-maintenance system (RMS) supports incremental formation and revision of beliefs. By viewing the construction of partial analyses of a text as analogous to forming beliefs about the meanings of its parts, a relation between parsing and reason maintenance can be conceived.

An RMS is coupled with an inference engine (for example, a parser) which makes inferences within the problem domain, and the overall, combined system can be seen as an inferential problem solver. The RMS makes use of two data structures, *nodes* and *justifications*. A node represents a datum provided by the inference engine, such as an assumption or an inferred proposition. Whenever a datum is inferred from a conjunction of other data, the RMS records this dependency as a justification which relates the respective nodes. The RMS thus keeps track of what data are believed and disbelieved, and why, given the inferences made so far.

The traditional approach, justification-based reason maintenance, JTMS (Doyle 1979), is to maintain a (global) belief by associating with each node a status of *in* (indicating belief in the corresponding datum) or *out* (indicating lack of belief in the datum) such that every justification is satisfied.[2] The entire set of (consistent) *in* data make up the cur-

This research has been supported by the National Swedish Board for Technical Development.

[1] For an excellent introduction to reason maintenance, see Reinfrank (1989).

[2] I prefer the term "reason maintenance" but use the standard abbreviations "ATMS" and "JTMS" (where "T" stands for "truth").

rent *context* (belief, interpretation[3]). In case of a contradiction, dependency-directed backtracking is invoked to identify the inconsistent assumptions and enable retraction of some of them.

A more recent approach is assumption-based reason maintenance, ATMS (de Kleer 1986), which supports problem solving in multiple contexts simultaneously — there is no need to keep the overall database consistent as in the JTMS. Thus, the ATMS is oriented towards finding all solutions, whereas a JTMS is oriented towards finding only one solution. In this and other respects, an ATMS resembles chart parsing (Kay 1980, Thompson and Ritchie 1984), something which will be further dealt with below. Each ATMS node has a label with information about the minimal sets of assumptions on which it ultimately depends. Thus, rather than associating explicit *in/out* information with nodes, the status of a node with respect to a context can be determined by comparing the label of the node with the assumptions that underlie the context. There is no dependency-directed backtracking in an ATMS since contradictions do not pose any problem and the assumptions underlying a contradiction are directly identifiable.

The rest of this paper is organized as follows: Sections 2 and 3 review and compare various ways of adopting JTMS and ATMS techniques in parsing and interpretation. Section 4 carries on with discussing specifically how an ATMS-style system can be used in various applications of incremental parsing and interpretation, and, finally, section 5 summarizes the conclusions.

2 JTMS-Style Approaches

2.1 Overview

An early example of adopting reason maintenance in parsing (more precisely, story analysis) is the system RESUND (O'Rorke 1983). O'Rorke considers the problem of correcting inferences[4] that conflict with subsequent information in the story. He combines the story processor with a JTMS, using dependency-directed backtracking to determine incompatible assumptions, and a collection of preference heuristics to choose from among candidate solutions.[5]

More recently, Zernik and Brown (1988) suggest a coupling of a DCG parser with a JTMS, both of which are embedded in a Prolog system. The DCG is extended with default rules to enable nonmonotonic reasoning. These defaults are used to guide the entire parsing and interpretation process. The division of labour is as follows: Given a new piece of input, the parser outputs an (extended) analysis in the form of a dependency network. This, in turn, is fed to the JTMS, resulting in an updated *in/out* labelling of the network corresponding to the currently believed interpretation. The purpose of the JTMS is to obtain a system which avoids (chronological) backtracking and instead handles inconsistent information (ambiguities) by choosing to believe a different portion of the previous inferences. For example, in parsing a sentence like "The child sold by his parents was found alive" (cf. Zernik and Brown 1988:802), the system initially assumes "the child" to be the agent since the "default voice" is considered to be "active". When later the word "by" is parsed, a nonmonotonic supporter of "active" becomes *in*, thus making "active" become *out* and "the child" be considered direct object.

2.2 Problems

This section discusses some general problems of JTMSs and JTMS-style approaches to incremental parsing with special reference to the framework of Zernik and Brown (1988) whose notion of parsing comes closest to the one considered here.

Perhaps the most important characteristic of a JTMS is that it insists on global consistency; in other words, it is limited to one single solution (one context) at a time:

> "At each point [presumably after each new word], the parser must deposit a hypothesis based on a partial set of clues, a hypothesis which might later be retracted."
> (Zernik and Brown 1986:802.)

Unfortunately, in a domain like natural-language analysis where local ambiguity constantly plagues the parser with inconsistent information, this becomes problematic. First of all, when the set of assumptions admits multiple solutions, these cannot be compared: since a JTMS only allows one context, there is simply no way to examine two sets of beliefs simultaneously in order to gauge their relative strengths.

Furthermore, upon each incremental change (i.e., parsing of a new word), new JTMS labellings have

[3] Sense-semantic interpretation; possibly also contextual interpretation.

[4] An inference here corresponds to a selection (assumption) of a schema, a script-like knowledge structure used for deriving information not explicitly mentioned in the text.

[5] Other work, though not adopting downright JTMSs, makes use of somewhat similar techniques for the purpose of recovering from erroneous inferences. For example, Jacobs (1988), in dealing with the problem of "concretion" — developing a most specific or metaphorical interpretation — lets his system, TRUMP, treat each interpretation as an assumption. If an assumed interpretation results in a contradiction, dependencies are used to discard (chains of) assumptions that conflict with the preferred interpretation. Conflicting information is simply thrown away, thus in a sense making the system even more insistent on consistency than a JTMS. Story processors like ARTHUR (Granger 1980), FAUSTUS (Norvig 1983), and ATLAST (Eiselt 1987) keep track of successive (candidate) inferences and reconsider rejected ones when faced with conflicting information. It could finally be mentioned that in the postscript to his book, Mellish (1985:114) suggests a combination of chart parsing and JTMS, but does not further develop this.

to be computed for the network. If, in the course of this, a contradiction arises — the probability of which increases with the size of the grammar, i.e., with the number of (potentially competing) default rules — dependency-directed backtracking has to be invoked to identify the sources of the contradiction and resolve the conflict. This requires extensive search and often results in new contradictions. Until all conflicts are resolved, the status of some nodes may have changed between *in* and *out* several times. Thus, "the machinery is cumbersome" (de Kleer 1986:139).[6]

A further problem is that the inference engine works on only one part of the search space at a time. For example, if a word has two senses, only one of them will be worked on by the parser. But all that is known in such a case is that both senses cannot be part of the same final solution; it may still be important to draw inferences from them independently. For further discussion of these and other problems in connection with JTMSs, see de Kleer (1986:138 ff.).

3 ATMS-Style Approaches

3.1 Overview

Charniak and Goldman (1988) make use of an ATMS for keeping track of alternatives arising in sense-semantic and contextual interpretation, for example, with respect to word sense, case, and noun-phrase reference. Each alternative is treated as an ATMS assumption and is furthermore assigned a probability. By comparing sets of assumptions underlying various potential interpretations, the system can choose the "best" alternative, i.e., the one with the highest probability.

Nagao (1989) provides an approach where different assumptions about sentence interpretations constitute different "worlds" of formulae in a way which resembles an ATMS representation. A characteristic of these frameworks is that they only handle semantic-interpretation alternatives and do not (attempt to) integrate this with parsing.

A different kind of ATMS-style approach, and one that is grounded in parsing, can be obtained by extending a chart parser with dependencies; more specifically, by recording for each edge its immediate and ultimate source edges (Wirén 1989). The next section develops this.

3.2 An ATMS-Style Chart Parser

How can a chart parser extended with edge dependencies be viewed as consisting of an inference engine and a (simplified) ATMS? This section develops an informal answer to this question.

To begin with, an ATMS, just like a chart parser, can be seen as a tool for organizing efficient search through a space of alternatives by providing a "cache" of partial results and by sharing these results across different branches of the search space. Both the (basic) ATMS and a chart parser are monotonic in the sense of only providing monotonic derivability (and, in case of the ATMS, monotonic justifications). Both frameworks are incremental, performing piecemeal updates in response to a constant stream of new assumptions, nodes, and justifications. Furthermore, the order in which updates are made does not affect the final outcome of the process. In particular, the following correspondances hold:

- A chart edge corresponds to an ATMS *node*.
- A preterminal (lexical) edge corresponds to an *assumption* node.[7]
- The immediate source information of an edge corresponds to a *justification*.
- Information about the set (of sets) of ultimate source edges of an edge corresponds to its ATMS *label*.
- The chart corresponds to an ATMS *network*.
- An analysis (or interpretation) of a phrase, sentence, etc. corresponds to an ATMS *context*, i.e., the theory of an *environment*, where the latter is a set of assumptions.
- The (standard) chart-parsing algorithm corresponds to the inference engine.

More precisely, information about source edges (i.e., justifications) can be derived as follows: An edge formed through a combination depends on the active–inactive edge pair that generated it. An edge formed through a prediction depends on the (one) edge that triggered it. (Alternatively, predicted edges can be left out of the dependency trails altogether: since they only represent inferential hypotheses and do not carry any analysis structure, they could be seen as belonging with the inference engine rather than with the ATMS. On this view, a prediction has neither dependants nor sources.[8]) A scanned edge does not depend upon any other

[6] Incidentally, the sole explicit example provided by Zernik and Brown (cited above) only involves one default rule. Since the original as well as the revised interpretation represent coherent sets of justified beliefs, both can be arrived at by straightforward (re)labelling, thus avoiding the more cumbersome process of dependency-directed backtracking — in fact, Zernik and Brown do not mention dependency-directed backtracking at all. They also do not state any systematic preference policy, so it is not clear how they generally gauge the relative strengths of incompatible assumptions when trying to resolve contradictions.

[7] Thus, each word sense corresponds to an assumption. In a system which handles noisy input (e.g., ill-formed or spoken input), one might instead let hypothesized word forms correspond to assumptions.

[8] This would also solve the problem with top-down parsing pointed out in Wirén (1989:245 f.). — Note that, if we want to introduce garbage collection of useless predictions, it would still be necessary to keep a record of their dependencies.

edge (but on an instance of a word or a lexicalized phrase).

Labels are likely to be simple in an ATMS-style chart parser. Normally, each edge has a unique, stable set of sources — zero, one, or two edges which are determined once and for all when the edge is created.[9] A potential exception to this uniqueness of source is the case of the parser attempting to regenerate an (existing) edge, something which is prohibited by a redundancy test. This attempted regeneration actually corresponds to introducing an additional justification for the edge. Allowing this would require a more elaborate machinery for computing ATMS labellings in accordance with de Kleer (1986). Whether this capability is needed or not would have to be decided with respect to the demands of the particular application of the ATMS-style parser (cf. section 4).

It could finally be noted that a chart parser, as normally conceived of, does not record "nogoods" (inconsistent combinations of assumptions) as does the ATMS. Instead, the chart parser by itself ensures that inconsistent combinations of edges do not get further worked on (through predictions, the agenda, etc.).

4 Applications

4.1 Incremental Parsing and Interpretation

A strong definition of incremental parsing would require that text can be added or deleted in a piecemeal fashion and, furthermore, that the system behaves monotonically in processing such updates; i.e., as words are added/deleted, the set of possible analyses increases/decreases monotonically. An attractive property which follows from this is that the amount of processing of an update is roughly proportional to the size of the update (cf. Wirén 1989:242, Earley and Caizergues 1972:1040). An ordinary chart parser as well as an ATMS-style chart parser as put forward above are incremental in this sense, whereas the nonmonotonic model of Zernik and Brown (1988) and, say, an ATN parser are not. In the latter frameworks, a previously determined analysis can be modified by a subsequent step in the parsing process.[10]

As for interpretation with respect to a model or a context, in order for a system to be incremental in the above sense, it should be compositional (and again monotonic) such that the interpretation of a phrase is a function of the interpretations of its syntactic constituents and their associated contexts. One computationally-oriented model which fulfils this requirement, and which is indeed taylored for incremental interpretation, is that of Haddock (1987, 1988, 1990). Haddock's model, which can be seen as a continuation and refinement of Mellish (1985), incrementally interprets singular noun-phrases which refer to known contextual entities. Translated into a chart-parsing framework, upon scanning a word, the corresponding set of predicates and potential referents (obtained from the context) is associated with the new edge, and upon combining edges, a constraint-satisfaction algorithm is run to narrow down the set of possible referents.

The possibility of using dependencies also to incrementally handle deletions of words (assumptions) is investigated in Wirén (1989). Actually, a machinery is developed to handle *arbitrary syntactic changes*, which can be thought of as edit operations (insertion, deletion, replacement), and which can be freely combined and applied to arbitrary portions of the text — for example, input can be entered in any order. Edge dependencies are used to propagate the effects of a change precisely to those edges that are affected by the change.[11] This has potential computational applications in interactive natural-language systems such as language-sensitive text editing. In psycholinguistic terms, an edit operation might correspond to correction of a misread or misheard passage.

Since Haddock's model for incremental interpretation is not in any way limited to left-to-right incrementality, it is possible to adopt it also within the system of Wirén (1989). Furthermore, it is possible to conceive of a semantic analogue to this processing of syntactic changes. Consider a dynamic context, for example a database representing the real-time state of some world. By maintaining dependencies between entities and relations in the contextual model and their counterparts in the linguistic analysis, a machinery for incrementally reevaluating previously made interpretations with respect to the changing context could be attained.

4.2 Comparison of Interpretations

The chart allows comparison of (competing) interpretations in the sense that any analyses can be simultaneously examined. What one cannot do in ordinary chart parsing is to ask which particular edges (assumptions, etc.) underlie a given analysis since edges are not labelled with justifications or assumptions (and all information from lower-level edges may not have been percolated upwards). Put differently, the chart cannot "explain" its analyses. Of course, extending a chart parser to record dependencies is a simple thing. The point is that, in doing so, one has in effect obtained a simple ATMS-style problem

[9] In bidirectional chart parsing (Satta and Stock 1989) an edge might have three sources.

[10] One might ask which grammatical formalisms enable incremental processing. Categorial grammar in its various incarnations is an obvious alternative (for example, Ades and Steedman 1982). Unification-based grammar formalisms (Shieber 1986) provide another alternative given that the unification component is capable of incremental processing (Bresnan and Kaplan 1982:xliv ff., Steedman 1985).

[11] This could also be achieved in a JTMS-style parser.

solver. Charniak and Goldman (1988) provide an example of how ATMS techniques could be used for comparisons (cf. section 3.1).

4.3 Revision of Interpretations

The basic ATMS does not provide nonmonotonic justifications and hence no machinery for actually revising a previously held interpretation through default reasoning, etc. (Similar effects are instead achieved by maintaining multiple contexts and by switching between these.) However, certain semantic and pragmatic phenomena, like anaphoric reference, seem to require a capability for default reasoning; processing such phenomena by gradually developing all possible analyses would lead to combinatorial explosion (Asher 1984). Thus, although nonmonotonic devices destroy the strong notion of incrementality discussed above (the effects of an update cannot in general be kept local, the amount of computation needed is not bounded by the size of the update, etc.), they are sometimes needed. A recent example of this is Pollack and Pereira (1988) who use a strict compositional semantics but with nonmonotonic interpretation rules; other examples are Asher (1984), Dunin-Keplicz (1984), and Appelt and Konolige (1988).

Dressler (1989) shows how to extend the basic ATMS with a new type of node which, in particular, allows the encoding of nonmonotonic justifications and default rules. Given the relationship between chart parsing and ATMS, it seems like there should be a way of translating such a default machinery to an ATMS-style parsing framework. Furthermore, given a framework which integrates parsing and interpretation by performing interpretation on-line to the parser, it might be advantageous to allow encoding of default reasoning at the level of the parser, as indeed Zernik and Brown do, but to use it in a much more restricted way.

5 Conclusion

This paper compares JTMS- and ATMS-based approaches to adopting reason-maintenance techniques in incremental parsing (and interpretation). A major problem with the JTMS-based approach is that it is fundamentally limited to representing one context (one hypothesized interpretation). One consequence of this is that competing interpretations cannot be compared. Another consequence is that the system is constantly forced to commit rash choices and to spend a large part of its time revising these choices when faced with conflicting information. This situation appears even worse in light of the relative inefficiency of JTMSs. Zernik and Brown (1988) in a sense take this approach to its extreme, using default reasoning and dependency-directed backtracking to guide the entire parsing and interpretation process. It is however not clear why one would want to confine oneself to a single hypothesis at each step of this process.[12] It is also not clear that default choices and dependency-directed backtracking is the best way to guide search in natural-language analysis with respect to phenomena like syntactic ambiguity. From a computational point of view, there are well-known and efficient techniques, such as chart parsing, that work by developing syntactic alternatives in parallel. From a psycholinguistic point of view, although the issue is under discussion, there are both theoretical arguments and empirical evidence supporting the claim that alternative interpretations are explored in parallel (Crain and Steedman 1985). Incidentally, one of the rationales for this claim is that it is a prerequisite for competing interpretations to be *compared*.

In contrast to this, an ATMS solves or circumvents the major problems posed by a JTMS. The ATMS supports efficient development and comparison of all legal (possibly contradictory) analyses; an ATMS-style parser thus seems more in accordance with computational practice as well as with psycholinguistic evidence. Furthermore, the relationship between ATMSs and chart parsing is in itself an advantage, because it facilitates cross-fertilization of the respective subfields. In particular, it might be possible to make use of recent advances within reason maintenance in encoding default reasoning for the purpose of handling certain problems in semantic and pragmatic interpretation. Ultimately, this might provide for a more uniform handling of syntax, semantics, and pragmatics.

References

Ades, Anthony E. and Mark J. Steedman (1982). On the Order of Words. *Linguistics and Philosophy* 4:517–558.

Appelt, Douglas and Kurt Konolige (1988). A Practical Nonmonotonic Theory for Reasoning about Speech Acts. *Proc. 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, New York: 170–178.

Asher, Nicholas (1984). Linguistic Understanding and Non-Monotonic Reasoning. *Proc. Non-Monotonic Reasoning Workshop*, New Paltz, New York: 1–20.

Bobrow, Robert J. and Bonnie Lynn Webber (1980). Knowledge Representation for Syntactic/Semantic Processing. *Proc. First Annual National Conference on Artificial Intelligence*, Stanford, California: 316–323.

Bresnan, Joan and Ronald M. Kaplan (1982). Introduction: Grammars as Mental Representations of Language. In: Joan Bresnan, ed., *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, Massachusetts: xvii–lii.

Charniak, Eugene and Robert Goldman (1988). A Logic for Semantic Interpretation. *Proc. 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, New York: 87–94.

Crain, Stephen and Mark Steedman (1985). On Not Being Led up the Garden Path: The Use of Context by the Psychological Syntax Processor. In: David R. Dowty, Lauri

[12] Note that Zernik and Brown also do not make use of lookahead or delayed processing to facilitate intelligent choices.

Karttunen, and Arnold M. Zwicky, eds., *Natural Language Parsing. Psychological, Computational, and Theoretical Perspectives.* Cambridge University Press, Cambridge, England: 320–358.

de Kleer, Johan (1986). An Assumption-based TMS. *Artificial Intelligence* 28(2):127–162.

Doyle, Jon (1979). A Truth Maintenance System. *Artificial Intelligence* 12(3):231–272.

Dressler, Oskar (1989). An Extended Basic ATMS. In: Michael Reinfrank, Johan de Kleer, Matthew L. Ginsberg, and Erik Sandewall, eds., *Proc. 2nd International Workshop on Non-Monotonic Reasoning.* Springer Lecture Notes in Computer Science 346, Heidelberg, FRG: 143–163.

Dunin-Keplicz, Barbara (1984). Default Reasoning in Anaphora Resolution. *Proc. Sixth European Conference on Artificial Intelligence*, Pisa, Italy: 157–166.

Earley, Jay and Paul Caizergues (1972). A Method for Incrementally Compiling Languages with Nested Statement Structure. *Communications of the ACM* 15(12):1040–1044.

Eiselt, Kurt P. (1987). Recovering from Erroneous Inferences. *Proc. Sixth National Conference on Artificial Intelligence*, Seattle, Washington: 540–544.

Granger, Richard H., Jr. (1980). When Expectation Fails: Towards a Self-Correcting Inference System. *Proc. First Annual National Conference on Artificial Intelligence*, Stanford, California: 301–305.

Haddock, Nicholas J. (1987). Incremental Interpretation and Combinatory Categorial Grammar. *Proc. Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy: 661–663.

Haddock, Nicholas J. (1988). Incremental Semantics and Interactive Syntactic Processing. Ph.D. thesis, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland.

Haddock, Nicholas J. (1990). Computational Models of Incremental Semantic Interpretation. *Language and Cognitive Processes* 4(3–4):337–368.

Jacobs, Paul S. (1988). Concretion: Assumption-Based Understanding. *Proc. 12th International Conference on Computational Linguistics*, Budapest, Hungary: 270–274.

Kay, Martin (1980). Algorithm Schemata and Data Structures in Syntactic Processing. Report CSL-80-12, Xerox PARC, Palo Alto, California. Also in: Sture Allén, ed. (1982), *Text Processing. Proceedings of Nobel Symposium 51.* Almqvist & Wiksell International, Stockholm, Sweden: 327–358.

Kobsa, Alfred, Jürgen Allgayer, Carola Reddig, Norbert Reithinger, Dagmar Schmauks, Karin Harbusch, and Wolfgang Wahlster (1986). Combining Deictic Gestures and Natural Language for Referent Identification. *Proc. 11th International Conference on Computational Linguistics*, Bonn, Federal Republic of Germany: 356–361.

Marslen-Wilson, William and Lorraine Tyler (1980). The Temporal Structure of Spoken Language Understanding. *Cognition* 8(1):1–74.

Mellish, Christopher S. (1985). *Computer Interpretation of Natural Language Descriptions.* Ellis Horwood, Chichester, England.

Nagao, Katashi (1989). Semantic Interpretation Based on the Multi-World Model. *Proc. Eleventh International Joint Conference on Artificial Intelligence*, Detroit, Michigan: 1467–1473.

Norvig, Peter (1983). Six Problems for Story Understanders. *Proc. Third National Conference on Artificial Intelligence*, Washington, D.C.: 284–287.

O'Rorke, Paul (1983). Reasons for Beliefs in Understanding: Applications of Non-Monotonic Dependencies to Story Processing. *Proc. Third National Conference on Artificial Intelligence*, Washington, D.C.: 306–309.

Pulman, Steven G. (1985). A Parser That Doesn't. *Proc. Second Conference of the European Chapter of the Association for Computational Linguistics*, Geneva, Switzerland: 128–135.

Pollack, Martha E. and Fernando C. N. Pereira (1988). An Integrated Framework for Semantic and Pragmatic Interpretation. *Proc. 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, New York: 75–86.

Reinfrank, Michael (1989). Lecture Notes on the Fundamentals of Truth Maintenance. Siemens Report INF 2 ARM-5-88, Version 2. Siemens AG, Munich, FRG.

Reps, Thomas and Tim Teitelbaum (1984). The Synthesizer Generator. *Proc. ACM SIGSoft/SIGPlan Symposium on Practical Programming Environments*, Pittsburgh, Pennsylvania: 42–48.

Reps, Thomas and Tim Teitelbaum (1987). Language Processing in Program Editors. *Computer* 20(11):29–40.

Satta, Giorgio and Oliviero Stock (1989). Formal Properties and Implementation of Bidirectional Charts. *Proc. Eleventh International Joint Conference on Artificial Intelligence*, Detroit, Michigan: 1480–1485.

Shieber, Stuart M. (1986). *An Introduction to Unification-Based Approaches to Grammar.* CSLI Lecture Notes No. 4. University of Chicago Press, Chicago, Illinois.

Steedman, Mark (1985). LFG and Psychological Explanation. *Linguistics and Philosophy* 8:359–385.

Thompson, Henry and Graeme Ritchie (1984). Implementing Natural Language Parsers. In: Tim O'Shea and Marc Eisenstadt, *Artificial Intelligence: Tools, Techniques, and Applications.* Harper & Row, New York, New York: 245–300.

Tyler, Lorraine and William Marslen-Wilson (1977). The On-line Effects of Semantic Context on Syntactic Processing. *Journal of Verbal Learning and Verbal Behavior* 16:683–692.

Wirén, Mats (1989). Interactive Incremental Chart Parsing. *Proc. Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester, England: 241–248. Also: Research report LiTH-IDA-R-89-24, Department of Computer and Information Science, Linköping University, Linköping, Sweden.

Zernik, Uri and Allen Brown (1988). Default Reasoning in Natural Language Processing. *Proc. 12th International Conference on Computational Linguistics*, Budapest, Hungary: 801–805.