Default Reasoning
in Natural Language Processing

Uri ZERNIK
Artificial Intelligence Program
GE, Research and Development Center

Allen BROWN
Systems Sciences Laboratory
Xerox, Webster Research Center

Abstract

In natural language, as in other computational task domains it is important to operate by default assumptions. First, many constraints required for constraint propagation are initially unspecified. Second, in highly ambiguous tasks such as text analysis, ambiguity can be reduced by considering more plausible scenarios first. Default reasoning is problematic for first-order logic when allowing *non-monotonic* inferences. Whereas in *monotonic logic* facts can only be asserted, in non-monotonic logic a system must be maintained consistent even as previously assumed defaults are being retracted.

Non-monotonicty is pervasive in natural language due to the serial nature of utterances. When reading text left-to-right, it happens that default assumptions made early in the sentence must be withdrawn as reading proceeds. Truth maintenance, which accounts for non-monotonic inferences, can resolve this issue and address important linguistic phenomena. In this paper we describe how in NMG (Non-Monotonic Grammar), by monitoring a logic parser, a truth maintenance system can significantly. enhance the parser's capabilities.

1. Introduction

Ironically, adding knowledge to a computational system does not always extend its power. When equipped with comprehensive linguistic knowledge the same system might fare worse than when equipped with impoverished linguistic knowledge, since additional rules might impair simple cases. For example consider the following two sentences:

(1) The child sold his parents' dearest ornament.
(2) The child sold to a stranger was found alive.

Linguistic systems must handle both sentences: sentence (1) which appears unambiguous, and requires only basic English grammar; and sentence (2) whose interpretation is obscured by its grammatic construction. However, it is crucial that rules required for handling sentence (2) will not complicate the analysis of sentence (1) by contributing spurious ambiguity and computational overhead.

This linguistic behavior is problematic for computer parsers. Sentence (2) involves the *garden path* phenomenon, in which the reader is led to a standard interpretation which is retracted towards the end of the sentence. On the other hand, sentence (1) is read "linearly" without any such deviations. For a computer parser, it is important to show (a) how the initial interpretation is flipped in parsing sentence (2), and (b), that for sentence (1) no such flipping occurs. However, this behavior cannot be be captured by PROLOG-based parsers* [Dahl77,Pereira80] for two reasons: (a) **representation:** all linguistic rules are viewed as equal citizens, (b) **computation:** PROLOG backtracking can search for any solution, yet it cannot retract partial solutions. Thus, this difference in parsing complexity must yet be manifested in computational terms.

Default reasoning [McCarthy80] and Truth Maintenance Systems (TMS's) were introduced to cope with the representational and computational limitations of first-order logic [Doyle79, Reiter87, Goodwin87, Brown87]. First, TMS's distinguish between default cases, and cases presenting deviations from the norm. Second, TMS's allow the retraction of default assumptions during the reasoning process itself. In this paper we explain how existing parsing mechanisms based on unification fail in capturing important linguistic behavior. Furthermore, by implementing NMG (Non-Monotonic Grammar), we demonstrate how by monitoring a logic parser, a TMS can significantly enhance the parser's performance without adding any excessive overhead.

2. The Theoretical Issues

Four theoretical issues must be considered when parsing text: ambiguity, non monotonicity, context dependency, and knowledge gaps.

(a) Ambiguity: Consider the following sentence:

Mary was tired, so **John showed her home.**

Even for this simple sentence which appears unambiguous to a human reader, a parser must span a variety of interpretations stemming from several linguistic levels:

Semantics: Who is "John": is it John Marberg or is it John Kepler
(where both exist in the context)?

Syntax: What part of speech is "her":
is it an objective pronoun, i.e. he showed **Mary** home
or is it a possessive, i.e. he showed **Mary's** home?

Lexicon: What is "to show", and what is the intended meaning:
he showed the house to a potential buyer,
or he showed her the way to her home?

Role-Binding: Does "show" (meaning make-visible) take 1 or 2 objects?
he showed <her>,
or he showed <her> <home>?

Context: What is the purpose of the act:
selling Mary's house to potential buyers,
or making sure she arrives safely home?

*We allude here to PROLOG as a parsing mechanism and not to the programming language PROLOG which is computationally universal.

A parser is required to negotiate this set of conflicting/cooperating clues and to deposit in a database a hypothesis about the meaning of the utterance. This problem is even aggravated when imposing left-to-right order on parsing. At each point, the parser must deposit a hypothesis based on a partial set of clues, a hypothesis which might be later retracted.

(b) Non-Monotonicity: Garden path sentences highlight the problem:

(3) The old man's glasses were filled with sherry.
(4) I saw the Grand Canyon flying to New York.
(5) The book sold yesterday was cheap.
(6) The horse raced past the barn fell.

In each one of these sentences, the initial hypothesis is retracted when an assumed default rule is violated:

(3) A semantic assumption–glasses stand for looking glasses–must be retracted at the end of the sentence–since glasses turn out to actually mean drinking containers.

(4) A syntactic assumption dictates the default sentence structure (**S -> VP, NP**). Thus the Grand Canyon is taken as the agent flying to NY. This interpretation is in conflict with existing world knowledge (since canyons cannot fly).

(5) A word sold is assumed to take the active voice. Backtracking occurs due to knowledge of the selling act (books do not normally sell days).

(6) There are two default assumptions which fail: (1) raced is taken by default to be active voice, and (2) the clause raced past the barn is taken as the verb phrase of the sentence. Both assumptions fail when the word fell is encountered, and raced is found to be the past participle in the unusual sense of "being driven".

Among these examples, some are impossible even for human listeners to comprehend; in others, the flipping of the meaning is hardly noticeable. How can a parsing mechanism reflect these degrees of difficulty?

(c) Context Dependency: Compare the interpretations of following pairs of sentences:

(7) John wanted a new car. He took it up with his dad.
(8) This is his new Porsche. He took it up with his dad.

The syntactic analysis was driven by the semantic context established by prior discourse: in (7), up is taken as an integral part of the entire lexical phrase to take it up with somebody, where in (8) it serves as a general adverb. This is due to the different interpretation of it: in (7) John discussed "an issue", while in (8) he probably drove "the car" itself up the hill accompanied by his dad. How can a parser reflect such context sensitivity?

(d) Lexical Gaps: Linguistic systems cannot be assumed complete, and so text processing must take place even in the presence of lexical gaps [Zernik87, Zernik88]. Consider the following sentence which includes a lexical unknown, the word plend.

(9) John **plended** Mary to come over.

How can a parser make even a partial sense out of this sentence? Can a parser use default lexicon rules in parsing such unknowns?

3. Truth Maintenance: The Architecture of NMG

What is a truth maintenance sytem and how can it be tailored for use in the linguistic domain? NMG is a system for employing truth maintenance in text comprehension. NMG's architecture is simple: it includes a parser (PAR), and a Truth Maintenance System (TMS). PAR is a DCG-style parser [Pereira80], implemented as a forward-chaining theorem prover. PAR produces a parse tree in the form of a logical *dependency-net*. This dependency-net is fed to a TMS whose output is an **IN/OUT** *labelling* presenting the currently believed interpretation. The basic computational paradigm is described by an example in parsing the sentence below:

(10) The baby sold by his parents was found alive.

Text interpretation can be pursued in two possible ways: either by propagating all possible grammatic interpretations of that utterance, as in logic parsers. This might introduce spurious interpretations for simple text. Or alternatively, by initially trying a default interpretation, and ***only when this default interpretation fails*** try out other interpretations. This is the approach taken by NMG. There are two interesting stages in the computation, according to the text already received:

(10a) The baby sold
(10b) by his parents was found alive.

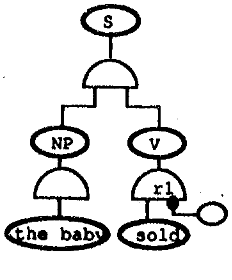
One basic default assumption was committed at stage (10a):

o Unless otherwise observed, the verb sold assumes the active voice past tense.

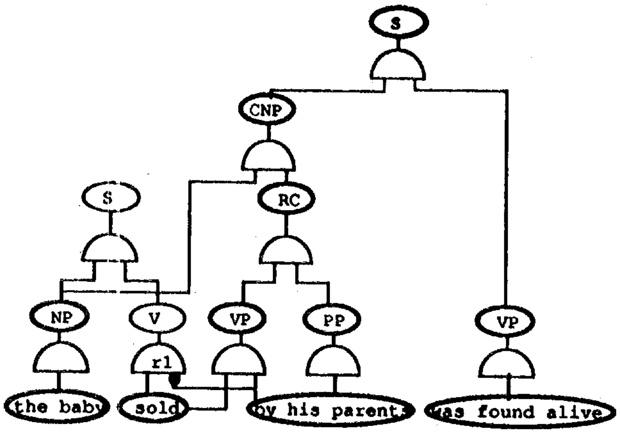
The rules related to this assumption are given in NMG as follows:

(r1) verb(sell,active,past) :- text(sold) O not-active(sold)
(r2) not-active(sold) :- preceded(sold, was)
(r3) not-active(sold) :- followed(sold, by-clause)

NMG extends DCG's Horn-clause notation by allowing an "unless" term, marked by the symbol O. Such an "unless" term appears in (r1), the default case for sold. Deviations can be established as aggregation of cases. (R2) presents the first deviation. When sold is preceded by the particle was, it is taken as a passive voice. (R3) presents the second deviation, when a by-clause follows sold. Other deviations may be added on as the linguistic system is enhanced. The diagram below describes two snapshots in the computation of sentence (10).



(a)



(b)

Notation: *ovals* in this scheme stand for facts; *AND gates* stand for rules; *dark* ovals are **IN**; *light* ovals are **OUT**. Consider part (a) which describes the parse of (10a): The **dependency-net** constructed by PAR is based on instantiated linguistic rules. There are three new instantiated facts: NP the noun phrase, V the verb, and S the entire sentence (some short cuts were made in drawing the parse tree due to space limitations). The associated **IN/OUT labelling** is produced by the TMS, and so far all the facts have been labelled **IN**. In particular, the output of the default rule **(r1)** is labelled **IN**, since its *inhibitive* port (marked by an inverter) is labeled **OUT**: no deviation of **(r1)** has yet been observed.

Part (b) describes the parse after the rest of the sentence (10b) has been read. (Notice that the dependencies of part (a) have been copied over for reference purposes, although in the model itself dependencies are *not* recalculated or copied). Reading (10b) causes the withdrawal of the previous interpretation and the construction of a new one. However, since new words were only added on, how in this scheme, could anything be withdrawn? In stage (b) too there are two orthogonal activities:

(1) PAR constructs new dependencies: The by-clause following sold justifies the inhibitive port of **(r1)**; the same by-clause also justifies an alternative role for sold (VP), as a passive voice verb; this fact plus the by-clause itself add up to a relative clause (RC); RC joins the old noun-phrase (NP) to form a composite noun-phrase (CNP); CNP now joins a new verb-phrase (VP) in forming a new sentence (S). Throughout this process *no dependencies have been modified or retracted; new dependencies were only added on.*

(2) The TMS relabels the network. First, the old interpretation is ruled out: since the inhibitive input of **(r1)** is now labeled **IN**, the output of **(r1)** becomes **OUT**, and so does the initial interpretation S. The rest of the new facts are labelled **IN**. Thus, *the non-monotonic effect is accomplished by relabeling nodes, and not by retracting dependencies.*

A PROLOG-based parser, at stage (b), must undo all its prior parsing inferences, retract the fact it has deposited in the database, and start processing from scratch, this time ruling out the incorrect default assumption. Using a TMS, a parser can recover by simply relabeling the parts of the parse which depend on the above assumptions, and proceed gracefully thereafter.

Problems which are naturally addressed by non-monotonic inference are pervasive in the linguistic domain, where parsing proceeds left-to-right, and they are epitomized by garden path sentences.

4. NMG: A Process Model

Non-monotonic reasoning is not confined only to garden path sentences. We show here an example of an apparently simple sentence, for which interpretations are asserted and retraced dynamically. This example also demonstrates the role of default reasoning in *lexical access* and in *context interaction*, where the context is the semantic structure yielded by prior discourse. Consider the initial interpretation constructed after reading the following utterance:

(11a) John needed a new battery. **He took it**

This text yields an initial hypothesis: "John ptransed a battery". This hypothesis is based on two default rules: (1) **Lexical access**: unless otherwise observed, a generic word such as take indexes the generic meaning *ptrans* (physical transfer) [Schank77] (2) **Context interaction**: Unless otherwise observed, it refers to the last physical object referred to in the discourse [Sidner79] (here it refers to the battery). However, as reading proceeds, these hypotheses are retracted:

(11b) John needed a new battery. He took it **up with his dad**

At this point, a more specific lexical entry is accessed: "X take Y up with Z" in the sense of "raising an issue" [Wilks75, Wilensky80]. The referent for it is switched now from the battery itself to "John's goal of getting a battery", due to selectional restrictions in the new lexical phrase. However, the initial interpretation is recovered when reading continues:

(11c) John needed a new battery.
He took it up with his dad **from the basement.**

At this point, the additional clauses in the sentence are used to augment the initial hypothesis which had been previously abandoned. Notice that the initial hypothesis is recovered simply by markling it IN and not by computing it from scratch.

5. Logic Programming: From CFG through DCG to NMG

Logic programming [Colmerauer73, Kowalski79, Dahl77, Pereira80] has mechanized many declarative linguistic systems [Kay79, Bresnan82, Shieber87], and provided a new computational paradigm. Definite-Clause Grammars (DCG) in particular, have exploited the advantages of Context-Free Grammars (CFG) by a simple extension. In DCG, a non-terminal may be any PROLOG term, rather than simply an atom as in CFG. The following example demonstrates how one particular rule has evolved from CFG to DCG.

```
CFG:  sentence               --> noun-phrase, verb-phrase
DCG:  sentence(s(NP,VP))     :-  noun-phrase(NP,N), verb-phrase(NP,N)
```

This extension has two features: (a) maintaining agreement rules–the argument N maintains the number in both the noun and the verb; (b) constructing semantic denotations–the arguments **NP** and **VP** contain the denotations of the constituents, from which the denotation of the entire sentence **(s(NP, VP))** is constructed.

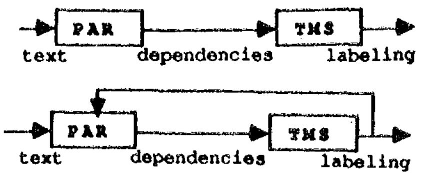
Logic programming has assumed a central role in language processing for two reasons: (a) It allowed the expression of declarative linguistic rules, and (b) Direct application of PROLOG operationalized grammars, using unification and backtracking as the mechanism. PROLOG also enabled other grammars (beside DCG) such as transformational grammar [Chomsky81] and case grammar [Fillmore68], to be emulated. However, the *direct application of PROLOG* has presented three limitations: (a) Parsing was driven by syntax, and the semantic interpretation was a by-product. (b) While PROLOG itself can be extended to express default (through the notion of *negation as failure* [Clark77]), PROLOG does not have an explicit notion of dependency so that a parser can diagnose what went wrong with the parse. (c) PROLOG itself does not facilitate default reasoning which can resolve lexical gaps.

Therefore, we have introduced NMG; a logic parser which enhances DCG's capabilities in three ways: non-monotonic reasoning, refinement and retraction, and diagnostic reasoning.

(a) **Non-Monotonic Reasoning:** NMG enables the parser to gracefully correct its parse tree by identifying parts of the reasoning structure which depend on retracted assumptions. Non-monotonicity is pervasive in language processing due to the serial nature of language.

(b) **Retraction and Refinement:** A main objective in text processing, required for left-to-right parsing, has been parsing by refinement. In reading a sentence, a parser should not deposit a single final meaning when a "full stop" is encountered. Rather, an initial concept must be asserted as early as possible, an assertion which must be refined as further words are provided at the input. However, in fulfilling this objective, existing models [Hirst86, Lytinen84, Jacobs87] have not dealt with the possibility that the entire hypothesis might be retracted, and replaced by a second hypothesis. NMG enables a parser to both refine an existing hypothesis, and to retract the entire hypothesis from the database, if contradictory evidence has been received.

(c) **Diagnostic Reasoning:** Consider a pair of operational modes, given in the diagram below:



While in (1), the system operates in an "open loop", and the TMS basically monitors which hypothesis is currently IN, in (2) the information produced by the TMS can be used in reasoning **about** the parse itself. This is important in learning and in parsing ill-formed text. We describe this mode in a later report.

6. Conclusions

We have presented NMG, a mechanism which can potentially enhance all logic parsers. NMG's advantages are emphasized in parsing complex sentences in which hypotheses are being retracted. However, its main advantage is in avoiding spurious activity when parsing simple sentences. Thus we have accomplished an objective laid down by Allan Kay: "Easy things should be easy; hard things should be possible".

References

[Bresnan82] Bresnan, J. and R. Kaplan, "Lexical-Functional Grammar," in *The Mental Representation of Grammatical Relations*, ed. J. Bresnan, MIT Press, MA (1982).

[Brown87] Brown, A., D. Gaucas, and D. Benanav, "An Algebraic Foundation for Truth Maintenance," in *Proceedings The 10th International Joint Conference on Artificial Intelligence*, Milano Italy (August 1987).

[Chomsky81] Chomsky, N., *Lectures on Government and Binding*, Foris, Dordrecht (1981).

[Clark77] Clark, K.L., "Negation as Failure," in *Logic and Data Bases*, ed. H. Gallaire, J. Minker, Plenum Press, New York and London (1977).

[Colmerauer73] Colmerauer, A., H. Kanoui, P. Roussel, and R. Pasero, "Un Systeme de Communication Homme-Machine en Francais," Universite d'Aix-Marseille, Marseille, France (1973). Tech Report.

[Dahl77] Dahl, Veronica, "Un System Deductif d'Interrogation de Banques de Donnees en Espgnol," Universite d'Aix-Marseille, Marseille, France (1977). PhD Dissertation.

[Doyle79] Doyle, J., "A Truth Maintenance System," *Artificial Intelligence* **12** (1979).

[Fillmore68] Fillmore, C., "The Case for Case," pp. 1-90 in *Universals in Linguistic Theory*, ed. E. Bach R. Harms, Holt, Reinhart and Winston, Chicago (1968).

[Goodwin87] Goodwin, J.W., "A Theory and System for Non-Monotonic Reasoning," Linkoping University, Linkoping, Sweden (1987). PhD Dissertation.

[Hirst86] Hirst, G. J., *Semantic Interpretation and the Resolution of Ambiguity*, Cambridge, New York, NY (1986).

[Jacobs87] Jacobs, P., "Concretion: Assumption-Based Understanding," in *COLING 88*, Budapest, Hungary (1988).

[Kay79] Kay, Martin, "Functional Grammar," pp. 142-158 in *Proceedings 5th Annual Meeting of the Berkeley Linguistic Society*, Berkeley, California (1979).

[Kowalski79] Kowalski, R., *Logic for Problem Solving*, Elsevier North Holland, New York (1979).

[Lytinen84] Lytinen, S., "The Organization of Knowledge in a Multi-lingual Integrated Parser," Yale University, New Haven, CT (1984). PhD Dissertation.

[Pereira80] Pereira, F. C. N. and David H. D. Warren, "Definite Clause Grammars for Language Analysis- A Survey of the Formalism and a Comparison with Augmented Transition Networks," *Artificial Intelligence* **13**, pp.231-278 (1980).

[Reiter87] Reiter, R. and J. deKleer, "Foundations of Assumption-Based Truth Maintenance Systems: Preliminary Report," in *Proceedings 6th National Conference on Artificial Intelligence*, Seattle WA (1987).

[Schank77] Schank, R. and R. Abelson, *Scripts Plans Goals and Understanding*, Lawrence Erlbaum Associates, Hillsdale, New Jersey (1977).

[Shieber87] Shieber, S., *An Introduction to Unification-Based Approaches*, Univ. of Chicago Press, Chicago, IL (1987).

[McCarthy80] McCarthy, J., "Circumscription--A Form of Non-Monotonic Reasoning," *Artificial Intelligence* 13 (1980).

[Sidner79] Sidner, C., "Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse," MIT, Cambridge, MA (1979). PhD Dissertation.

[Wilensky80] Wilensky, R. and Y. Arens, "PHRAN: A Knowledge-Based Approach to Natural Language Analysis," in *Proceedings 18th Annual Meeting of the Asosciation for Computational Linguistics*, Philadelphia, PA (1980).

[Wilks75] Wilks, Y., "Preference Semantics," in *The Formal Semantics of Natural Language*, ed. E. Keenan, Cambridge, Cambridge Britain (1975).

[Zernik87] Zernik, U., "Learning Phrases in a Hierarchy," in *Proceedings 10th International Joint Conference on Artificial Intelligence*, Milano Italy (August 1987).

[Zernik88] Zernik, U. and M. G. Dyer, "The Self-Extending Phrasal Lexicon," *The Journal of Computational Linguistics: Special Issue on the Lexicon* (1988). to appear.