# PARSING WITH ASSERTION SETS AND INFORMATION MONOTONICITTT

C. Edward Barton, Jr. and Robert C. Berwick

MIT. Artificial Intelligence Laboratory
545 Technology Square
Cambridge, MA 02139

## ABSTRACT

We propose a new approach to parsing ambiguity in which a parser always moves forward with the common elements of competing syntactic analyses. The approach involves *assertion sets* constrained so that in formation is monotonically preserved throughout a parse Asseriion sets have several advantages over trees as a parsing representation. They may also lead to better computational understanding of the attention-shifting mechan ism

## 1 INTRODUCTION

Recent linguistic theories divide linguistic constraints into subsystems each having its own character. The complex surface character of a language is ultimately generated by the interactions among a few fundamental processes and constraints. We are most interested in the GB-theory framework of Chomsky (1981), which identifies subtheones concerned with locality, government, assignment of semantic roles, pronoun binding, case, control, and X-constraints, but some developments in other frameworks also tend toward modularity. For instance, Shieber (1983:2fF) describes a version of the GPSG formalism that separates immediate-dominance rules, linear-order constraints, and metarules, while the TAG formalism (Kroch and Joshi, 1985) factors recursion apart from co-occurrence restrictions.

The surface complexity of parsing should be decomposed in the same way as the surface complexity of language. Principles that are common to all languages should not have their effects repeatedly redescribed in the descriptions of particular languages, but should instead be exploited as part of parser design. Similarly, a single underlying process within an individual language should not have its effects spelled out separately in each surface manifestation; ideally the process should be encoded just once, in such a way that, the parser can work out what surface appearances to look for Beyond syntactic theory, parser design provides additional opportunities for such factoring, not only linguistic principles but also aspects of parser control structure may be factored out However, the general effect on the description of a language is the same Less information is needed to describe a language, if redundancy can be factored out and if control-structure elements can be removed from the grammar and incorporated in basic parser design

Our goal is thus a dual one. We aim to build a parser that bases its operation on modular subcomponents instead of a welter of surface-oriented rules; in so doing, we will reduce the amount of language-particular syntactic information that must be supplied by the designer of a natural-language system.** As one component of this effort, we are considering possible ways to use a "stripped-down" parsing representation that is based as much as possible on the predicates of linguistic theory. We hope to reduce the amount of grammatically extraneous information that the parser manipulates

## II ASSERTION SETS

It is doubtful that traditional parse trees are ideal for representing syntactic structure, for in general the range of structural information that a tree makes explicit may not correspond to the information that is grammatically relevant. For example, X-theory suggests that the head-projection relationship may be more important grammatically than the immediate-domination relationship that a tree displays. In a different vein, it has been hypothesized (Lasnik and Kupin, 1977:178f) that linguistic theory is insensitive to characteristics of trees that cannot be recovered from information about the range of terminals spanned by each constituent.

We are investigating the use of *monotonically growing assertion sets* as a parsing representation. A constituent is represented by a triple , where is a bundle of syntactic features *(eg* one that we might abbreviate with the usual label NP) and i, *j* are the input positions defining the left and right edges of the constituent* For example, if we assume that Adj" N forms a constituent that we will call NBAR, the structure of the NP *the red block* might be represented by the assertion set

{(NP 0 3),(Det 0 l),(Adj 1 2),(N 2 3),(NBAR 1 3)}.

With this representation, parsing is the construction of such an assertion set — closely akin to a "phrase-marker" in the sense of Chomsky's (1955) early work (In the early stages of parsing, the assertion should actually appear as (NP 0 *) to indicate that the right edge of the constituent has not yet been encountered.)

An assertion-set parser develops its analyses *drterministically* if changes in its (global) assertion set are always *refinements* in the information-theoretic sense that is, if information is monotonically preserved. Under determinism, the only possible refinements are adding a new assertion, changing a * to a specific value, and adding features to an underspecified category. For example, operating under a rudimentary X-theory, it would be possible to change the features of a constituent from [+-Nj to i+N, V,+rnax], i.e. from an underspecified category to NP, but impossible to change NP to VP. Monotonic.ity would also rule out the usual notion of nondeterministic chart parsing; the parser would be unable to remove initially plausible analyses that failed to pan out.

Monotonically growing assertion sets are attractive in several ways for representing syntactic structure. For example, beyond the device of using initially underspecified feature bundles, there are some useful *structural modifications* that are information-preserving when applied to assertion sets, but not when applied

Thin report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology, Support for the Laboratory's artificial intelligence research has been provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-80-O-0805. During a portion of this research Ed Barton* graduate studies were supported by the Fannie and John Hertz Foundation.

See Barton (1064) for more information on this general research program.

This representation takes a cut? from Lasnik and Kupin as well as from the representations used in chart parsing. Nirenburg and Attiya (1984) use a similar representation, but do not add the constraint of information monotonicity. Although we use numeric input indices here for simplicity, a representation based instead on the actual words of the terminal string is better in cases of movement and, under an new analysis by Goodall (1984), in cases of conjunction.

to trees * Changing the tree for / *told John a ghost story* into the tree for / *told John a ghost story was th< lust thing 1 wanttd to hear* requires (non-monotonically) breaking the link between VP and ;$_{Nr}$ *a ghost story* J and replacing it with a link between VP and S. In the same way, a tree link is deleted when *John* is moved one level deeper in going from *see John* to *sit John and Bill* In the assertion-set representation, each of these changes can be described as the addition of an assertion An S assertion is added in the first change, an NP assertion in the second; in each case all previous structure assertions remain valid when the new assertion is added, if the change is made before the right edge of VP has been declared. Assertion sets can thus allow a deterministic parser to be *partially noncommittal* about the exact attachment level of a constituent.**

The ability of assertion sets to represent partial information can also be useful in handling PP-attachment ambiguity. If it is not clear whether to attach an adjunct PP under NP or VP, for instance, the various structural possibilities will still agree on the existence and internal structure of PP. If NP with adjunct PP is analyzed as [$_{NI}$. NP PP ], they will also agree on the lower NP.*** This example illustrates the fact that assertion sets support co-called Chomsky-adjunction more naturally than sister-adjunction For example, on some analyses of the rightward movement called Heavy NP Shift, NP is Chomsky-adjoined to the end of VP to produce the structure |$_{VI}$ VP NP ] With assertion sets, the representation of the lower VP is (monotonically) preserved when the assertion is added that describes the upper VP. Some linguists have argued that the preservation of information about constituent structure makes this form of adjunction the appropriate one for describing the structural changes wrought by transformations.**** With trees, it is *sister-adjunction* that is information-preserving, as in a hypothetical replacement of |vr V *t* PP ] With |$_{vp}$ V *t* PP NP ).

## III THE THEORY OF ATTENTION-SHIFTING

Because of its atomistic character, the assertion-set representation may also pave the way to a better understanding of the attention-shifting mechanism of the deterministic Marcus parser (Marcus, 1980:175) The attention-shifting mechanism implements a "wait-and-see" strategy for dealing with some of the cases in which the parser cannot tell which possible step to take next. Interpreted abstractly, the strategy allows the parser to move forward with those elements of the structural analysis that it can be sure of. When attention-shifting rules begin to build a constituent, it may be unclear how it will fit into the final parse tree. However, a deterministic parser cannot be justified in building the constituent unless all competing analyses agree on its existence and internal structure.*****

The possibility exists that a parser could deal with parsing ambiguity by explicity observing the operating principle: always go ahead with the *common elements* of competing syntactic analyses. Under this principle, it would not be necessary to write attention shifts into the rule system explicitly; attention-shifting, when necessary, would be automatic. Such a parser could *explain* attention-shifting behavior by deriving it from a principled treatment of parsing ambiguity, could validate the informal characterization of attention-shifting as the implementation of a *wait-and-see* strategy, and would clarify the *computational problem* that is solved by the attention-shifting mechanism. In brief, it would contribute to the *computational theory* of the Marcus parser in Marrs (1980:25) sense in addition to serving the goal of removing control-structure elements from language descriptions.

Assertion sets are superior to trees for use in such a parser It is unclear how to *intersect* trees in order to take the com mon elements of different analyses, while ordinary set intersec tion roughly suffices for assertion sets because of their atomistic character * It is also unclear how the tree representing common elements of analysis could be *partially noncommittal* about attachment point in the cases mentioned earlier. Finally, in the presence of left-recursion as with possessive NPs, there will be an *infinite number* of ways to extend a tree downward to encompass a new element; with at most bounded lookahead, it is impossible to say how many tree nodes lie between the new input and the point of attachment to the existing tree. It will thus be difficult to envision all possible syntactic analyses in a tree-based parser. Left-recursion causes no more of a problem with assertion sets than it does with the related representation of a chart parser, since the ^-notation collapses an infinite number of nodes into a single assertion.

Several issues must be addressed in the design of any parser that proceeds by moving ahead with those elements of the syntactic analysis that are known for certain. The parsing representation must be *decomposable* into smaller elements that have meaning when separated; assertion sets fit the bill here. Different aspects of parsing rules and actions must also be *separable* in the sense that some parsing actions can still be licensed even when knowledge is insufficient to license all of them. It should sometimes be possible for parsing ambiguity to be eventually *resolved* when disambiguating evidence is encountered; elements of analysis that are correct but were initially discarded because of uncertainty should not remain forever absent. A decision must be made about the *stringency* of rule matching; it is customary to require the parsing representation to explicitly list the features mentioned in a rule, but because taking the common elements of competing analyses will result in feature underspecification, it may be better to require only feature compatibility between the rule pattern and the parsing representation. One must ensure that the implementation strategy does not allow *combinatorial blowup* to creep into the rule-matching process. Finally, it will be necessary to impose some *coherence* requirement on the collection of analytic possibilities; if they diverge too widely, the parser cannot sensibly integrate their common elements.

## IV SHAPE COMPATIBILITY

The application of this parsing method is not completely worked out, either in the Marcus framework or in a standard context-free parsing framework. However, some of the intent can be suggested by sketching out a simplified model based on CFGs. The basic parsing cycle is to involve three steps: matching rules against the parsing representation, running the matching rules to produce several possible extensions of the representation, and intersecting the possibilities to produce the next parsing representation. The fundamental problem is to intertwine analysis and control in such a way that the monotonically growing syntactic analysis is always sufficient to support rule matching, while the control component always runs a set of rules that will advance the analysis one step further in a coherent way — all the while operating under the strategy of taking the common elements of competing syntactic analyses.

As an initial approximation, suppose predictions define the unit of licensed parse continuation Then if *a* is ambiguously either $A_1$ or $A2$ and the rules $S$ ~» $A_1$ $B$ $x$ and $S$ —> $A_2$ $B$ $y$ are being considered while parsing a string that begins with a, it will

* Marcus *et al*. (1983) describe a parsing representation that also differs from trees in its possibilities for information-preserving structural modification.

This should be especially helpful in devising a data driven treatment of conjunction that does not predict conjoined NPs except when prompted by specific cues. The close relation between some variant of assertion sets and the monostrings that Lasnik and Kupin have described also makes assertion sets promising for the implementation of Goodall's new (1984) monostring-based theory of conjunction.

The assertion-set framework is compatible with having the assertion sets filtered by extrasyntactic information in order to ultimately resolve the attachment ambiguity.

Chomsky (1981:141), among others, has tentatively argued this. However, "Chomsky-adjunction" as a name for this operation is historically accidental.

Marcus's actual mechanism includes nothing to guarantee such agreement, with consequences that become more severe as lexical ambiguity increases.

Some modification is necessary in order to accommodate analyses that disagree only on node features.

initially be unclear whether to reduce *a* as $A_1$ or $A_2$ However, in either case it is predicted that a *B* will come next* in the input This common prediction can license the construction of a *B* and eventually allow the parser to see the disambiguating *x* at the end of the string. The same principle operates in a more complex way when parsing the VP *know that big n d blocks* . . given a determiner-noun agreement mechanism; if *thai* can be either Comp or Det. it is initially unclear which interpretation to take However, the competing syntactic analyses agre* that an NBAK is possible after *that.* Thus the construction of NBAK is licensed, and the agreement mechanism can rule out the determiner interpretation once the NBAK is built

Closer scrutiny reveals that intersection of predictions is not actually the appropriate operation here. Suppose *B* above surfaces as *k'b,* another constituent *C* surfaces as A-'r, and the rule *S* —» *A\Cz* is possible. The construction of either *B* or *C* should be licensed after the initial a; in fact, *B* and *C* can always be distinguished by the time they have been completely scanned However, only *B* is licensed if we take the intersection of all predictions. In seeming contradiction to the principle of taking common elements of competing analyses, predictions should be subject to *union* rather than intersection.

The contradiction is resolved by noticing that in any deterministic parser, it is necessary to say what counts as part of the syntactic analysis subject to determinism In Marcus's parser, packet activations don't count; in contrast to features, they may be both added and removed. Packet activation *licenses* the interpretation of certain elements in certain ways if they occur, but it does not *commit* the parser to expecting those elements (Node creation and attachment, on the other hand, *arc* subject to determinism ) Predictions in the CFG framework license possible interpretations in the same way, hence they should not be subject to intersection; rather, the interpretations that are actually imposed should be intersected.

If some operations involve the union rather than the intersection of possibilities, an immediate question is why the parsing method does not degenerate to the full Earley algorithm. In this sketch, that question is where the *coherence* requirement comes in. In the assertion-set framework, the simplest requirement to impose is one of *shape compatibility.* In the case of *that*-ambiguity, even though we cannot initially decide whether *that* is a complementizer or a determiner, in either case it bears the same structural relationship to the next constituent (commanding the NBAR of NP, or commanding the S of SBAR).* If different analyses must place constituent boundaries in the same place but may disagree about constituent identity, dotted-rule items will not re* quire a return address as they do in the full Earley algorithm. In the above example, the possibilities (A1 0 1) and (A2 0 1) are shape-compatible and can intersect to the featureless lump (* 0 1). When the end of a dotted rule is reached, such lumps can be back-traversed to find the left edge of the completed constituent Thus the coherence requirement of shape compatibility allows the parser to use a *finite* "packet structure," as indeed it must if information monotonicity is not to be vacuous.

Preliminary investigation suggests that shape compatibility can help in many troublesome cases, *e.g.* t/iaf-ambiguity and /or-ambiguity. Other cases such as PP-attachment ambiguity will require mechanisms to be extended. As one possibility, limited lookahead promotes coherence by filtering out shape-incompatible possibilities that would the soon anyway. Extensions will also be required for full attention-shifting behavior.

Ideally, CFGs and dotted rules should be dispensed with and the feature bundles of assertions should drive the entire analysis. By separability, this is feasible only if the feature system that is used can support a parsing interpretation for individual features. A feature system under development by Reuland (1984) is especially interesting in this regard, since each of his features describes a separate aspect of the combinatorial possiblities of a syntactic category. However, the standard X features [iN,±V| also are individually relevant to Case-assignment rules and other constraints that could be exploited in parsing In addition, some c ;IM-: of parsing ambiguity that have been thought to require .mention-shifting might be analyzed in terms of other common elements between competing analyses besides those mentioned here. For example, a fact closely related to shape compatibility is that English *that* must *begin* some kind of X projection, whether it 1- the specifier (Det) or the head (Comp) of that projection.

* The possibilities share the same skeleton, in the sense of Levy and Joshi (1978).

## REFERENCES

1 Barton, E. (1984) "Toward a Principle-Based Parser,v A.I. Memo No. 788, MIT. Artificial Intelligence Laboratory, Cambridge, Mass

[2] Chomsky, N. (1955). *The Logical Structure of Linguistic Theory,* published in 1975 by Plenum Press (New York).

\'A\ Chomsky, N. (1981). *Lectures on Government and Binding* Dordrecht, Holland: Foris Publications.

!-I| Goodall, G. (1984) "Coordination," PhD dissertation, Department of Linguistics, University of California at San Diego

|5] Kroch, A , and A. Joshi (1985) "The Linguistic Relevance of Tree Adjoining Grammars," Technical Report No. MS-C1S-85-10, Department of Computer and Information Science, Moore School, University of Pennsylvania, Philadephia, Penn.

|6I Lasnik, H , and J. Kupin (1977) "A Restrictive Theory of Transformational Grammar," *Theoretical Linguistics* 4:3 J 73- 196.

|7j Levy, L, and A Joshi (1978) "Skeletal Structural Descriptions, " *Information and Control* 39:192 211.

[8j Marcus, M (1980) *A Theory of Syntactic Recognition for Natural Language.* Cambridge, Mass.: MIT. Press

J9i Marcus, M., D. Hindle, and M. Fleck (1983) "D-Theory: Talking about Talking about Trees," ACL '83 proceedings.

|l0i Marr, D. (1982). *Vision.* San Francisco. W. H. Freeman and Company

[Il] Nirenburg, S , and C. Attiya (1984). "Interruptible Transition Networks," COLING '84 proceedings, pp. 393-397

112) Reuland, E. (1984). "Features for the Set of Categonal Heads," ms.