

LICENSING AND TREE ADJOINING GRAMMAR IN GOVERNMENT BINDING PARSING

Robert Frank*

Department of Computer and Information Sciences
University of Pennsylvania
Philadelphia, PA 19104
email: frank@linc.cis.upenn.edu

Abstract

This paper presents an implemented, psychologically plausible parsing model for Government Binding theory grammars. I make use of two main ideas: (1) a generalization of the licensing relations of [Abney, 1986] allows for the direct encoding of certain principles of grammar (e.g. Theta Criterion, Case Filter) which drive structure building; (2) the working space of the parser is constrained to the domain determined by a Tree Adjoining Grammar elementary tree. All dependencies and constraints are localized within this bounded structure. The resultant parser operates in linear time and allows for incremental semantic interpretation and determination of grammaticality.

1 Introduction

This paper aims to provide a psychologically plausible mechanism for putting the knowledge which a speaker has of the syntax of a language, the competence grammar, to use. The representation of knowledge of language I assume is that specified by Government Binding (GB) Theory introduced in [Chomsky, 1981]. GB, as a competence theory, emphatically does not specify the nature of the language processing mechanism. In fact, "proofs" that transformational grammar is inadequate as a linguistic theory due to various performance measures are fundamentally flawed since they suppose a particular connection between the grammar and parser [Berwick and Weinberg, 1984]. Nonetheless, it seems desirable to maintain a fairly direct connection between the linguistic competence and

its processing. Otherwise, claims of the psychological reality of this particular conception of competence become essentially vacuous since they cannot be falsified but for the data on which they are founded, i.e. grammaticality judgments. Thus, in building a model of language processing, I would like to posit as direct a link as is possible between linguistic competence and the operations of the parser while still maintaining certain desirable computational properties.

What are the computational properties necessary for psychological plausibility? Since human syntactic processing is an effortless process, we should expect that it take place efficiently, perhaps in linear time since sentences do not become more difficult to process simply as a function of their length. Determinism, as proposed by Marcus [1980], seems desirable as well. In addition, the mechanism should operate in an incremental fashion. Incrementality is evidenced in the human language processor in two ways. As we hear a sentence, we build up semantic representations without waiting until the sentence is complete. Thus, the semantic processor should have access to syntactic representations prior to an utterance's completion. Additionally, we are able to perceive ungrammaticality in sentences almost immediately after the ill formedness occurs. Thus, our processing mechanism should mimic this early detection of ungrammatical input.

Unfortunately, a parser with the most transparent relationship to the grammar, a "parsing as theorem proving" approach as proposed by [Johnson, 1988] and [Stabler, 1990], does not fare well with respect to our computational desiderata. It suffers from the legacy of the computational properties of first order theorem proving, most notably undecidability, and is thus inadequate for our purposes. The question, then, is how much we must retreat from this direct instantiation so that we can maintain the requisite properties. In this paper, I attempt to provide

*I would like to thank the following for their valuable discussion and suggestions: Naoki Fukui, Jamie Henderson, Aravind Joshi, Tony Kroch, Mitch Marcus, Michael Niv, Yves Schabes, Mark Steedman, Enric Vallduví. This work was partially supported by ARO Grants DAAL03-89-C0031 PRI and DAAG29-84-K-0061 and DARPA grant N00014-85-K-0018. The author is supported by a Unisys doctoral fellowship.

an answer. I propose a parsing model which represents the principles of the grammar in a fairly direct manner, yet preserves efficiency and incrementality. The model depends upon two key ideas. First, I utilize the insight of [Abney, 1986] in the use of *licensing relations* as the foundation for GB parsing. By generalizing Abney's formulation of licensing, I can directly encode and enforce a particular class of the principles of GB theory and in so doing efficiently build phrase structure. The principles expressible through licensing are not all of those posited by GB. Thus, the others must be enforced using a different mechanism. Unfortunately, the unbounded size of the tree created with licensing makes any such mechanism computationally abhorrent. In order to remedy this, I make use of the Tree Adjoining Grammar (TAG) framework [Joshi, 1985] to limit the working space of the parser. As the parser proceeds, its working structure is bounded in size. If this bound is exceeded, we reduce this structure by one of the operations provided by the TAG formalism, either substitution or adjunction. This results in two structures, each of which form independent elementary trees. Interestingly, the domain of locality imposed by a TAG elementary tree appears to be sufficient for the expression of the remaining grammatical principles. Thus, we can check for the satisfaction of the remaining grammatical principles in just the excised piece of structure and then send it off for semantic interpretation. Since this domain of constraint checking is bounded in size, this process is done efficiently. This mechanism also works in an incremental fashion.

2 Abney's Licensing

Since many grammatical constraints are concerned with the licensing of elements, Abney [1986] proposes utilizing licensing structure as a more concrete representation for parsing. This allows for more efficient processing yet maintains "the spirit of the abstract grammar."

Abney's notion of licensing requires that every element in a structure be licensed by performing some syntactic function. Any structure with unlicensed elements is ill-formed. Abney takes theta role assignment to be the canonical case of licensing and assumes that the properties of a general licensing relation should mirror those of theta assignment, namely, that it be *unique*, *local* and *lexical*. The uniqueness property for theta assignment requires that an argument receives one and only one theta role. Correspondingly, licensing is unique: an element is licensed via exactly one licensing relation. Locality demands that theta assignment, and correspondingly licensing, take place under a strict definition of government: sisterhood. Finally,

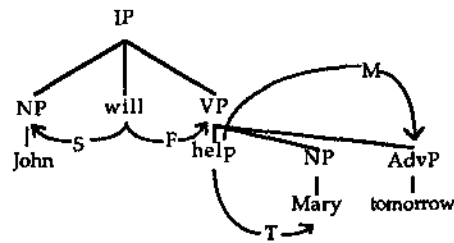


Figure 1: Abney's Licensing Relations in Clausal Structure (S = subjecthood, F = functional selection, M = modification, T = theta)

theta assignment is lexical in that it is the properties of the the theta assigner which determine what theta assignment relations obtain. Licensing will have the same property; it is the licenser that determines how many and what sort of elements it licenses.

Each licensing relation is a 3-tuple $\langle D, Cat, Type \rangle$. D is the direction in which licensing occurs. Cat is the syntactic category of the element licensed by this relation. $Type$ specifies the linguistic function accomplished by this licensing relation. This can be either *functional selection*, *subjecthood*, *modification* or *theta-assignment*. Functional selection is the relation which obtains between a functional head and the element for which it subcategorizes, i.e. between C and IP, I and VP, D and NP. Subjecthood is the relation between a head and its "subject". Modification holds between a head and adjunct. Theta assignment occurs between a head and its subcategorized elements. Figure 1 gives an example of the licensing relations which might obtain in a simple clause. Parsing with these licensing relations simply consists of determining, for each lexical item as it is read in a single left to right pass, where it is licensed in the previously constructed structure or whether it licenses the previous structure.

We can now re-examine Abney's claim that these licensing relations allow him to retain "the spirit of the abstract grammar." Since licensing relations talk only of very local relationships, that occurring between sisters, this system cannot enforce the constraints of binding, control, and ECP among others. Abney notes this and suggests that his licensing should be seen as a single module in a parsing system. One would hope, though, that principles which have their roots in licensing, such as those of theta and case theory, could receive natural treatments. Unfortunately, this is not true. Consider the theta criterion. While this licensing system is able to encode the portion of the constraint that requires theta roles to be assigned uniquely, it fails to guarantee that all NPs (arguments) receive a theta

role. This is crucially not the case since NPs are sometimes licensed not by theta but by subject licensing. Thus, the following pair will be indistinguishable:

- i. It seems that the pigeon is dead
- ii. * Joe seems that the pigeon is dead

Both *It* and *Joe* will be appropriately licensed by a subject licensing relation associated with *seems*. The case filter also cannot be expressed since objects of ECM verbs are “licensed” by the lower clause as subject, yet also require case. Thus, the following distinction cannot be accounted for:

- i. Carol asked Ben to swat the fly
- ii. * Carol tried Ben to swat the fly

Here, in order to get the desired syntactic structure (with *Ben* in the lower clause in both cases), *Ben* will need to be licensed by the inflectional element *to*. Since such a licensing relation need be unique, the case assigning properties of the matrix verbs will be irrelevant. What seems to have happened is that we have lost the modularity of the syntactic relations constrained by grammatical principles. Everything has been conflated onto one homogeneous licensing structure.

3 Generalized Licensing

In order to remedy these deficiencies, I propose a system of *Generalized Licensing*. In this system, every node is assigned two sets of licensing relations: *gives* and *needs*. Gives are similar to the Abney’s licensing relations: they are satisfied locally and determined lexically. Needs specify the ways in which a node must be licensed.¹ A need of type theta, for example, requires a node to be licensed by a theta relation. In the current formulation, needs differ from gives in that they are always directionally unspecified. We can now represent the theta criterion by placing theta gives on a theta assigner for each argument and theta needs on all DPs. This encodes both that all theta roles must be assigned and that all arguments must receive theta roles.

In Generalized Licensing, we allow a greater vocabulary of relation types: case, theta assignment, modification, functional selection, predication, f-features, etc. We can then explicitly represent many of the relations which are posited in the grammar and preserve the modularity of the theory. As a result, however, certain elements can and must be multiply licensed. DPs, for instance, will have needs for both theta and case as a result of the case filter and theta criterion. We therefore relax the requirement that

all nodes be uniquely licensed. Rather, we demand that all gives and needs be uniquely “satisfied.” The uniqueness requirement in Abney’s relations is now pushed down to the level of individual gives and needs. Once a give or need is satisfied, it may not participate in any other licensing relationships.

One further generalization which I make concerns the positioning of gives and needs. In Abney’s system, licensing relations are associated with lexical heads and applied to maximal projections of other heads. Phrase structure is thus entirely parasitic upon the reconstruction of licensing structure. I propose to have an independent process of lexical projection. A lexical item projects to the correct number of levels in its maximal projection, as determined by theta structure, f-features, and other lexical properties.² Gives and needs are assigned to each of these nodes. As with Abney’s system, licensing takes place under a strict notion of government (sisterhood). However, the projection process allows licensing relations determined by a head to take place over a somewhat larger domain than sisterhood to the head. A DP’s theta need resulting from the theta criterion, for example, is present only at the maximal projection level. This is the node which stands in the appropriate structural relation to a theta give. As a result of this projection process, though, we must explicitly represent structural relations during parsing.

The reader may have noticed that multiple needs on a node might not all be satisfiable in one structural position. Consider the case of a DP subject which possesses both theta and case needs. The S-structure subject of the sentence receives its theta role from the verb, yet it receives its case from the tense/agreement morpheme heading IP. This is impossible, though, since given the structural correlate of the licensing relation, the DP would then be directly dominated both by IP and by VP. Yet, it cannot be in either one of these positions alone, since we will then have unsatisfied needs and hence an ill-formed structure. Thus, our representation of grammatical principles and the constraints on give and need satisfaction force us into adopting a general notion of chain and more specifically the VP internal subject hypothesis. A chain consists of a list of nodes $\langle a_1, \dots, a_n \rangle$ such that they share gives and needs and each a_i c-commands each a_{i+1} . The first element in the chain, a_1 , the head, is the only element which can have phonological content. Others must be empty categories.

Now, since the elements of the chain can occupy different structural positions, they may be governed and hence licensed by distinct elements. In the simple sentence:

[_{IP} John_i t_{ns/agr} [_{VP} t_i smile]]

¹These bear some similarity to the anti-relations of Abney, but are used in a rather different fashion.

²I assume the relativized X-bar theory proposed in [Fukui and Speas, 1986].

the trace node which is an argument of *smile* forms a chain with the DP *John*. In its V' internal position, the theta need is satisfied by the theta give associated with the V. In subject position, the case need is satisfied by the case give on the I' projection of the inflectional morphology.

Now, how might we parse using these licensing relations? Abney's method is not sufficient since a single instance of licensing no longer guarantees that all of a node's licensing constraints are satisfied. I propose a simple mechanism, which generalizes Abney's approach: We proceed left to right, project the current input token to its maximal projection *p* and add the associated gives and needs to each of the nodes. These are determined by examination of information in the lexical entries (such as using the theta grid to determine theta gives), examination of language specific parameters (using head directionality in order to determine directionality of gives, for example), and consultation of UG parameters (for instance as a result of the case filter, every DP maximal projection will have an associated case need). The parser then attempts to combine this projection with previously built structure in one of two ways. We may attach *p* as the sister of a node *n* on the right frontier of the developing structure, when *p* is licensed by *n* either by a give in *n* and/or a need in the node *p*. Another possibility is that the previously built structure is attached as sister to a node, *m*, dominated by the maximal projection *p*, by satisfying a give in *m* and/or a need on the root of the previously built structure. In the case of multiple attachment possibilities, we order them according to some metric such as the one proposed by Abney, and choose the most highly ranked option.

As structure is built, nodes in the tree with unsatisfied gives and needs may become closed off from the right frontier of the working structure. In such positions, they will never become satisfied. In the case of a need in an internal node *n* which is unsatisfied, we posit the existence of an empty category *m*, which will be attached later to the structure such that (n, m) form a chain. We posit an element to have been moved into a position exactly when it is licensed at that position yet its needs are not completely satisfied. After positing the empty category, we push it onto the *trace stack*. When a node has an unsatisfied give and no longer has access to the right frontier, we must posit some element, not phonologically represented in the input, which satisfies that give relation. If there is an element on the top of the trace stack which can satisfy this give, we pop it off the stack and attach it.³ Of course, if the trace has any remaining needs, it is returned to the trace stack since its new position is isolated from the right frontier. If no such element appears on top of the trace stack, we

³Note that the use of this stack to recover filler-gap structures forbids non-nested dependencies as in [Fodor, 1978].

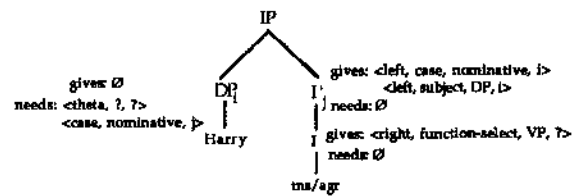


Figure 2: Working Space after "Harry tns/agr"

posit a non-trace empty category of the appropriate type, if one exists in the language.⁴

Let's try this mechanism on the sentence "Harry laughs." The first token received is *Harry* and is projected to DP. No gives are associated with this node, but theta and case needs are inserted into the need set as a result of the theta criterion and the case filter. Next, *tns/agr* is read and projected to I', since it possesses f-features (cf. [Fukui and Speas, 1986]). Associated with the I' node is a rightward functional selection give of value V. On the I' node is a leftward nominative case give, from the f-features, and a leftward subject give, as a result of the Extended Projection Principle. The previously constructed DP is attached as sister to the I' node, thereby satisfying the subject and case gives of the I' as well as the case need of the DP. We are thus left with the structure in figure 2.⁵ Next, we see that the theta need of the DP is inaccessible from the right frontier, so we push an empty category DP whose need set contains this unsatisfied theta need onto the trace stack. The next input token is the verb *laugh*. This is projected to a single bar level. Since *laugh* assigns an external theta role, we insert a leftward theta give to a DP into the V' node. This verbal projection is attached as sister to I', satisfying the functional selection give of I. However, the theta give in V' remains unsatisfied and since it is leftward, is inaccessible. We therefore need to posit an empty category. Since the DP trace on top of the trace stack will accept this give, the trace stack is popped and the trace is attached via Chomsky-adjunction to the

⁴Such a simplistic approach to determining whether a trace or non-trace empty category should be inserted is clearly not correct. For instance, in "tough movement"

Alvin_i is tough PRO to feed t_i

the proposed mechanism will insert the trace of *Alvin* in subject position rather than PRO. It remains for future work to determine the exact mechanism by which such decisions are made.

⁵In the examples which follow, gives are shown as 4-tuples $\langle D, Type, Val, SatBy \rangle$ where *D* is the direction, *Type* is the type of licensing relation, *Val* is the licensing relation value and *SatBy* is the node which satisfies the give (marked as ?, if the relation is as yet unsatisfied). Needs are 3-tuples $\langle Type, Val, SatBy \rangle$ where these are as in the gives. For purposes of readability, I remove previously satisfied gives and needs from the figure. Of course, such information persists in the parser's representation.

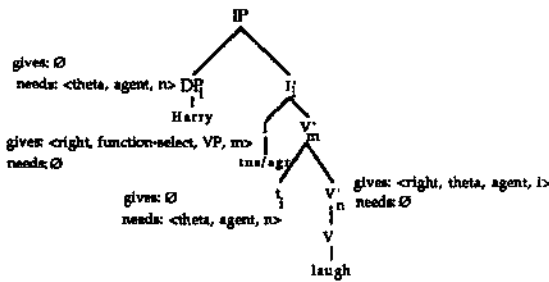


Figure 3: Working space after "Harry tns/agr laugh"

V' node yielding the structure in figure 3. Since this node forms a chain with the subject DP, the theta need on the subject DP is now satisfied. We have now reached the end of our input. The resulting structure is easily seen to be well-formed since all gives and needs are satisfied.

We have adopted a very particular view of traces: their positions in the structure must be independently motivated by some other licensing relation. Note, then, that we cannot analyze long distance dependencies through successive cyclic movement. There is no licensing relation which will cause the intermediate traces to exist. Ordinarily these traces exist only to allow a well-formed derivation, i.e. not ruled out by subjacency or by a barrier to antecedent government. Thus, we need to account for constraints on long distance movement in another manner. We will return to this in the next section.

The mechanism I have proposed allows a fairly direct encoding for some of the principles of grammar such as case theory, theta theory, and the extended projection principle. However, many other constraints of GB, such as the ECP, control theory, binding theory, and bounding theory, cannot be expressed perspicuously through licensing. Since we want our parser to maintain a fairly direct connection with the grammar, we need some additional mechanism to ensure the satisfaction of these constraints.

Recall, again, the computational properties we wanted to hold of our parsing model: efficiency and incrementality. The structure building process I have described has worst case complexity $O(n^2)$ since the set of possible attachments can grow linearly with the input. While not enormously computationally intensive, this is greater than the linear time bound we desire. Also, checking for satisfaction of non-licensing constraints over unboundedly large structures is likely to be quite inefficient. There is also the question of when these other constraints are checked. To accord with incrementality, they must be checked as soon as possible, and not function as post-processing "filters." Unfortunately, it is not easily determinable when a given constraint can apply such that further input will not

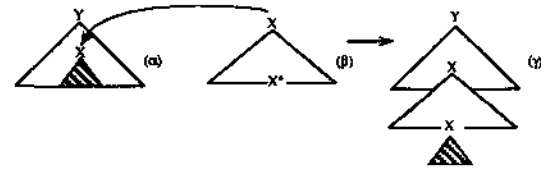


Figure 4: Adjunction of auxiliary tree β into elementary tree α to produce γ

change the status of the satisfaction of a constraint. We do not want to rule a structure ungrammatical simply because it is incomplete. Finally, it is unclear how we might incorporate this mechanism which builds an ever larger syntactic structure into a model which performs semantic interpretation incrementally.

4 Limiting the Domain with TAG

These problems with our model are solved if we can place a limit on the size of the structures we construct. The number of licensing possibilities would be bounded yielding linear time for structure construction. Also, constraint checking could be done in a constant amount of processing. Unfortunately, the productivity of language requires us to handle sentences of unbounded length and thus linguistic structures of unbounded size.

TAG provides us with a way to achieve this paradise. TAG accomplishes linguistic description by factoring recursion from local dependencies [Joshi, 1985]. It posits a set of primitive structures, the *elementary trees*, which may be combined through the operations of *adjunction* and *substitution*. An elementary tree is a minimal non-recursive syntactic tree, a predication structure containing positions for all arguments. I propose that this is the projection of a lexical head together with any of the associated functional projections of which it is a complement. For instance, a single elementary tree may contain the projection of a V along with the I and C projections in which it is embedded.⁶ Along the frontier of these trees may appear terminal symbols (i.e. lexical items) or non-terminals. The substitution operation is the insertion of one elementary tree at a non-terminal of same type as the root on the frontier of another elementary tree. Adjunction allows the insertion of one elementary tree of a special kind, an auxiliary tree, at a node internal to another (cf. figure 4). In

⁶This definition of TAG elementary trees is consistent with the Lexicalized TAG framework [Schabes *et al.*, 1988] in that the lexical head may be seen as the anchor of the elementary trees. For further details and consequences of this proposal on elementary tree well-formedness, see [Frank, 1990].

auxiliary trees, there is a single distinguished non-terminal on the frontier of the tree, the *foot node*, which is identical in type to the root node. Only adjunctions, and not substitutions, may occur at this node.

TAG has proven useful as a formalism in which one can express linguistic generalizations since it seems to provide a sufficient domain over which grammatical constraints can be stated [Kroch and Joshi, 1985] [Kroch and Santorini, 1987]. Kroch, in two remarkable papers [1986] and [1987], has shown that even constraints on long distance dependencies, which intuitively demand a more “global” perspective, can be expressed using an entirely local (i.e. within a single elementary tree) formulation of the ECP and allows for the collapsing of the CED with the ECP. This analysis does not utilize intermediate traces, but instead the link between filler and gap is “stretched” upon the insertion of intervening structure during adjunctions. Thus, we are relieved of the problem that intermediate traces are not licensed, since we do not require their existence.

Let us suppose a formulation of GB in which all principles not enforced through generalized licensing are stated over the local domain of a TAG elementary tree. Now, we can use the model described above to create structures corresponding to single elementary trees. However, we restrict the working space of the parser to contain only a single structure of this size. If we perform an attachment which violates this “memory limitation,” we are forced to reduce the structure in our working space. We will do this in one of two ways, corresponding to the two mechanisms which TAG provides for combining structure. Either we will undo a substitution or undo an adjunction. However, all chains are required to be localized in individual elementary tree. Once an elementary tree is finished, non-licensing constraints are checked and it is sent off for semantic interpretation. This is the basis for my proposed parsing model. For details of the algorithm, see [Frank, 1990]. This mechanism operates in linear time and deterministically, while maintaining coarse grained (i.e. clausal) incrementality for grammaticality determination and semantic interpretation.

Consider this model on the raising sentence “Harry seemed to kiss Sally.” We begin as before with “Harry tns/agr” yielding the structure in figure 2. Before we receive the next token of input, however, we see that the working structure is larger than the domain of an elementary tree, since the subject DP constitutes an independent predication from the one determined by the projection of I. We therefore unsubstite the subject DP and send it off to constraint checking and semantic interpretation. At this point, we push a copy of the subject DP node onto the trace stack due to its unsatisfied theta need.

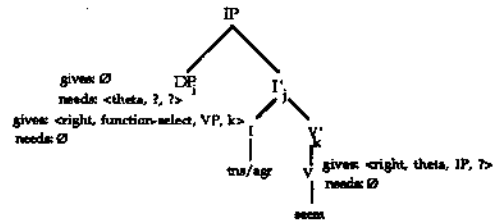


Figure 5: Working space after “Harry tns/agr seem”

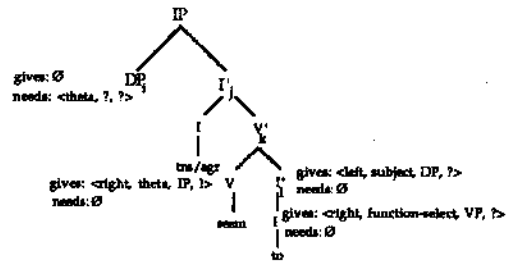


Figure 6: Working space after “Harry tns/agr seem to”

We continue with the verb *seem* which projects to V' and attaches as sister to I satisfying the functional selection give yielding the structure in figure 5. There remains only one elementary tree in working space so we need not perform any domain reduction. Next, *to* projects to I' since it lacks f-features to assign to its specifier. This is attached as object of *seem* as in figure 6. At this point, we must again perform a domain reduction operation since the upper and lower clauses form two separate elementary trees. Since the subject DP remains on the trace stack, it cannot yet be removed. All dependencies must be resolved within a single elementary tree. Hence, we must unadjoin the structure recursive on I' shown in figure 7 leaving the structure in figure 8 in the working space. This structure is sent off for constraint checking and semantic interpretation. We continue with *kiss*, projecting and attaching it as functionally selected sister of I and popping the DP from the trace stack to serve as external argument. Finally, we

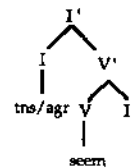


Figure 7: Result of unadjunction

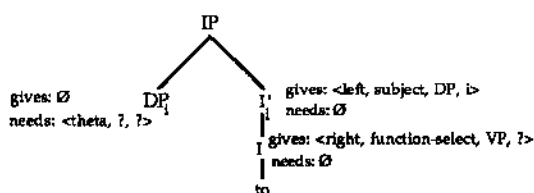


Figure 8: Working space after unadjunction

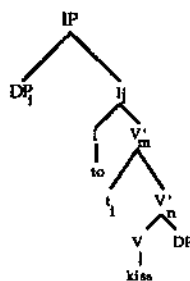


Figure 9: Working Structure after entire sentence

project and attach the DP *Sally* as sister of V, receiving both theta role and case in this position. This DP is unsubstituted in the same manner as the subject and is sent off for further processing. We are left finally with the structure in figure 9, all of whose gives and needs are satisfied, and we are finished.

This model also handles control constructions, bare infinitives, ECM verbs and binding of anaphors, modification, genitive DPs and others. Due to space constraints, these are not discussed here, but see [Frank, 1990].

5 Problems and Future Work

The parsing model which I have presented here is still rather preliminary. There are a number of areas which will require further development before this can be considered complete.

I have assumed that the process of projection is entirely determined from lexical lookup. It is clear, though, that lexical ambiguity abounds and that the assignment of gives and needs to the projections of input tokens is not determinate. An example of such indeterminacy has to do with the assignment to argument maximal projections of theta needs as a result of the theta criterion. DPs need not always function as arguments, as I have been assuming. This problem might be solved by allowing for the statement of disjunctive constraints or a limited form of parallelism. If the duration of such parallelism could be tightly

constrained, we might be able to retain the efficient nature of the current model. Other strategies for resolving such indeterminacies using statistical reasoning or hard coded rules or templates might also be possible, but these constructs are not the sort of grammatical knowledge we have been considering here and would entail further abstraction from the competence grammar.

Another problem with the parser has to do with the incompleteness of the algorithm. Sentences such as

Boris knew that Tom ate lunch

will not be parsed even though there exist well-formed sets of elementary trees which can derive them. The problem results from the fact that the left to right processing strategy we have adopted is a bit too strict. The complementizer *that* will be attached as object of *know*, but *Tom* is not then licensed by any node on the right frontier. Ultimately, this DP is licensed by the *tns/agr* morpheme in the lower clause whose IP projection is licensed through functional selection by C. Similarly, the parser would have great difficulty handling head final languages. Again, these problems might be solved using extra-grammatical devices, such as the attention shifting of [Marcus, 1980] or some template matching mechanism, but this would entail a process of “compiling out” of the grammar that we have been trying to avoid.

Finally, phonologically empty heads and head movement cause great difficulties for this mechanism. Heads play a crucial role in this “project and attach” scheme. Therefore, we must find a way of determining when and where heads occur when they are either dislocated or not present in the input string at all, perhaps in a similar manner to the mechanism for movement of maximal projections I have proposed above.

6 Conclusion

In this paper, I have sketched a psychologically plausible model for the use of GB grammars. The currently implemented parser is a bit too simple to be truly robust, but the general approach presented here seems promising. Particularly interesting is that the *computationally* motivated use of TAG to constrain processing locality provides us with insight on the nature of the meta-grammar of possible grammatical constraints. Thus, if grammatical principles are stated over such a bounded domain, we can guarantee the existence of a perspicuous model for their use, thereby lending credence to the cognitive reality of this competence grammar.

References

- [Abney, 1986] Steven Abney. Licensing and parsing. In *Proceedings of NELS 16*, Amherst, MA.
- [Berwick and Weinberg, 1984] Robert Berwick and Amy Weinberg. *The Grammatical Basis of Linguistic Performance*. MIT Press, Cambridge, MA.
- [Chomsky, 1981] Noam Chomsky. *Lectures on Government and Binding*. Foris, Dordrecht.
- [Fodor, 1978] Janet D Fodor. Parsing strategies and constraints on transformations. *Linguistic Inquiry*, 9.
- [Frank, 1990] Robert Frank. *Computation and Linguistic Theory: A Government Binding Theory Parser Using Tree Adjoining Grammar*. Master's thesis, University of Pennsylvania.
- [Fukui and Speas, 1986] Naoki Fukui and Margaret Speas. Specifiers and projection. In Naoki Fukui, T. Rappaport, and E. Sagey, editors, *MIT Working Papers in Linguistics 8*, MIT Department of Linguistics.
- [Johnson, 1988] Mark Johnson. Parsing as deduction: the use of knowledge of language. In *The MIT Parsing Volume, 1987-88*, MIT Center for Cognitive Science.
- [Joshi, 1985] Aravind Joshi. How much context-sensitivity is required to provide reasonable structural descriptions: tree adjoining grammars. In D. Dowty, L. Karttunen, and A. Zwicky, editors, *Natural Language Processing: Psycholinguistic, Computational and Theoretical Perspectives*, Cambridge University Press.
- [Kroch, 1986] Anthony Kroch. Unbounded dependencies and subjacency in a tree adjoining grammar. In A. Manaster-Ramer, editor, *The Mathematics of Language*, John Benjamins.
- [Kroch, 1987] Anthony Kroch. Assymetries in long distance extraction in a tree adjoining grammar. manuscript, University of Pennsylvania.
- [Kroch and Joshi, 1985] Anthony Kroch and Aravind Joshi. *The Linguistic Relevance of Tree Adjoining Grammar*. Technical Report MS-CS-85-16, University of Pennsylvania Department of Computer and Information Sciences. To appear in *Linguistics and Philosophy*.
- [Kroch and Santorini, 1987] Anthony Kroch and Beatrice Santorini. The derived constituent structure of the west germanic verb raising construction. In R. Freidin, editor, *Proceedings of the Princeton Conference on Comparative Grammar*, MIT Press, Cambridge, MA.
- [Marcus, 1980] Mitchell Marcus. *A Theory of Syntactic Recognition for Natural Language*. MIT Press, Cambridge, MA.
- [Schabes et al., 1988] Yves Schabes, Anne Abeillé, and Aravind K. Joshi. Parsing strategies with 'lexicalized' grammars: application to tree adjoining grammars. In *COLING Proceedings*, Budapest.
- [Stabler, 1990] Edward Stabler. Implementing government binding theories. In Levine and Davis, editors, *Formal Linguistics: Theory and Implementation*. forthcoming.