University of Pennsylvania Working Papers in Linguistics

Volume 4 | Issue 3 Article 7

1-1-1997

Exceptional Case Marking in the Xtag System

Seth Kulick

Exceptional Case Marking in the Xtag System

Exceptional Case Marking in the Xtag System

Seth Kulick

1. Introduction

This paper describes an implementation of Exceptional Case Marking (ECM) verbs for the Xtag system[1]. Xtag is a parser based on an implemention of Tree Adjoining Grammar (TAG) extended with a feature-based unification mechanism. A TAG analysis of ECM verbs and raising passives was first set out, in rough form, in Kroch and Joshi (1985). However, the exceptional nature of ECM verbs causes certain problems when actually implementing such an analysis in Xtag, and we describe here the problems encountered and their solution. Furthermore, we extend the earlier analysis to handle verbs that take bare infinitive complements. The Xtag implementation allows for a clean solution to the problem of the lack of a bare infinitive passive.

As a quick review, ECM verbs are those which appear to mark the subject of the infinitival complement with accusative Case, as in (1a), in contrast with control verbs such as in (1b), in which *Bob* is a thematic object of the verb. The passivization of an ECM verb causes the external theta role assignment to be suppressed along with the the exceptional accusative Case assignment. This results in a "raising passive", as in (2a), analogous to a regular raising verb such as in (2b).

(1) a. Van expects [Bob to talk]
    b. Van persuaded Bob [PRO to talk]
(2) a. Bob was expected to talk
    b. Bob seems to talk

Now consider the class of verbs that take bare infinitive complements, as in (3a)[2]. A surprising feature of these verbs is that in the passive, they can no longer take the bare infinitive, but must take a full infinitive, as illustrated in (3bc).

---

[1]I would like to thank the members of the Xtag project for their advice, and their insistence that I get this work done, and two anonymous reviewers. This work was supported by NSF grant SBR8920230 and ARO grant DAAH04-94-G-0426.

[2]These are like ECM verbs in that the subject of the complement receives Case from the matrix verb while not being theta-marked by the matrix verb, but unlike ECM in that complement is not usually considered to have as large a projection (say, only VP) and so the Case assignment is not so "exceptional".

(3) a. Bob sees the harmonica fall
b. * The harmonica was seen fall
c. The harmonica was seen to fall

In section 2 we give a brief overview of TAG and the associated feature system used in Xtag, with examples of raising and sentential complementation. In section 3 we describe the Xtag implementation of ECM verbs with infinitival complements and of verbs taking bare infinitive complements. Section 4 describes how the passive for both classes is implemented, and how the bare infinitive passive problem is handled. In conclusion, Section 5 discusses how the Xtag implementation compares to some recently proposed analyses in the Minimalist framework.

2. Basics of the Xtag System

The Xtag system is based on the TAG formalism developed in Joshi et al. (1975), Kroch and Joshi (1985), extended with a feature-based unification system as in Vijay-Shanker and Joshi (1991), The XTAG-Group (1995). These references should be consulted for more detail than can be presented here.

TAG was introduced in Joshi et al. (1975) as a formalism with interesting mathematical properties, and it has since (e.g., Kroch and Joshi (1985) and Frank (1992)) been argued that TAG allows linguistically attractive analyses to be stated in natural ways. The essential idea is that TAG allows for recursion to be separated from the specification of a grammar, thus allowing, and requiring, the substantive theory of syntax to be confined to the domain of ELEMENTARY TREES, the primitive elements of the TAG formalism.

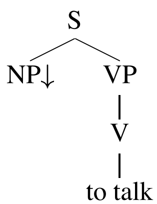
The ELEMENTARY TREES, are of two types: INITIAL TREES and AUXILIARY TREES. In a TAG grammar for natural language, INITIAL TREES are phrase structure trees of simple sentences containing no recursion, while recursive structures are represented by AUXILIARY TREES[3]. Elementary trees are combined by the operations of SUBSTITUTION and ADJUNCTION. Substitution inserts elementary trees into substitution nodes that appear on the

[3]A reviewer asks how this notion of recursion is "linked to GB/minimalist analyses which only have finite rules of phrase structure combined with movement." As will hopefully become clear, adjunction of auxiliary trees allow the elimination of cyclic movement. So instead of a derivation of *John*$_i$ *seems* t_i *to be certain* t_i *to like pizza* with *John* moving successively from each clause, in TAG (and Xtag) there is an elementary tree for *John to like pizza*, with *seems* and *to be certain* both adjoining in. There is no "movement" at all of *John*. This is discussed in more detail shortly.

frontier of other elementary trees. Adjunction grafts auxiliary trees into elementary trees at the node whose label is the same as the root and foot labels of the auxiliary tree.

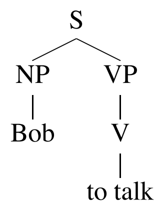
2.1. An Example of Raising in TAG

(4)(a)



(b)

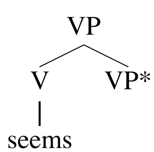
(5)



The trees in (4a) and (4b) are both initial trees. We are assuming a simplified phrase structure for the purpose of explaining the TAG system. Nodes on the frontier of initial trees are marked as substitution sites by a (↓). The tree (4b) substitutes into the NP substitution nodes in (4a) to give the result in (5).

This use of substitution (essentially a tree-substitution grammar) is clearly no big deal, and by itself would be of no interest as a grammar formalism. What makes TAG interesting is the use of the adjunction operation, which can be used to give the effect of movement by "stretching" components of one tree away from the rest of the tree. Exactly one node on the frontier of an auxiliary tree, whose label matches the label of the root of the tree, is marked as a foot node by a (∗). The adjunction operation takes an auxiliary tree and inserts it into the body of another tree at a node of the same label as the foot and root nodes of the auxiliary, as shown in Figure 1.

(6)



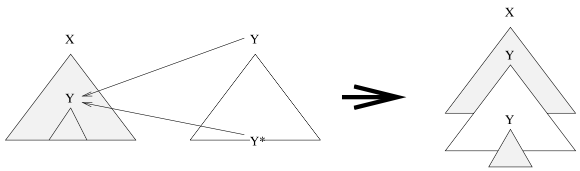
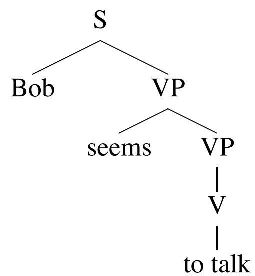


Figure 1: The Adjunction operation

(7)



For example, again with a simplified phrase structure, (6) is a tree for the raising verb *seems* and is an auxiliary tree, with both the root and foot nodes being VP. When (6) is adjoined at the VP node of (5), the result is the tree in (7). There is no operation of "movement" from one tree to another. Furthermore, the operation can be recursive, as in *Bob seems to be certain to talk*. The derivation of this would proceed by having a raising tree for *to be certain* that is roughly the same as the one for *seems* (leaving aside issues of tense), and the tree for *seems* would adjoin into that for *to be certain*, deriving a complex auxiliary tree for *seems to be certain*, which would then adjoin into (5)[4].

This example abstracted away from many necessary components of a grammar. For example, there is no indication of tense, agreement, Case assignment, etc. TAG as a formalism makes no claims about how these are to be handled, and in general has nothing to say about the character of the elementary trees. TAG only provides the machinery to combine elementary trees once they are specified.

[4]This is a simplified description. Technically, TAG only allows adjunction of a single tree into another, not of a complex derived tree into another tree. So a precise description of the derivation consists of two simple adjunctions - *seems* into *to be certain*, and *to be certain* into *Bob to talk*.

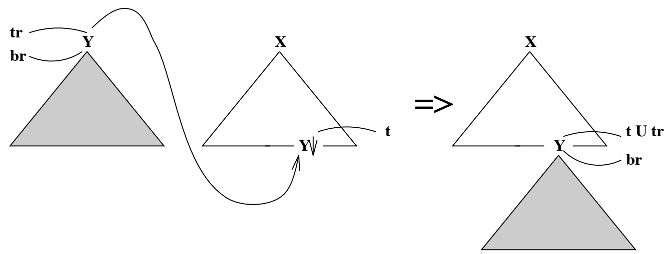


Figure 2: Schemata for feature formation upon substitution
[t=top b=bottom r=root f=foot U=unification]

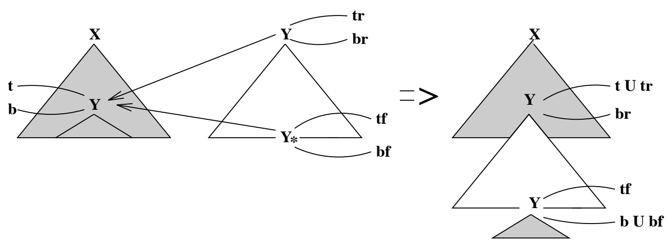


Figure 3: Schemata for feature formation upon adjunction
[t=top b=bottom r=root f=foot U=unification]

The Xtag system takes one particular approach to how these details should be handled. Feature structures are added to the basic TAG formalism by associating a feature structure with each node in an elementary tree. It consists of a top part, which expresses the constraints specified by the structure above the node, and a bottom part, which expresses the constraints specified by the subtree associated with the node. Substitution nodes, however, have only the top features, since the tree substituting in carries the bottom features. When substitution is performed at a node, the features are formed as shown in Figure 2. When adjunction is performed, the node is "split", with the features formed as shown in Figure 3. At the end of a derivation, the top and bottom features of each node must unify.

2.2. Raising in Xtag

We'll now reconsider the previous example as it is derived in the Xtag system, with feature values. Consider the trees in Figure 4. Aside from the features,

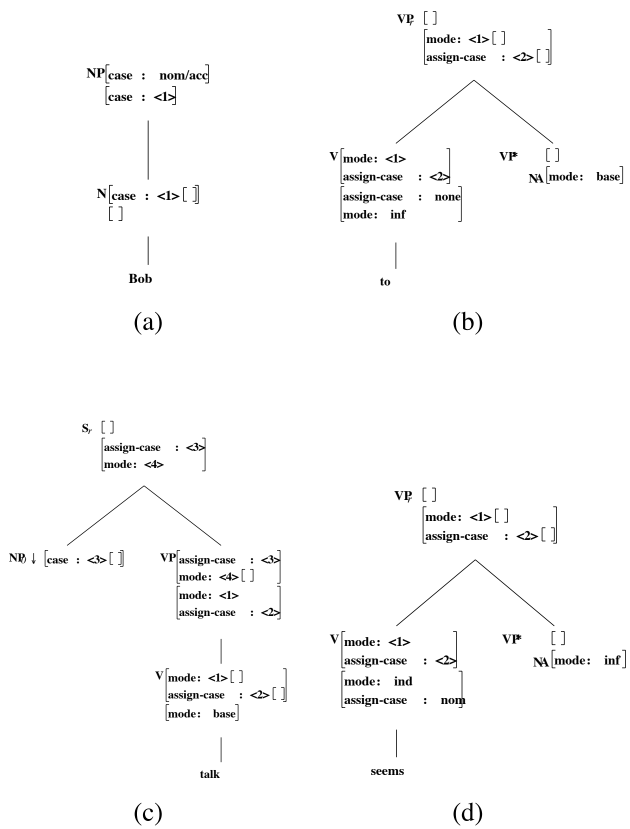


Figure 4: Component trees for *Bob seems to talk*

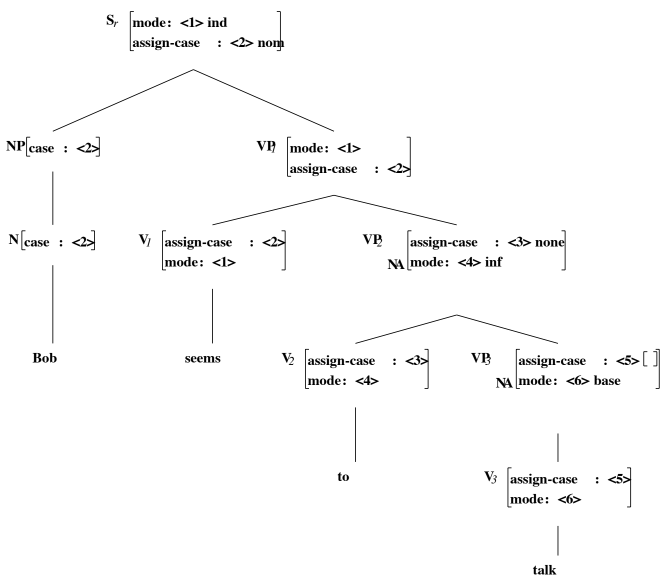


Figure 5: Derived tree for *Bob seems to talk*

the only difference between these trees and the earlier ones is that *to* is now treated as a separate auxiliary tree, instead of being part of the *talk* elementary tree. The derivation proceeds as follows:

1. The tree for *Bob* in Figure 4(a) substitutes into the NP node of the tree for *talk* in Figure 4(c), to produce a tree for *Bob talk*[5].

2. The tree for *to* in Figure 4(b) adjoins into the VP node of the resulting tree to produce a tree for *Bob to talk*.

3. The tree for *seems* in Figure 4(d) adjoins into the VP node of the resulting tree to produce a tree for *Bob seems to talk*.

The resulting tree is shown in Figure 5. Case assignment of the subject is handled by using the <**assign-case**> and <**case**> features. The basic idea is

[5]If the derivation stopped at this point, then a sentence for *Bob talk* would be produced with no unification errors. It is stipulated that every sentence must be indicative or imperative, which is indicated by the <**mode**> feature. In the current example *Bob talk* would have mode <**mode**>**=base**, and therefore fail as derivation.

that Case assigners have an <**assign-case**> feature, and noun phrases come with a <**case**> feature indicating whether they are nominative or accusative. The <**assign-case**> and <**case**> features must unify, thus "licensing" the case of the NP.

The *talk* tree, being uninflected, specifies no value for <**assign-case**>, since by itself it has no case to assign. The *to* tree has a <**assign-case**> value of **none**, which gets unified with the <**assign-case**> value on the top of the VP node in the *talk* tree. At this point, in the tree for *Bob to talk*, the <**case**> feature at the NP node is **none**, and so without any further operations this derivation would crash due to a unification failure. As discussed in more detail below, this corresponds to the licensing of PRO and not lexical subjects in infinitivals. However, when *seems* adjoins in, and unification of top and bottom features takes place, the <**assign-case**>**=nom** value on the V node of the *seems* tree gets percolated to the S_r node of the *talk* tree, which then unifies with the <**case**> value of the NP node to ensure that the subject has case **nom**.

The <**mode**> feature is used to indicate the mode of the sentence built so far. An uninflected verb, such as *talk* in this example, has <**mode**>**= base**, and the adjunction of the *to* tree percolates a value of **inf** for the feature <**mode**>, indicating that it is infinitive. A mode of value **ind** stands for indicative.

2.3. Sentential Complementation in Xtag

Consider the derivation of a sentence with a non-ECM verb with an infinitival complement, as in (8), in which, as shown, we are assuming an analysis with PRO as the subject of the complement.

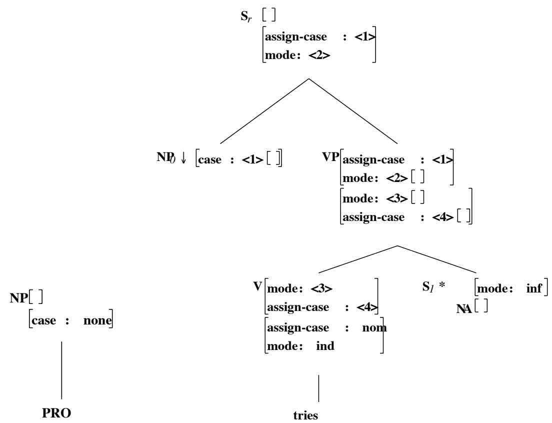
(8) Bob tries [PRO to talk]

We use the same trees for *Bob*, *to*, and *talk* as in Figure 4, and also the trees for *tries* and *PRO* in Figure 6.

Sentence (8) is derived using the trees in Figures 4 and 6 as follows:

1. The tree for *to* in Figure 4(b) adjoins at the VP node and the tree for *PRO* in Figure 6(a) substitutes at the NP node of the tree for *talk* in Figure 4(c), to produce a tree for *PRO to talk*.

2. The tree for *Bob* in Figure 4(a) substitutes at the NP node of the tree for *tries* in Figure 6(b).



(a) (b)

Figure 6: Component trees for *Bob tries to talk*

3. The result of step (2) adjoins at the S root node of the result of step (1).

The derived tree is shown in Figure 7, leaving out this time the feature values.

This derivation illustrates a slightly different case of nominative Case assignment, and also an example of how Case assignment is used to control the distribution of PRO. This latter case will be important for the ECM analysis.

For the subject Case assignment, the finite verb in Figure 6(b) has an <**assign-case**> value of **nom**, and since nothing gets adjoined into the V, VP, or S nodes, unification causes the <**assign-case**> value at the S node to be coindexed with the **nom** value. Since <**case**> at the NP substitution node is unified with that <**assign-case**> feature, then only an NP with nominative Case can substitute into the NP substition node of the *tries* tree. The tree for *Bob* in Figure 4(a) can unify with either the **nom** or **acc** value for the feature <**case**>, and so can successfully substitute in.[6]

The Case assignment for the embedded clause illustrates how the distribution of PRO is handled. As before, the embedded verb *talk* is only of <**mode**>**=base**, to signify a clause with no inflection, and the inflection is supplied by the adjunction of the *to* tree in Figure 4(b) into the VP node in

[6]A tree for a NP with explicit Case marking would have a specific Case value; e.g., a tree for *him* would have the feature <**case**>**=acc**.

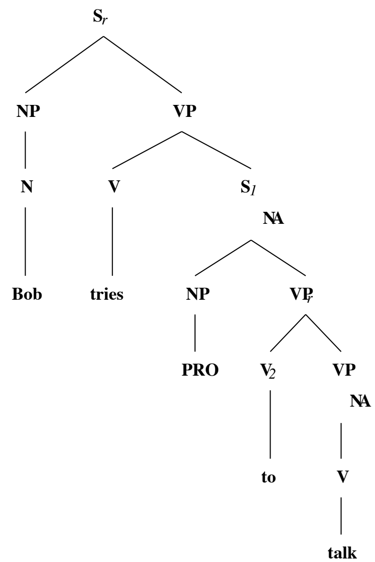


Figure 7: Derived tree for *Bob tries to talk*

Figure 4(c).

The adjunction of the *to* tree passes up a **none** value for the <**assign-case**> feature. This value is eventually unified with the <**case**> feature on the NP node in Figure 4(c), as a result of the rules for feature unification upon adjunction, as illustrated in the previous section, and since nothing else adjoins onto the VP node in the *talk* tree. This means that for an NP to substitute into Figure 4(c), it must be able to unify with <**assign-case**>**=none**. The only such NP which has this assignment is PRO, as shown in Figure 6(a). This ensures that the subject of the infinitival can only be PRO. (Adjunction of the *to* tree also blocks the **base** value for <**mode**> from being passed up, instead passing up a value of **inf**, to indicate that the adjunction of the *to* tree makes it into an infinitival clause.[7])

Note that it is crucial that *to* assign Case **none**, as opposed to not assigning a Case at all, with an empty value for the feature <**assign-case**>. If the latter were the case, then *any* NP could substitute in, since either a **nom** or

[7] Note that the specification of <**mode**>**=base** on the foot node limits this tree to adjoining only onto a VP node with **base** mode.

acc Case value could unify with an unspecified Case assignment value. The use of feature unification thus requires the use of the **none** value, as opposed to saying that PRO receives no Case at all.

Note also that the Xtag system uses a phrase structure that is less articulated than in some other approaches. There is no distinction between IP and CP nodes. Instead, both are just S nodes with feature values used to represent the usual I/C distinction. There are also no single-bar levels. Also, complementizers are not built in as part of the elementary trees, but rather are handled as auxiliary trees that only adjoin if required for a derivation. Thus, in a sense each projection is as minimal as it needs to be. For example, the clause *PRO to talk* in the derivation just considered only projects to an S that is the immediate projection of the embedded verb, with no need for a complementizer with another projection. Since the distribution of PRO is handled by the assignment of Case **none**, this does not cause a problem.

3. ECM Verbs and Bare Infinitives in XTAG

3.1. ECM Verbs in XTAG

Now consider how an ECM verb might be specified. Since it also takes an infinitival sentential complement, it will have the same form as the tree for *tries* in Figure 6(b). Since the features of the foot S node will unify with the S node on the infinitival complement, it will also have an <**assign-case**>**=acc** feature, which would pass down the Case feature to the NP of the complement via the root of the complement tree. However, this will not work, because at the same time the <**assign-case**>**=none** value from the *to* tree is unified with the root of the complement, as before, and so the <**assign-case**> value at the root of the complement tree would have to unify both with **acc** from the ECM verb, and **none** from the *to* tree.

This problem was faced earlier in the Xtag system for infinitival clauses with the complementizer *for*, as in *For Mona to drive the train is a good idea*. The solution developed then[8] is used here for the ECM verb problem.

The technique used is to create another tree for *to*, one that allows the unification clash to be avoided. Figure 8 shows the trees for *expects* and *to* that are used in a parse of *Van expects Bob to talk*. The trees for *talk* and *Bob* are the same as before, and the tree for *Van* is of course the same as for *Bob*.

[8] Which, to make sure credit is properly placed, the author of this paper was not involved in.

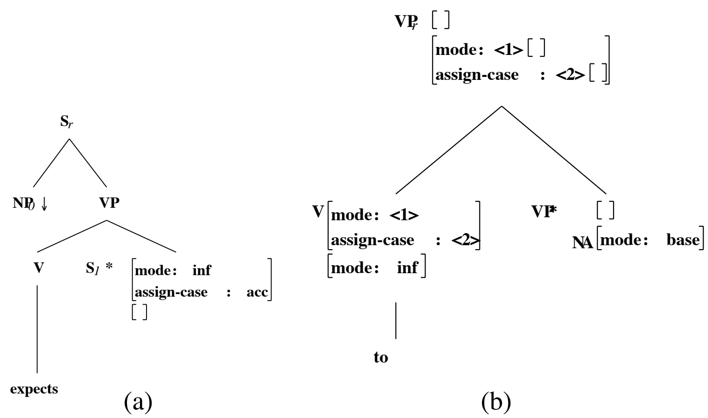


Figure 8: Component trees for *Bob expects Van to talk*

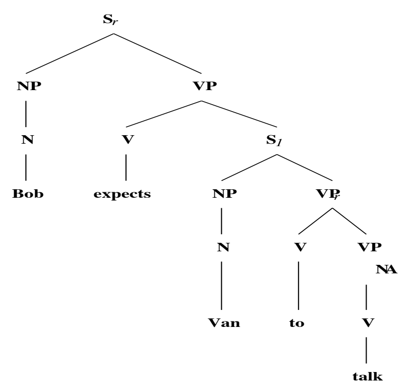


Figure 9: Derived tree for *Bob expects Van to talk*

The structure of the derivation is the same as before, leaving aside of course the Case assignment. The tree for *to* in Figure 8(b) adjoins into the VP node of the *talk* tree, and the tree for *Van* substitutes into the NP node for *talk*, giving *Van to talk*. The tree for *expects* in Figure 8 adjoins onto the root node of the *talk* node, and *Bob* substitutes into the NP node to give the result, as shown in Figure 9.

Unlike before, the *to* tree no longer passes up the <**assign-case**> value from the embedded verb, but instead an unspecified value is passed up. The foot node of the ECM verb has <**assign-case**>**=acc**, and as desired this will unify with the root node of the complement tree, and therefore with the subject of the complement, thus enforcing an accusative Case on the subject.

Note also that since the ECM tree is an auxiliary tree, and adjoins in, the same tree is used for a sentence in which the subject of the complement is extracted, as in *Who did Bob expect to talk*, with the ECM tree adjoining into a derivation of *who to talk*. For details on how Xtag handles long-distance extraction, see The XTAG-Group (1995).

3.2. Bare Infinitives in Xtag

Verbs with no inflection are selected from the lexicon with a value of **base** for the feature <**mode**>, and the adjunction of a tree for *to* (either of the two trees for *to* discussed above), gives the resulting structure a mode **inf** at the node where the *to* tree adjoins (VP). If no *to* tree adjoins, then the sentence will have no inflection and mode **base**.

Of course, this is exactly the case for bare infinitives, such as *harmonica fell* in *Bob sees the harmonica fall*. Therefore, the trees for verbs that take bare infinitives are straightforward to implement. As shown in Figure 10, the tree for *sees* is just like the earlier ECM tree for *expect* except that the complement is of mode **base**. The overall parse is simpler than the earlier ECM case, since there is no complication with *to* since there is, of course, no *to* in the lower clause. An example derived tree is shown in Figure 11. The base form of the embedded verb leaves the Case of its subject unspecified, but since the root node of the upper verb tree unifies with the root of the lower clause, the lower subject can only be accusative, thus ruling out **John saw he eat* or **John saw PRO eat*.

It's important to note that since the same tree is used for both sets of verbs, the lexical entries for *expect* will select the ECM tree and specify that the S foot node will have the feature <**mode**>**=inf**, while *see* will also select the ECM tree and specify that it should have the feature <**mode**>**=base**.

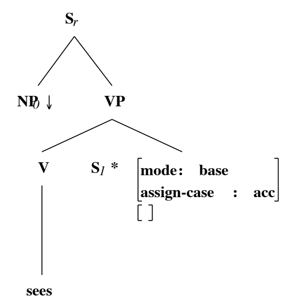


Figure 10: Tree for *sees*

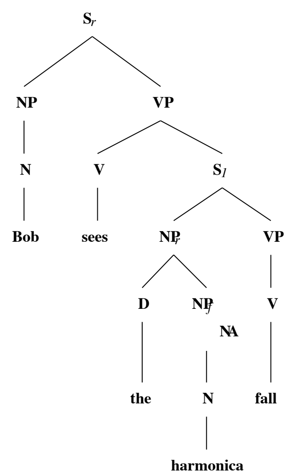


Figure 11: Derived Tree for *Bob sees the harmonica fall*

Furthermore, since Xtag groups together trees belonging to the same categorization frame, both verbs will actually select the entire ECM family of trees. So they will also select related trees that handle other contexts in which ECM verbs occur - for example, subject extraction, as in *who expects Van to talk?*. The crucial point is that for all of the trees in the ECM family, the <**mode**> feature of **inf** or **base** will be placed on the S foot node. A principled exception is the passive tree, as seen in the next section.

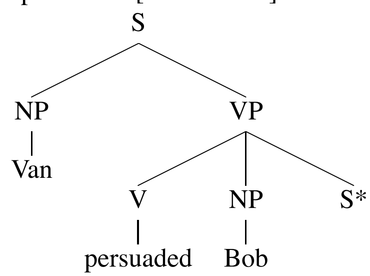
4. Passives and Raising Passives in XTAG

4.1. Passives in Xtag

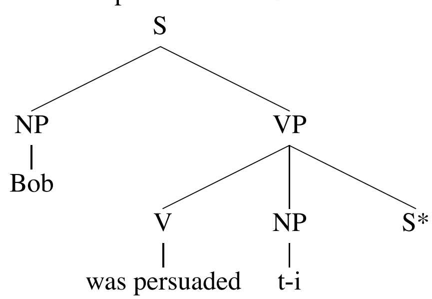
The examples in the previous section illustrate how adjunction allows recursive structures to be separated from the domain of dependencies of a verbal element. A consequence of this is that the elementary trees define the domain of locality over which constraints and thematic roles are specified. One of the interesting aspects of TAG is that transformational-type analyses can be used in TAG, but only as mappings from one elementary tree to another. One important example of this is of course the passive, which is treated as an operation on the tree for the active sentence. For example, the trees for the active and passive sentences in (9ab) are as shown in (10) and (11), respectively.

(9) a. Van persuaded Bob [PRO to talk]
b. Bob was persuaded [PRO to talk]

(10)



(11)



4.2. Raising Passives in Xtag

As discussed in the introduction for (2), the passive of an ECM verb is a "raising passive", and (3) shows that the passive for a verb that ordinarily takes a bare infinitive in its declarative use must take a full, not bare, infinitive.

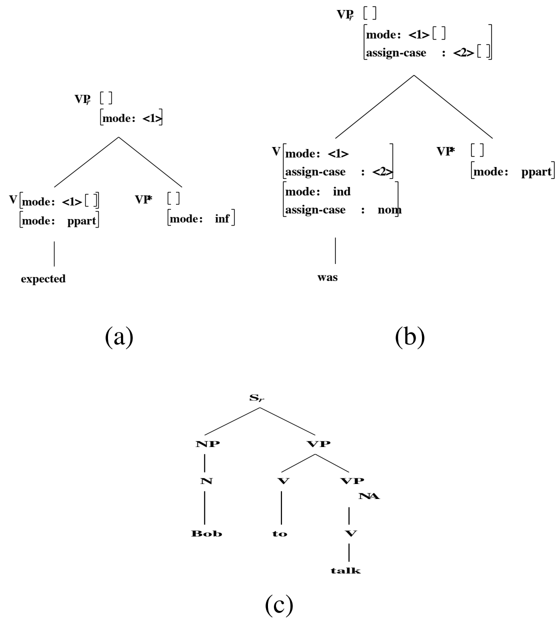


Figure 12: Component trees for *Bob was expected to talk*

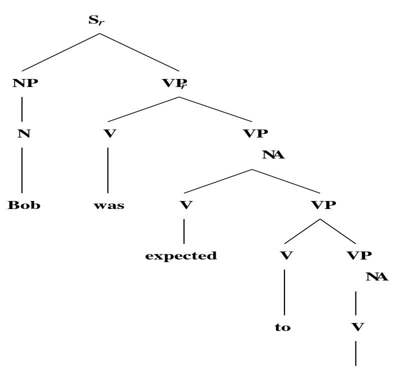


Figure 13: Derived tree for *Bob was expected to talk*

The component trees for *Bob was expected to talk* are shown in Figure 12. As discussed earlier, in TAG a subject is "raised" by the process of adjunction, which results in the raised item getting "stretched" away from the other components of its elementary tree, and the derivation of *Bob was expected to talk* is exactly parallel, except for the minor addition of also needing to adjoin the tree for *was* onto the the tree for *expected* at the VP node. This requirement is handled by some feature values that are peripheral to the topic of this paper and so are not discussed here. The adjunction of the *was* tree onto the root of *expected* tree and of the *expected* tree onto the VP node of Figure 12(c) gives the derived tree in Figure 13.

Now consider the raising passive of one of the verbs that take bare infinitives, such as *sees* in (3). Although verbs like *expects* and *sees* differ on the mode of their clausal complements, they both share the property of not thematically selecting the subject of that complement, which becomes the matrix subject of the corresponding passive sentences. Therefore, the passive versions of *sees* and other verbs that take bare infinitives are *also* VP auxiliary trees, exactly the same in structure as the one for *expected* in Figure 12(a).

Recall that the ECM verbs and verbs that take bare infinitive complements differ only by the **mode** feature on the foot S node, and that this feature is put onto all the trees in the tree family. However, there is no S node in the passive tree, and so the passive remains unaffected by this feature specification. But this is of course exactly the desired situation. All that needs to be done is to specify that the foot node of the passive tree has **<mode>=inf**, as shown in Figure 12(a). A derivation of **Bob was seen talk* is ruled out because the **<mode>** value at the VP adjunction site would be **base**, and not **inf**. At the same time, since it's the S node where the difference between the two classes of verbs is located, *Bob sees the harmonica fall* can be derived[9].

5. Conclusion and Comparison to Other Work

We have described an implementation in the Xtag system of the TAG analysis of ECM verbs first set out in Kroch and Joshi (1985). We presented a solution to a problem raised by the ECM analysis for Xtag, and also extended the analysis and implementation to handle verbs that take bare infinitive complements, allowing for a nice account of the problem of the bare infinitive passive.

[9] I am extremely hesitant to regard this as anything more than a nice implementational "trick". See Santorini and Heycock (1988) for arguments that the *to*-infinitive passive bears no syntactic relationship to bare infinitive actives, and are instead related to *to*-infinitive complements that previously existed in English.

As discussed in Section 3, the Xtag system is forced to use an analysis in which the distribution of PRO is controlled by the use of a feature with value **none**, instead of just a lack of assignment. Since noun phrases are drawn from the lexicon with trees already instantiated for Case values that those NPs are compatiable with, then a lack of Case assignment would mean that any NP could appear where only PRO was desired. The assignment of Case **none** to PRO allows the distribution of PRO to be properly handled

It is of interest to note that this technique, undertaken for computational issues brought on by unification, has intriguing parallels to some other current approaches to the distribution of PRO, such as use of the null Case assignment in infinitivals that was independently proposed in Chomsky and Lasnik (1991). This was not done for reasons of unification, but in part for conceptual reasons regarding the apparent movement of PRO in passives of control infinitives such as *Bob tried PRO to be arrested*.

Furthermore, once the move to **null** Case is made, the same consequences follow in both analyses. As described in section 3, the Xtag analysis requires two different trees rooted by *to*, one to be used in ECM constructions, and the other to be used in control constructions. The Chomsky and Lasnik (1991) approach has been similarly refined in recent work, such as that of Boskovic (1995) and Martin (1992)[10]. In this work, the null Case assignment story is modified to handle the case of the difference between ECM and control infinitivals[11]. It is proposed that in complements to control verbs, the INFL has a [+Tense] feature that assigns **null** Case, while complements to ECM infinitivals have only a [-Tense] feature which does not assign Case, thus allowing the subject to move to the matrix clause to get Case checked off in [Spec, AgrO], whether overtly or covertly being a matter of debate. Borrowing from an earlier analysis by Stowell (1982), it is further argued that ECM and control infinitivals have different temporal properties, which I won't go into here, with only the ECM complements being truly "tenseless". The use of null Case assignment allows the CP vs. IP distinction to be done away with to some extent, allowing both control and ECM verbs to select an IP, with the distribution of PRO being controlled not by government, but by Case assignment in the lower clause[12].

[10] I have not yet seen Martin (1992).

[11] The implications of the null Case assignment to PRO for ECM and control infinitivals do not appear to be discussed in Chomsky and Lasnik (1991).

[12] In Stowell (1982), however, the tense argument of control infinitivals was assumed to be in C, and so if the CP vs. IP distinction for control and ECM complements is assumed, this derives that only control complements have the [+Tense] feature. If both

There is an obvious parallel with the use of *two* to's forced by implementation considerations in Xtag. The *to* in Figure 4b can be seen as the [+Tense] *to*, and the *to* in Figure 8b can be seen as the [-Tense] *to*. Just as with these recent analyses, Xtag is also able to let both ECM and control verbs select a complement of infinitival S (the equivalent of IP).

Under both analyses, there is the need to distinguish ECM from control verbs in terms of whether the complement can license a **null** Case. To some extent, this is taken care of in Xtag by the presence of the **<assign-case>=acc** feature on the foot node of the ECM trees, which would cause an unification clash if the "wrong" *to* was used - that is, the one meant for control clauses. The **<assign-case>=none** value on the embedded verb's tree would conflict with the **<assign-case>=acc** value on the ECM verbs' tree. However, this is not a sufficient solution, since it does not take care of the ambiguity with raising verbs. Consider again the derivation of *Bob seems to talk* shown in Figures 4 and 5. This derivation used the tree for the "control" *to*, the one with **<assign-case>=none**. No conflict arose because the adjunction of the *seems* tree caused its **<assign-case>=nom** feature to percolate up instead of the **<assign-case>=none** feature from the *to* feature. However, there is nothing that prevents the *to* tree with an empty **<assign-case>** feature (the one used in an ECM complement), from also being used, thus resulting in two derivations. So it has to be specified that one or the other should be used. This is roughly equivalent to the need, in a minimalist framework, to specify that the complement of a raising verb has [-Tense], since otherwise it would allow a sentence such as *It seems PRO to be happy*. Presumably this follows from common semantic properties of the raising and ECM complements as opposed to the control complements[13].

References

Boskovic, Zeljko (1995) "Null Case and Certain Differences between French and English," in Marek Przezdziecki and Lindsay Whaley, eds., *Proceedings of the Twelfth Eastern States Conference on Linguistics*, 13–24.

Chomsky, Noam, and Howard Lasnik (1991) "The Theory of Principles and Parameters," in J. Jacobs, A. von Stechow, W. Sternefeld, and T. Vennemann, eds.,

complement types are IP, this account no longer holds, and it is argued that the tense difference is a semantic constraint on the complement.

[13]This indicates that the "wrong" *to* was used in the derivation of *Bob seems to talk*, and that instead the "ECM" *to* should be used. I leave as an open problem for now how this can be accomplished in an unification framework.

Syntax: An International Handbook of Contemporary Research, MIT Press, Cambridge, Massachusetts.

Frank, Robert (1992) *Syntactic Locality and Tree Adjoining Grammar: Grammatical, Acquisition and Processing Perpectives*, Doctoral dissertation, University of Pennsylvania.

Joshi, Aravind K., L. Levy, and M. Takahashi (1975) "Tree Adjunct Grammars," *Journal of Computer and System Sciences* 10:1, 136–163.

Kroch, Anthony, and Aravind K. Joshi (1985) "The Linguistic Relevance of Tree Adjoining Grammars," Technical Report MS-CIS-85-16, University of Pennsylvania.

Martin, R. (1992) "On the Distribution and Case Features of PRO," manuscript, University of Connecticut.

Santorini, Beatrice, and Caoline Heycock (1988) "Remarks on Causatives and Passive," Technical Report MS-CIS-88-33, University of Pennsylvania.

Stowell, T. (1982) "The Tense of Infinitives," *Linguistic Inquiry* 13, 561–570.

The XTAG-Group (1995) "A Lexicalized Tree Adjoining Grammar for English," Technical Report IRCS 95-03, University of Pennsylvania.

Vijay-Shanker, K., and Aravind K. Joshi (1991) "Unification Based Tree Adjoining Grammars," in J. Wedekind, ed., *Unification-based Grammars*, MIT Press, Cambridge, Massachusetts.

Institute for Research in Cognitive Science
University of Pennsylvania
3401 Walnut Street, Suite 400A
Philadelphia, PA 19104-6228
skulick@linc.cis.upenn.edu