A Multi-Level TAG Approach to Dependency

Mark Dras
Inst. for Research in Cognitive Science
University of Pennsylvania
Suite 400A, 3401 Walnut Street
Philadelphia, PA 19104-6228
`madras@linc.cis.upenn.edu`

David Chiang, William Schuler
Computer and Information Science Dept.
University of Pennsylvania
Philadelphia, PA 19103
{`schuler,dchiang`}`@linc.cis.upenn.edu`

Abstract

This paper looks at integrating dependency and constituency into a common framework, using the TAG formalism and a different perspective on the metagrammar of (Dras, 1999) in which the meta level models dependencies and the object level model constituency. This framework gives consistent dependency analyses of raising verbs interacting with bridge verbs, solving a problem in Synchronous TAG. And in a completely different area, the framework gives appropriate analyses of subject-auxiliary inversion; moreover, in doing this, a neat representation of case assignment falls out of the analyses. This and other evidence suggests the integration of dependency and constituency is a useful avenue to explore.

1 Introduction

1.1 Background

English-language linguistics has long been dominated by grammar formalisms based on constituency. However, dependency-based formalisms have a long history, arguably longer than constituency grammars. They also have a strong body of work in modern times by Europeans such as (Tesnière, 1959), (Sgall et al., 1986) and (Mel'čuk, 1988); and there is now something of a resurgence of interest in them in the English-speaking world—in particular, in combining the two in some way to take advantage of both.

The use of insights from dependency grammars for English can be grouped into three broad categories. First, there are practical applications that just use dependency to achieve their ends: for example, the statistical parsing of (Magerman, 1995) and others which uses dependency relations between words; the translation methodology of (Palmer, Rosenzweig, and Schuler, 1998); the multilingual generation of (Iordanskaja et al., 1992); and so on.

Second, there are systems and formalisms which are 'augmented' by selecting aspects of dependency and incorporating them into constituency, and vice versa. This has a long history as well: the modistic school of grammar of Paris in the late 13th and 14th centuries, which was the school from which the notion of government originated (Covington, 1984), was a fundamentally dependency-based formalism,[1] but several of its proponents acknowledged that constituency was necessary to handle aspects of language like conjunction, sentence embedding, and impersonal constructions. More recent attempts include the work of Gladkij (1980) on Eastern European languages, reported by Mel'čuk (Mel'čuk and Pertsov, 1987). In the other direction, and more recently, Abney (1995) describes a technique of robust parsing, 'chunking', which partially groups words into constituents and then joins these by dependency links; his work covers both a theoretical foundation for this combination based on psycholinguistic and prosodic evidence, and a system for broad-coverage parsing. Other robust parsing systems, such as those based on supertags (Srinivas, 1997), can be viewed as operating similarly. And on the formal side, Hudson's (1976) daughter-dependency grammar adds dependency to constituency; and more generally, the notions of government and of heads of construction have been adapted from dependency grammar into standard Chomskyan analyses (Robinson, 1970).

Finally, there are attempts to integrate constituency and dependency into a single formalism. Early in the recent prominence of constituency-based linguistics, Gaifman (1965) and Hays (1964) express dependency grammars using phrase structure rules, and restrict themselves only to projective dependency grammars[2], showing that these are weakly equivalent to context-free grammars. Then, as part of the Transformational Grammar program, there were proposals to use dependency grammars, rather than context-free or context-sensitive grammars, as the base of a transformational grammar (Robinson, 1970; Vater, 1975), on various grounds including that it allows a neater description of, for example, case phrases than a phrase-structure grammar does (Anderson, 1971), and that it is a weaker theory (Hays, 1964). This was criticized in, for instance, (Bauer, 1979), who notes that it is difficult to determine what should be the ultimate head of the sentence, although the determination of the distinguished symbol in phrase structure grammars, and later headedness, has been the subject of similar discussion. He also notes that, given the result of (Peters and Ritchie, 1973), where transformations are shown to make Transformational Grammar unrestricted in formal power, the base is actually not significant as it is dominated by the transformations. It is, however, interesting to note that some incarnations of Chomskyan theory have a D-Structure component which has properties quite similar to those of dependency grammar: e.g. "We have been tacitly assuming throughout that D-structure is a 'pure' representation of theta structure, where all and only the θ-positions are filled by arguments." (Chomsky, 1986); c.f. the labeled dependency structures of (Mel'čuk, 1988), which are structures representing headedness and arguments.

In this paper we explore an integration of formalisms into a common framework in the spirit of this last type of melding of constituency and dependency grammars.

Tree Adjoining Grammar (TAG) is a good candidate for such a framework. Although it does not intrinsically say anything about dependency, TAG assigns *derivation trees* to sentences which are commonly interpreted as dependency structures (Rambow and Joshi, 1997). However, many cases have been pointed out for which TAG derivation trees do not successfully capture linguistic dependencies (Becker, Joshi, and Rambow, 1991; Schabes and Shieber, 1994; Rambow, Weir, and Vijay-Shanker,

---

[1] This use of a dependency formalism was fairly common before the 20th-century focus on English, and here could be seen as a natural consequence of the fact that the modistic grammarians studied the relatively free word order syntax of Latin: Mel'čuk (1988) comments that constituency-based formalisms only came to be seen as natural ways to describe language because of the English-language focus, and resulting fixed word order bias, of much of modern linguistics.

[2] Roughly, an arc representing a relation between two words x and its dependent y is projective if, for every word w between x and y, w is a dependent (immediately or transitively) of x; a grammar is projective if all of its arcs are projective. More precise definitions are found in the original Lecerf (Lecerf, 1960) and the more frequently quoted Robinson (Robinson, 1970). Some natural language phenomena, however, such as English *wh*-extraction and clitic climbing in Romance, are known to require non-projective analyses.

1995). Schabes and Shieber (Schabes and Shieber, 1994) revised the standard notion of TAG derivation to better match derivation trees to dependencies, but mismatches still remain.

A related line of research in finding such a framework has been in formalisms which are more powerful than TAG, like set-local multicomponent TAG (Weir, 1988), V-TAG (Rambow, 1994), and most recently, D-tree Substitution Grammar (Rambow, Weir, and Vijay-Shanker, 1995). These have been quite successful, but their added formal power makes them more difficult to process.

Yet another line of research has focused on squeezing as much strong generative capacity as possible out of weakly TAG-equivalent formalisms (see, for example, (Joshi, 2000) on what it means to extract more strong generative capacity out of a formalism without increasing its weak generative capacity): tree-local multicomponent TAG (Weir, 1988), nondirectional composition (Joshi and Vijay-Shanker, 1999), and segmented adjunction (Kulick, 2000). We follow this approach.

1.2 Multi-level TAGs

Problems related to the mismatch of TAG derivations and linguistic dependencies arise for synchronous TAG as defined in (Shieber, 1994), because mappings are induced by isomorphisms between derivation trees. Therefore, just as TAG's ability to describe dependencies is limited by its strong generative capacity, synchronous TAG's ability to define mappings is as well. These limitations are serious in practice, both for translation (Shieber, 1994) and paraphrase (Dras, 1999). Dras (Dras, 1999) showed that these difficulties could be resolved by the use of a *meta-level grammar.*

A TAG is generally thought of as a set of elementary trees which combine by substitution and adjunction to form a *derived tree*. The process of combining the elementary trees together is recorded in a *derivation tree*.

Let us refine this view somewhat. Weir (1988) showed that the derivation trees of a TAG can be generated by a context-free grammar. We can therefore think of the derivation process as the building up of a context-free derivation tree, followed by the application of a *yield function* f_G, dependent on the grammar G, to produce a derived tree.

Now, since a TAG yield function maps from trees to trees, nothing prevents us from applying more than one of them. A k*-level TAG* (Weir, 1988) has k yield functions $f_G, f_{G'}, \ldots, f_{G^{(k)}}$ (dependent on grammars $G, G', \ldots G^{(k)}$) which apply successively to trees generated by a context-free grammar.

In the case of 2-level TAG, we call G the *meta-level grammar* (or *meta-grammar*) and G' the *object-level grammar*, because the meta-level grammar generates the derivation trees for the object-level grammar. A *regular form 2-level TAG* (RF-2LTAG) is a 2-level TAG whose meta-grammar is in the regular form of (Rogers, 1994). Since the object-level derivation trees will then form a recognizable set, RF-2LTAG is weakly equivalent to TAG (Dras, 1999).

In synchronous RF-2LTAG (Dras, 1999), mappings are induced by isomorphisms between meta-level derivation trees. Even though RF-2LTAG is weakly equivalent to TAG, its extra strong generative capacity enables synchronous RF-2LTAG to generate more mappings than synchronous TAG can (Chiang, Schuler, and Dras, 2000).

In this paper we seek to use this extra strong generative capacity to better describe linguistic dependencies. To do this, we interpret the meta-level derivation trees as dependency structures, instead of the object-level derivation trees as in (Dras, 1999; Chiang, Schuler, and Dras, 2000). In doing this, our approach has similarities to the use of dependency grammars as a base for transformational grammars, in that a dependency representation and a constituency representation are related by TAG yield functions on the one hand and transformations on the other. However, our approach is computationally tractable, and, moreover, can be seen as integrating the two representations into a single multidimensional structure, in the sense of Rogers (1997).

We discuss the formal details below, in Section 2, and then some linguistic applications of this combination of dependency and constituency in Section 3.

2 The formalism

2.1 TAGs

A TAG is a tuple of $\langle \Sigma, NT, I, A, F, V, S \rangle$ where Σ is a set of terminal symbols, NT is a set of non-terminal symbols, I and A are sets of initial and auxiliary elementary trees, F is a finite set of features, V is a finite set of feature values, and $S \subset NT$ is a set of distinguished symbols, respectively.[3] Elementary trees are trees whose internal nodes are labeled with symbols from NT, and whose leaf nodes are labeled with symbols from both NT and Σ. These elementary trees may be composed into larger trees by the the operations of substitution and adjunction (see Figure 1). Substitution is the attachment of an initial tree (such as that for ‘John’) at the frontier of another tree, by identifying the root of the initial tree with one of the host tree’s leaf nodes (marked with $\downarrow$). Adjunction is the attachment of an auxiliary tree (such as that for ‘quietly’) into the interior of another tree, by removing the entire subtree at one of the host tree’s internal nodes, inserting the auxiliary tree in its place, and re-attaching the removed subtree at one of the auxiliary tree’s leaf nodes (marked with $*$). Nodes also have non-recursive top and bottom feature structures which must unify for a complete derivation.

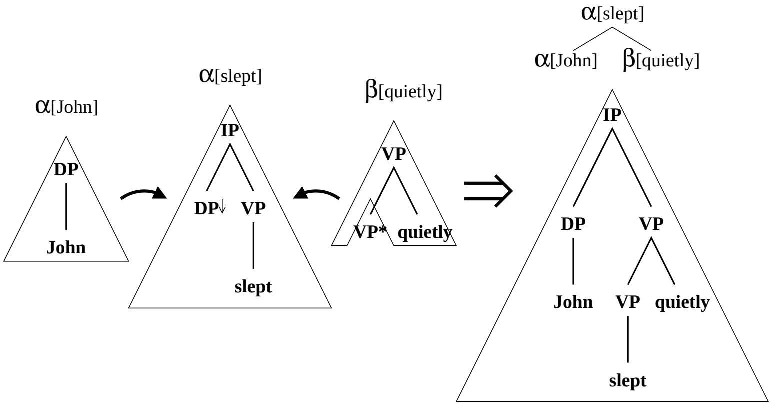


Figure 1: Substituting and adjoining TAG elementary trees.

Substitution and adjunction are typically used in linguistic analyses to represent the attachment of arguments and modifiers to the predicates they modify (as in the example above), but adjunction can also be used in the other direction, to represent the attachment of bridge and raising predicates to the arguments they predicate over. For example, in an analysis of the sentence,

(1) What does Mary think John seems to like?

the raising construction ‘seems’ adjoins into the tree for ‘like’ between the subject and the verb, as shown in Figure 2. Then the bridge construction ‘Mary thinks’ adjoins onto the initial tree on the other

---

[3] This is a simplified formulation of the Feature-based TAGs defined in (Vijay-Shanker, 1987), which are used as standard in the world of TAGs; see e.g. (XTAG, 1998).

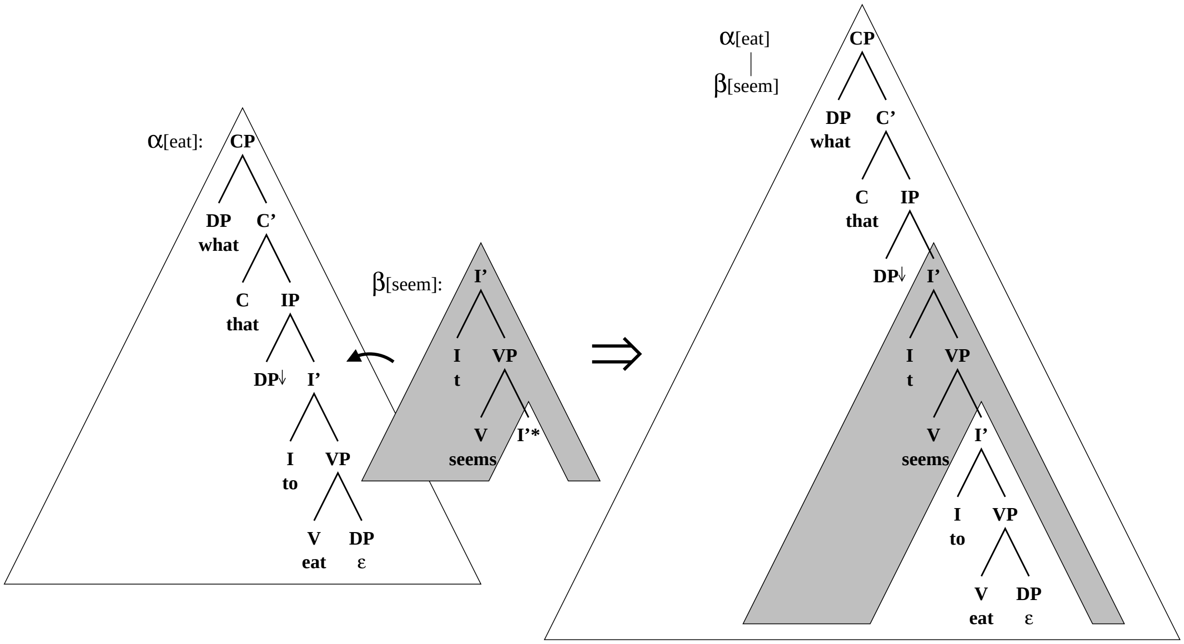


Figure 2:

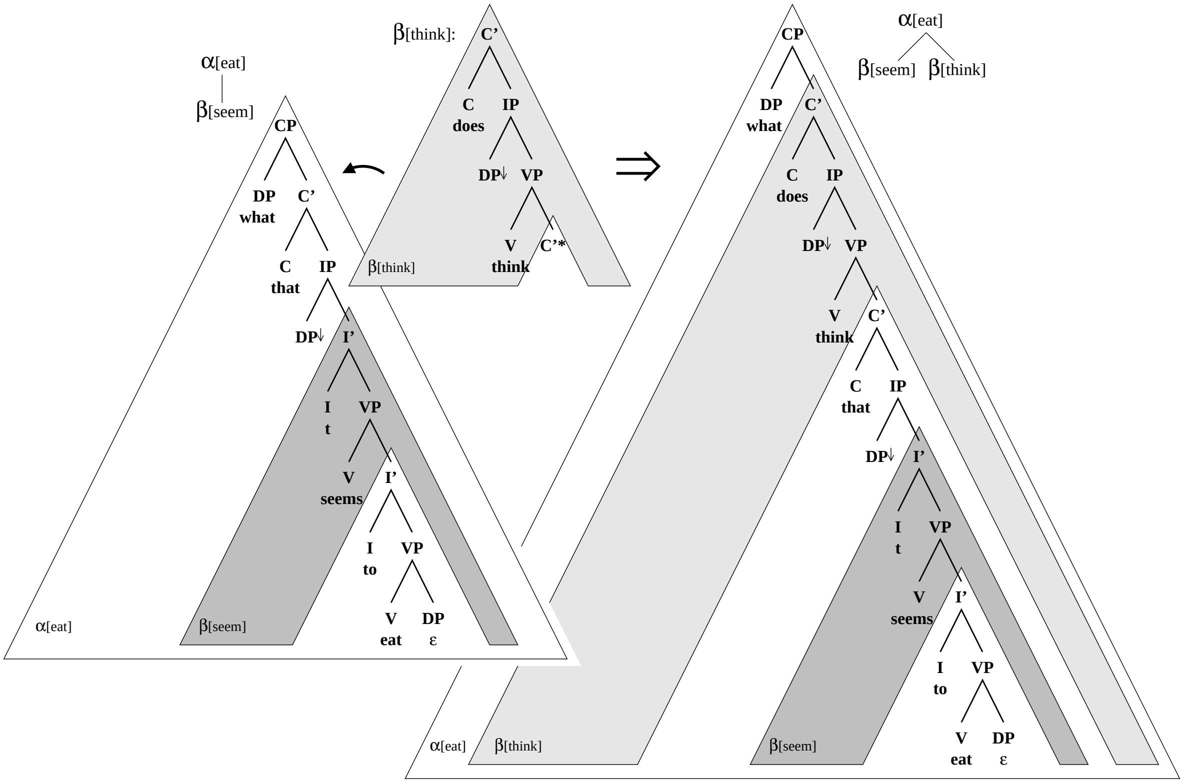


Figure 3:

side of the subject, as shown in Figure 3. A derivation tree, shown above each of the derived trees in the figure, represents the process by which each derived tree was obtained, using nodes for elementary trees and arcs for the substitutions and adjunctions that occured between them.

One problem with this kind of derivation, first pointed out in (Rambow, Weir, and Vijay-Shanker, 1995), is that although it comes close to matching the traditional notion of dependency, the derivation for a sentence such as (1) will connect the bridge verb and the lower verb, between which there are no semantic dependencies, and will not connect the bridge verb and the raising verb, between which a semantic dependency should exist.

2.2 2LTAGs

A 2LTAG is a pair of TAGs $\langle G, G' \rangle = \langle \langle \Sigma, NT, I, A, F, V, S \rangle, \langle X, X, I', A', F', \bar{I} \cup \bar{A} \cup \mathbb{N}^*, S' \rangle \rangle$. We call the first member of the pair the *object-level* grammar, and the second member the *meta-level* grammar. Both grammars have the same standard TAG composition operations of substitution and adjunction. The object-level grammar has no special characteristics; the meta-level grammar has the following properties:

- the set of labels, X, consists of only a single element;
- the set of features, F', has only two elements, one of which resides in the bottom feature structure of each node (for labels of elementary trees of G), and the other of which resides in the top feature structure of each node (for Gorn addresses[4]);
- the set of feature values consists of the labels of the trees of G and of Gorn addresses in G;
- the yield function $f_{G'}$ reads the feature values of the nodes, rather than their labels, in derived trees in G' in order to produce derived trees of G.

We write meta-level elementary trees using the following shorthand:

$$\overset{|^{\eta}}{\gamma} \equiv X \begin{matrix} [\mathit{addr} : \eta] \\ [\mathit{tree} : \gamma] \end{matrix}$$

Furthermore, we write γ for a term which can unify with any term of the form $\gamma[\cdot]$.

The result that we want from this definition is that the trees produced by G' look like derivation trees of G. We define the *tree set* of $\langle G, G' \rangle$, $\mathcal{T}(\langle G, G' \rangle)$, to be $f_G[\mathcal{T}(G')]$, where f_G is the yield function of G and $\mathcal{T}(G')$ is the tree set of G'. Thus, when the elementary trees in the meta-level grammar G' are combined, using the substitution and adjunction operations as defined for TAG, the derived trees can be interpreted as derivations for the object-level grammar G.

We can now produce a meta-level derivation tree for (1) which represents the desired dependencies. First, the meta-level auxiliary tree $\mathcal{B}[seem]$ adjoins into the initial tree $\mathcal{A}[like]$ to derive a tree where the node labeled $\beta[seem]$ is between $\alpha[like]$ and $\beta[like]$ (see Figure 4). Viewed as an object-level derivation, this tree has $\beta[seem]$ adjoined at node 2 of $\alpha[like]$, and $\beta[like]$ adjoined at node 2 of $\beta[seem]$. Then $\mathcal{A}[think]$ substitutes into $\mathcal{B}[seem]$ to complete the meta-level derivation, adjoining $\beta[think]$ at the root (address ϵ) of $\beta[seem]$ in the object-level derivation (Figure 5).

By way of explanation about the notation: we choose to encode the labels of trees in G as feature values in G' because we want 'fundamentally related' nodes to be able to be identified by substitution

---

[4] The Gorn address of a root node is ϵ; if a node has Gorn address η, then its ith child has Gorn address $\eta \cdot i$.

or adjunction—for example, we want to allow the $\beta[\textit{seem}]$-rooted tree $\mathcal{B}[seem]$ to adjoin at $\beta[\textit{eat}]$ in Figure 4—and if their labels were the strings '$\beta[\textit{seem}]$' and '$\beta[\textit{eat}]$' this would preclude strict adjunction. Whether a particular substitution or adjunction is licit—whether $\beta[\textit{eat}]$ can adjoin into $\beta[\textit{seem}]$, as the resulting object-level derivation tree of Figure 4 indicates—is determined in the object-level grammar G. The choice of Gorn addresses on nodes versus on arcs is a minor notational variant: the original on nodes, from (Weir, 1988), is more suitable for our purposes definitionally, although in diagrams we have used the notationally more popular arc labeling.

Also, although the above analysis produces the correct dependency links, the directions are inverted in some cases. This is a disadvantage compared to, for example, DSG; but since the directions are consistently inverted, for applications like translation or statistical modeling, the particular choice of direction is usually immaterial.

2.3 RF-2LTAGs

By itself, 2LTAG has more generative capacity and recognition complexity than TAGs, but if the meta-level derivations are restricted to a regular form (Rogers, 1994), the object-level derivations will be restricted to a context free form like ordinary TAG derivations, so the generative capacity and recognition complexity of the formalism will be constrained to that of TAG.

The regular form condition of (Rogers, 1994) holds for any TAG if the elementary trees of that grammar do not allow any cycles of possible internal spine adjunction in derivation—that is, adjunction on the path from the root to the foot but not at the root or the foot of an elementary tree. Since the only auxiliary meta-level trees used in this analysis (the trees for raising constructions) do not have any internal nodes, our grammar meets this condition.

3 Additional linguistic applications

3.1 Raising and subject-Aux inversion

A related problem occurs with the sentence

(2) Does Gabriel seem to eat gnocchi?

We assume, following Frank's Condition on Extended Tree Minimality (1992), that the functional head *does* must occur in the same elementary tree as the lexical head *seem* with which it is associated. But this is impossible in ordinary TAG because the subject stands in the way. This is exactly parallel to the previous example, where we wanted the complementizer *that* to be in the same tree as *seems*. The solution employed there works here as well (Figure 6): we simply assume that the CETM applies to meta-level elementary trees instead of object-level elementary trees.

A more familiar solution would be to use tree-local MCTAG (Figure 7), in which a set of trees adjoins simultaneously into a single elementary tree (assuming that the CETM applies to elementary tree sets instead of individual elementary trees). But this solution does not extend to the following:

(3) a. I think that Gabriel seems to be likely to eat gnocchi.
b. Does Gabriel seem to be likely to eat gnocchi?

In both cases tree-locality is violated unless the the tree for *likely* adjoins at the foot of the tree for *seem(s)*, which is normally prohibited. More importantly for present purposes, the derivation would

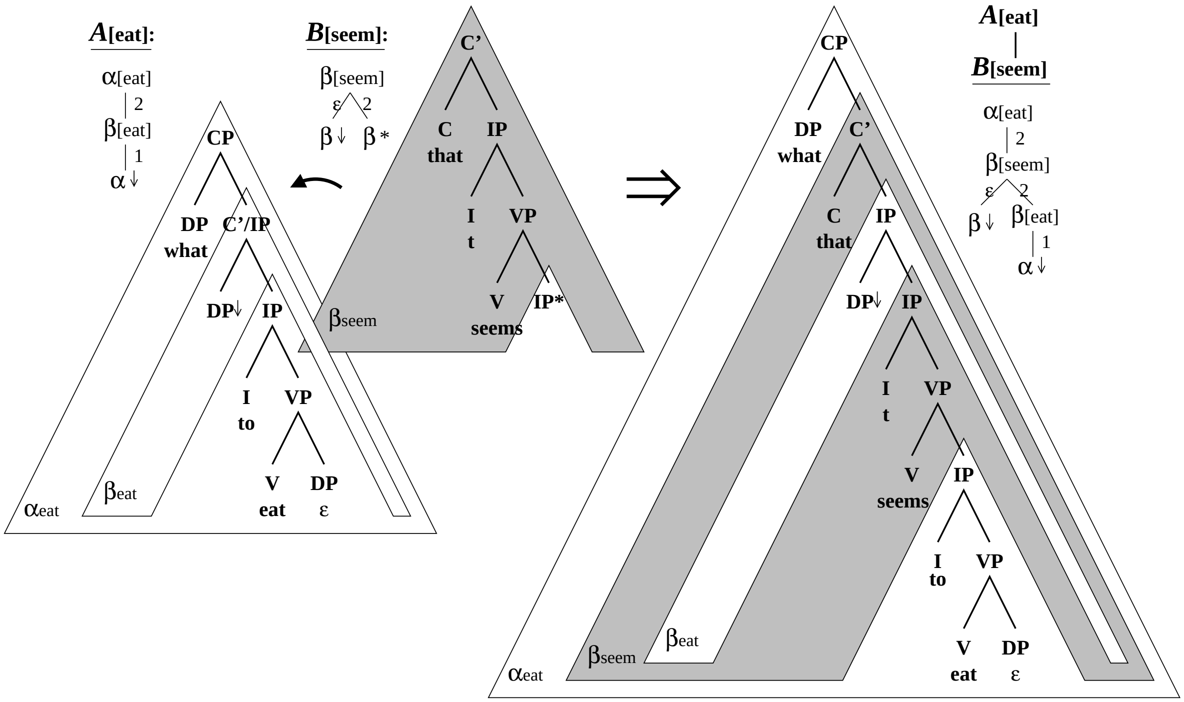


Figure 4:

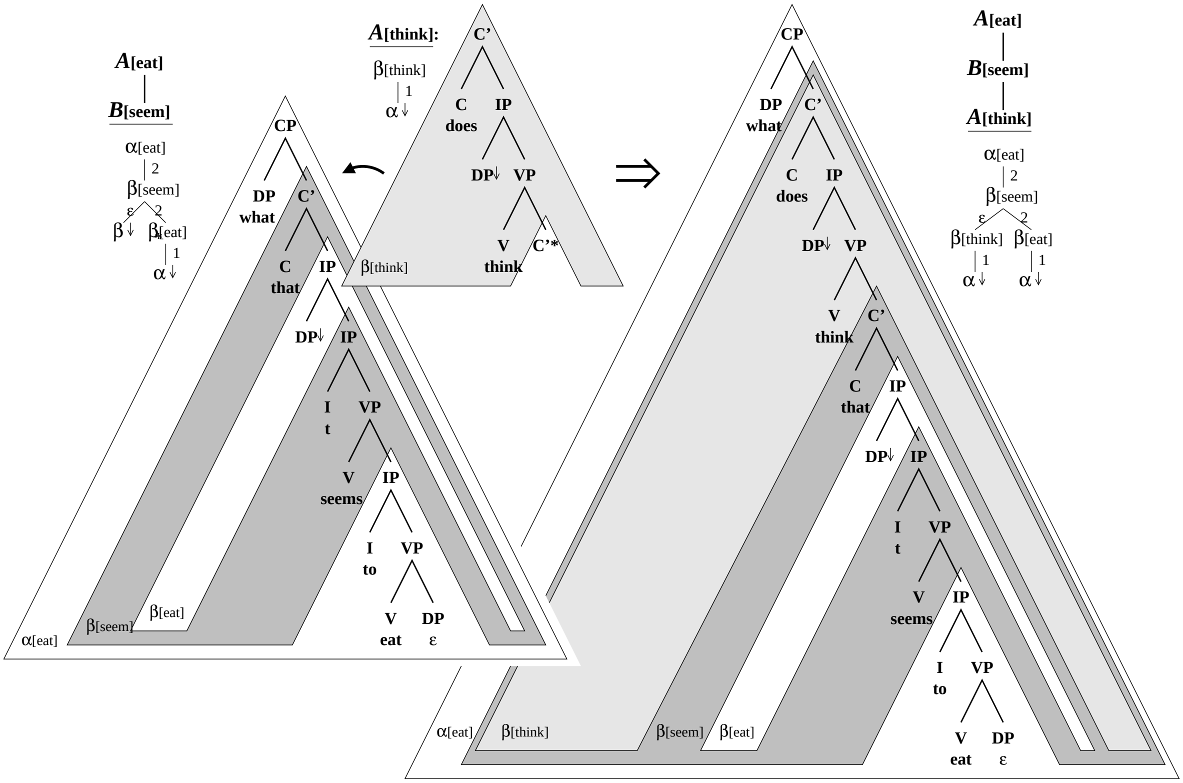


Figure 5:

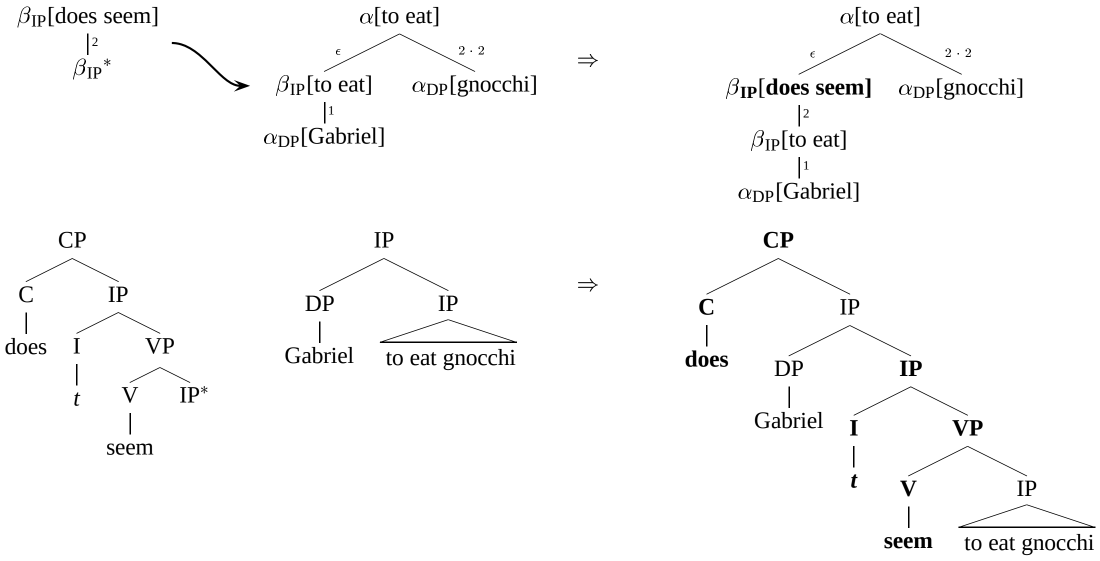


Figure 6: 2LTAG derivation for sentence (2).

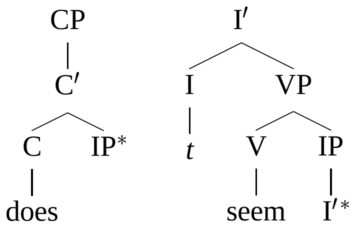


Figure 7: A multicomponent tree set for sentence (2).

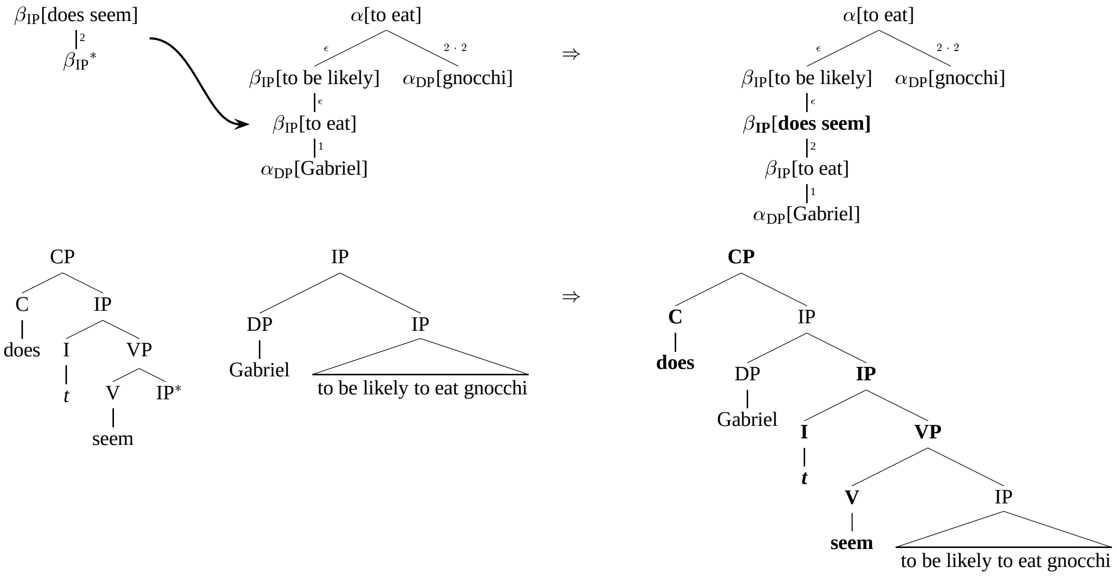


Figure 8: 2LTAG derivation for sentence (3b).

not correctly reflect the dependencies: *eat* would compose with the very highest raising verb.

But such sentences are not problematic for our 2LTAG analysis, as shown in Figure 3.1.

3.2 The object-level derivation tree

It is tempting to think of the object-level derivation tree as an intermediate structure, produced halfway through the transmogrification of the meta-level derivation tree into the object-level derived tree.

But consider the case of context-free grammars. In a CFG derivation, rewrite rules combine to form a derived string, and this process is recorded in a derivation tree. It is reasonable to think of a CFG derivation as the building of a derivation tree, followed by the application of a yield function to produce a derived string. But it would be strange to conclude that the derivation tree comes prior to the derived string, because the derivation tree contains the derived string on its leaves. All the yield function does is read the leaves off in order; being grammar-independent, it is not an extra step in the derivation but the means of recovering the result of the derivation.

Similarly, Rogers (1997) introduces a notion of TAG derivation trees as three-dimensional trees in which each node is not labeled with the name of an elementary tree, but has a tree of children corresponding to an elementary tree, just as each node in a CFG derivation tree has a string of children corresponding to a rewrite rule. The derived tree (and derived string) are recoverable from these three-dimensional trees by means of a grammar-independent yield function. Thus they integrate derivation trees and derived trees into a single structure. Under the view that TAG derivation trees should represent dependencies, these structures provide an integrated representation of dependency and constituency.

Rogers generalizes this idea further to an infinite hierarchy of multidimensional trees and corresponding formalisms. This hierarchy of formalisms corresponds to Weir's multilevel TAG hierarchy. Thus we can think of a 2LTAG derivation as the building of a four-dimensional structure, followed by successive applications of grammar-independent yield functions to recover the information stored within them.

Like the three-dimensional trees produced by TAG, these four-dimensional trees provide an integrated representation of dependency and constituency. But the additional level does not just provide a looser coupling between the two so that both will come out right; the derived structures of 2LTAG actually contain additional information which the grammar writer can exploit if he so chooses.

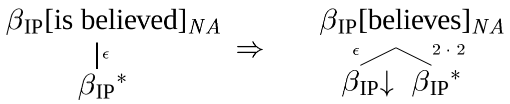
3.3 Raising and exceptional case marking

The object-level derivation which our analysis generates, motivated by dependency considerations, looks a bit unusual. In particular, the elementary tree for the subject (in this case, α_{DP}[Gabriel]) can stretch arbitrarily far away from the elementary tree for its predicate (in this case, α[to eat]). Where it ends up is below the elementary tree for the finite verb (in this case, β_{IP}[does seem]), that is, the verb which assigns it case.

A similar thing happens when we try to analyze exceptional case marking. Observe that passivizing an ECM verb yields a "raising passive":

(4) a. There are believed to be two accomplices.
    b. The cat was thought to be out of the bag.

So we can simply take the meta-level elementary tree for *seems* and "depassivize" it by relabeling the subject-position arc (ϵ) with the object position ($2 \cdot 2$), and adding a substitution node for the subject:



Then just as with a raising verb, the elementary tree for the ECM subject will end up below the elementary tree for the ECM verb, again, the verb which assigns it case.

So case is always assigned downward on the object-level derivation tree. Unfortunately unlexicalized trees like β_{IP}[to eat] prevent us from saying that case is only assigned under *immediate* domination. But we can propose the following generalization:

(5) Every DP receives case from the head of the lowest lexicalized elementary tree which dominates it on the object-level derivation tree.

In GB case was assigned under government, which was defined as a somewhat complicated common-ancestor relationship. In XTAG case is assigned using features which get passed in various ways: usually it is assigned from the case-assigner down to the argument, but for raising and ECM verbs, case is passed up through the embedded verb's elementary tree and then back down to the subject (XTAG, 1998; Kulick, 1997).

Under the present 2LTAG analysis, however, the object-level derivation provides a much more straightforward way of characterizing the configurations under which case assignment takes place. This constraint limits the number of ways the grammar writer can handle case features to a few simple possibilities: in the object-level grammar, elementary trees headed by nouns receive case; elementary trees headed by case assigners pass case features to whatever substitutes or adjoins at certain nodes; and unlexicalized trees transmit case features.

4 Related approaches

RF-2LTAG follows other work in reconciling dependency and constituency approaches to modeling natural language. One such early integration involved work by (Hays, 1964) and (Gaifman, 1965), which showed that projective dependency grammars could be represented by CFGs. However, it is known that there are common phenomena which require non-projective dependency grammars (Kahane, Nasr, and Rambow, 1998), so looking only at projective dependency grammars is inadequate. Following the observation of TAG derivations' similarity to dependency relations, other formalisms have also looked at relating dependency and constituency approaches to grammar formalisms.

A more recent instance is D-Tree Substitution Grammars (DSG) (Rambow, Weir, and Vijay-Shanker, 1995). In this formalism the derivations are also interpreted as dependency relations, and there is an object-level representation which combines via the operations of subsertion and sister-adjunction. Thought of in the terms of this paper, there is a clear parallel with RF-2LTAG, with a local set having some yield function applied to it, although in the case of DSG it is not a composition of TAG yield functions; the idea of non-immediate dominance also appears in both formalisms. The difference between the two is in the kinds of languages that they are able to describe: DSG is both less and more restrictive than RF-2LTAG. DSG can generate the language COUNT-k for some arbitrary k (that is, $\{{a_1}^n {a_2}^n \ldots {a_k}^n\}$), which makes it extremely powerful (by comparison, RF-2LTAG can only generate COUNT-4; and even if the metagrammar is not in regular form, 2LTAG can only generate COUNT-8). However, unlike RF-2LTAG it cannot generate the copy language (that is, $\{ww \mid w \in \Sigma^*\}$ with Σ some terminal alphabet); this may be problematic for a formalism modeling natural language, given the key role of the copy language in demonstrating that natural language is not context-free (Shieber, 1985). RF-2LTAG is thus a more constrained relaxation of the notion of immediate dominance in favor of non-immediate dominance than is the case for DSG.

Another formalism of particular interest here is the Segmented Adjoining Grammar of (Kulick, 2000). This generalization of TAG is characterized by an extension of the adjoining operation, motivated by evidence in scrambling, clitic climbing and subject-to-subject raising. Most interestingly, this extension to TAG, proposed on empirical grounds, is defined by a composition operation with constrained non-immediate dominance links that looks quite similar to the formalism described in this paper, which began purely from formal considerations and was then applied to data. This confluence suggests that the ideas described here might be reaching towards some deeper connection.

5 Conclusion

From a theoretical perspective, integrating dependency and constituency into a common framework is an interesting exercise. It also, however, proves to be useful in modeling otherwise problematic constructions, such as subject-auxiliary inversion and bridge and raising verb interleaving, one application of which resolves difficulties with the Synchronous TAG formalism. Moreover, the formalism developed from theoretical considerations, presented in this paper, has similar properties to work developed on empirical grounds, suggesting that this is worth further exploration.

References

Abney, Steven. 1995. Chunks and dependencies: Bringing processing evidence to bear on syntax. In *Computational Linguistics and the Foundations of Linguistic Theory*. CSLI.

Anderson, J. 1971. *The Grammar of Case: Towards a Localistic Theory*. Cambridge University Press, Cambridge, UK.

Bauer, Laurie. 1979. Some thoughts on dependency grammar. *Linguistics*, 17:301–315.

Becker, Tilman, Aravind Joshi, and Owen Rambow. 1991. Long distance scrambling and tree adjoining grammars. In *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics (EACL'91)*, pages 21–26.

Chiang, David, William Schuler, and Mark Dras. 2000. Some Remarks on an Extension of Synchronous TAG. In *Proceedings of TAG+5*, Paris, France.

Chomsky, Noam. 1986. *Knowledge of language*. Praeger, New York, NY.

Covington, Michael. 1984. *Syntactic Theory in the High Middle Ages*. Cambridge University Press, London, UK.

Dras, Mark. 1999. A meta-level grammar: redefining synchronous TAG for translation and paraphrase. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL '99)*.

Frank, Robert. 1992. *Syntactic locality and tree adjoining grammar: grammatical acquisition and processing perspectives*. Ph.D. thesis, Computer Science Department, University of Pennsylvania.

Gaifman, Haim. 1965. Dependency Systems and Phrase-Structure Systems. *Information and Control*, 8:304–337.

Gladkij, Aleksej. 1980. Opisanie sintaksičeskoj struktury predloženija s pomošč'ju sistem sintaksičeskix grupp. *Slavica*, 17:5–38.

Hays, David. 1964. Dependency Theory: A Formalism and Some Observations. *Language*, 40:511–525.

Hudson, Richard. 1976. *Arguments for a Non-transformational Grammar*. Chicago University Press, Chicago, IL.

Iordanskaja, Lidja, Myunghee Kim, Richard Kittredge, Benoit Lavoie, and Alain Polguhre. 1992. Generation of Extended Bilingual Statistical Reports. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 1019–1023, Nantes, France.

Joshi, Aravind. 2000. Relationship between strong and weak generative power of formal systems. In *Proceedings of the Fifth International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+5)*, pages 107–114, Paris, France.

Joshi, Aravind and K. Vijay-Shanker. 1999. Compositional Semantics with Lexicalized Tree-Adjoining Grammar (LTAG): How Much Underspecification is Necessary? In *Proceedings of the 2nd International Workshop on Computational Semantics*.

Kahane, Sylvain, Alexis Nasr, and Owen Rambow. 1998. Pseudo-projectivity: A polynomially parsable non-projective dependency grammar. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL '98)*, Montreal, Canada.

Kulick, Seth. 1997. Exceptional case marking in the xtag system. In Alexis Dimitriades, editor, *Current Work in Linguistics*, volume 4.3 of *University of Pennsylvania Working Papers in Linguistics*.

Kulick, Seth. 2000. A uniform account of locality constraints for clitic climbing and long scrambling. In *Proceedings of the Penn Linguistics Colloquium*.

Lecerf, Yves. 1960. Programme des conflits, modèle des conflits. *Bulletin bimestrial de l'ATALA*, 4.

Magerman, David. 1995. Statistical decision-tree models for parsing. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL'95)*.

Mel'čuk, Igor. 1988. *Dependency syntax: theory and practice* . State University of NY Press, Albany.

Mel'čuk, Igor and Nikolaj Pertsov. 1987. *Surface Syntax of English: A Formal Model within the Meaning-Text Framework*. John Benjamins, Amsterdam, The Netherlands.

Palmer, Martha, Joseph Rosenzweig, and William Schuler. 1998. Capturing Motion Verb Generalizations with Synchronous TAG . In Patrick St. Dizier, editor, *Predicative Forms in NLP*. Kluwer Press.

Peters, P. and R. Ritchie. 1973. On the generative power of transformational grammars. *Information Sciences*, 6:49–83.

Rambow, Owen. 1994. *Formal and computational aspects of natural language syntax*. Ph.D. thesis, Computer Science Department, University of Pennsylvania.

Rambow, Owen and Aravind Joshi. 1997. A Formal Look at Dependency Grammars and Phrase-Structure Grammars, with Special Consideration of Word-Order Phenomena. In Leo Wanner, editor, *Current Issues in Meaning-Text Theory*. Pinter, London.

Rambow, Owen, David Weir, and K. Vijay-Shanker. 1995. D-tree grammars. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL '95)*.

Robinson, Jane. 1970. Dependency structures and transformational rules. *Language*, 46(2):259–285.

Rogers, James. 1994. Capturing CFLs with tree adjoining grammars. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL '94)*.

Rogers, James. 1997. A unified notion of derived and derivation structures in tag. In *Proceedings of the Fifth Meeting on Mathematics of Language (MOL5)*, Dagstuhl, Germany.

Schabes, Yves and Stuart M. Shieber. 1994. An alternative conception of tree-adjoining derivation. *Computational Linguistics*, 20(1):91–124.

Sgall, Petr, Eva Hajičová, J. Panevová, and J. L. Mey, editors. 1986. *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*. Academia, Prague.

Shieber, Stuart. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343.

Shieber, Stuart M. 1994. Restricting the weak-generative capability of synchronous tree adjoining grammars. *Computational Intelligence*, 10(4).

Srinivas, B. 1997. *Complexity of Lexical Descriptions and its Relevance to Partial Parsing*. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania.

Tesnière, Lucien. 1959. *Éléments de syntaxe structurale*. Librarie Klincksieck, Paris.

Vater, H. 1975. Toward a generative dependency theory. *Lingua*, 36:121–145.

Vijay-Shanker, K. 1987. *A study of tree adjoining grammars*. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania.

Weir, David. 1988. *Characterizing Mildly Context-Sensitive Grammar Formalisms*. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania.

XTAG. 1998. A lexicalized tree adjoining grammar for english. Technical Report IRCS-98-18, Institute for Research in Cognitive Science, University of Pennsylvania.