Separating Dependency from Constituency in a Tree Rewriting System*

Anoop Sarkar
Department of Computer and Information Science
University of Pennsylvania
200 South 33rd St, Philadelphia PA 19104
anoop@linc.cis.upenn.edu

1 Introduction

We define a new grammar formalism called Link-Sharing Tree Adjoining Grammar (LSTAG) which arises directly out of a concern for distinguishing the notion of constituency from the notion of relating lexical items in terms of linguistic dependency[1](Mel'ĉuk, 1988; Rambow and Joshi, 1992). This work derives directly from work on Tree Adjoining Grammars (TAG) (Joshi, Levy, and Takahashi, 1975) where these two notions are conflated. The set of derived trees for a TAG correspond to the traditional notions of constituency while the derivation trees of a TAG are closely related to dependency structure (Rambow and Joshi, 1992). A salient feature of TAG is the extended domain of locality it provides for stating these dependencies. Each elementary tree can be associated with a lexical item giving us a lexicalized TAG (LTAG)(Joshi and Schabes, 1991). Properties related to the lexical item such as subcategorization, agreement, and certain types of word-order variation can be expressed directly in the elementary tree (Kroch, 1987; Frank, 1992). Thus, in an LTAG all of these linguistic dependencies are expressed locally in the elementary trees of the grammar. This means that the predicate and its arguments are always topologically situated in the same elementary tree.

However, in coordination of predicates, e.g. (1), the dependencies between predicate and argument cannot be represented in a TAG elementary tree directly, since several elementary trees seem to be 'sharing' their arguments.

(1) a. Kiki frolics, sings and plays all day.
b. Kiki likes and Bill thinks Janet likes soccer.

*Thanks to Christy Doran, Aravind Joshi, Nobo Komagata, Owen Rambow, and B. Srinivas for their helpful comments and discussion.

[1]The term dependency is used here broadly to include formal relationships such as case and agreement and other relationships such as filler-gap.

The idea behind LSTAG is that the non-local nature of coordination as in (1) (for TAG-like grammar formalisms) can be captured by introducing a restricted degree of *synchronized* parallelism into the TAG rewriting system while retaining the existing *independent* parallelism[2](Engelfriet, Rozenberg, and Slutzki, 1980; Rambow and Satta, to appear). We believe that an approach towards coordination that explicitly distinguishes the dependencies from the constituency gives a better formal understanding of its representation when compared to previous approaches that use tree-rewriting systems which conflate the two issues, as in (Joshi, 1990; Joshi and Schabes, 1991; Sarkar and Joshi, 1996) which have to represent sentences such as (1) with either unrooted trees or by performing structure merging on the derived tree. Other formalisms for coordination have similar motivations: however their approaches differ, e.g. CCG (Steedman, 1985; Steedman, 1997b) extends the notion of constituency, while generative syntacticians (Moltmann, 1992; Muadz, 1991) work with three-dimensional syntactic trees.

2 Synchronized Parallelism

The terms *synchronized* parallelism and *independent* parallelism arise from work done on a family of formalisms termed parallel rewriting systems that extend context-free grammars (CFG) by the addition of various restrictive devices (see (Engelfriet, Rozenberg, and Slutzki, 1980))). Synchronized parallelism allows derivations which include substrings which have been generated by a common (or shared) underlying derivation process[3]. Independent parallelism corresponds to the instantiations of independent derivation processes which are then combined to give the entire derivation of a string[4]. What we are exploring in this paper is an example of a mixed system with both independent and synchronous parallelism.

In (Rambow and Satta, to appear) it is shown that by allowing an unbounded degree of synchronized parallelism we get systems that are too unconstrained. However, interesting subfamilies arise when the synchronous parallelism is bounded to a finite degree, i.e. only a bounded number of subderivations can be synchronized in a given grammar. The system we define has this property.

[2]It is important to note that while the adjunction operation in TAGs is "context-free", synchronized parallelism could be attributed to the TAG formalism due to the string wrapping capabilities of adjunction, since synchronized parallelism is concerned with how strings are derived in a rewriting system. We note this as a conjecture but will not attempt to prove it here.

[3]The Lindenmayer systems are examples of systems with only synchronous parallelism and it is interesting to note that these L systems have the anti-AFL property (where none of the standard closures apply).

[4]CFG is a formalism that only has independent parallelism.

3 LSTAG

We first look at the formalism of Synchronous TAG (STAG)(Shieber and Schabes, 1990) since it is an example of a tree-rewriting system that has synchronized parallelism.

As a preliminary we first informally define Tree Adjoining Grammars (TAG). For example, Figure 1 shows an example of a tree for a transitive verb *cooked*. Each node in the tree has a unique address obtained by applying a Gorn tree addressing scheme. For instance, the object *NP* has address 2.2. In the TAG formalism, trees can be composed using the two operations of *substitution* (corresponds to string concatenation) and *adjunction* (corresponds to string wrapping). A history of these operations on elementary trees in the form of a derivation tree can be used to reconstruct the derivation of a string recognized by a TAG. Figure 2 shows an example of a derivation tree and the corresponding parse tree for the derived structure obtained when $\alpha(John)$ and $\alpha(beans)$ substitute into $\alpha(cooked)$ and $\beta(dried)$ adjoins into $\alpha(beans)$ giving us a derivation tree for *John cooked dried beans*. Trees that adjoin are termed as *auxiliary trees*, trees that are not auxiliary are called *initial*. Each node in the derivation tree is the name of an elementary tree. The labels on the edges denote the address in the parent node where a substitution or adjunction has occured.

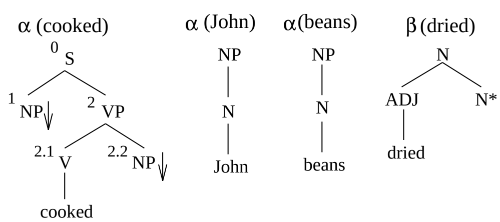

Figure 1: Example of a TAG

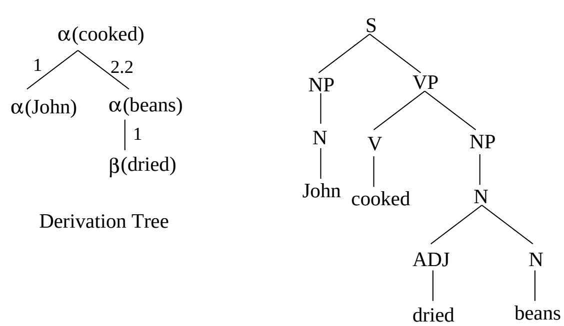

Figure 2: Example of a derivation tree and corresponding parse tree

Definition 1 *In a* TAG $G = \{\gamma \mid \gamma$ *is either an* initial *tree or an* auxiliary *tree* $\}$*, we will notate adjunction (similarly substitution) of trees* $\gamma_1 \dots \gamma_k$ *into tree* γ *at addresses* $a_1 \dots a_k$ *giving a derived tree* γ' *as*

$$\gamma' = \gamma[a_1, \gamma_1] \dots [a_k, \gamma_k]$$

Definition 2 *Given two standard* TAG*s* G_L *and* G_R *we define (from (Shieber, 1994)) a* STAG *as* $\{\langle \gamma, \gamma', \frown \rangle \mid \gamma \in G_L, \gamma' \in G_R\}$*, where* $\frown$ *is a set of* links *from a node address in* γ *to a node address in* γ'*. A derivation proceeds as follows:*

- *for* $\gamma = \langle \gamma_L, \gamma_R, \frown \rangle$*, pick a link member* $a_L \frown_i a_R$*, where the a's are node addresses and* $\frown_i \in \frown$*. For simplicity, we refer to* $\frown$ *as* link *and its elements* $\frown_i$ *as link* members.
- *adjunction (similarly substitution) of* $\langle \beta_L, \beta_R, \frown' \rangle$ *into* γ *is given by*

$$\langle \gamma'_L, \gamma'_R, \frown'' \rangle = \langle \gamma_L[a_L, \beta_L], \gamma_R[a_R, \beta_R], \frown'' \rangle$$

  where all links in $\frown$ *and* $\frown'$ *are included in* $\frown''$ *except* $\frown_i$*.*
- $\langle \gamma'_L, \gamma'_R, \frown'' \rangle$ *is now a derived structure which can be further operated upon.*

In (Abeillé, 1992; Abeillé, 1994) STAGs have been used in handling non-local dependencies and to seperate syntactic attachment from semantic roles. However, STAG cannot be used to seperate the dependencies created in (pairs of) derivation trees for coordinate structures from the constituency represented in these derivation trees. In this particular sense, STAG has the same shortcomings of a TAG. Also the above definition of the inheritance of links in derived structures allows STAG to derive strings not generable by TAG (Shieber, 1994). We look at a modified version of STAGs which is weaker in power than STAGs as defined in Defn 2. We call this formalism *Link-Sharing* TAG (LSTAG).

Definition 3 *An* LSTAG G *is defined as a 4-tuple* $\langle G_L, G_R, \Delta, \Phi \rangle$ *where* G_L, G_R *are standard* TAG*s,* Δ *and* Φ *are disjoint sets of sets of links and for each pair* $\gamma = \langle \gamma_L, \gamma_R \rangle$*, where* $\gamma_L \in G_L$ *and* $\gamma_R \in G_R$*,* $\delta_\gamma \in \Delta$ *is a subset of* links *in* γ *and* $\phi_{\gamma_R} \in \Phi$ *is a distinguished subset of* links *with the following properties:*

- *for each link* $\frown \in \phi_{\gamma_R}$*,* $\eta \frown \eta$*, where* η *is a node address in* γ_R*. i.e.* ϕ_{γ_R} *is a set of reflexive links.*
- δ_R *and* ϕ_{γ_R} *have some canonical order* $\prec$*.*
- *adjunction (similarly substitution) of* $\langle \beta_L, \beta_R \rangle$ *into* γ *is given by*

$$\langle \gamma'_L, \gamma'_R \rangle = \langle \gamma_L[a_L, \beta_L], \gamma_R[a_R, \beta_R] \rangle$$

  and for all $\gamma_i \in \delta_\gamma, \beta_i \in \phi_{\beta_R} (1 \leq i \leq n)$ *(card(*δ_γ*)* $\geq$ *card(*β_R*))*

$$\delta_\gamma \sqcup \phi_{\beta_R} \stackrel{def}{=} \frown_{\gamma_1} \sqcup \frown_{\beta_1} \cup \dots \cup \frown_{\gamma_n} \sqcup \frown_{\beta_n}$$

where

$$\frown_{\gamma_1} \prec \frown_{\gamma_2}, \ldots, \frown_{\gamma_{n-1}} \prec \frown_{\gamma_n}$$

and

$$\frown_{\beta_{R_1}} \prec \frown_{\beta_{R_2}}, \ldots, \frown_{\beta_{R_{n-1}}} \prec \frown_{\beta_{R_n}}$$

- $\frown_i \sqcup \frown_j$ *is a set of links defined as follows. If* $a_{L_i} \frown_i a_{R_i}$ *and* $a_{R_j} \frown_j a_{R_j}$*, then*

$$\frown_i \sqcup \frown_j \stackrel{def}{=} \{a_{L_i} \frown a_{R_i}\} \cup \{a_{L_i} \frown a_{R_j}\}$$

- $\langle \gamma'_L, \gamma'_R \rangle$ *is the new derived structure with new set of links* $\delta_\gamma \sqcup \phi_{\beta_R}$.

Φ is used to derive synchronized parallelism in G_R. The ordering $\prec$ is simply used to match up the links being shared via the (non-local) sharing operation $\sqcup$.

This ordering $\prec$ can be defined in terms of node addresses or "first argument $\prec$ second argument", i.e. ordering the arguments of the two predicates being coordinated.

It is important to note that only the links in Φ are used non-locally and they are always exhausted in a single adjunction (or substitution) operation. No links from Δ are ever inherited unlike STAGs. Hence, non-locality is only used in a restricted fashion for the notion of 'sharing'.

4 Linguistic Relevance

To explain how the formalism works consider sentence (2).

(2) John cooks and eats beans.

Consider a LSTAG $G = \{\gamma, \beta, \alpha, \upsilon\}$ partially shown in Fig. 3(a) and Fig. 3(b). α and υ are analogously defined for *John* and *beans* respectively (see Fig. 1). In Fig. 3(a) $\delta_\gamma = \{1, 2\}$[5]and $\phi_{\gamma_R} = \{\}$, while for Fig. 3(b) $\delta_\gamma = \{\}$ and $\phi_{\gamma_R} = \{1, 2\}$.

It is important to note that our initial motivation about seperating dependency from the constituency information is highlighted in β (see Fig. 3(b)) where the first projection will only contribute information about constituency in a derivation tree while the second projection will contribute only dependency information in a derivation tree. We conjecture that this is true for all the structures defined in an LSTAG. the kind of questions addressed in (Rambow, Vijay-Shanker, and Weir, 1995) can perhaps be answered within the framework of LSTAG[6].

[5]We are just using numbers $1, 2, \ldots$ to denote the links rather than use the Gorn notation to make the trees easier to read. Here, link number 1 stands for $1 \frown 1$ and 2 stands for $2.2 \frown 2.2$

[6]In (Rambow, Vijay-Shanker, and Weir, 1995) a new formalism called D-Tree Grammars was introduced in order to bring together the notion of derivation tree in a TAG with the notion of dependency grammar (Mel'ĉuk, 1988). Perhaps the kind of questions addressed in (Rambow, Vijay-Shanker, and Weir, 1995) can also be handled using the current framework. Such an application of the formalism would motivate the need for trees like γ in Fig. 3 independent of the coordination facts since they would be required to get the dependencies right.

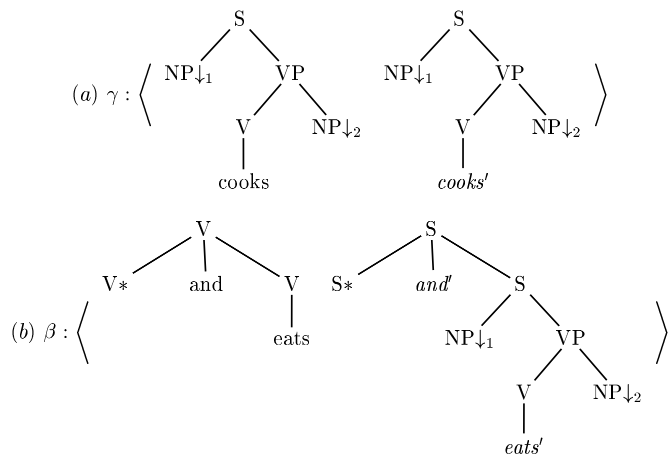

Figure 3: Trees γ and β from LSTAG G

The derived structure after β adjoins onto γ is shown in Fig. 4(a). Fig. 5(a) shows the derived tree after the tree α (for *John*) substitutes into γ. Notice that due to link sharing, substitution is shared, effectively forming a "tangled" derived tree[7]. In Figs. 4 and 5 the derivation trees are also given (associated with each element). The derivation structure for the second element in Fig. 5(b) is a directed acyclic derivation graph which gives us information about dependency we expect. The derivation tree of the first element in Fig. 5(b), on the other hand, gives us information about constituency.

The notion of link sharing is closely related to the schematization of the coordination rule in (Steedman, 1997b) shown below in combinatory notation.

$$\begin{aligned} bxy &\equiv bxy \\ bfg &\equiv \lambda x.b(fx)(gx) \\ bfg &\equiv \lambda x.\lambda y.b(fxy)(gxy) \\ \ldots & \end{aligned}$$

Link sharing is used to combine the interpretation of the predicate arguments f and g (e.g. *cooks*, *eats*) of the conjunction b with the interpretation of the arguments of those predicates $x, y, \ldots$. However, it does this within a tree-rewriting system, unlike the use of combinators in (Steedman, 1997b).

[7] While this notion of sharing bears some resemblance to the notion of *joining node* in the three-dimensional trees used in (Moltmann, 1992; Muadz, 1991) the rules for semantic interpretation of the derivations produced in a LSTAG is considerably less obscure than the rules needed to interpret 3D trees; crucially because elementary structures in a TAG-like formalisms are taken to be semantically minimal without being semantically void.

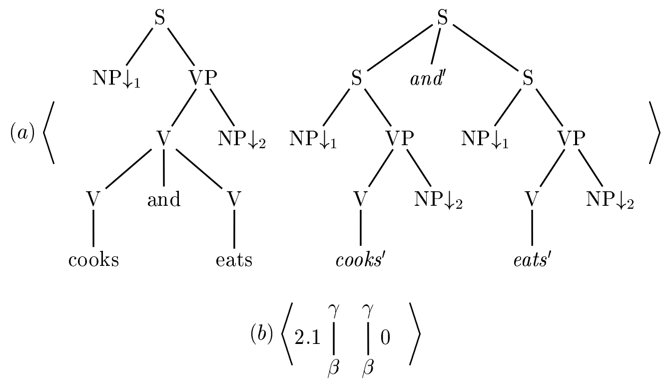

Figure 4: Derived and derivation structures after β adjoins into γ.

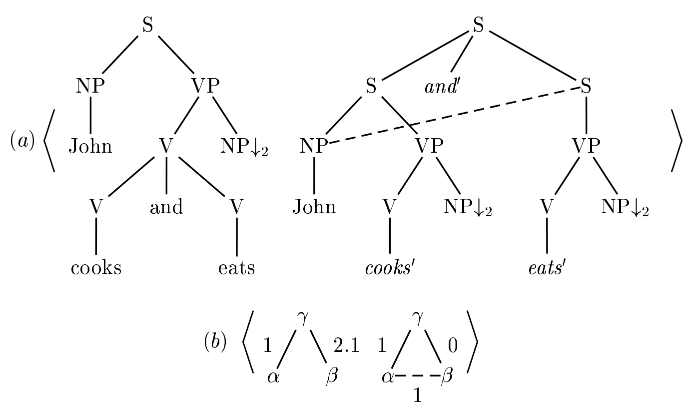

Figure 5: Substitution of α

5 Restrictions

Having defined the formalism of LSTAG, we now define certain restrictions on the grammar that can be written in this formalism in order to capture correctly certain facts about coordinate structures in English.

For instance, we need to prohibit elementary structures like the one in Fig. 6 because they give rise to ungrammatical sentences like (3).

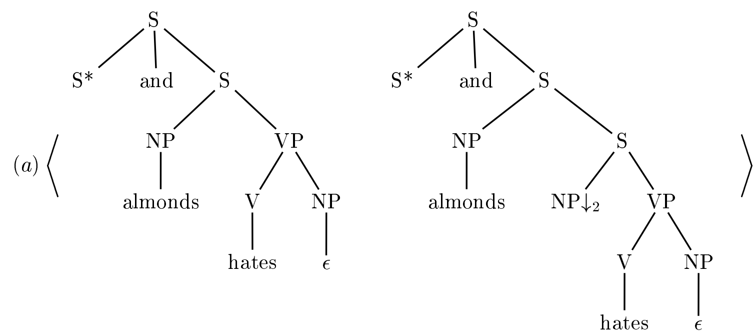

Figure 6: Discontiguous elementary structure

(3) *Peanuts John likes and almonds hates. (Joshi, 1990)

However, such restrictions in the context of TAGs have been discussed before. (Joshi, 1990) rules out (3) by stating a requirement on the lexical string spelled out by the elementary tree. If the lexical string spelled out is not contiguous then it cannot coordinate. This requirement is stated to be a phonological condition and relates the notion of an *intonational phrase* (IP) to the notion of appropriate fragments for coordination (in the spirit of (Steedman, 1997a)). It is important to note that the notions of phrase structure for coordination and intonational phrases defined in (Joshi, 1990) for TAG are not identical, whereas they *are* identical for CCG (Steedman, 1997a).

We can state an analogous restriction on the formation of elementary structures in a LSTAG, one that is motivated by the notion of link sharing. The left element of an elementary structure in a LSTAG cannot be composed of discontinuous parts of the right element. For example, in Fig. 6 the segment $[_S[_{NP_\downarrow}][_{VP}]]$ from the right element has been excised in the left element. This restriction corresponds to the notion that the left element of a structure in a LSTAG represents constituency.

6 Conclusion

We have presented a new tree-rewriting formalism called Link-Sharing Tree Adjoining Grammar (LSTAG) which is a variant of synchronous TAGs (STAG).

Using LSTAG we defined an approach towards coordination where linguistic dependency is distinguished from the notion of constituency. Appropriate restrictions on the nature of elementary structures in a LSTAG were also defined. Such an approach towards coordination that explicitly distinguishes dependencies from constituency gives a better formal understanding of its representation when compared to previous approaches that use tree-rewriting systems which conflate the two issues (see (Joshi and Schabes, 1991; Sarkar and Joshi, 1996)). The previous approaches had to represent coordinate structures either with unrooted trees or by performing structure merging on the parse tree. Moreover, the linguistic analyses presented in (Joshi and Schabes, 1991; Sarkar and Joshi, 1996) can be easily adopted in the current formalism.

References

Abeillé, Anne. 1992. Synchronous TAGs and French Pronominal Clitics. In *Proc. of COLING-92*, pages 60–66, Nantes, Aug 23–28.

Abeillé, Anne. 1994. Syntax or Semantics? Handling Nonlocal Dependencies with MCTAGs or Synchronous TAGs. *Computational Intelligence*, 10(4):471–485.

Engelfriet, J., G. Rozenberg, and G. Slutzki. 1980. Tree transducers, *L* systems, and two-way machines. *Journal of Computer and System Science*, 43:328–360.

Frank, Robert. 1992. *Syntactic locality and Tree Adjoining Grammar: grammatical, acquisition and processing perspectives.* Ph.D. thesis, University of Pennsylvania,IRCS-92-47.

Joshi, A. and Y. Schabes. 1991. Tree adjoining grammars and lexicalized grammars. In M. Nivat and A. Podelski, editors, *Tree automata and languages*. North-Holland.

Joshi, Aravind. 1990. Phrase Structure and Intonational Phrases: Comments on the papers by Marcus and Steedman. In G. Altmann, editor, *Computational and Cognitive Models of Speech*. MIT Press.

Joshi, Aravind and Yves Schabes. 1991. Fixed and flexible phrase structure: Coordination in Tree Adjoining Grammar. In *Presented at the DARPA Workshop on Spoken Language Systems*, Asilomar, CA.

Joshi, Aravind K., L. Levy, and M. Takahashi. 1975. Tree Adjunct Grammars. *Journal of Computer and System Sciences*.

Kroch, A. 1987. Subjacency in a tree adjoining grammar. In A. Manaster-Ramer, editor, *Mathematics of Language*. J. Benjamins Pub. Co., pages 143–172.

Mel'ĉuk, I. 1988. *Dependency Syntax: Theory and Practice*. State University of New York Press, Albany.

Moltmann, Friederike. 1992. On the Interpretation of Three-Dimensonal Syntactic Trees. In Chris Barker and David Dowty, editors, *Proc. of SALT-2*, pages 261–281, May 1-3.

Muadz, H. 1991. *A Planar Theory of Coordination*. Ph.D. thesis, University of Arizona, Tucson, Arizona.

Rambow, O. and A. Joshi. 1992. A formal look at dependency grammars and phrase-structure grammars, with special consideration to word-order phenomena. In

Intern. Workshop on the Meaning-Text Theory, pages 47–66, Arbeitspapiere der GMD 671. Darmstadt.

Rambow, O. and G. Satta. to appear. Independent parallelism in finite copying parallel rewriting systems. *Theor. Comput. Sc.*

Rambow, O., K. Vijay-Shanker, and D. Weir. 1995. D-Tree Grammars. In *Proceedings of the 33rd Meeting of the ACL*.

Sarkar, Anoop and Aravind Joshi. 1996. Coordination in TAG: Formalization and implementation. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING'96)*, Copenhagen.

Shieber, S. 1994. Restricting the weak generative capacity of synchronous tree adjoining grammars. *Computational Intelligence*, 10(4):371–385, November.

Shieber, Stuart and Yves Schabes. 1990. Synchronous Tree Adjoining Grammars. In *Proceedings of the* 13^{th} *International Conference on Computational Linguistics (COLING'90)*, Helsinki, Finland.

Steedman, Mark. 1985. Dependency and coordination in the grammar of Dutch and English. *Language*, 61:523–568.

Steedman, Mark. 1997a. Information Structure and the Syntax-Phonology Interface. manuscript. Univ. of Pennsylvania.

Steedman, Mark. 1997b. *Surface Structure and Interpretation: Unbounded and Bounded Dependency in Combinatory Grammar*. Linguistic Inquiry monograph. MIT Press.