A Meta-Level Grammar: Redefining Synchronous TAG for Translation and Paraphrase

Mark Dras
Microsoft Research Institute
Department of Computer Science
Macquarie University, Australia
markd@ics.mq.edu.au

Abstract

In applications such as translation and paraphrase, operations are carried out on grammars at the meta level. This paper shows how a meta-grammar, defining structure at the meta level, is useful in the case of such operations; in particular, how it solves problems in the current definition of Synchronous TAG (Shieber, 1994) caused by ignoring such structure in mapping between grammars, for applications such as translation. Moreover, essential properties of the formalism remain unchanged.

1 Introduction

A grammar is, among other things, a device by which it is possible to express structure in a set of entities; a grammar formalism, the constraints on how a grammar is allowed to express this. Once a grammar has been used to express structural relationships, in many applications there are operations which act at a 'meta level' on the structures expressed by the grammar: for example, lifting rules on a dependency grammar to achieve pseudo-projectivity (Kahane *et al*, 1998), and mapping between synchronised Tree Adjoining Grammars (TAGs) (Shieber and Schabes, 1990; Shieber 1994) as in machine translation or syntax-to-semantics transfer. At this meta level, however, the operations do not themselves exploit any structure. This paper explores how, in the TAG case, using a meta-level grammar to define meta-level structure resolves the flaws in the ability of Synchronous TAG (S-TAG) to be a representation for applications such as machine translation or paraphrase.

This paper is set out as follows. It describes the expressivity problems of S-TAG as noted in Shieber (1994), and shows how these occur also in syntactic paraphrasing. It then demonstrates, illustrated by the relative structural complexity which occurs at the meta level in syntactic paraphrase, how a meta-level grammar resolves the representational problems; and it further shows that this has no effect on the generative capacity of S-TAG.

2 S-TAG and Machine Translation

Synchronous TAG, the mapping between two Tree Adjoining Grammars, was first proposed by Shieber and Schabes (1990). An application proposed concurrently with the definition of S-TAG was that of machine translation, mapping between English and French (Abeillé *et al*, 1990); work continues in the area, for example using S-TAG for English-Korean machine translation in a practical system (Palmer *et al*, 1998).

In mapping between, say, English and French, there is a lexicalised TAG for each language (see XTAG, 1995, for an overview of such a grammar). Under the definition of TAG, a grammar contains elementary trees, rather than flat rules, which combine together via the operations of substitution and adjunction (composition operations) to form composite structures—derived trees—which will ultimately provide structural representations for an input string if this string is grammatical. An overview of TAGs is given in Joshi and Schabes (1996).

The characteristics of TAGs make them better suited to describing natural language than Context Free Grammars (CFGs): CFGs are not adequate to describe the entire syntax of natural language (Shieber, 1985), while TAGs are able to provide structures for the constructions problematic for CFGs, and without a much greater generative capacity. Two particular characteris-

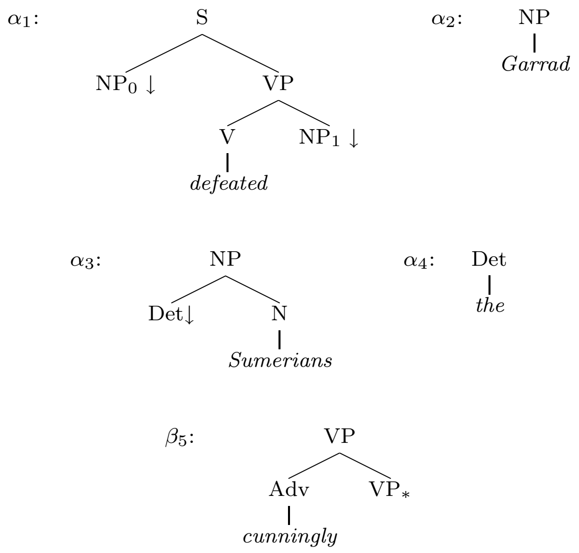

Figure 1: Elementary TAG trees

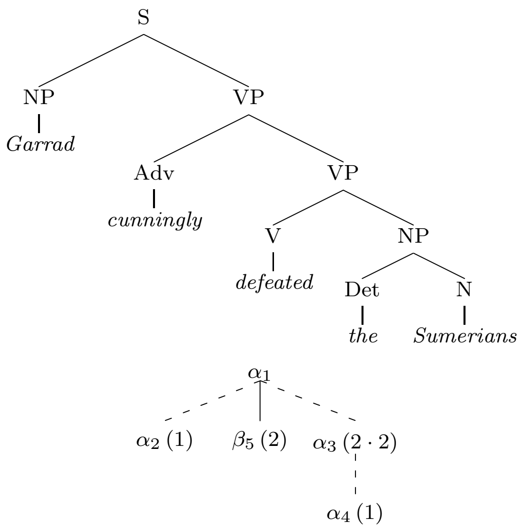

Figure 2: Derived and derivation trees, respectively, for Figure 1

tics of TAG that make it well suited to describing natural language are the extended domain of locality (EDL) and factoring recursion from the domain of dependencies (FRD). In TAG, for instance, information concerning dependencies is given in one tree (EDL): for example, in Figure 1,[1] the information that the verb *defeated* has subject and object arguments is contained in the tree α_1. In a CFG, with rules of the form **S** → **NP VP** and **VP** → **V NP**, it is not possible to have information about both arguments in the same rule unless the VP node is lost. TAG keeps dependencies together, or local, no matter how far apart the corresponding lexical items are. FRD means that recursive information—for example, a sequence of adjectives modifying the object noun of *defeated*—are factored out into separate trees, leaving dependencies together.

A consequence of the TAG definition is that, unlike CFG, a TAG derived tree is not a record of its own derivation. In CFG, each tree given as a structural description to a string enables the rules applied to be recovered. In a TAG, this is not possible, so each derived tree has an associated derivation tree. If the trees in Figure 1 were composed to give a structural description for *Garrad cunningly defeated the Sumerians*, the derived tree and its corresponding derivation tree would be as in Figure 2.[2]

Weir (1988) terms the derived tree, and its component elementary trees, OBJECT-LEVEL TREES; the derivation tree is termed a META-LEVEL TREE, since it describes the object-level trees. The derivation trees are context free (Weir, 1988), that is, they can be expressed by a CFG; Weir showed that applying a TAG yield function to a context free derivation tree (that is, reading the labels off the tree, and substituting or adjoining the corresponding object-level trees as appropriate) will uniquely specify a TAG tree. Schabes and Shieber (1994) characterise this as a function $\mathcal{D}$ from derivation trees to derived trees.

The idea behind S-TAG is to take two TAGs and link them in an appropriate way so that when substitution or adjunction occurs in a tree in one grammar, then a corresponding composition operation occurs in a tree in the other grammar. Because of the way TAG's EDL captures dependencies, it is not problematic to have translations more complex than word-for-word mappings (Abeillé *et al*, 1990). For example, from the Abeillé *et al* paper, handling argument swap, as in (1), is straightforward. These would be represented by tree pairs as in Figure 3.

[1] The figures use standard TAG notation: ↓ for nodes requiring substitution, ∗ for foot nodes of auxiliary trees.

[2] In derivation trees, addresses are given using the Gorn addressing scheme, although these are omitted in this paper where the composition operations are obvious.

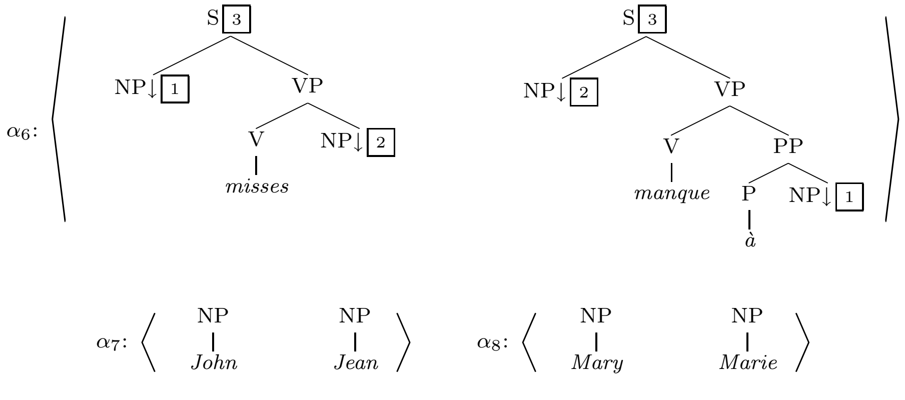

Figure 3: S-TAG with argument swap

(1) a. John misses Mary.
b. Marie manque à Jean.

In these tree pairs, a diacritic ($\boxed{\text{n}}$) represents a link between the trees, such that if a substitution or adjunction occurs at one end of the link, a corresponding operation must occur at the other end, which is situated in the other tree of the same tree pair. Thus if the tree for *John* in α_7 is substituted at $\boxed{1}$ in the left tree of α_6, the tree for *Jean* must be substituted at $\boxed{1}$ in the right tree. The diacritic $\boxed{3}$ allows a sentential modifier for both trees (e.g. *unfortunately / malheureusement*).

The original definition of S-TAG (Shieber and Schabes, 1990), however, had a greater generative capacity than that of its component TAG grammars: even though each component grammar could only generate Tree Adjoining Languages (TALs), an S-TAG pairing two TAG grammars could generate non-TALs. Hence, a redefinition was proposed (Shieber, 1994). Under this new definition, the mapping between grammars occurs at the meta level: there is an isomorphism between derivation trees, preserving structure at the meta level, which establishes the translation. For example, the derivation trees for (1) using the elementary trees of Figure 3 is given in Figure 4; there is a clear isomorphism, with a bijection between nodes, and parent-child relationships preserved in the mapping.

In translation, it is not always possible to have a bijection between nodes. Take, for example, (2).

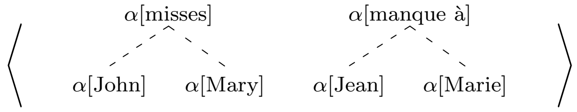

Figure 4: Derivation tree pair for Fig 3

(2) a. Hopefully John misses Mary.
b. On espère que Marie manque à Jean.

In English, *hopefully* would be represented by a single tree; in French, *on espère que* typically by two. Shieber (1994) proposed the idea of bounded subderivation to deal with such aberrant cases—treating the two nodes in the derivation tree representing *on espère que* as singular, and basing the isomorphism on this. This idea of bounded subderivation solves several difficulties with the isomorphism requirement, but not all. An example by Shieber demonstrates that translation involving clitics causes problems under this definition, as in (3). The partial derivation trees containing the clitic *lui* and its English parallel are as in Figure 5.

(3) a. The doctor treats his teeth.
b. Le docteur lui soigne les dents.

A potentially unbounded amount of material intervening in the branches of the righthand tree means that an isomorphism between the trees cannot be established under Shieber's specification even with the modification of bounded subderivations. Shieber suggested that the isomorphism requirement may be overly stringent;

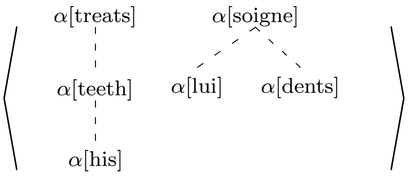

Figure 5: Clitic derivation trees

but intuitively, it seems reasonable that what occurs in one grammar should be mirrored in the other in some way, and this reflected in the derivation history.

Section 3 looks at representing syntactic paraphrase in S-TAG, where similar problems are encountered; in doing this, it can be seen more clearly than in translation that the difficulty is caused not by the isomorphism requirement itself but by the fact that the isomorphism does not exploit any of the structure inherent in the derivation trees.

3 S-TAG and Paraphrase

Syntactic paraphrase can also be described with S-TAG (Dras, 1997; Dras, forthcoming). The manner of representing paraphrase in S-TAG is similar to the translation representation described in Section 2. The reason for illustrating both is that syntactic paraphrase, because of its structural complexity, is able to illuminate the nature of the problem with S-TAG. In a specific parallel, a difficulty like that of the clitics occurs here also, for example in paraphrases such as (4).

(4) a. The jacket which collected the dust was tweed.

b. The jacket was tweed. It collected the dust.

Tree pairs which could represent the elements in the mapping between (4a) and (4b) are given in Figure 6. It is clearly the case that the trees in the tree pair α_9 are not elementary trees, in the same way that *on espère que* is not represented by a single elementary tree: in both cases, such single elementary trees would violate the Condition on Elementary Tree Minimality (Frank, 1992). The tree pair α_9 is the one that captures the syntactic rearrangement in this paraphrase; such a tree pair will be termed the STRUCTURAL MAPPING PAIR (SMP). Taking as a basic set of trees the XTAG standard grammar of English (XTAG, 1995), the derivation tree pair for (4) would be as in Figure 7.[3] Apart from α_9, each tree in Figure 6 corresponds to an elementary object-level tree, as indicated by its label; the remaining labels, indicated in bold in the meta-level derivation tree in Figure 7, correspond to the elementary object-level trees forming α_9, in much the same way that *on espère que* is represented by a subderivation comprising an *on* tree substituted into an *espère que* tree.

Note that the nodes corresponding to the left tree of the SMP form two discontinuous groups, but these discontinuous groups are clearly related. Dras (forthcoming) describes the conditions under which these discontinuous groupings are acceptable in paraphrase; these discontinuous groupings are treated as a single block with SLOTS connecting the groupings, whose fillers must be of particular types. Fundamentally, however, the structure is the same as for clitics: in one derivation tree the grouped elements are in one branch of the tree, and in the other they are in two separate branches with the possibility of an unbounded amount of intervening material, as described below in Section 4.

4 Meta-Level Structure

Example (5) illustrates why the paraphrase in (4) has the same difficulty as the clitic example in (3) when represented in S-TAG: because unbounded intervening material can occur when promoting arbitrarily deeply embedded relative clauses to sentence level, as indicated by Figure 8, an isomorphism is not possible between derivation trees representing paraphrases such as (4) and (5). Again, the component trees of the SMP are in bold in Figure 8.

(5) a. The jacket which collected the dust which covered the floor was tweed.

b. The jacket which collected the dust

[3] Node labels, the object-level tree names, are given according to the XTAG standard: see Appendix B of XTAG (1995). This is done so that the component trees of the aggregate α_9 and their types are obvious. The lexical item to which each is bound is given in square brackets, to make the trees, and the correspondence between for example Figure 6 and Figure 7, clearer.

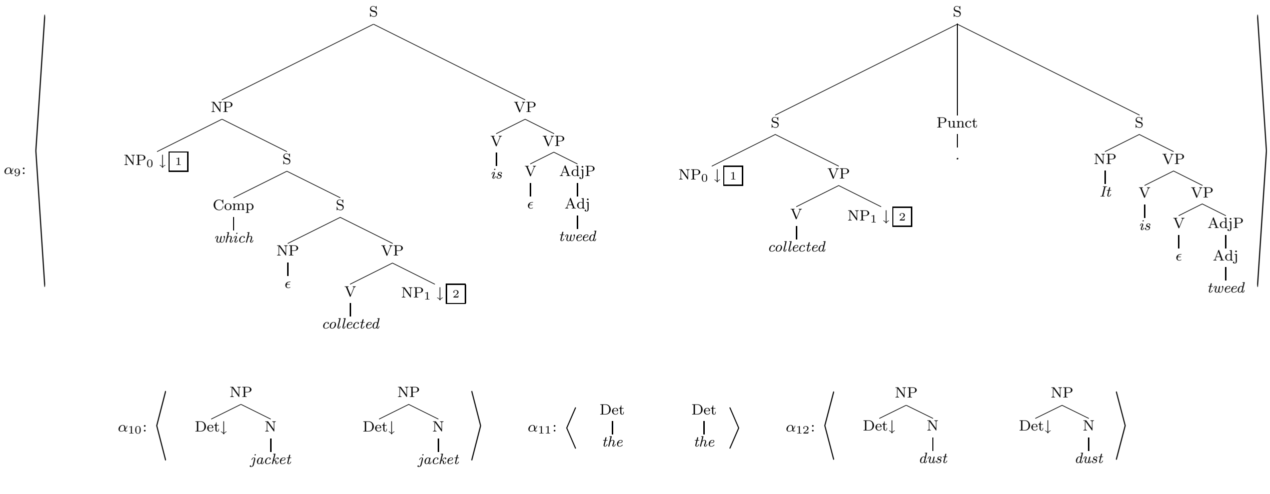

Figure 6: S-TAG for (4)

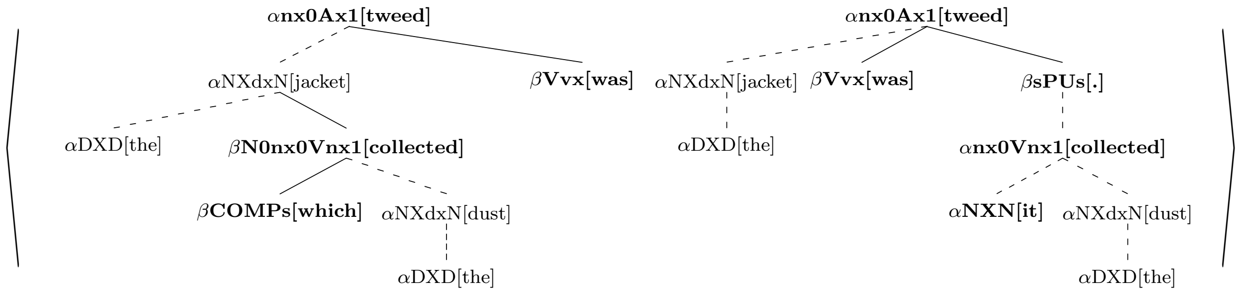

Figure 7: Derivation tree pair for example (4)

was tweed. The dust covered the floor.[4]

The paraphrase in (4) and in Figures 6 and 7, and other paraphrase examples, strongly suggest that these more complex mappings are not an aberration that can be dealt with by patching measures such as bounded subderivation. It is clear that the meta level is fundamentally not just for establishing a one-to-one onto mapping between nodes; rather, it is also about defining structures representing, for example, the SMP at this meta level: in an isomorphism between trees in Figure 8, it is necessary to regard the SMP components of each tree as a unitary substructure and map them to each other. The discontinuous groupings should form these substructures regardless of intervening material, and this is suggestive of TAG's EDL.

In the TAG definition, the derivation trees are context free (Weir, 1988), and can be expressed by a CFG. The isomorphism in the S-TAG definition of Shieber (1994) reflects this, by effectively adopting the single-level domain of locality (extended slightly in cases of bounded subderivation, but still effectively a single level), in the way that context free trees are fundamentally made from single level components and grown by concatenation of these single levels. This is what causes the isomorphism requirement to fail, the inability to express substructures at the meta level in order to map between them, rather than just mapping between (effec-

[4] The referring expression that is the subject of this second sentence has changed from *it* in (4) to *the dust* so the antecedent is clear. Ensuring it is appropriately coreferent, by using two occurrences of the same diacritic in the same tree, necessitates a change in the properties of the formalism unrelated to the one discussed in this paper; see Dras (forthcoming). Assume, for the purpose of this example, that the referring expression is fixed and given, as is the case with *it*, rather than determined by coindexed diacritics.

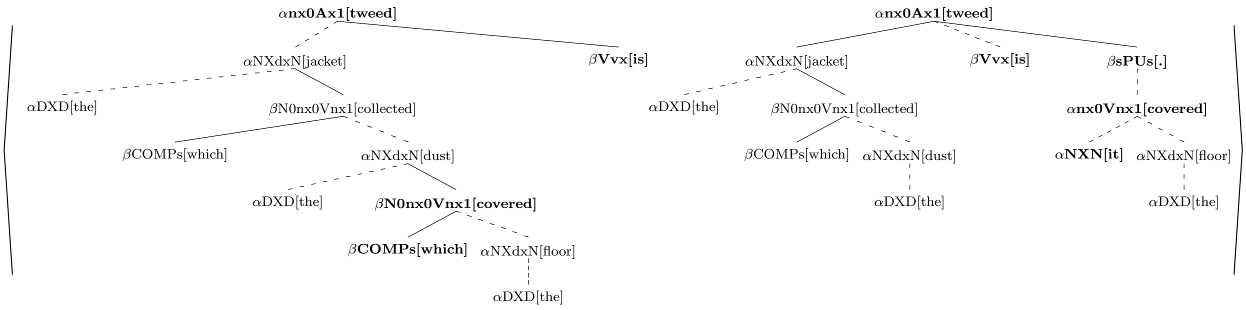

Figure 8: Derivation tree for example (5)

tively) single nodes.

To solve the problem with isomorphism, a meta-level grammar can be defined to specify the necessary substructures prior to mapping, with minimality conditions on what can be considered acceptable discontinuity. Specifically, in this case, a TAG meta-level grammar can be defined, rather than the implicit CFG, because this captures the EDL well. The TAG yield function of Weir (1988) can then be applied to these derivation trees to get derived trees. This, of course, raises questions about effects on generative capacity and other properties; these are dealt with in Section 5.

A procedure for automatically constructing a TAG meta-grammar is as follows in Construction 1. The basic idea is that where the node bijection is still appropriate, the grammar retains its context free nature (by using single-level TAG trees composed by substitution, mimicking CFG tree concatenation), but where EDL is required, multi-level TAG initial trees are defined, with TAG auxiliary trees for describing the intervening material. These meta-level trees are then mapped appropriately; this corresponds to a bijection of nodes at the meta-meta level. For (5), the meta-level grammar for the left projection then looks as in Figure 9, and for the right projection as in Figure 10. Figure 11 contains the meta-meta-level trees, the tree pair that is the derivation of the meta level, where the mapping is a bijection between nodes. Adding unbounded material would then just be reflected in the meta-meta-level as a list of β nodes depending from the β_{15}/β_{18} nodes in these trees.

The question may be asked, Why isn't it the case that the same effect will occur at the meta-meta level that required the meta-grammar in the first place, leading perhaps to an infinite (and useless) sequence? The intuition is that it is the meta-level, rather than anywhere 'higher', which is fundamentally the place to specify structure: the object level specifies the trees, and the meta level specifies the grouping or structure of these trees. Then the mapping takes place on these structures, rather than the object-level trees; hence the need for a grammar at the meta-level but not beyond.

Construction 1 *To build a TAG metagrammar:*

1. *An initial tree in the metagrammar is formed for each part of the derivation tree corresponding to the substructure representing an SMP, including the slots so that a contiguous tree is formed. Any node that links these parts of the derivation tree to other subtrees in the derivation tree is also included, and becomes a substitution node in the metagrammar tree.*

2. *Auxiliary trees are formed corresponding to the parts of the derivation trees that are slot fillers along with the nodes in the discontinuous regions adjacent to the slots; one contiguous auxiliary tree is formed for each bounded sequence of slot fillers within each substructure. These trees also satisfy certain minimality conditions.*

3. *The remaining metagrammar trees then come from splitting the derivation tree into single-level trees, with the nodes on*

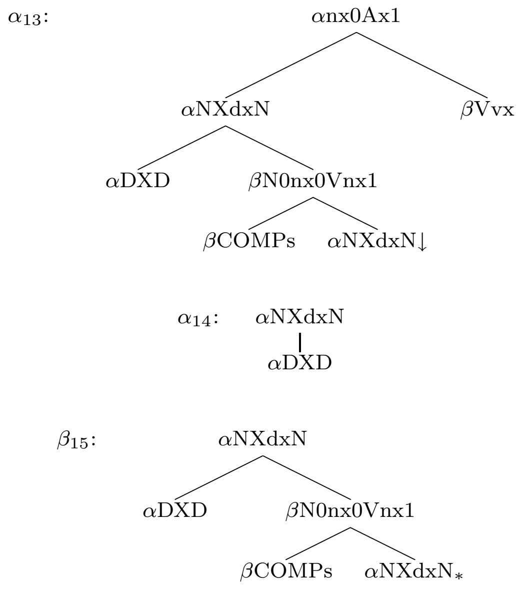

Figure 9: Meta-grammar for (5a)

these single-level trees in the metagrammar marked for substitution if the corresponding nodes in the derivation tree have subtrees.

The minimality conditions in Step 2 of Construction 1 are in keeping with the idea of minimality elsewhere in TAG (for example, Frank, 1992). The key condition is that meta-level auxiliary trees are rooted in α-labelled nodes, and have only β-labelled nodes along the spine. The intuition here is that slots (the nodes which meta-level auxiliary trees adjoin into) must be α-labelled: β-labelled trees would not need slots, as the substructure could instead be continuous and the β-labelled trees would just adjoin in. So the meta-level auxiliary trees are rooted in α-labelled trees; but they have only β-labelled trees in the spine, as they aim to represent the minimal amount of recursive material. Notwithstanding these conditions, the construction is quite straightforward.

5 Generative Capacity

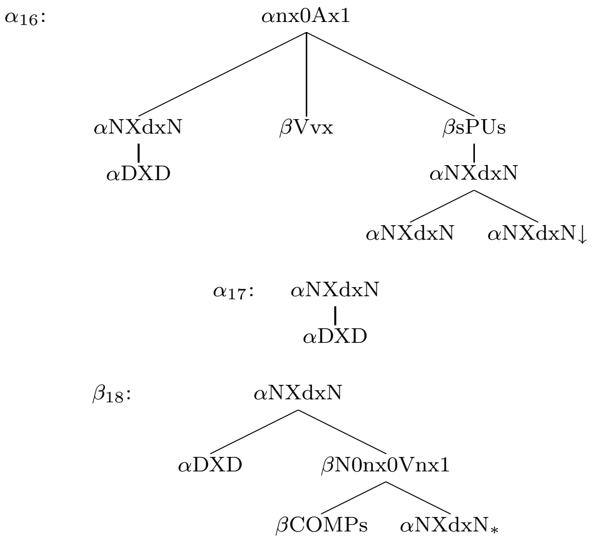

Figure 10: Meta-grammar for (5b)

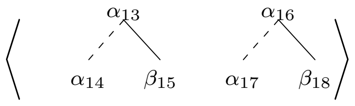

Figure 11: Derivation tree pair for Fig 3

Weir (1988) showed that there is an infinite progression of TAG-related formalisms, in generative capacity between CFGs and indexed grammars. A formalism $\mathcal{F}_i$ in the progression is defined by applying the TAG yield function to a derivation tree defined by a grammar formalism $\mathcal{F}_{i-1}$; the generative capacity of $\mathcal{F}_i$ is a superset of $\mathcal{F}_{i-1}$. Thus using a TAG meta-grammar, as described in Section 4, would suggest that the generative capacity of the object-level formalism would necessarily have been increased over that of TAG.

However, there is a regular form for TAGs (Rogers, 1994), such that the trees of TAGs in this regular form are local sets; that is, they are context free. The meta-level TAG built by Construction 1 with the appropriate conditions on slots is in this regular form. A proof of this is in Dras (forthcoming); a sketch is as follows. If adjunction may not occur along the spine of another auxiliary tree, the grammar is in regular form. This kind of adjunction does not occur under Construction 1 because all meta-level auxiliary trees are rooted in α-labelled trees (object-level initial trees), while their spines consist only of β-labelled trees (object-level auxiliary trees).

Since the meta-level grammar is context free, despite being expressed using a TAG grammar, this means that the object-level grammar is still a TAG.

6 Conclusion

In principle, a meta-grammar is desirable, as it specifies substructures at a meta level, which is necessary when operations are carried out that are applied at this meta level. In a practical application, it solves problems in one such formalism, S-TAG, when used for paraphrase or translation, as outlined by Shieber (1994). Moreover, the formalism remains fundamentally the same, in specifying mappings between two grammars of restricted generative capacity; and in cases where this is important, it is possible to avoid changing the generative capacity of the S-TAG formalism in applying this meta-grammar.

Currently this revised version of the S-TAG formalism is used as the low-level representation in the Reluctant Paraphrasing framework of Dras (1998; forthcoming). It is likely to also be useful in representations for machine translation between languages that are structurally more dissimilar than English and French, and hence more in need of structural definition of object-level constructs; exploring this is future work.

References

Abeillé, Anne, Yves Schabes and Aravind Joshi. 1990. Using Lexicalized TAGs for Machine Translation. *Proceedings of the 13th International Conference on Computational Linguistics*, 1–6.

Dras, Mark. 1997. Representing Paraphrases Using S-TAGs. *Proceedings of the 35th Meeting of the Association for Computational Linguistics*, 516–518.

Dras, Mark. 1998. Search in Constraint-Based Paraphrasing. *Natural Language Processing and Industrial Applications (NLP+IA98)*, 213–219.

Dras, Mark. forthcoming. *Tree Adjoining Grammar and the Reluctant Paraphrasing of Text.* PhD thesis, Macquarie University, Australia.

Joshi, Aravind and Yves Schabes. 1996. Tree-Adjoining Grammars. In Grzegorz Rozenberg and Arto Salomaa (eds.), *Handbook of Formal Languages, Vol 3*, 69–123. Springer-Verlag. New York, NY.

Kahane, Sylvain, Alexis Nasr and Owen Rambow. 1998. Pseudo-Projectivity: A Polynomially Parsable Non-Projective Dependency Grammar. *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, 646–652.

Palmer, Martha, Owen Rambow and Alexis Nasr. 1998. Rapid Prototyping of Domain-Specific Machine Translation Systems. *AMTA-98*, Langhorne, PA.

Rogers, James. 1994. Capturing CFLs with Tree Adjoining Grammars. *Proceedings of the 32nd Meeting of the Association for Computational Linguistics*, 155–162.

Schabes, Yves and Stuart Shieber. 1994. An Alternative Conception of Tree-Adjoining Derivation. *Computational Linguistics*, 20(1): 91–124.

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

Shieber, Stuart and Yves Schabes. 1990. Synchronous Tree-Adjoining Grammars. *Proceedings of the 13th International Conference on Computational Linguistics*, 253–258.

Shieber, Stuart. 1994. Restricting the Weak-Generative Capacity of Synchronous Tree-Adjoining Grammars. *Computational Intelligence*, 10(4), 371–386.

Weir, David. 1988. *Characterizing Mildly Context-Sensitive Grammar Formalisms.* PhD thesis, University of Pennsylvania.

XTAG. 1995. *A Lexicalized Tree Adjoining Grammar for English.* Technical Report IRCS95-03, University of Pennsylvania.