Synchronous Parallelism Between Different Grammar Formalisms

Mark Dras

Department of Computing
Macquarie University
NSW. 2109. Australia
markd@ics.mq.edu.au

Coordination has posed a problem in TAG-related formalisms, with a recent proposal to solve it suggesting that constituency and dependency be treated separately within a synchronously parallel formalism. This leads to 'tangled' trees—graphs—and it is not clear how these structures satisfy the definition of synchronised TAGs. Naive alternatives which do not result in tangled trees, however, do not satisfy the weak language preservation property (WLPP). This paper shows how the idea of synchronised parallelisation of grammars can be extended so that grammars of different types are synchronised, how this extension has the WLPP, and how this relates to coordination.

1 Introduction

The idea of parallelised grammars is a widely used one, and one which has been applied in many different ways, both in terms of grammars parallelised—context free grammars (CFGs), mildly context sensitive grammars such as Tree Adjoining Grammars (TAGs) and so on—and in terms of type of parallelisation, synchronous or independent (Rambow and Satta, 1994). What these parallelised formalisms do have in common is that they all tend to relate grammars of the same type: two CFGs, two standard TAGs, etc; and this can be unnecessarily restrictive, and in particular cases lead to difficulties in the formalism.

The particular case this paper looks at is Synchronous Tree Adjoining Grammar (S-TAG) (Shieber and Schabes, 1990; Shieber, 1994) and variants. In this formalism, there are two TAGs parallelised in a synchronous fashion, with the weak language preservation property (Rambow and Satta, 1996) holding, that is, in such a way that the generative capacity of each of the two grammars is not altered by the synchronisation. S-TAG has been used to represent machine translation (Abeillé *et al*, 1990), syntax-semantics mapping (Shieber and Schabes, 1990), and in a modified form as Link Sharing TAG (LSTAG) (Sarkar, 1997) for dealing with coordination in a TAG-related formalism. This paper explores how, by extending the range of formalisms that can take part in a synchronous parallelisation, it is possible to resolve some of the difficulties that arise in the representation of coordination using synchronous parallelism, notably the 'tangling' of derived and derivation trees.

This paper is set out as follows. It reviews the definitions of TAG, the parallelised variants Multi-Component TAG (MCTAG), S-TAG and LSTAG, and shows how 'tangling' occurs in the last of these. It then shows how similar tangling could occur in representing paraphrase—the transduction between syntax trees representing text-to-text mapping—with coordination in S-TAG, demonstrating that it is a property of the synchronisation, rather than a characteristic of LSTAG, that causes the problem; and that other obvious alternative solutions do not have the weak language preservation property. It then goes on to demonstrate how synchronisation of differing formalisms, with differing generative capacities, can rectify the problem, and how such a redefinition of S-TAG can still retain the essential properties of the formalism such as the weak language preservation property.

2 Review of TAG Formalisms

2.1 TAG

Under the definition of Tree Adjoining Grammar (TAG), a grammar contains elementary trees, rather than flat rules as in a Context Free Grammar (CFG); these trees combine together via composition operations (the operations of substitution and adjunction) 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. An example of a TAG is give in Figure 1;[1] this contains elementary trees which can be composed together via substitution and adjunction. 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.

2.2 MCTAG

An instance of independent parallelism in the domain of TAG-related grammars, first introduced in Joshi *et al* (1975) and investigated more fully in Weir (1988), is Multi-Component TAG (MCTAG). Here, the domain of locality is extended even further than for standard TAGs, by grouping trees together in a SEQUENCE, with this sequence representing some relationship between trees which can be seen as a kind of locality.

Weir (1988: 32) notes that, given these sequences, there are a number of ways in which

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

[2]The derivation tree is annotated with the address at which the composition occurred, using a Gorn addressing scheme.

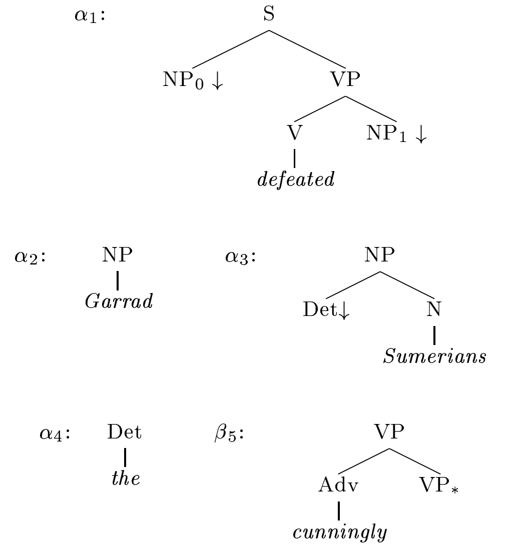

Figure 1: Elementary TAG trees

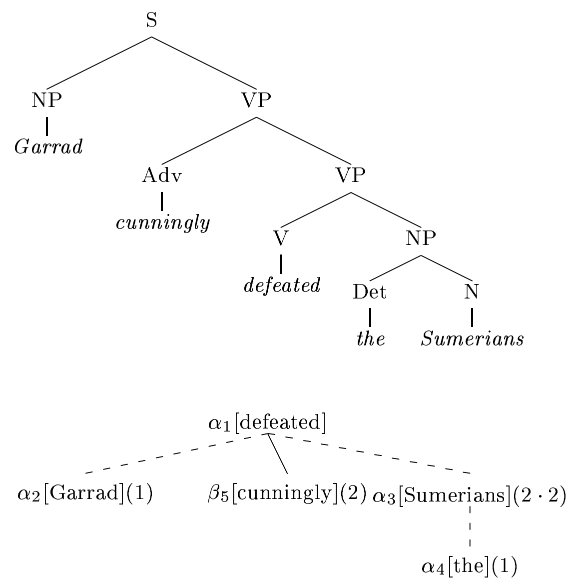

Figure 2: Derived and derivation trees for Figure 1

the sequences can be composed. These lead to the idea of different kinds of locality, tree-local and set-local, so termed in Frank (1992). In tree-local MCTAG, all of the elements of a sequence are composed with a single tree. In set-local, all of the elements of a sequence are composed with the elements of another sequence. In both cases, there is a sequence with only one element, which the other must ultimately be composed into, so that the result is a single derived tree. Of interest here is set-local MCTAG, which because of the independent parallelism is more powerful than standard TAG. For example, the MCTAG in Figure 3 generates the language COUNT-8, $\{a_1^n a_2^n \dots a_8^n \mid n \geq 0\}$, which, by the Pumping Lemma for TALs (Vijay-Shanker, 1987: 96), is beyond the generative capacity of standard TAGs. In addition, unlike for standard TAGs, the path sets of the derived trees may not be context free, and the paths are not independent.

The derivation trees for MCTAGs are similar to those for TAGs: they are similarly context free, this characteristic making them one of the class of Linear Context Free Rewriting Systems (LCFRSs) defined in Weir (1988). The only difference is that the derivation tree nodes are additionally annotated to identify the element of the parent sequence with which composition occurred.

2.3 S-TAG

S-TAG is a formalism which synchronously parallelises two TAGs and links 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. An early use of this was in machine translation between English and French (Abeillé *et al*, 1990). Because of the way TAG captures dependencies, it is not problematic to have translations more complex than word-for-word mappings. 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 4.

(1) a. John misses Mary.

b. Marie manque à Jean.

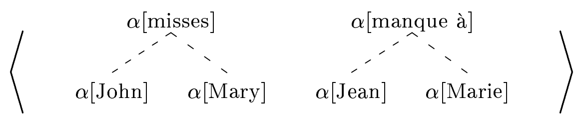

Figure 5: Derivation tree pair for Figure 4

In these tree pairs, a diacritic ($\boxed{n}$) 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* 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, did not have the weak language preservation property (WLPP); instead, an S-TAG 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, licensed by the diacritics in the object-level trees, which establishes the translation. For example, the derivation trees for (1) using the elementary trees of Figure 4 is given in Figure 5; there is a clear isomorphism preserving dominance, with a bijection between nodes.

References to S-TAG in this paper are to this second definition as the standard, unless specified otherwise as 'the rewriting definition of S-TAG'.

2.4 LSTAG

The idea behind synchronous TAG, that of synchronised parallelism of grammars, has also been used in the modelling of coordination in TAG. Coordination poses difficulties for TAG, with a number of analyses proposed (Joshi, 1990; Joshi and Schabes, 1991; Sarkar and Joshi, 1996; Sarkar, 1997). The last of these

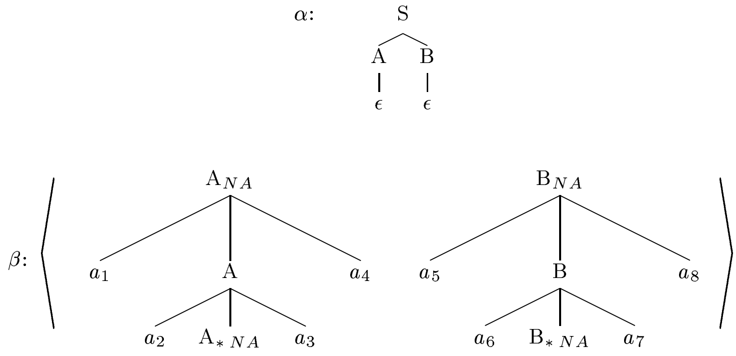

Figure 3: An MCTAG generating COUNT-8

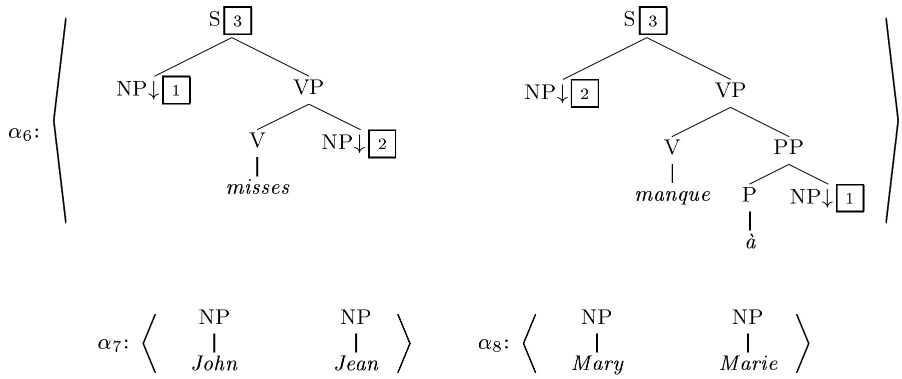

Figure 4: S-TAG with argument swap

(Sarkar, 1997) notes that previous attempts are somewhat inelegant, and do not follow the spirit of the formalism, either ending up with unrooted trees or requiring structure merging on derived trees. According to Sarkar, the difficulties occur because TAG conflates the ideas of constituency and dependency: the derivations do not, for coordination under these earlier approaches, reflect dependency of argument structure appropriately. Sarkar thus proposes splitting the representation of constituency and dependency, but keeping them linked via synchronisation. He defines a variant of S-TAG, Link Sharing TAG (LSTAG), which divides links into two disjoint sets Δ and Φ, where links from Φ are inherited, as under the rewriting definition of S-TAG, but only for one composition, while those from Δ are not inherited. For example, in describing (2), the tree pair in Figure 6 could be used. In α_2, the links are both in Δ (that is, they are not inherited), while those in β are both in Φ (that is, they are inherited for the single adjunction of β into α_2).

(2) John cooks and eats beans.

Adjoining β into α gives derived and derivation trees as in Figures 7 and 8, which is not problematic. However, if an NP-rooted tree, such as α_1 from Figure 6, is substituted at the subject NP slot, the result is as in Figures 9 and 10. As Sarkar discusses, this 'tangled' tree is a directed acyclic graph; this poses particular difficulties given that the definition of S-TAG depends on an isomorphism between well-formed derivation trees. It is also unclear what the formal properties of such a representation are.

3 S-TAG and Paraphrase

Difficulties such as those of coordination as described above also occur in paraphrase: as well as mapping between similar structures in different languages, as for machine translation, synchronous parallelism as in S-TAG can be used

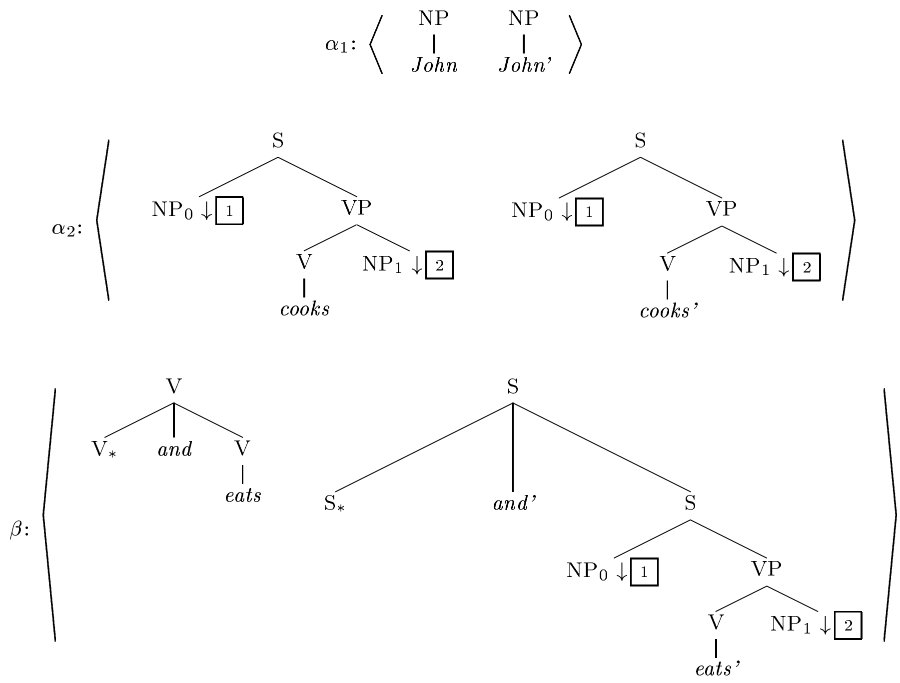

Figure 6: Coordination using LSTAG: elementary tree pairs

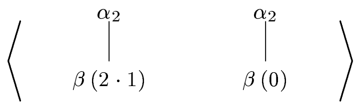

Figure 8: Coordination using LSTAG: derivation tree pair

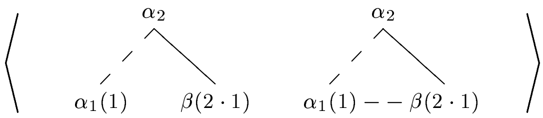

Figure 10: Coordination using LSTAG: derivation tree pair

to map between different syntactic structures in the same language, as for paraphrasing (Dras, forthcoming). As an example, take

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

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

In this paraphrase, the relative clause in (3a) is effectively promoted to be a separate sentence in (3b); in (3b) the clauses share the same subject.[3] An S-TAG pair representing the fundamental aspects of the paraphrase mapping is as in Figure 11. Each projection is a complex derived tree, comprising several elementary trees, unlike in machine translation where the projections tend to be elementary trees or fairly simple derived trees; this is because each projection represents the syntactic combination that is transformed as part of a paraphrase pair such as (3) (Dras, forthcoming). We will term such a pair a STRUCTURAL MAPPING PAIR (SMP).[4]

Links between trees are now no longer one-to-one but many-to-one; note the two 1 links at $\mathbf{NP}_0$ and $\mathbf{NP}_1$ in the right tree: both have the same NP substituted here as in the left tree.

[3]The paraphrase alternative (3b) is a clumsy-sounding one; it is structured this way to illustrate a point about many-to-one links. More natural-sounding paraphrases in the same vein are possible.

[4]The projections' derived trees are composed of elementary trees from the standard XTAG grammar (XTAG, 1995): for the left projection, α**nx0Ax1**[tweed], β**Vvx**[was], β**N0nx0Vnx1**[collected], β**COMPs**[which]; for the right projection, α**nx0Ax1**[tweed], β**Vvx**[was], α**nx0Vnx1**[collected], α**sCONJs**[and]. Section 4 and the derivation tree pair of Figure 16 explain how these standard trees were composed to give the SMP.

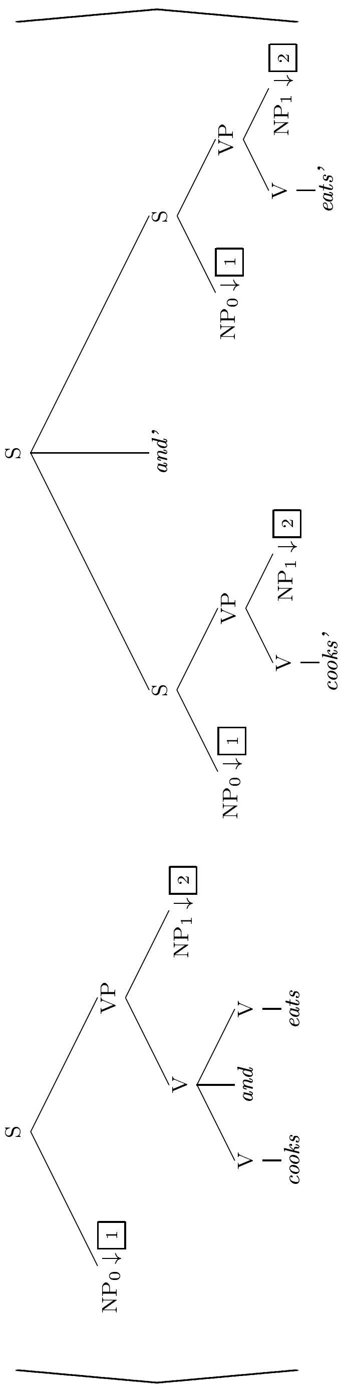

Figure 7: Coordination using LSTAG: derived tree pair

There are two obvious alternatives regarding what to do in the case of such a substitution. The first, following Sarkar (1997), is to substitute the *same child* (a tree for *jacket*) into each parent marked 1, so that both parents have an edge connecting them to the one single child. This causes the same sort of 'tangles' as in LSTAG: it is a feature of many-to-one links in synchronous parallelism, rather than one of LSTAG specifically, that these tangles occur.

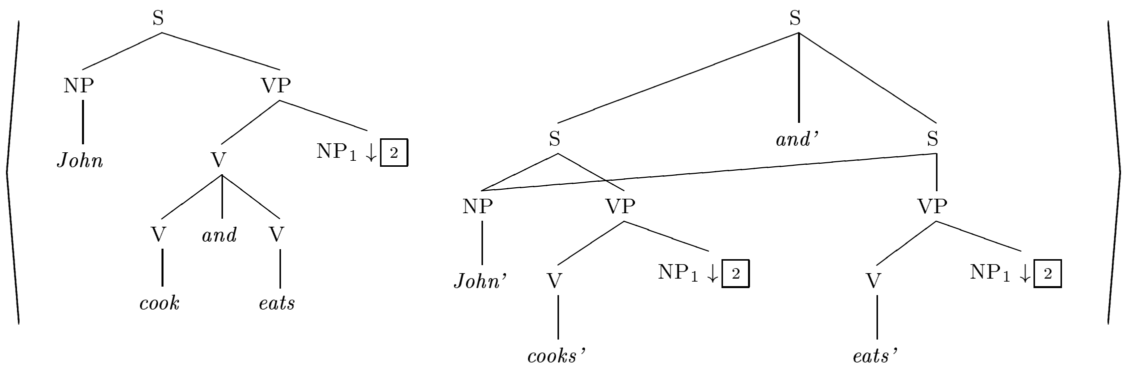

Figure 9: Coordination using LSTAG: derived tree pair

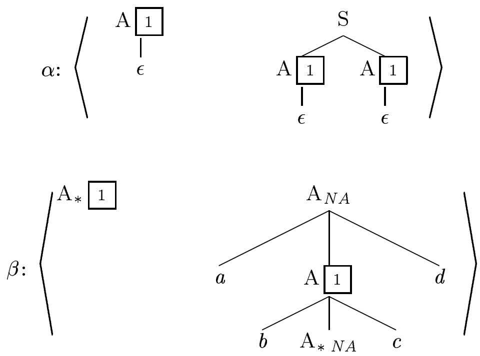

Figure 12: An S-TAG with many-to-one links generating $\{a^n b^n c^n d^n a^n b^n c^n d^n \mid n \geq 0\}$

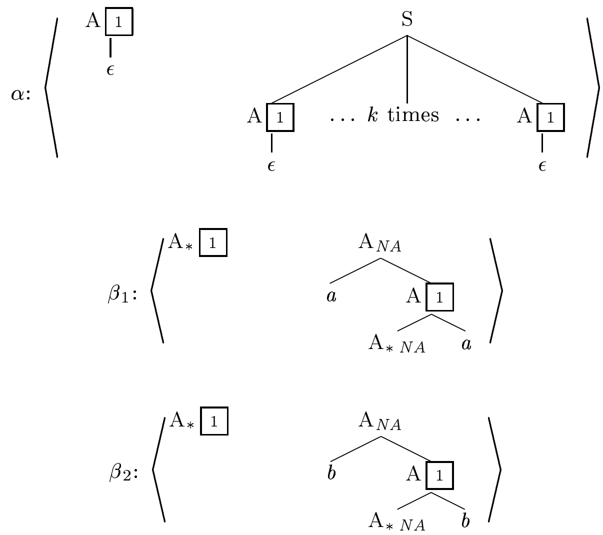

Figure 13: An S-TAG with many-to-one links generating $\{w^k \mid w \in \{a, b\}^*, k \geq 0\}$

An obvious alternative to this tangling is to remove the requirement that both nodes marked with the same diacritic share the same child, which is essentially what causes the tangling. Instead, just insert the child tree separately into each occurrence of the nodes sharing diacritics. Doing so with these many-to-one links, however, changes the properties of the S-TAG formalism, so that the WLPP no longer holds. This can be shown by the following example. Take the TAG tree pairs of Figure 12. It will be clear that, if separate occurrences of β are adjoined into α, the right projection of this S-TAG generates the string language $\{a^n b^n c^n d^n a^n b^n c^n d^n \mid n \geq 0\}$, which is not a TAL; this can be confirmed by application of the Pumping Lemma for TALs (Vijay-Shanker, 1987: 96), which states that TALs must be of the form $u_1 v_1^i w_1 v_2^i u_2 v_3^i w_2 v_4^i u_3$, with $u_i, v_i, w_i \in V^*$, with V the alphabet of terminal and non-terminal symbols.

In fact, for some arbitrary value k, we can get the language $\{(a^n b^n c^n d^n)^k \mid k, n \geq 0\}$ by having k nodes in one tree linked together by the same diacritic. Similarly, we can obtain the k-copy language $\{w^k | w \in \{a, b\}^*, k \geq 0\}$ by the tree pairs of Figure 13.

4 Handling Many-to-One Links

The most natural way of adapting the S-TAG formalism to handle this phenomenon in a predictable way, retaining the well-formedness of the derivation trees, is to note that what is actually happening when these trees are being synchronously composed is that *multiple copies* of the trees are being composed into the parent.

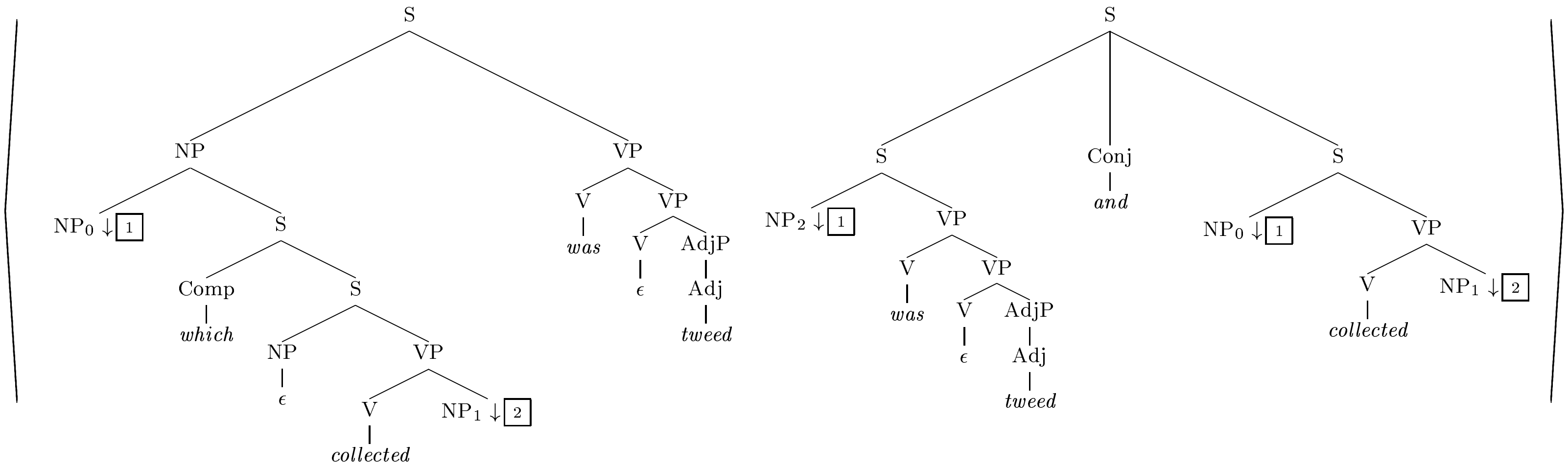

Figure 11: SMP with many-to-one links

That is, there is a set of identical trees being composed ultimately into the single tree at the root of the derivation. The neatest representation for this is thus an MCTAG, with identical elements in the sequence; because the elements of a sequence are composed into elements of another sequence, the MCTAG is set-local. Figure 12 can thus be re-represented as in Figure 14.[5]

As mentioned in Section 2.2, set-local MCTAG has a generative capacity beyond that of standard TAG. With a sequence of k elements, MCTAGs are able to generate the language COUNT-$4k$, that is $L_{c4k} = \{a_1^n a_2^n \dots a_{4k}^n \mid k, n \geq 0\}$. If these elements are all the same, then the language generated is $L'_{c4k} = \{(a_1^n a_2^n a_3^n a_4^n)^k \mid k, n \geq 0\}$; similarly, they can generate $\{w^k \mid k \geq 0, w \in \{a, b\}^*\}$. We do not need the full generative capacity of MCTAG; at this stage, only sequences with identical elements are used. A characterisation of this will be presented further on; first, however, it is necessary to ask whether the mapping between standard TAG and MCTAG is a valid one. For example, is it reasonable to map between grammars with differing generative capacities? And if so, does the WLPP hold?

General answers are given here, with more formal proofs in Dras (forthcoming).

In order to answer the questions, it is first necessary to look at exactly why S-TAG has the WLPP. Although it was argued that in Shieber (1994) that S-TAG does have the WLPP by framing S-TAG in terms of MCTAG, the paper did not give a formal proof that this was so.

It is possible to show that it is so by considering one projection of some S-TAG T. The derivation trees corresponding to this projection can be represented by a CFG G (Weir, 1988). Applying the TAG yield function to these derivation trees gives a TAG, which generates a Tree Adjoining Language (TAL). Now, an isomorphism from the derivation trees of T—the key condition for S-TAG is that there is an isomorphism preserving dominance between derivation trees—corresponds to an operation of substitution on the grammar G: each symbol in the grammar is translated to one of a limited set of alternatives,[6] and by this each node in the trees which represent strings generated by G will correspond to a node in the trees defined by G', which is G under substitution. Now, CFGs are closed under substitution (Salomaa, 1973: 23), thus G' is a CFG also. And applying the TAG yield function to the trees generated by G' again gives a TAL.

The key here is that both grammars' derivation structures are context free, and CFGs are closed under substitution (corresponding to isomorphism between meta-level trees). The TAG yield function (reading off nodes to get object-level trees which are composed appropriately) is in some sense independent of this—all it requires is a context free structure, and it will produce trees whose string language is a TAL.

But this key characteristic is one that is common to all LCFRSs; all have context free derivation trees, and it is the object-level structures and the yield function which are different for each. Thus a proof along the lines of the argument above will hold also for LCFRSs in general, and the formalism that synchronises between any two such will have the WLPP.

What we will be interested in here is the synchronisation of a standard TAG and an MCTAG. The MCTAG is used where there are many-to-one links, and following there will be a discussion of what implications this extra generative capacity has. But first, an example follows, looking at a paraphrase with coordination in S-TAG, where the component grammars are a TAG (left projection) and an MCTAG (right projection); representing paraphrase in this way raises some issues which will be discussed after the example has been presented. Given (3) from Section 3 and the SMP from Figure 11, we will use the tree pairs of Figure 15 to give the pairs for the arguments fitting into the substitution slots.

In Figure 15, the pairs that correspond to a one-to-one link in Figure 11—α_1 and α_2—have simple TAG trees as the right projection, while

[5]Note that angle brackets have been used by convention for both S-TAG and MCTAG. Here the inner set represent an MCTAG and the outer set an S-TAG.

[6]This is a more general case of a homomorphism on G: under a homomorphism each symbol in G would be mapped to a particular given symbol. Substitution, rather than homomorphism, is necessary as a given symbol in G does not always map to the same symbol in G'.

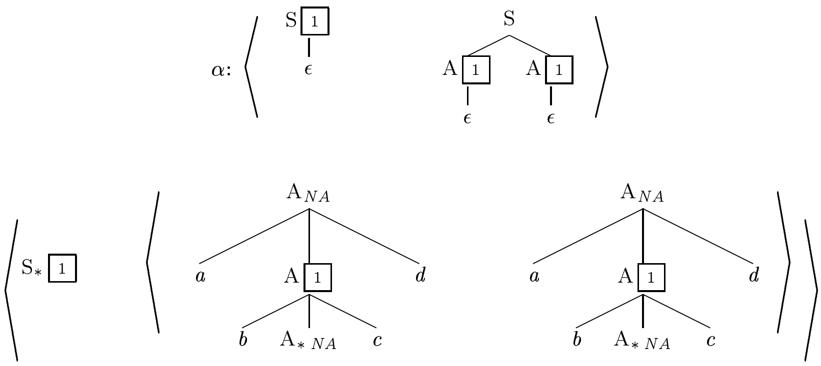

Figure 14: An S-TAG with many-to-one links represented via MCTAG

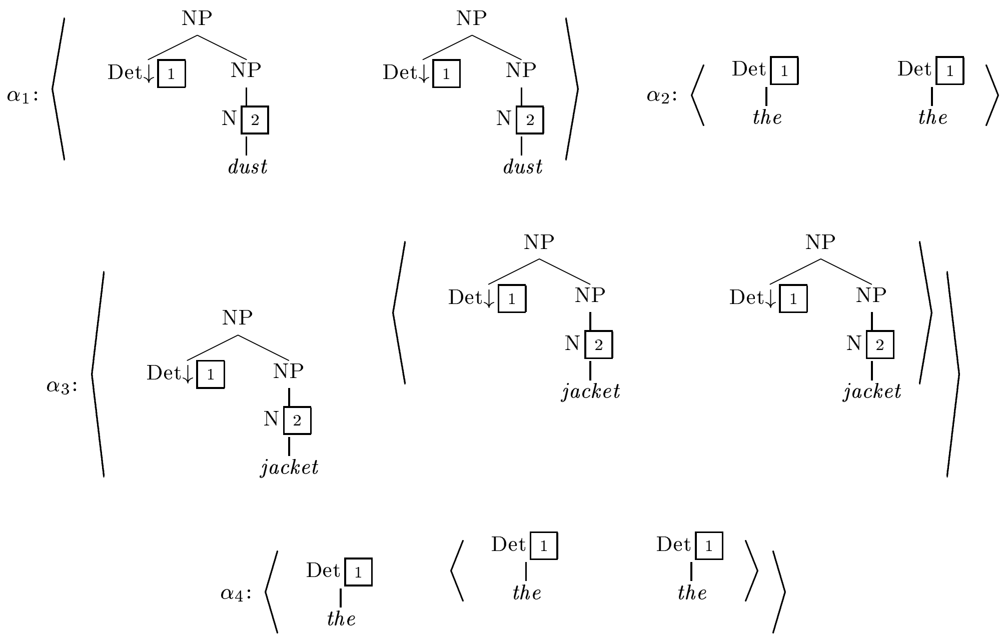

Figure 15: Tree pairs for (3)

those corresponding to a many-to-one link—α_3 and α_4—have a sequence of two identical trees as the right projection. Note that the elements of the sequence in α_4 substitute into different elements of the sequence in α_3, making this a set-local use of MCTAG for the right projection.

A possible derivation tree pair is as in Figure 16; the parts corresponding to the SMP are in bold. For the right tree, the sequences in the MCTAG are given in Table 1.

In the derivation tree for the MCTAG in Figure 16, the addresses annotating each node are of the form $(t_1, a_1; t_2, a_2; \ldots; t_k, a_k)$, where, for

S_1	=	{α**nx0Ax1**[tweed], α**nx0Vnx1**[collected]}
S_2	=	{α**NXdxN**[jacket], α**NXdxN**[jacket]}
S_3	=	{α**DXD**[the], α**DXD**[the]}
S_4	=	{β**Vvx**[was]}
S_5	=	{α**sCONJs**[and]}
S_6	=	{α**NXdxN**[dust]}
S_7	=	{α**DXD**[the]}

Table 1: MCTAG grammar sequences for Figure 16

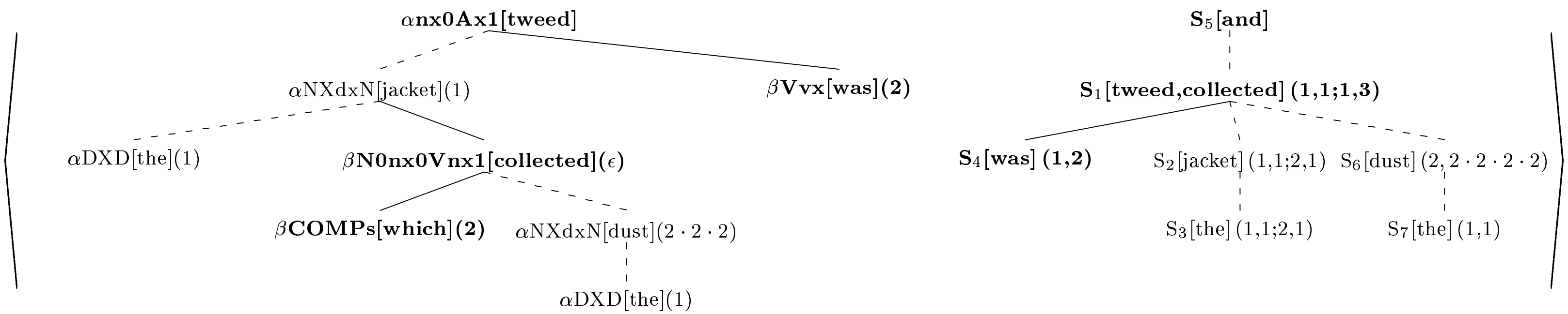

Figure 16: Derivation tree pair for (3)

a sequence of k trees, t_i is the tree in the parent sequence into which tree i is composed, and a_i is the address at which this occurs. The 'copy' sequences S_2 and S_3 are substituted in parallel into the sequence S_1 representing the two component sentences, and this sequence in turn substituted into the discourse structure tree S_5, which combines the two sentences. An isomorphism that is natural in the paraphrase context is to treat the bolded regions within the derivation trees, which represent the SMP, as single nodes that correspond to each other; and for sequences of (one of more) identical items to correspond to the single occurrence of that item in the left tree, as S_2 and S_3 to the singular α**NXdxN**[dust] and α**DXD**[the] (in the case with one item) in the MCTAG sequence.

5 Discussion

In the example, there are two points in particular which deserve comment. The first is that it may not appear that the derivation tree pair of Figure 16 satisfies the condition for S-TAG, specifically preservation of dominance inthe isomorphism, since in the left tree *jacket* is the parent of *collected*, but the reverse is true in the right tree. By treating the discontinuous bolded regions as a single unit, where the children of any of these elements is considered a child of the unit, this is not problematic. A neater theoretical explanation of this, using the derivation level for grouping of constituents by a TAG grammar, is given in Dras (1999) and Dras (forthcoming).

The second point to note is that for this derivation tree pair, it is necessary to have a sequence like S_1, so that the copied items of S_2 can be composed into a sequence; if this were not the case, and the copied items were then necessarily copied into a derived tree, then the MCTAG would be non-local, which would be undesirable from the point of view of constraining generative capacity.[7]

The question could now be asked, How appropriate is it to use extra generative capacity just to handle, for example, coordination?

These sorts of many-to-one links have been used in this section to model the coreference of two NPs, caused by splitting one sentence into two (or combining two into one). That is, the ability to perform a k-copy, or to count to $4k$, occurs when there are multiple independent clauses, which are effectively separate sentences. In an informal way, it is possible to conceive of this 'extra power' as being 'distributed' over the multiple sentences so that each sentence has the 'power' of a standard TAG allocated to it. That is, if there are k sentences, and k-to-one links allowing languages like COUNT$-4k$ and k-COPY, this can be seen as it being possible to allocate the ability to count to 4 to each of the k sentences—the power of a standard TAG—or having one copy of a string $\{a, b\}^*$ per sentence. Given this, it is appropriate to put a restriction on the many-to-one links. Conceptually, only one link indicating a given coreference is allowed per sentence (or independent clause); each sentence can thus be viewed as the expression of a standard TAG.

Now, even given that the 'copy' technique of this paper avoids tangles, it might be argued that it is still better to use the LSTAG approach for linguistic reasons related to coordination. For example, *who hates football and love cricket* does not means *who hates football and who loves cricket*. LSTAG provides an obvious way of showing that the subject of the coordinated predicates is the same entity, through links to the same node. However, the 'copy' approach does also keep a record, by virtue of the fact that the shared subjects are contained in the same MCTAG set of identical items (for example, S_2 in Table 1).

6 Conclusion

Modelling coordination—or more generally, problems where many-to-one links occur in the synchronous parallelisation of grammars—leads

[7]This rules out some structures allowed when mapping between standard TAGs. For example, if we wanted to use the β**sCONJs**[and] tree, the second element of the S_1 sequence would need to be substituted into this, which would in turn need to be adjoined into the first element of the sequence S_1, and this is not possible. This would mean that the elements of S_1 could not be in a sequence; and in order for the MCTAG to avoid being non-local, the entire unit corresponding to the SMP would need to be treated as elementary. This would lead to problems with attachment—raising the question, Which part of this unit should the sequence S_2 attach to?—and with addressing.

to difficulties in representation: either the trees in the representation become 'tangled', or the generative capacity of the synchronous formalism is increased and the weak language preservation property no longer holds.

This paper has shown that synchronous parallelism does not have to be restricted to being between the same types of grammars. Here, if synchronisation is extended to take place between a standard TAG and an MCTAG, coordination does not involve tangled tres, and the generative capacity of the synchronous formalism is not greater than that of its component grammars.

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. 1999. A Meta-Level Grammar: Redefining Synchronous TAG for Translation and Paraphrase. In *Proceedings of the Thirty-Seventh Annual Meeting of the Association for Computational Linguistics.*

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

Frank, Robert. 1992. *Syntactic Locality and Tree Adjoining Grammar: Grammatical, Acquisition and Processing Perspectives.* PhD thesis, University of Pennsylvania.

Joshi, Aravind, Leon Levy and Masako Takahashi. 1975. Tree Adjunct Grammars. *Journal of Computer and System Sciences*, 10, 136–163.

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

Joshi, Aravind and Yves Schabes. 1991. Fixed and flexible phrase structure: Coordination in Tree Adjoining Grammar. *Proceedings of the DARPA Workshop on Spoken Language Systems.*

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.

Rambow, Owen and Giorgio Satta. 1994. *A Two-Dimensional Hierarchy for Parallel Rewriting Systems.* University of Pennsylvania Technical Report IRCS94-02.

Rambow, Owen and Giorgio Satta. 1996. Synchronous Models of Language. *Proceedings of the 34th Meeting of the Association for Computational Linguistics*, 116–123.

Salomaa, Arto. 1973. *Formal Languages.* Academic Press. Orlando, FL.

Sarkar, Anoop and Aravind Joshi. 1996. Coordination in TAG: Formalization and Implementation. *Proceedings of the 16th International Conference on Computational Linguistics*, 610–615.

Sarkar, Anoop. 1997. Separating Dependency from Constituency in a Tree Rewriting System. *Proceedings of the 5th Meeting on Mathematics of Language.*

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.

Vijay-Shanker, K. 1987. *A Study of Tree Adjoining Grammars.* PhD thesis, University of Pennsylvania.

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.