Systematic Grammar Development in the XTAG Project

Carlos A. Prolo
University of Pennsylvania

1. Introduction

The XTAG Project (Joshi, 2001) is an ongoing project at the University of Pennsylvania since about 1988, aiming at the development of natural language resources based on Tree Adjoining Grammars (TAGs) (Joshi and Schabes, 1997). Perhaps the most successful experience in it has been the construction of a wide-coverage Lexicalized TAG for English (LTAG) (Doran *et al.*, 2000; XTAG Research Group, 2001), based on ideas initially developed in (Krock and Joshi, 1985).

As the grammar grew larger, the process of consistent grammar development and maintenance became harder (Vijay-Shanker and Schabes, 1992). Driven by locality principles, each elementary tree for a given lexical head is expected to contain its projection, and slots for its arguments (e.g., (Frank, 2001)). As consequence, the number of required elementary trees grows huge for a grammar with reasonable coverage of syntactic phenomena. Under the XTAG project, for engineering reasons, the grammar has been split up in (roughly) two main components[1]: a set tree templates lexicalized by a syntactic category, and a lexicon with each word selecting its appropriate tree templates. Although various syntactic categories have multiple syntactic frames available (e.g., prepositions may have different kinds of arguments, nouns and adjectives may have arguments or not, etc.), it is the verbs that exhibit the most wild variety of domains of locality: from the 1226 template trees in the XTAG grammar, 1008 are for verbs, more than 82%. That happens because the grammar tries to capture in elementary trees the locality for each of the diverse syntactic structures related transformationally to each other (the effect of long distance movement is captured by adjunction of the intervening material). Examples of required tree templates are: declarative transitive; ditransitive passive with wh-subject moved; and intransitive with PP object with the PP-object relativized.

As early noticed by (Vijay-Shanker and Schabes, 1992) the information regarding syntactic structure and feature equations in (feature-based) LTAGs is repeated across templates trees in a quite regular way, that perhaps could be more concisely captured than by just having a plain set of elementary trees. Besides the obvious linguistic relevance, as a pure engineering issue, the success of such enterprise would result in enormous benefits for grammar development and maintenance.

Several approaches have been proposed in the literature describing compact representations methods for LTAGs, of which, perhaps the best known are (Vijay-Shanker and Schabes, 1992), (Candito, 1996; Candito, 1998), (Evans, Gazdar and Weir, 1995; Evans, Gazdar and Weir, 2000), (Xia *et al.*, 1998; Xia, 2001), and (Becker, 1993; Becker, 1994; Becker, 2000). We describe in this paper how we combined Becker's metarules with a hierarchy of rule application to generate the verb tree templates of the XTAG English grammar, from a small initial set of trees.

2. Metarules

A subsystem for interpreting metarules was initially introduced into the XTAG development system by Tilman Becker, from 1993 to 1995, based on his ideas in (Becker, 1993; Becker, 1994) with subsequent additions over the years, reaching a stable form as documented (by this author) in (XTAG Research Group, 1998). Although it has been topically used since then, as an auxiliary tool to reduce the effort spent in grammar development (e.g., to generate the trees for an updated analysis of relative clauses, using the former trees as starting point), this paper describes the first effort to effectively use them to generate the full XTAG grammar verb trees.[2]

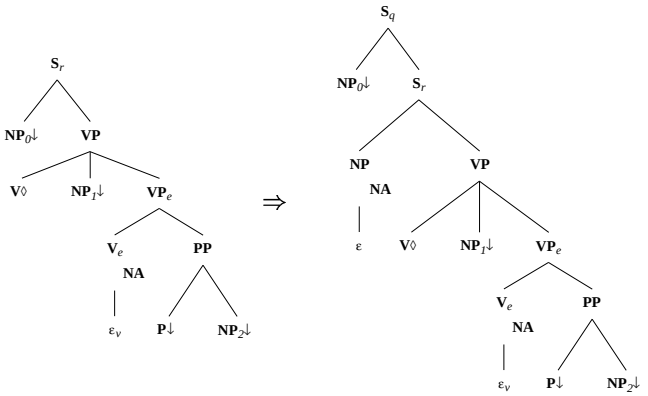
* We are indebted to all members of the XTAG Group that participated of the valuable discussions during the realization of this work, and in particular to Alexandra Kinyon for her comments on this paper.

1. For a more accurate description of the architecture, see (XTAG Research Group, 2001) or (Doran *et al.*, 2000).

2. This effort is already mentioned in (Doran *et al.*, 2000, p. 388). There has been some confusion on the issue, perhaps driven by a somewhat ambiguous statement in (Becker, 2000, p. 331): "In this paper, we present the various patterns which are used in the implementation of metarules which we added to the XTAG system (Doran et al. 2000)". The work of Becker conceived and developed the idea of metarules for TAGs and also created the original implementation of the metarule interpreter as part of the XTAG software. However, he only created the necessary example patterns to support the concepts of metarules while the work described here is the first to actually evaluate metarules in-the-large as part of the XTAG project.

 Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+6), pp. 181–186. Universitá di Venezia.

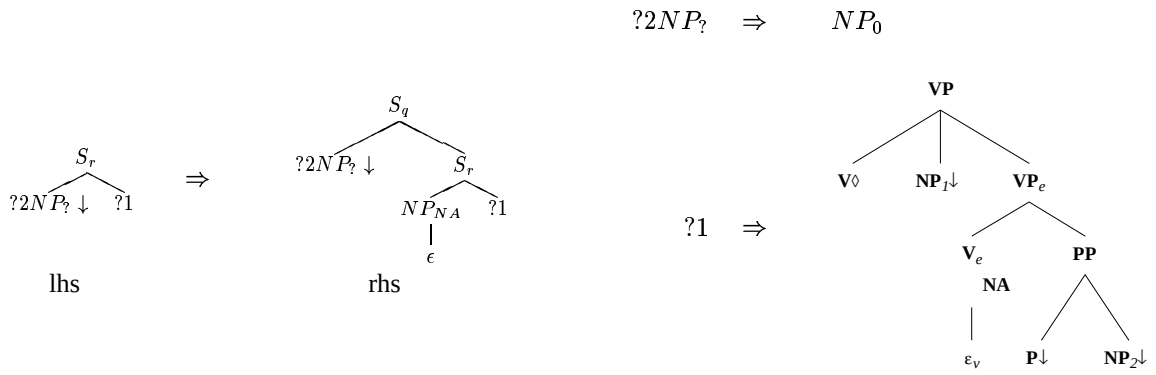
We present in this section a brief introduction to Becker's metarules.[3] Consider the trees in Figure 1 anchored by verbs that take as arguments an NP and a PP (e.g., *put*). The one to the left corresponds to its declarative structure; the other to the wh-subject extracted form. Despite their complexity, they share most of their structure: the only differences being the wh-site (higher NP in the right tree) and the trace at subject position. That observation would not be very useful if the differential description we have made was idiosyncratic to this pair, which is not the case. Clearly, many other pairs all over the grammar will share the same differential description.



(a) declarative (b) subject extracted

Figure 1: Some related trees for the verb put

Figure 2.a shows a metarule for wh-subject extraction, that captures the similarities mentioned above. It describes how to automatically generate the tree in Figure 1.b, given as input the tree in Figure 1.a. Here is how it works. First the input tree has to match the *left-hand side* of the metarule, *lhs* in Figure 2.a, starting from their roots. In the example the lhs tree, requires the candidate tree to have its root labeled S_r. Then, its leftmost child has to be an NP, as indicated by the node $?2NP_?$ in lhs: $?2$ indicates it is the variable $\#2$; $NP_?$ indicates we need an NP, regardless of the subscript. Next, the lhs tree requires the rest of the tree to match variable $?1$. That is trivial, because such variables with just an identification number are "wild cards" that match any range of subtrees. The matches of each variable in lhs, for the application to the input tree in Figure 1.a, are shown in Figure 2.b.



a) Metarule for wh-movement of subject b) Variables Matching for the tree in Figure 1.a

Figure 2: A metarule for wh-movement of subject applied to the tree of Figure 1.a

Had the matching process failed the metarule would not apply and no new tree would have been generated. Since in the example above the matching succeeded, the processor move to the final step, which is to generate the new tree. We look at the *right-hand side* of the metarule *rhs* and just replace the instances of the variables there

3. For a more comprehensive view, including linguistic motivations and the sort of patterns it allows, we refer the reader to (Becker, 2000). The actual set of metarules we used can be obtained upon request to this author.

SUBCATEGORIZATION GROUP	No. OF FAMS.	TOTAL No. OF TREES
Intransitive	1	12
Transitive	1	39
Adjectival complement	1	11
Ditransitive	1	46
Prepositional complement	4	182
Verb particle constructions	3	100
Light verb constructions	2	53
Sentential Complement (full verb)	3	75
Sentential Subject (full verb)	4	14
Idioms (full verb)	8	156
Small Clauses/Predicative	20	187
Equational "be"	1	2
Ergative	1	12
Resultatives	4	101
It Clefts	3	18
Total	57	1008

Table 1: Current XTAG Grammar Coverage

with their matched values, obtaining the tree in Figure 1.b. The same process can be applied for the many other pairs related by the same metarule.

Variables like ?1 above, with no category specification, are indeed more powerful than the above example allow us to see. For instance, they allow us to express dominance relations. Additionally a single metarule application may result multiple matchings and therefore multiple output trees.

In a feature-based grammar as the one we are focusing on, to create tree structures without the proper feature equations is of little use. On the other hand, experience has shown that feature equations are much harder to maintain correct and consistent in the grammar than the tree structures. The XTAG metarules use features in two ways: as matching requirements, and for transformation purposes. Both positive and negative matching can be specified (that is, one can state that match will happen only if the input tree does not have a certain feature equation). Regarding the transformations, feature equations can be inserted, deleted, maintained, or modified, when generating the new tree from the matched one. A few imperative commands have proved very useful, e.g. "replace all NPs with index 1 by NPs with index w in all equations".

3. A hierarchy for the application of the metarules

The set of verbal trees can be seen as a subset of the Cartesian product of three dimensions: subcategorization (e.g., transitive, intransitive), redistribution (e.g., passive), and realization (e.g., wh-subject movement) – discounted, of course, combinations blocked by linguistic constraints (e.g., there can not be object movement in intransitives). The verb trees in the XTAG English grammar are organized in families that roughly reflect a subcategorization frame. Hence, each family contains trees for each combination of redistribution and realization alternatives compatible with the subcategorization frame. The *base* tree of a family is the one corresponding to its declarative usage (no redistribution, arguments in canonical position). Table 1 summarizes the current coverage of the XTAG English grammar. The grouping of the families is just for presentational convenience.

Becker(1993; 1994; 2000) proposes that a grammar is the closure of the set of base trees under metarule application, raising a heated discussion on the unboundedness of the process of recursive application. We understand the issue is artificial and we show in this section that a simple ordering mechanism among the metarules suffices.[4]

Our strategy for generation of the verbal trees is as follows. There is a unique ordered set of 21 metarules (Table 2). For each family, we start with the base, declarative tree, apply the sequence of metarules, and the result is the whole family of trees. The sequence of metarules are applied in a way we call cumulative mode of application represented in Figure 3. The generated set start with the declarative tree. The first metarule is applied to the set, generating new trees, which are themselves included in the generated set. Then the second rule is applied, and so on, until the sequence is finished.

Redistribution rules are applied before realization rules. It is usual for a metarule to fail to apply to many of the already generated trees. Partly, this is due to the obvious fact that not all rules are compatible with any given

4. Notice that in the context of TAGs, metarules are used "off-line" to generate a finite grammar, a bounded process, which is radically different from their use in the Transformational Grammar tradition or in any other "on-the-fly" environment.

Metarule	Description
passive	Generate the passive form
passive-fromPP	Passive form for PP complements: "The results were accounted for by the theory"
dropby	Passive without by-clause
gerund	Trees for NPs like in "John eating cake (is unbelievable)"
wh-subj	Wh-subject movement
wh-sentsubj	Wh-subject movement for sentential subjects
wh-npobj	NP extraction from inside objects
wh-smallnpobj	NP extraction from inside objects for small clauses
wh-apobj	AP complement extraction
wh-advobj	ADVP complement extraction
wh-ppobj	PP complement extraction
rel-adj-W	Adjunct relative clause with wh-NP
rel-adj-noW	Adjunct relative clause with (possibly empty) complementizer
rel-subj-W	Subject relative clause with wh-NP
rel-subj-noW	Subject relative clause with complementizer
rel-subj-noW-forpassive	Subject relative clause with complementizer for passives
rel-obj-W	NP Object relative clause with wh-NP
rel-obj-noW	NP Object relative clause with complementizer
rel-ppobj	PP Object relative clause
imperative	Imperative
PRO	PRO Subject

Table 2: Metarules used to generate the verb families of the XTAG English Grammar

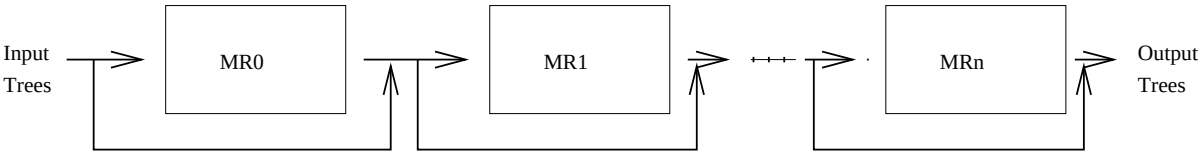


Figure 3: Cumulative application of metarules

subcategorization frame or after a redistribution metarule has been applied to it. But also, because the linear order is a simplification of what in fact should be a partial order, e.g. subject relativization should not apply to a wh-subject extracted tree. The constraints expressed in the metarules are responsible for blocking such applications.

We chose one of the largest families, with 52 trees, for verbs like *put* that take both an NP and a PP as complements, to detail the process of generation. For the sake of simplicity we omit the 26 relative clause trees. The remaining 25 trees [5] are described in Table 3, and the generation graph is shown in Figure 4. Numbers assigned to the trees in the Table are used to refer to them in the Figure.

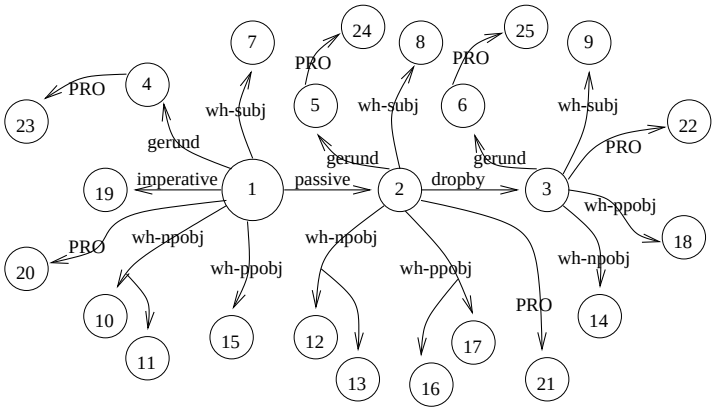


Figure 4: Partial generation of the *put* family using Metarules

5. There is one tree, for nominalization with determiner, we have found not worth generating. We comment on that ahead.

Number	Description	Example
1	Declarative	He put the book on the table
2	Passive w. by	The book was put on the table by him
3	Passive w.o. by	The book was put on the table
4	Gerundive nominals	He putting the book on the table was unexpected
5	Gerundive for passive w. by	The book being put on the table by him ...
6	Gerundive for passive w.o. by	The book being put on the table ...
7	Subject extraction	Who put the book on the table ?
8	Subj. extr. from passive w. by	What was put on the table by him ?
9	Subj. extr. from passive w.o. by	What was put on the table ?
10	1st obj extraction	What did he put on the table ?
11	2nd obj NP extraction	Where did he put the book on ?
12	2nd obj NP extr. from pass. w. by	Where was the book put on by him ?
13	Agent NP extr. from pass. w. by	Who (the hell) was this stupid book put on the table by ?
14	2nd obj NP extr. from pass. w.o. by	Where was the book put on ?
15	PP obj extr	On which table did he put the book ?
16	PP obj extr. from pass. w. by	On which table was the book put by him ?
17	By-clause extr. from pass. w. by	By whom was the book put on the table ?
18	PP obj extr. from pass. w.o. by	On which table was the book put ?
19	Imperative	Put the book on the table !
20	Declarative with PRO subject	I want to [PRO put the book on the table]
21	Passive w. by w. PRO subject	The cat wanted [PRO to be put on the tree by J.]
22	Passive w.o. by w. PRO subject	The cat wanted [PRO to be put on the tree]
23	Gerundive nominals with PRO subject	John approved of [PRO putting the cat on the tree]
24	Gerundive nominals for passive w. by w. PRO subject	The cat approved of [PRO being put on the tree by J.]
25	Gerundive nominals for passive w.o. by w. PRO subject	The cat approved of [PRO being put on the tree]

Table 3: Partial view of the trees from the *put* family

4. Evaluation and final remarks

An important methodological issue is that the grammar was generated towards a pre-existent English Grammar. So we can claim that the evaluation was quite accurate. Differences between the generated and pre-existent trees had to be explained and discussed with the group of grammar developers. Often this led to the discovery of errors and better ways of modeling the grammar. The purpose of this work was to generate the 57 verb families (1008 trees) from only the corresponding 57 declarative trees (or so) plus 21 metarules, a quite compact initial set. More importantly a compact set that can be effectively used for grammar development.[6] We turn now to a short inventory of problems found as well as some interesting observations:

1. We undergenerate:

   (a) There are about 20 idiosyncratic trees not generated, involving trees for "-ed" adjectives, restricted to transitive and ergative families, and Determiner Gerund trees, which lack a clear pattern across the families.[7] These trees should be separately added to the families. Similarly, there are 10 trees involving punctuation in the sentential complement families which are not worth generating automatically.

   (b) We overlooked the it-cleft families with peculiar tree structures, and the equational *be* family with two trees.

   (c) We do not handle yet: the passivization of the second object (from inside a PP) in families for idiomatic expressions ("The warning was taken heed of"); the occurrence of the "*by phrase*" before sentential complements ("I was told by Mary that ..."); and wh-extraction of sentential complements and of exhaustive PPs. Except for the first case all can be easily accounted for.

2. We overgenerate: we generate 1200 trees (instead of 1008).[8] However things are not so bad as they look at first: 206 of them are for passives related to idioms and they are fruit of some pragmatism in the group. It is acknowledged the existence of a certain amount of overgeneration in the tree families due to the separation between the lexicon and the tree templates. For instance, it is widely known that not all transitive verbs can undergo passivization. But the transitive family contains passive trees. The reconciliation can be made through

6. Of course we would not be very happy with a compact representation resembling a "zipped" file.
7. For instance, the nominalization of the transitive verb *find* selects a prepositional complement introduced by the preposition *of*: "The finding of the treasure (by the pirates) was news for weeks." But the "of" insertion is not uniform across families: cf. "the accounting for the book."
8. Which means more than an excess of 192 trees since there is also some undergeneration, already mentioned.

features assigned to verbs that allow blocking the selection of the particular tree. However in the family for verb particle with two objects (e.g., for "John opened up Mary a bank account"), the four lexical entries were judged not to undergo passivization and the corresponding trees (64) were omitted from the family. It is not surprising then that the metarules overgenerate them. Still, 100 out of the 206 are for passives in idiom families which are currently not in the grammar and are definitely lexically dependent. The other 42 overgenerated passives are in the light verb families. There a few other cases of overgeneration due to lexically dependent judgments, not worth detailing. Finally, a curious case involved empty elements that could be generated at slightly different positions that are not distinguished at surface (e.g., before or after a particle). The choice for having only one alternative in the grammar is of practical nature (related to parsing efficiency) as opposed to linguistic.

3. Limitations to further compaction: All the metarules for wh-object extraction do essentially the same, but currently they cannot be unified. Further improvements in the metarule system implementation could solve the problem at least partially, by allowing to treat symbols and indices as separate variables. A more difficult problem is some subtle differences in the feature equations across the grammar (e.g., causing the need of a separate tree for relativization of the subject in passive trees). By far, feature equations constitute the hardest issue to handle with the metarules.

4. A metarule shortcoming: currently they do not allow for the specification of negative structural constraints to matching. There is one feature equation related to punctuation that needed 5 separate metarules (not described above) to handle (by exhaustion) the following constraint: the equation should be added if and only if the tree has some non-empty material after the verb which is not a "by-phrase".

5. Other cases: a separate metarule was needed to convert foot nodes into substitution nodes in sentential complement trees. This families departs from the rest of the grammar in that their base tree is an auxiliary tree to allow extraction from the sentential complement. But the corresponding relative clauses have to have the S complement as a substitution node. This is more an engineering than a conceptual problem.

References

Abeille, Anne and Owen Rambow, editors. 2000. *Tree Adjoining Grammars: formalisms, linguistic analysis and processing*. Stanford, CA, USA: CSLI.

Becker, Tilman. 1993. *HyTAG: A new Type of Tree Adjoining Grammars for Hybrid Syntactic Representation of Free Word Order Lang uages*. Ph.D. thesis, Universität des Saarlandes.

Becker, Tilman. 1994. Patterns in metarules. In *Proceedings of the 3rd TAG+ Conference*, Paris, France.

Becker, Tilman. 2000. Paterns in Metarules for TAG. In Abeille and Rambow (Abeille and Rambow, 2000), pages 331–342.

Candito, Marie-Helene. 1996. A Principle-Based Hierarchical Representation of LTAGs. In *Proceedings of the 16th CoLing (COLING'96)*, pages 194–199, Copenhagen, Denmark.

Candito, Marie-Helene. 1998. Building Parallel LTAG for French and Italian. In *Proceedings of the 36th Annual Meeting of the ACL and 16th CoLing*, pages 211–217, Montreal, Canada.

Doran, Christine, Beth Ann Hockey, Anoop Sarkar, B. Srinivas and Fei Xia. 2000. Evolution of the XTAG System. In Abeille and Rambow (Abeille and Rambow, 2000), pages 371–404.

Evans, Roger, Gerald Gazdar and David Weir. 1995. Encoding Lexicalized Tree Adjoining Grammars with a Nonmonotonic Inheritance Hierarchy. In *Proceedings of the 33rd Annual Meeting of the ACL*, pages 77–84, Cambridge, MA, USA.

Evans, Roger, Gerald Gazdar and David Weir. 2000. 'Lexical Rules' are just lexical rules. In Abeille and Rambow (Abeille and Rambow, 2000), pages 71–100.

Frank, Robert. 2001. *Phrase Structure Composition and Syntactic Dependencies.* to be published.

Joshi, Aravind K. 2001. The XTAG Project at Penn. In *Proceedings of the 7th International Workshop on Parsing Technologies (IWPT-2001)*, Beijing, China. Invited speaker.

Joshi, Aravind K. and Yves Schabes. 1997. Tree-Adjoining Grammars. In *Handbook of Formal Languages, vol. 3*. Springer-Verlag, Berlin, pages 69–123.

Krock, Anthony S. and Aravind K. Joshi. 1985. The linguistic relevance Tree Adjoining Grammar. Technical Report MS-CIS-85-16, University of Pennsylvania.

Vijay-Shanker, K. and Yves Schabes. 1992. Structure Sharing in Lexicalized Tree-Adjoining Grammars. In *Proceedings of the 14th CoLing (COLING'92)*, pages 205–211, Nantes, France.

Xia, Fei. 2001. *Investigating the Relationship between Grammars and Treebanks for Natural Languages*. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania.

Xia, Fei, Martha Palmer, K. Vijay-Shanker and Joseph Rosenzweig. 1998. Consistent Grammar Development Using Partial-Tree Descriptions for Lexicalized Tree-Adjoining Grammars. In *Proceedings of the 4th International Workshop on Tree Adjoining Grammars (TAG+4)*, Philadelphia, USA.

XTAG Research Group, The. 1998. A Lexicalized Tree Adjoining Grammar for English. Technical Report IRCS 98-18, University of Pennsylvania.

XTAG Research Group, The. 2001. A Lexicalized Tree Adjoining Grammar for English. Technical Report IRCS 01-03, University of Pennsylvania.