A Compact Encoding of a DTG Grammar

Martine Smets
Cognitive and Computing Sciences
University of Sussex
Brighton, BN1 9QN, UK
martines@cogs.sussex.ac.uk

Roger Evans
Information Technology Research Institute
University of Brighton
Brighton, BN2 4GJ, UK
Roger.Evans@itri.brighton.ac.uk

Abstract

This paper describes the use of a compact encoding scheme to represent the trees of the wide-coverage DTG grammar currently being developed in the LEXSYS project (Caroll et al 1998). The encoding scheme is derived from the scheme for LTAG grammars described in Evans, Gazdar and Weir (1995), but the LEXSYS grammar is the first attempt to apply these ideas on a larger scale. In this paper we report on the approach taken and discuss some technical improvements to the encoding scheme that we have introduced to overcome problems of scaling.

1 Compact encoding of LTAG/ DTG trees

Evans, Gazdar and Weir (1995) describe a compact representation of LTAG trees using the default inheritance language DATR (Evans and Gazdar 1996). This representation uses two techniques to make tree grammars more compact. First, inheritance between trees allows them to share common structure: for example a transitive verb tree can inherit the structure of an intransitive verb, adding a direct object argument, and a ditransitive can inherit all this structure from the transitive, adding a further indirect argment. Second, the grammar includes rules which derive new trees from old: tree relations such as passive, dative movement and topicalisation are encoded as rules, allowing the full grammar to be encoded as a set of base trees plus a set of such generative rules.

The representation of rules is *internal*, in the sense that they are expressed as part of the tree definitions themselves, rather than externally as a set of rules in a separate representation system. For example, passive is represented as a constraint between an 'input' tree with a direct object and an 'output' tree without one. This relation is part of the definition of any transitive verb and can be 'invoked' by setting the input to be the base tree for the verb and reading off the outpu as the resultin surface tree (see Evans, Gazdar and Weir 1995 for full details).

This approach has a number of advantages. Rule definitions can directly access test or modify any part of the tree under consideration, and they can themselves use the inheritance mechanisms to share structure between rules. In addition rules can be positioned in the main tree hierarchy so that they are only visible to trees they can sensibly be applied to. For example, the passive rule definition can be located as part of the transitive verb tree definition, and so will be inheritied by ditransitives (which also passivise) but not by intransitives, sentential complement verbs etc., which cannot. Thus by internalising rules, one can simply express generalisations about their scope.

However, the specific proposal in Evans, Gazdar and Weir (1995) also had some less desirable features. In order to apply a rule, it was necessary to 'plumb together' inputs and outputs using inheritance statements in a new DATR node. In addition, rule definitions included specific references to their own names, making it difficult for rules to share definitions in practice, and difficult to apply a rule more than once to the same tree. The present approach uses an improved version of this scheme which addresses these issues:

1. rule application is achieved by adding path prefixes (specifying rules to be applied) to queries on the basic tree definition, rather than creating a new node and 'plumbing';
2. rule definitions are no longer dependent on the name of the rule they define, they are properly modular, making it easier to generalise across rules;
3. as a consequence it is now possible to apply a rule more than once to the same tree if required;

These improvements make it more feasible to consider a more realistic set of rules with more complex

interactions, as required for a large scale grammar such as the LEXSYS grammar.

2 Application to LEXSYS

2.1 The rules

There are at present 35 rules in the grammar that we are developing as part of the LEXSYS system. Roughly half of these rules are movement rules, because there is currently a different movement rule for each possible extraction site: e.g. there is a rule for *wh*-questions on the subject, another one for questions on the first object, different rules for extraction of prepositional objects depending of whether or not the preposition is stranded, etc. Other rules are the passive rules (with or without a *by*-phrase), the rules concerning the order of particles and complements, the inversion rule, the rule deriving VP complements from S, etc.; there are also rules for nouns, determiners, adjectives and adverbs.

Rules which share common characteristics are coded as a hierarchy, which allows them to share much of their structure. For example, for movement rules[1] the top of the hierarchy is the *topic* rule, which specifies the top structure of the derived tree (where the 'extracted' element is localised). The rule *topobj1* (topicalization of the first object) inherits the information in *topic* and specifies the position in the tree of the null category coindexed with the 'extracted' element. Finally, the rules *whobj1* and *relobj1* inherit from the rule *topobj1* and specify the type of the topicalized category: *wh*-word or relative word.

This organization in an inheritance hierarchy allows to capture linguistic generalizations: the wh-movement rules[2](topicalization, *wh*-questions, relative clauses) 'move' a constituent to the same position, the front of the clause; the fronted constituent can be a NP (if the 'extracted' element is the subject, a direct object or the object of a preposition), a PP (a prepositional object with no preposition stranding), an AP (adjective phrase) or an AdvP (adverb phrase). This constituent is associated with a gap corresponding to one of the arguments of the verb, and it shares the syntactic and semantic information of the gap: for example, a *wh*-pronoun can be an accusative form only if it corresponds to the object of the verb or of a preposition.

[1]We are only discussing the rules referred to in the HPSG literature as *filler-gap* constructions or strong unbounded dependency constructions.

[2]These constructions are discussed as a separate class of unbounded dependencies in the literature (Pollard and Sag 1996, see previous footnote).

The only information which is not shared is the type of the preposed constituent (unmarked, *wh*-word or relative word), which determines the type of the unbounded dependency.

Another example of rule organization is given by the passive rules: information is inherited along two different dimensions. First, the rule for simple passive, defined at the tree for transitive verbs, is inherited by trees lower in the hierarchy: for example, the tree for verbs with prepositional objects (V+PP, such as *look after*), inherits the information provided by the general passive rule, and need only specify idiosyncratic information (about the preposition of the original complement).

Second, the rule for passive with *by* inherits the information provided by the rule for simple passive and adds information relative to the prepositional phrase.

This hierarchical organization of rules captures the fact that there is one passive rule, which can vary depending on the object of the original transitive verb, and whether the agent is expressed or not. This cannot be captured if the different passive trees are represented independently of each other, with no more connection between them than between a passive tree and, for example, the gerund tree.

2.2 Application of rules

Not all rules are applicable to all trees, and not all orders of rule application are valid for all trees. There are three ways in which we constrain rule application:

- by the rule's position in the main hierarchy – as discussed above, rules applying to just a subset of trees can be located at the most general node defining that subset, and no other trees will be able to access the rule. The fact that inheritance is non-monotonic allows the expression of exceptions to rules: for example, transitive verbs which cannot passivize inherit from the general definition for transitive verbs, but add that the passive rule does not apply.

- by specifying conditions directly within the rule – for example, the passive rule can check that the first complement really is a noun phrase and fail to apply if not. Note that this may not be achievable pnrely by method (1) due to the possibility of applying other rules first: a transitive verb from which the direct object has been extracted will still be within the scope of the passive rule, but the rule will not be able to apply to it because the object has disappeared. Another example is the *wh*-question on the subject: the

rule should apply only if the subject is not an expletive pronoun, and this has to be checked by the rule itself.

- by explicitly specified constraints – although the previous two methods provide theoretically adequate mechanisms for all constraints, for efficiency reasons we also maintain a separate model of which rules can apply in which combination. The rules defined at each node are first grouped into sets if they enter into a paradigmatic relationship (if they cannot apply simultaneously on the same tree): this is the case in English for the extractions rules discussed earlier, and for the passive rules, for example. Rules and sets of rules are then ordered, according to a partial ordering of the rules, and all possible rule application sequences which respect that ordering are computed off-line. Not all these sequences will apply in all cases (due to the constraints of type (1) and (2)) but this is still much more efficient than blind search through *all* possible rule combinations.

This situation is reminiscent of the debate about rule ordering which took place in transformational grammar in the seventies (Soames and Perlmutter (1979)). One position defended an ordering of transformations, the other position maintained that ordering the rules is unnecessary, because rules should be allowed to apply whenever their structural description is met. In practical applications, however, this means computing and testing all possible rule combinations, which in the case at hand is impractical[3].

2.3 Grammar expansion and parsing

The LEXSYS grammar currently includes 44 basic trees and 35 rules which together expand to 619 trees. This is work in progress, and we predict that the number of trees will quickly grow. Also, we do not allow disjunctive feature values, but use multiple instances of the same tree, and this will also increase the number of trees. We currently expand the grammar as an off-line process before parsing. The high number of trees resulting from this expansion might be seen as a drawback for parsing, but techniques described in Evans and Weir (1998) can be applied to optimise the parsing of such large grammars by converting them to automata which can be merged and minimised.

[3]Rule application is an issue also in computational applications of lexicalist grammars which use rules, such as HPSG (Meurers and Minnen 1995).

3 Related work

Other work in this general area includes Becker (1993, 1994) and Vijay-Shanker and Schabes (1992). Somewhat more recently, Candito (1996) presented an approach which is somewhat different in spirit. In her approach, LTAG is viewed as the compilation of what she calls a *metagrammar*. This metagrammar is based on the notion of syntactic function and hierarchically organizes information along three dimensions: initial predicate-argument structure, redistribution of functions and surface realization of syntactic functions. These three types of information are combined to yield cross-classes, and there is a step of translation of these resulting classes into trees. Inheritance is monotonic, except for functional information (the redistribution of functions can overwrite the initial distribution of functions). This scheme does not provide for an efficient way to handle exceptions or subregularities: if a predicate does not select some trees belonging to its tree family, these trees have to be stipulated in the lexical entry of the predicate.

On the other hand, trees, in our approach, are directly organized in a non-monotonic inheritance hierarchy, so that there is no translation step. Our use of nonmonotonicity enables us to capture exceptions easily, but also contributes to the succinctness of some of our generalisations. A detailed comparison of the two approaches on a significant grammar fragment would therefore be very interesting.

References

Tilman Becker. 1993. HyTAG: *A new type of Tree Adjoining Grammar for hybrid syntactic representation of free word order languages.* Ph.D. thesis, Univ. des Saarlandes.

Tilman Becker. 1994. Patterns in metarules. In *Proceedings of the Third International Workshop on Tree Adjoining Grammars*, 9–11.

Marie-Hélène Candito. 1996. A principle-based hierarchical representation of LTAGs. *COLING-96*, Copenhagen, 194-199.

John Carroll, Nicolas Nicolov, Olga Shaumyan, Martine Smets and David Weir. 1998. The LEXSYS Project. In *Proceedings of the fourth TAG+ workshop.*

Roger Evans and Gerald Gazdar. 1996. DATR: A language for lexical knowledge representation. *Computational Linguistics* **22.2**, 167-216.

Roger Evans, Gerald Gazdar and David Weir. 1995. Encoding lexicalized tree adjoining grammars with a nonmonotonic inheritance hierarchy. *33rd*

Annual Meeting of the Association for Computational Linguistics, 77-84.

Roger Evans and David Weir. 1998. A structure-sharing parser for lexicalised grammars. *COLING/ACL 98.*

Walt Detmar Meurers and Guido Minnen. 1995. A Computational treatment of HPSG lexical rules as covariation in lexical entries. *Proceedings of the Fifth International Workshop on Natural Language Understanding and Logic Programming.*

K. Vijay-Shanker and Yves Schabes. 1992. Structure sharing in lexicalized tree-adjoining grammar. In *COLING-92*, 205–211.

Scott Soames and David Perlmutter. 1979. *Syntactic Argumentation and the Structure of English* University of California Press.