CDL–TAGs: A grammar formalism for flexible and efficient syntactic generation

Anne Kilger and Peter Poller

DFKI GmbH, Stuhlsatzenhausweg 3
D-66123 Saarbrücken, Germany

Abstract

During the last decade we developed and continuously improved CDL–TAGs, *an extension of* TAGs *for incremental syntactic generation. This paper presents the current state of development and gives details of the definition of context dependent linearization rules.*

1. Introduction

This paper presents *Tree Adjoining Grammars with Context-Dependent Disjunctive Linearization Rules (CDL–TAG)* that have been developed for incremental syntactic generation in the system WIP (WAF$^+$93). CDL–TAGs were successfully used in the projects PERFECTION (Fin96), EFFENDI (PH 96), PRACMA (JKN$^+$94), and VERBMOBIL (Wah93).

A fully incremental system is characterized by realizing interleaved input consumption, processing and output production, so that first output elements may even be produced before the input is complete. So, decisions based on the data at hand impose assumptions about the outstanding input, thereby reducing the set of input increments that can be consistently integrated into processing. For syntactic generation, two different processing levels can be distinguished. First, for each new element the hierarchical structure of the sentence under construction has to be expanded. Second, elements have to be positioned in the final utterance thereby constraining any further positioning. According to that, it is essential to choose a syntactic representation formalism that facilitates the dynamic construction of the hierarchical structure and the stepwise linearization and utterance production for its substructures. The grammar formalism must be flexible enough to preserve word order variations as long as possible during generation. Thereby, it should be easy to handle the prefix of the sentence already uttered as constraining the set of applicable linearization rules. Additionally, the grammar formalism should support linearization rules that describe situational factors (e.g., time or space restrictions).

The separation of a grammar into Hierarchical and Positional constraints (in the following called *H/P paradigm*) fulfills these requirements. Such a grammar (e.g., LD/LP–TAGs (Jos87)) consists of two distinct sets of rules, one merely describing mother–daughter relations only hierarchically, while the other describes positional constraints by referring to elements of the hierarchical structures.

This paper presents *CDL–TAGs*, that almost perfectly reflect the required different levels of processing for incremental syntactic generation and thereby strongly facilitate the implementation of the incrementality effects on syntactic generation (FS92).

2. Definition of CDL–TAGs

TAG with Context–Dependent Disjunctive Linearization Rules (CDL–TAG) is an extension of Tree Adjoining Grammar (JLT75) that helps to design an extremely compact grammar by avoiding redundant descriptions without extending the power of the formalism.

2.1. *The Standard TAG Formalism*

Standard TAG combines elementary (initial and auxiliary) trees by *adjoining*, an operation which makes the grammar mildly context–sensitive and adequate for the representation of natural language. The TAG formalism has been extended by a second combination operation called *substitution* which has only context–free power (SAJ88). In order to allow compact representations of complex syntactic dependencies, TAG has been extended furthermore by feature structures (*TAGs with unification*, (Kil92), (Kil94), or *Feature Structure Based TAG*, (VJ88)).
The H/P–paradigm was applied to TAG by (Jos87). He defined *Local Dominance/Linear Precedence–TAG (LD/LP–TAG)* by "taking the elementary trees as domination structures over which linear precedences can be defined." The descriptive power of LP–rules in LD/LP–TAGs is not sufficient to describe all linearization alternatives of one hierarchical structure locally, i.e., without duplicating the hierarchical structure (e.g., for German verbal phrases subject–verb–object, object–verb–subject, . . .). Furthermore, there is no means to associate different LP–rules with contextual (semantic and pragmatic) constraints. To get more flexible linearization, we developed a new extension of TAG on the basis of LD/LP–TAGs.

2.2. *CDL–TAG*

CDL–TAG is defined according to the H/P paradigm, i.e., domination structures are used as elementary structures instead of trees. The possible orderings of sister nodes are restricted by *linearization rules* which are associated with the mother node. They have the form: "(<" {"("context lin-rule* ")"}* ")". The rules are initiated by the key "<". Each alternative starts with the name of a *context* in which the rule is valid. The value of *context* is matched with a feature *lin-context* of the feature structure associated with the respective node.
The left part of Figure 1 illustrates a VP–node whose subtree represents a German verbal phrase.

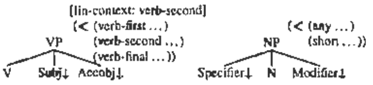

Figure 1: Examples for German Linearization Rules

Its linearization rules include statements about verb–first, verb–second and verb–final word order while `verb-second' is the actual `lin-context' inside its feature structure. Other contexts (like `any' or `short' at the NP–node in the right part of the figure) distinguish word order rules that differ with respect to their suitability for specific situational — non–syntactic — factors which is useful for a generation system with globally set `parameters'. E.g., the value `short'[1] is used for word orders that permit to save space and time in the final utterance.
Each $lin - rule$ is encoded as a list that contains linearization elements $lin - el$. The order of the list elements defines the order of the elements of the TAG tree they refer to. A symbol sym is a $lin - el$ and refers to a daughter of the node the linearization rule is associated with.
In order to describe constraints on sister nodes which include complements as well as optional elements, we extended the formalism by a combination operation that allows to add sister nodes without introducing additional depth into the tree. The operation of *furcation* has been defined by (DK88) as the unification of two root nodes of structures to one root node with two substructures. We adapted it to CDL–TAGs by defining an new kind of elementary tree, namely a *furcation auxiliary tree* whose foot node is leftmost or rightmost daughter of the root node, as a structure leaving away the foot node[2].

[1] The key `short' is meant in the sense of saving space and time in the final utterance when using this alternative. This may be meaningful under time pressure or when the space for the written text is restricted.

[2] This is comparable to the *modifier auxiliary tree* in contrast to the *predicative auxiliary tree* introduced by

Some symbols sym in $lin - rule$ denote adjuncts. They have to be detailed enough to express all aspects that influence word positions. For English, e.g., different adverb classes have to be defined according to their different linearization constraints. Symbols referring to adjuncts always appear inside disjunctions with `regular–like' expressions. They permit to describe exactly one occurrence of one element of a list by $((sym_1...sym_n)^1)$, one or zero occurrences by $((sym_1...sym_n)^{1/0})$, at least one occurrence of elements by $((sym_1...sym_n)^+)$ or an arbitrary (or zero) number of elements by $((sym_1...sym_n)^*)$.
The following expression is a possible linearization rule of the VP–node of Figure 1:

```
(verb-second
    (subj v ...(advp)* ...accobj ...)        ((advp)^1 v subj ...(advp)* ...accobj ...)
...)
```

It shows two alternatives to fill the first position of a verbal phrase in the linearization context *verb–second*, namely a complement (`subj' refers to the subject), or exactly one optional element (`advp' refers to an optional adverbial phrase). After the first element, the finite part of the inflected verb (referred to by `v' in the linearization rule) has to follow. The second expression prescribes that the subject directly follows the verb in case of a topicalized adverbial phrase.
Furthermore, the selection of adequate linearization rules may be restricted by features of the subtree to be linearized. CDL–TAGs use *child-info* that is inherited from the daughters of the node the linearization rule is associated with. The resulting structure for LP–rules is "(<" {"("context child-info lin-rule* ")"}* ")". The entry *child–info* realizes a specific test (identified by the key `test') for feature–value–combinations which have to hold for some of the daughters of the actual node. The LP–rule

```
(short (test (mod (cat) name))
        (...mod ...(adjp)* ...n ...)
...)
```

might be associated with the NP–node on the right in Figure 1. It describes a possible linearization of a Specifier–Noun–Modifier construction in German: Instead of "Die Werke Goethes" (the works of Goethe) it is also possible to say "Goethes Werke" (Goethe's works). The presupposition for choosing this `brief' linearization alternative is that the modifier is realized as a proper name which is tested by referring to the third daughter of NP (the Modifier↓ node, referred to in the test above by `mod') and then checking the equality of feature–value of `cat' and the atomic value `name'.
The generative power of CDL–TAGs is equivalent to standard TAG (with constraints) because the only addition to standard TAG is the combination operation "furcation" which has only context–free power. So, CDL–TAGs are not sufficient to describe all linearization phenomena that include adjuncts. E.g., there is no easy way to describe scrambling without mixing hierarchical and positional information. Nevertheless, we use it as a promising starting point, concentrating on its usefulness for (incremental) syntactic generation.

3. Conclusions and Future Work

In this paper we presented CDL–TAGs, a highly compact grammar formalism, that is especially well–suited for the representation of grammar sources for (incremental) natural language generation. Furthermore, the lexicalization allows the grammar to consider a subset of word class specific elementary trees (tree families) for each lexical entry.
The TAG–GEN generator (Kil94) makes use of the CDL–rules by preferring linearization alternatives that reflect the order of input elements so that the output can start as early as possible,

(SS92). In this sense, furcation auxiliary trees are the CDL–TAG variant of sister adjunction in, e.g., DTG (RVW95) and furcation in, e.g., TFG (Cav98).

e.g., by fronting elements which are given early in the input. It also sorts linearization alternatives according to some generation parameters such as time pressure and style.
Although the formalism has been successfully used in several different application systems, there is no grammar developing tool yet. So, the most important task for future work is the development and implementation of a CDL–TAG parser, e.g., as an extension of the work described in (Pol94).

References

M. Cavazza. An integrated parser for TFG wich Explicit Tree Typing In *TAG+4'98*, Institute for Research in Cognitive Science (IRCS), University of Pennsylvania, Philadelphia, PA, 1998.

K. De Smedt and G. Kempen. The representation of grammatical knowledge in a model for incremental sentence generation. In *INLG'88*, Santa Catalina Island, CA, 1988.

W. Finkler. Automatische Selbstkorrektur bei der inkrementellen Generierung gesprochener Sprache unter Realzeitbedingungen. Dissertation. Universität des Saarlandes, Saarbrücken, 1996.

W. Finkler and A. Schauder. Effects of Incremental Output on Incremental Natural Language Generation. In B. Neumann, editor, *ECAI'92*, p. 505–507, Vienna, Austria, August 1992.

A. Jameson, B. Kipper, A. Ndiaye, B. Schäfer, J. Simons, T. Weis, D. Zimmermann. Cooperating to be Noncooperative: The Dialog System PRACMA In B. Nebel and L. Dreschler-Fischer (Ed.), *Proceedings of the 18th German Annual Conference on Artificial Intelligence : KI-94: Advances in Artificial Intelligence*, Saarbrücken, 1994. LNAI 861. Springer, Berlin (1994). 106-117.

A. K. Joshi, L. Levy, and M. Takahashi. Tree adjunct grammars. *Journal of the Computer and Systems Science*, 10(1):136–163, 1975.

A. K. Joshi. Word-order variation in natural language generation. In *AAAI'87*, p. 550–555, Seattle, USA, 1987.

A. Kilger. Realization of tree adjoining grammars with unification. DFKI Technical Memo TM–92–08, German Research Center for Artificial Intelligence – DFKI GmbH, 1992.

A. Kilger. Using utags for incremental and parallel generation. *Computational Intelligence*, 10(4):591–603, November 1994.

P. Poller. Incremental parsing with LD/TLP-TAGs. *Computational Intelligence*, 10(4):549–552, November.

P. Poller and P. Heisterkamp. EFFENDI – Effizientes Formulieren von Dialogbeiträgen – Bericht zum Projektende – Handbuch zur SIL–Schnittstelle. Technischer Bericht Nr. F3–96–014, Daimler–Benz Forschung und Technik, Ulm, 1996.

O. Rambow, K. Vijay–Shanker, and D. Weir D–Tree Grammars Proceeding of *ACL'95*, MIT, Cambridge, MA, 1995.

Y. Schabes, A. Abeillé, and A. K. Joshi. Parsing strategies with lexicalized grammars: Application to tree adjoining grammar. In *COLING'88*, p. 578–583, Budapest, Hungary, 1988.

Y. Schabes and S. M. Shieber. An alternative conception of tree–adjoining derivation. In *ACL'92*, p. 167–176, Newark, DW, 1992.

K. Vijay-Shanker and A.K. Joshi. Feature Structure Based Tree Adjoining Grammars. In *COLING'88*, Budapest, Hungary, 1988.

W. Wahlster, E. André, W. Finkler, H.-J. Profitlich, and T. Rist. Plan-based integration of natural language and graphics generation. *Artificial Intelligence*, 63:387–427, 1993.

W. Wahlster. Verbmobil: Translation of face–to–face dialogs. Research Report RR-93-34, DFKI GmbH, Saarbrücken, FRG, 1993.