and de Smedt's **Segment Grammar**, the forcing of adjunction on a semantic basis in analysis of complement attachment as adjunction (as Santorini & Kroch would do), and even the relation of PF, SS, DS and LF in government-binding theory.

The expressive power of synchronous TAGs extends that of pure TAGs. In an attempt to understand the source of this power, I developed an alternative formalization of adjunction, and of related operations like synchronous adjunction, that allowed a definition of a notion of a **monotonic** operation. I noted that adjoining constraints and link updating in synchronous TAGs are both nonmonotonic in this sense, and it appears to be the interaction between two nonmonotonic operations that underlies the extended power.

TAGs with Unification

Bela Buschauer, Peter Poller, Anne Schauder, Karin Harbusch
DFKI
Stuhlsatzenhausweg 3
W-6600 Saarbrücken 11
buschau or poller or schauder or harbusch@dfki.uni-sb.de

The presented definition of Tree Adjoining Grammars with Unification (UTAG) is an approach to embed TAGs in a feature structure based unification system. In the feature structures associated with the elementary trees, constraints and relations among the dependent nodes can be stated directly. The use of variables within feature structures makes it possible to represent a grammar (especially a grammar for natural language) in a more compact way.

We define an integrated mechanism of adjoining with unification. The feature structures (DAGs) are specified at the nodes of elementary trees in form of specification lists according to the PATR-formalism. In order to allow inheritance of information all over the trees there may be links between the DAGs of neighboring nodes (father-son-relations). The main problem with this combination of the two formalisms "TAG" and "unification" is the question, how to manage such links in case of adjoining. If a node becomes an adjoining node, it has to be erased during adjoining and be replaced by an auxiliary tree. It is unavoidable to cut already existing links and newly connect them to be able to fit in the auxiliary tree. This is done dynamically and automatically during adjoining. By this process the unification loses its "monotonicity property".

This approach has the advantage that in each phase of the construction of a tree starting from an initial tree to the complete syntax tree the grammar designer is able to see the effects of the information flow through the connected DAG structure. In contrast to our solution for the problem of adjoining with unification, Vijay-Shanker and Joshi define a static splitting of the DAGs (into top- and bottom-features) for their definition of FTAG (Feature Structure based Tree Adjoining Grammar) that allows adjoining without cutting off existing links. The disadvantage of their approach seems to be that the top- and bottom-features at the nodes of elementary and derivated trees are not unified until all adjoinings have been done. So there is no information flow throughout the tree during the computation of the complete syntax tree.

Further discussion has to show whether there exists a clear difference regarding the practical usefulness of the two definitions especially for incremental computations.

Metarules in Tree Adjoining Grammars

Tilman Becker
Department of Computer and Information Science
3401 Walnut Street 4C
Philadelphia, PA 19104, USA
tilman@grad1.cis.upenn.edu

This talk discusses metarules as an extension to the TAG formalism. Metarules allow for a more compact representation of grammars, especially for natural languages. They also capture generalizations that can not be expressed in the original framework.

Metarules consist of an "input-pattern" and an "output-pattern". If a grammar rule matches the output-pattern (i.e. there is a substitution for the variables in the pattern that makes it equal to the grammar rule), the application of the metarule generates a new grammar rule (i.e. the output-pattern with its variables substituted according to the matching).

Other grammar formalisms like GPSG, HPSG, Categorial Grammars and Van Wijngarden Grammars have used metarules for compactification and generalizations. But they all encountered the problem of the generative power of metarules. If metarules are allowed to be applied recursively (and thereby produce infinite sets of grammar rules), the resulting formalism can generate every r.e. language.

This talk presents two different approaches to avoid this problem with metarules for TAGs. The first approach is a restriction of the form of metarules to one variable that can match only one subtree. For this definition it has been shown that it does not increase the generative power if such metarules apply recursively. The restricted form of metarules, however, is a drawback because it does not allow for a compact description of some generalizations. A second approach allows unrestricted patterns and variables for metarules, but restricts arbitrary recursive application of metarules. This is based on two properties of TAGs: 1) The adjoining operation already factors recursion in a compact way. 2) The extended domain of locality of an elementary tree has a bounded size. Property 1) rules out arbitrary recursive application and property 2) motivates a boundary on the size of elementary trees. The proposed definition allows the output of a metarule as a new elementary tree only if it is smaller than a given boundary (e.g. it contains at most one predicate-argument structure). This also rules out arbitrary recursive application of metarules. On the other hand the descriptive power of metarules can be enlarged to handle a large set of generalizations.