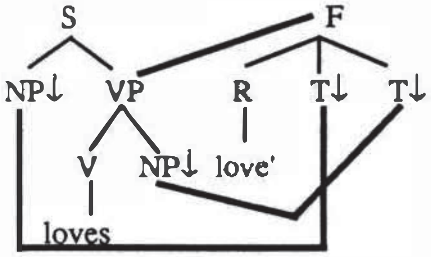
Formal Properties
of
Synchronous Tree-Adjoining Grammars

Stuart Shieber
Aiken Computation Laboratory
33 Oxfort Street, No. 14
Harvard University
Cambridge, MA, USA
shieber@harvard.harvard.edu

Tree-adjoining grammars constitute a grammatical formalism with attractive properties for the strong characterization of the syntax of natural languages. These properties, however, present a challenge for the application of TAGs beyond the limited confines of syntax—to the task of semantic interpretation, for instance, or automatic translation of natural language.

Previous work in the TAG framework has recognized, implicitly at least, the importance of semantics in its exhibition of certain intuitions about the semantic ramifications of TAG analyses. The formalism of **synchronous TAGs** makes these intuitions explicit.

Synchronous TAGs are a formalism based on pure variants of TAGs for describing not a single language, but a relation between two languages—a natural language and its associated logical form language, or two natural languages, for example. The relation is stated through the pairing of elementary trees in two base TAGs. For example, the following pairing describes the relation between a verb "loves"' and its semantic interpretation "*love*".



The dashed links pair corresponding nodes in the trees, and serve to mark node pairs at which synchronous adjunction (or substitution) can occur.

In the talk at Dagstuhl, I showed how synchronous TAGs make explicit many of the semantic intuitions implicit in previous work on TAG analyses of natural language, and discussed several applications of synchronous TAGs, including the analysis of idioms, quantifier scope, machine translation and generation. (During the workshop, other potential applications arose, such as the declarative codification of the procedure relating constituent structure and functional structure in Kempen 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