Karin Harbusch, Wolfgang Wahlster (editors)

Tree Adjoining Grammars
1st. International Workshop on TAGs: Formal Theory and Applications

Dagstuhl-Seminar-Report; 2
15. - 17.8.1990 (9033)

ISSN 0940-1121

Tel.: +49-6871 - 2458
Fax: +49-6871 - 5942

Das Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI) ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.

Bezugsadresse: Geschäftsstelle Schloß Dagstuhl
Informatik, Bau 36
Universität des Saarlandes
W - 6600 Saarbrücken
Germany
Tel.: +49 -681 - 302 - 4396
Fax: +49 -681 - 302 - 4397
e-mail: office@dag.uni-sb.de

First International Workshop on Tree Adjoining Grammars: Formal Theory and Applications

organized by :
Karin Harbusch (DFKI, FRG)
Wolfgang Wahlster (DFKI, FRG)

Wednesday, August 15 - Friday, August 17
1990

Overview

Karin Harbusch, Wolfgang Wahlster

The topic of the workshop was a grammar formalism - the **Tree Adjoining Grammars (TAGs)** - which has interesting formal properties (e.g., mild context-sensitivity) as well as a wide range of application domains, especially in the field of natural language processing. Thus, it was very fruitful for the discussions to bring together researchers from both areas of interest in TAGs.

TAGs were introduced in 1975 by Joshi, Levy and Takahashi ([Joshi et al. 75]). To get a first intuition of the formalism - for a good introduction see [Joshi 85] - one can think of TAG rules as combined context-free rules building a context-free derivation tree. These trees are called *initial trees*. A second class of rules - the *auxiliary trees* - which are necessary for describing arbitrary large TAG-derivation trees - are characterized by a special nonterminal leave - the *foot node*) - in the context-free derivation tree which carries the same label as the root node. The *adjoining* operation replaces a nonterminal node in an initial tree (which can be modified by former adjoinings) by an auxiliary tree. This means that the incoming edge in the root node will end in the root node of the auxiliary tree and all outgoing edges of the eliminated node will start in the foot node of the auxiliary tree. Obviously a derivation tree results again.

To get an idea of such a grammar Figure 1 describes a fragment of a natural language grammar. The initial tree α can produce sentences like, e.g., "Children play" where "Children" is a lexical entry with the terminal category **N** and "play" is of category **V**. The auxiliary trees β_1, β_2 and β_3 modify the NP node by a determiner, adjectives and relative clauses, respectively. The auxiliary trees β_4 and β_5 modify the verbal complex (VP) by a prepositional object (e.g., "with balls") or a direct or indirect object (e.g., "tennis").

The similarity with context-free grammars can lead to the conclusion that TAGs are simply an equivalent description for context-free grammars. But one important property of TAGs is that they are more powerful than context-free grammars (e.g., there exists a TAG for $a^n\ b^n\ c^n$ or the copy language ww). This additional power is called *mild context-sensitivity* because not the complete set of context-sensitive languages is covered by TAGs (e.g., the languages $a^n\ b^n\ c^n d^n\ e^n$ or the copy language www).

The TAG formalism was introduced as an adequate formalism for encoding natural language grammars referring to the property of mild context-sensitivity. There is strong evidence in the linguistic community that this is the right complexity for natural language description.

The workshop delt with various problems in the formal area, e.g., extensions for the pure TAG formalism, automata models for the grammar representation or efficient parsing algorithms. Most investigations were motivated by specific applications (e.g., natural language parsing and generation, help systems).

In this interdisciplinary field of computer science, computational linguistics and

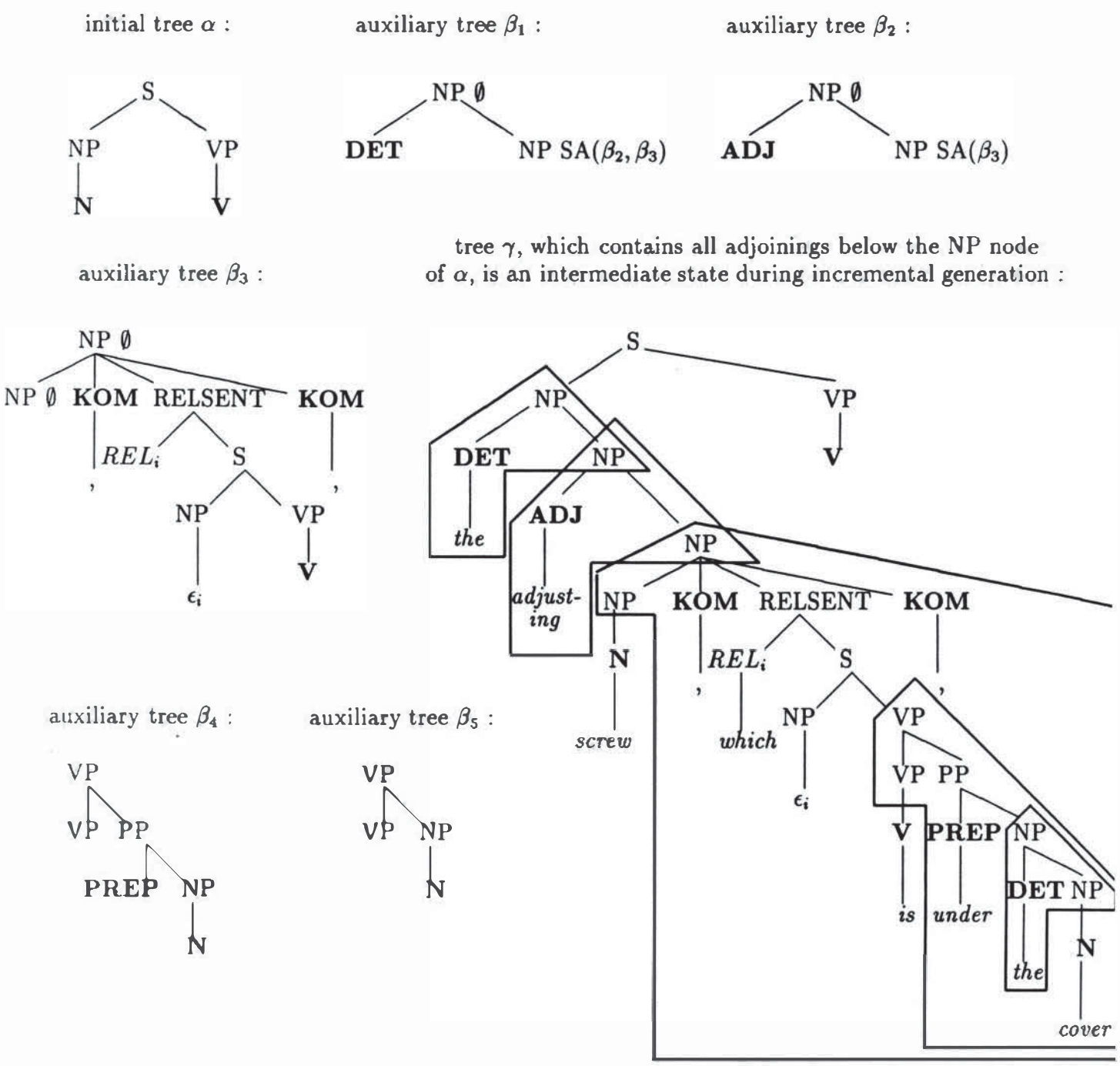


Figure 1: Example for adjoinings with the resulting sentence fragment **'The adjusting screw, which is under the cover, ...'**

psycholinguistics the talks found interesting feedback and a lot of very fruitful discussions went on during the three days.

References

[Joshi et al. 75] **A. K. Joshi, L. S. Levy, M. Takahashi :** *Tree Adjoining Grammars,* Journal of Computer and Systems Science 10:1, Seite 136-163, 1975.

[Joshi 85] **A. K. Joshi :** *An Introduction to Tree Adjoining Grammars,* Technical Report MS-CIS-86-64, Department of Computer and Information Science, Moore School, University of Pennsylvania, Philadelphia, Pennsylvania, 1985.

Acknowledgements

We want to thank the International Conference and Research Center for Computer Science (IBFI) for the financial support and the IBFI stuff for their assistance with arranging the workshop.

We want to thank the German AI Center (DFKI) for additional funding of our workshop. This made it possible to support people from overseas to attend the workshop.

List of Participants

Anne Abeillé, University of Paris 7 - Jussieu, France
Tilman Becker, University of Pennsylvania, USA
Bela Buschauer, DFKI, FRG
Jerome Chiffaudel, University of Paris 7 - Jussieu, France
Sharon Cote, University of Pennsylvania, USA
Koenraad DeSmedt, NICI - University of Nijmegen, Netherlands
Yonggang Guan, Universität des Saarlandes, FRG
Wolfgang Finkler, DFKI, FRG
Karin Harbusch, DFKI, FRG
Günter Hotz, Universität des Saarlandes, FRG
Mark Johnson, Brown University, USA
Aravind Joshi, University of Pennsylvania, USA
Gerard Kempen, NICI - University of Nijmegen, Netherlands
Anthony Kroch, University of Pennsylvania, USA
Bernard Lang, INRIA, France
David McDonald, Content Technonlogies, Inc., USA
Michael Palis, University of Pennsylvania, USA
Peter Poller, DFKI, FRG
Beatrice Santorine, University of Pennsylvania, USA
Yves Schabes, University of Pennsylvania, USA
Anne Schauder, DFKI, FRG
Stuart Shieber, Harvard University, USA
Kuniaki Uehara, Kobe University, Japan
K. Vijay-Shanker, University of Delaware, USA
Wolfgang Wahlster, DFKI, FRG

Program

Wednesday, August 15:
Welcome by Reinhard Wilhelm (IBFI) and Wolfgang Wahlster (DFKI)

Formal Properties of Synchronous Tree-Adjoining Grammars, *S. Shieber*
TAGs with Unification, *B. Buschauer, P. Poller, A. Schauder, K. Harbusch*
Metarules in Tree Adjoining Grammars, *T. Becker*
Multicomponent TAGs, *D. Weir* - Talk given by K. Vijay-Shanker
Embedded Pushdown Automata, *K. Vijay-Shanker*
TAGs by Interpreting Context Free Tree Languages, *Y. Guan, G. Hotz*

Thursday, August 16:
The systematic construction of Earley Parsers:: Application to the production of an $O(n^6)$ Earley Parser for Tree Adjoining Grammars, *B. Lang*
The Valid Prefix Property and Parsing Tree Adjoining Grammars, *Y. Schabes* Parallel TAG Parsing on the Connection Machine, *M. Palis, D. Wei*
Tree Adjoining Grammar, Segment Grammar and Incremental Sentence Generation, *G. Kempen, K. DeSmedt*
Incremental Natural Language Generation with TAGs in the WIP Project, *W. Finkler*
Implications of Tree Adjoining Grammar for Natural Language Generation, *D. McDonald, M. Meteer*

Friday, August 17:
Features in a Lexicalized TAG for English, *Sharon Cote*
A TAG analysis of the Third construction in German, *Anthony Kroch, Beatrice Santorini, Aravind Joshi*
French and english determiners: Interaction of morphology, syntax and semantics in Lexicalized Tree Adjoining Grammars, *Anne Abeillé*
Japanese Tree Adjoining Grammar and its Application to On-Line Help System NeoAssist, *Kuniaki Uehara*
Coordination in TAG in the manner of CCG (Combinatory Category Grammars): Fixed vs Flexible Phrase Structure, *Aravind Joshi*

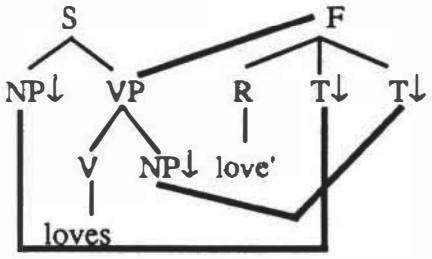
Formal Properties
of
Synchronous Tree-Adjoining Grammars

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

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.

Multicomponent Tree Adjoining Grammars

David Weir
Department of EECS
Northwestern University
USA
weir@weir.eecs.nwu.edu

Multicomponent Tree Adjoining Grammars (MCTAG) are intended as a way to extend the domain of locality of Tree Adjoining Grammars. A MCTAG is made up of a set of elementary tree sets and at each step in a derivation all of the trees in a set must be adjoined together. This operation is called multicomponent adjunction. In TAG relationships can be stated between nodes within the same elementary tree. In MCTAG additional expressive power is achieved since relationships can be stated between nodes in different trees that are in the same elementary tree set. Several alternative notions of a MCTAG derivation are possible depending on the size of the domain into which the trees in a set are adjoined during a derivation.

When the domain of multicomponent adjunction is a single elementary tree we show that the system has the same generative capacity as TAG. Many of the cases in which MCTAG have been used to give linguistic analyses assume this version of multicomponent adjunction.

When multicomponent adjunction is local to the nodes of trees in an elementary tree set (we call this local MCTAG) then additional generative power results. We show that the class of string languages generated is larger and depends on the number of trees in the largest elementary tree set. The class of tree sets generated is larger: we show that it is possible to generate tree sets with dependent branches and sets whose path sets are more complex than those of TAG. We have shown (Weir 1988) that this form of MCTAG is weakly equivalent to the Linear Context-Free Rewriting Systems (Vijay-Shanker, Weir and Joshi 1987). One consequence of this is that polynomial recognition of local MCTAL is known to be possible. The derivations of local MCTAG can be represented with trees using a similar approach to that used for TAG. We show that the set of derivation trees for a local MCTAG is a local set, i.e., can be generated by a Context-Free Grammars.

The final case is one in which the domain of multicomponent adjunction is unconstrained (we call this non-local MCTAG). At this point, it is an open question as to how the generative power of this versions of MCTAG relates to the others. We show how the derivations of non-local MCTAG can be naturally represented with acyclic multigraphs.

References

Vijay-Shanker, K., Weir, D. J., and Joshi, A. K., Characterizing structural descriptions produced by various grammatical formalisms. In Proceedings of 25th meeting of Association of Computational Linguistics, July 1987.

Weir, D. J. Characterizing Mildly Context-Sensitive Grammar Formalisms, Ph.D. Thesis, University of Pennsylvania, 1988.

Embedded Pushdown Automata

K. Vijay-Shanker
Department of Computer and Information Science
University of Delaware
Newark, DE 19716, USA
vijay@udel.edu

This talk discussed a class of automata that recognize exactly the class of TALs. The definition of EPDA can be motivated by noting the difference between the structures derived in CFG formalism and TAG.

The EPDA can be considered as a second order PDA whose storage is a push-down of pushdown of symbols. The aim of the talk was to informally explain the relationship between TAGs and EPDA as well as other weakly equivalent formalisms, Head Grammars and Combinatory Categorial Grammars. Finally the addition of nonlinearity to EPDA move can be shown to be similar to addition of coordination schema to Categorial Grammars.

TAGs by Interpreting Context Free Tree Languages

Yonggang Guan, Günter Hotz
Fachbereich 14 - Informatik
Universität des Saarlandes
Im Stadtwald 15
W-6600 Saarbrücken 11, FRG
guan or hotz@sbsvax

By functional multilinear interpretations of context free tree languages, we are able to define infinite hierarchies

$$\mathcal{L}_0 \subsetneq \mathcal{L}_1 \subsetneq \cdots \subsetneq \mathcal{L}_k \subsetneq \cdots$$

of languages. $\mathcal{L}_0$ is identical with the class of context free languages, $\mathcal{L}_1$ with the class of TAGs with constraints. k is the maximal indegree of the nodes of the trees. These classes of languages appear as natural generalizations of the context free languages. Each $\mathcal{L}_k$ satisfies a pumping lemma and for

$$D_k = \{a_1^n a_2^n \cdots a_k^n \mid n \in \mathbf{N}\}$$

it is

$$D_{2k+2} \in \mathcal{L}_k \text{ but } D_{2k+2} \notin \mathcal{L}_{k-1}.$$

Each $L \in \mathcal{L}_k$ is semilinear under the Parikh mapping. Each class $\mathcal{L}_k$ is closed under union, concatenation, and the Kleene $*$-operation.

There is a second form of representations for these classes. $\mathcal{L}_k$ can be generated by semi-Dyck controlled coupled substitutions. These representations allow to reduce the parsing problem to the parsing problem of context free languages.

The systematic construction of Earley Parsers: Application to the production of an $O(n^6)$ Earley Parser for Tree Adjoining Grammars

Bernard Lang
INRIA
B.P. 105
78153 Le Chesnay, CEDEX, France
lang@margaux.inria.fr

Logic Programming languages (Prolog) were originally introduced as an extension of context-free (CF) languages. Conversely CF grammars may be seen as logic programs (or Horn clauses) where the predicates are the CF grammar symbols, and where these predicates have arguments corresponding to the boundaries of the input string fragments they derive into.

Earley's parsing algorithm can be generalized to Horn Clauses as a dynamic programming evaluation technique. Keeping in mind the relation between Horn clauses and CF grammars, we suggest encoding similarly in Horn Clauses other syntactic formalisms so as to take advantage of this generalisation of Earley's algorithm to obtain for free efficient parsers for these encoded formalisms.

As an example, we consider the problem of TAG parsing. We show that any TAG can be encoded into a logic program for which there is an evaluation in time $O(n^6)$. We show on this example how the general dynamic programming procedure can be adapted to conform the constraint that sentences be parsed from left to right, even in the presence of interleaved constituents as is the case for TAGs.

The Valid Prefix Property and Parsing Tree Adjoining Grammars

Yves Schabes
Department of Computer and Information Science
R-555 Moore School
University of Philadelphia
220 South Street 33rd Street
Philadelphia, PA 19104-6389, USA
schabes@linc.cis.upenn.edu

The valid prefix property (VPP), capability of a left to right parser to detect errors as soon as possible, is often unobserved in parsing CFGs. Earley's parser for CFG maintains the VPP and obtains a worst case complexity ($O(n^3)$) as good as parsers that do not maintain VPP (as the CKY parser). Contrary of CFGs, maintaining the valid prefix property for TAGs seems costly.

The aim of talk was to informaly explain why the VPP for TAGs seems expensive to maintain and also to introduce a new Earley-style parser for TAGs which has $O(n^6)$ worst case time complexity. The new parser does not maintain VPP but it can

behave in linear time on some grammars, in $O(G^2n^4)$ worst time for unambiguous TAGs and in general in $O(G^2n^6)$-time in the worst case. An earlier Earley-type parser that we proposed in 1988 maintains the VPP but at its cost of its worst case complexity ($O(G^2n^9)$-time). To our knowledge, it is the only known polynomial-time general TAG parser that maintains the VPP. Both Earley-style parsers for TAGs use top-down filtering and therefore their behaviors are in practice superior to pure bottom-up parsers (as Joshi's and Vijay-Shanker's adaptation of CKY algorithm to TAG).

In practice, the importance of the VPP varies from grammars and is currently being evaluated on natural language TAG grammars for English and French.

Parallel TAG Parsing on the Connection Machine

Michael Palis, David Wei
Department of Computer and Information Science
School of Engineering and Applied Sciences
R-555 Moore School
University of Philadelphia
220 South Street 33rd Street
Philadelphia, PA 19104-6389, USA
palis@linc.cis.upenn.edu

We present a parallel parsing algorithm for Tree Adjoining Grammars (TAGs) and its implementation on the Connection Machine (CM). The CM TAG parser is designed to handle TAGs of arbitrary size without significant decrease in performance. Specifically, the expected run-time of the parallel algorithm is logarithmic in the grammar size (as opposed to quadratic in a serial implementation).

The CM TAG parser is an emulation of the CRCW PRAM algorithm. The PRAM algorithm is characterized by frequent communication between processors via the shared memory. Moreover, the pattern of inter-processor communication does not have the regular structure often found in many parallel numerical algorithms. Because the CM has a distributed memory, the emulation of the PRAM algorithm can only be realized by explicit message-passing, albeit between non-adjacent processors. Unfortunately, routing messages between non-adjacent processors is time-consuming on the CM. The CM uses a deterministic oblivious routing strategy, which, in the worst-case, can introduce $\sqrt{p}$ delay per emulated step, where p is the number of processors used.

To obtain a more efficient emulation, we employ randomization: i.e., grammar nodes of the TAG are mapped randomly to corresponding CM processors. In theory, this reduces the delay per emulated step to $O(log(p))$ with high probability. In practice, we use randomization as part of a pre-processing step: given a fixed TAG, we generate several random mappings of the TAG to the CM, then choose the most efficient mapping. The most efficient mapping is obtained experimentally by running

the CM parser for the different mappings. Note that the pre-processing step is only performed once – at the time the grammar is defined.

In the current CM implementation, a "coarse-grain" emulation of the PRAM algorithm is used. More specifically, the number of CM processors used in $| G |^2$ and the run-timeis $O(n^6 log | G |)$. The motivation behind this coarse-grain mapping is that for NL parsing, $| G | \gg n$; in particular, n is rarely more than 20. (This is direct contrast to parsing programming languages where $| G |$ is small but n, the length of the program to be parsed, can be arbitrarily large.)

The CM parser currently being developed is for a small grammar consisting of 55 trees which expands to approximately 200 grammar nodes (Yves Schabes' Small English Lexicalised TAG). Initial performance measurements indicate that the run-time is linear in n, rather than the theoretical $O(n^6)$ run-time. The next stage of the project is to enlarge the TAG and to measure the run-time of the CM with respect to both grammar size and sentence length. ¿From this experimental data, we hope to verify the logarithmic behavior of the run-time with respect to grammar size.

Tree Adjoining Grammar, Segment Grammar and Incremental Sentence Generation

Gerard Kempen, Koenraad DeSmedt
NICI
Department of Psychology
University of Nijmegen
NL/6525 HR Nijmegen, Netherlands
KEMPEN or DESMEDT@KUNPV1.PSYCH.KUN.NL

The cognitive process of syntactic structure formation is **lexically guided**, both in production and in parsing. "Lexicalized" grammars are therefore likely to figure prominently in psycholinguistic processing models. Tree Adjoining Grammars (TAG) and Segment Grammars (SG) are two such formalisms. They are similar in that they both use subsentential structures as building blocks: elementary trees (or mobiles) which are larger than individual nodes. At least one terminal node of a building block is a lexical node (as implied by the definition of lexicalized grammars).

A second property of human syntactic structure formation is **incremental generation**. This feature imposes special demands on the syntactic processor and its associated grammar. In our talk we evaluated TAG and SG from the point of view of the following three demands:

1. The processor should be capable of incrementing the current (incomplete) syntactic structure in any direction (leftward or rightward) and by any method (upward expansion, downward expansion and insertion).

2. Not only phrase- and clause-sized incrementsshould be allowed, but word-sized increments as well.

3. Syntactic coordination (inclusive of reduction phenomena such as gapping) should closely resemble the treatment of self-repair in spontaneous speech. (For example, the repair text can **refer back** to the reparandum text; this suggests that, during the computation of the repair text, the reparandum's structure is not destroyed. We hypothesize that reparandum and repair are "coordinated" in a way similar to the members of a conjunction).

We explained the workings of SG, compared it to lexicalized TAG and evaluated both in terms of the three demands.

The discussion and informal conversation revealed that the most important difference between SG and TAG resides in the fact that TAG uses only one level of syntactic representation, whereas SG distinguishes two levels: Functional (or F-) structures and Constituent (or C-) structures. Y. Schabes suggested informally that the mapping between C- and F-structures could be formalized in terms of S. Shieber's & Y. Schabes' **Synchronous TAG**. We added to this t he suggestion that one might consider a system performing a double mapping: Between Semantic and F-structures, and hetween F- and C-structures, and that this, in turn, could considerably simplify the complexity of the (TAG-style) syntactic structures in the 'middle' layer. For instance, we suspect that only 'canonical' trees suffice (as in SG F-structure), and that their expansion to tree families is no longer needed: this work is replaced by the F-to-C-structure mapping. These ideas deserve further scrutiny.

Incremental Natural Language Generation with TAGs in the WIP Project

Wolfgang Finkler
DFKI
Stuhlsatzenhausweg 3
W-6600 Saarbrücken 11
finkler@dfki.uni-sb.de

In my talk, I argued that lexicalized Tree Adjoining Grammars with unification are useful for the incremental processing at the syntactic level of description. In order to motivate the need for incremental natural language generation in the WIP project I gave a short overview of the system to present the specific requirements upon its natural language generation component.

Incremental generation means the immediate verbalization of the parts of a stepwise computed message. It is psychologically evident that humans often start speaking before they know exactly what the whole contents of their utterance will be. Because the WIP system shall be usable in scenarios where information to be presented is continuously supplied by an application system and where such information must be simultaneously presented in a condensed way to assist human decision makers – there is a need for incremental presentation.

The syntax generation module's architecture was presented. Thereby it was argued that knowledge about local dominance relations should be separated from knowledge about linear precedence. Especially for languages with a relatively free word order – like German – one should avoid during incremental verbalization building up unnecessary syntactic paraphrases resulting from ordering variations in the input.

It was demonstrated how the three expansion operations that are needed during incremental generation – known from the literature as upward expansion, insertion, downward expansion – are realized for a lexicalized TAG with unification.

I argued that in contrast to the level of descriptions, where a verb directs the creation of an elementary structure including all its arguments, processing should consider parts of those structures to ensure incremental processing. The predicate called 'local completeness' for the lexical head can be used to enforce processing of parts. In contrast to De Smedt and Kempen, I argue that the linguistic module should demand missing information from the conceptualizer: Firstly, to ensure a fast utterance (instead of waiting or using defaults immediately), secondly, to ensure grammatically well–formed utterances.

Finally I presented a preliminary idea to handle phenomena caused by conceptual addition of input elements by using auxiliary trees as modifying filter for propagated information. This was possible because of our nonmonotonic unification operation (UTAG).

Implications of Tree Adjoining Grammar for Natural Language Generation

David McDonald, Mary Meteer
Content Technologies, Inc.
14 Brantwood Road
Arlington, MA 02174/8004, USA

Modelling a cognitive process such as the production of utterances is in large part a problem of design. There is no direct evidence to which one can appeal for the representation of grammar in the mind or the mechanisms for selecting what is to be said or how it is to be organized. Instead one must adopt guiding frameworks and employ indirect evidence, especially aesthetic principles, from other disciplines. This paper considered such a case: adopting the TAG formalism for formulating grammars, as developed in mathematical and theoretical linguistics, to the processing model implemented in the natural language generation system, Mumble.

In our work, the TAG formalism is taken as given, and thus provides a means of reducing the degrees of design freedom within the rest of the generation process to just those possibilities that are consistent with TAGs. The greatest impact of the formalism comes from the fact that it provides only a single packaging for all linguistic information, the elementary tree. This means that the text planner's

decisions can be only which trees to select; it cannot get access to smaller units of linguistic structure, and larger ones can only be formed by the combination of entire trees.

This primary fact can be leveraged for corollaries applying to incremental generation, to criteria by which trees are grouped into families, and to the relationship between the content of individual trees and the speaker's conceptual representation. One can also couple the properties of TAG with a particular approach to generation, for example message-directed processing. We can then project back from this to draw conclusions about how information may be structured in the mind, and then again forward to suggest how trees are composed through adjunction and substitution.

Features in a Lexicalized TAG for English

Sharon Cote
Department of Linguistics
University of Pennsylvania
618 Williams HALL
Philadelphia, PA 19104, USA
cote@linc.cis.upenn.edu

This talk is an overview of the current state of the English LTAG and a discussion of some issues that have arisen in designing features for this grammar.

I explore the possibility that the only types of features required in a LTAG are those that specify the properties of lexical items (Lexical feature Principal). These features are characterized as either **Anchor Features**, which are bottom features, or **"Argument" Features** which are top features. **Structural Features** would be used only to carry information that is relevant above the level of sentence grammar.

I also consider the special nature of the category feature and suggest that auxiliary trees do not necessarily have to be defined as trees with a root and foot node of the same, fully pre-specified category.

A TAG analysis of the Third construction in German

Anthony Kroch, Beatrice Santorini, Aravind Joshi
Department of Computer and Information Science
R-555 Moore School
University of Philadelphia
220 South Street 33rd Street
Philadelphia, PA 19104-6389, USA
kroch or beatrice or joshi@linc.cis.upenn.edu

In this paper, we consider the so-called third construction in German, illustrated in (1):

(1) Der Lehrer hat das Theorem versucht zu beweisen.
the teacher has the theorem attempted to prove

'the teacher attempted to prove the theorem'

While syntactically distinct from the well-known West Germanic verb raising construction, the third construction is similar to it in that it exhibits cross-serial dependencies and is hence not context-free. Recently, Joshi 1990 has proposed an analysis of the parsing of verb sequences using extended push down automata (EPDA) which presents a formal model of the differential psycholinguistic processing complexity of cross-serial vs. nested dependencies, as reported by Bach, Brown and Marslen-Wilson 1989. Interestingly, den Besten and Rutten 1989 have proposed an analysis of the third construction (in Dutch) according to which it reflects two independently motivated syntactic processes: long distance scrambling (leftward movement) and extraposition. Joshi's EPDA for cross-serial dependencies corresponds directly to den Besten and Rutten's grammer of the third construction - a result that is striking since the motivation for Joshi's EPDA lies in the explanation of processing complexity, while the motivation for den Besten and Rutten's analysis lies in distributional generalizations of the conventional linguistic type. We presented two TAG analysis of the third construction. The first analysis requires only one-part trees; however, it has certain linguistic drawbacks - in particular, it requires relaxing the important constraint that traces be c-commanded by their antecedents, and it is unable to derive instances of pure long-distance scrambling, which German (like many verb-final languages) allows. As a result, we present an analysis of the third construction using multicomponent adjunction which does not have the above-mentioned drawbacks. Even this analysis, however, is unable to derive certain instances of long-distance scrambling (in particular, one in which a long-distance scrambled constituent interrupts two matrix arguments). We propose a multicomponent adjunction analysis which relies crucially on introducing arguments of the verb on a par with adjuncts. We conclude by presenting linguistic evidence based on facts concerning weak crossover and parasitic gaps, which support the last multicomponent adjunction analysis presented.

French and english determiners: Interaction of morphology, syntax and semantics in Lexicalized Tree Adjoining Grammars

Anne Abeillé
LADL & UFRL
University of Paris 7 - Jussieu
F-75005 Paris, France
abeille@franz.ibp.fr

Tree adjoining grammars have proved quite relevant for handling numerous linguistic phenomena, for example unbounded dependencies (A. Kroch and A. Joshi 1985, A. Kroch 1987), light-verb constructions (A. Abeillé 1988) and idioms (A. Abeillé and Y. Schabes 1989, 1990). Two sizable grammars have been written for French and English (A. Abeillé 1988, A. Abeillé, K. Bishop, S. Cote, Y. Schabes

1990). They result from common work at University of Pennsylvania and Université de Paris 7-Jussieu. We present recent work which has been done focusing on the interactions between morphology, syntax and semantics. A case study of French and English determiners, involving such interactions, is also presented.

Interaction of Morphology and Syntax A lexicalized TAG grammar is organized into two lexicons: a "morphological" one which lists for all the lemmas the corresponding inflected forms (with the associated morphological features) and a "syntactic" one which lists for all autonomous lexical items the corresponding elementary tree structures they head (these elementary trees are usually gathered into Tree Families, which express the possible syntactic variation of a given predicate argument structure).A given lemma has in the syntactic lexicon as many entries as it has different subcategorization frames, associated to different meanings. As first shown by M. Gross 1975, it is thus possible to perform a lot of semantic disambiguation on syntactic grounds. For example, 'voler' means either 'to fly' or 'to steal': the first one is intransitive, the second one transitive. They thus have different entries in the syntactic lexicon. Adjectives and nouns are disambiguated in the same way. Such disambiguations are usefull in the perspective of machine translation (Abeillé, Schabes, Joshi 1990). Notice that the subcategorization frame (i.e. the syntactic category of the predicate) may interfere with some of its morphological properties. In French, as noticed by M. Gross 1989. when a verb can be both transitive and intransitive (with different meanings) it will lack inflected past participle forms in its intransitive use, since the past participle usually agrees with the preposed object in French. Thus the set of inflected forms corresponding to the intransitive VOLER (fly) is smaller than that of the transitive VOLER (steal). This is done here by allowing morphological features in the entries of the syntactic lexicon.

Syntactic Flexibility and Semantic Non-Compositionality Lexicalized Tags, which associate sets of elementary trees to lexical items, define linguistic units of extended domain of locality that have both syntactic and semantic relevance. Such a formalism offers very natural representations for constituents that follow regular syntactic composition rules, and may exhibit internal discontinuities, but lack semantic compositionality (Abeillé and Schabes 1990). Examples, for French and English, are idioms, light verb constructions, and verb particle combinations. We require that all entries be syntactically and semantically autonomous in the syntactic lexicon. We thus allow entries in the syntactic lexicon to be comprised of several lexical items (or lemmas). This is made possible by the extended domain of locality offered by TAGs. When some word is not autonomous semantically (as most idiom chunks, English particles or case marking prepositions) then it cannot be an autonomous entry by itself and is considered part of the entry of the expression it belongs to. Our 'syntactic' lexicons are in fact semantico-syntactic ones.

Case study: French and English determiners As an application of the principles relating morphology, syntax and semantics described above, we suggest a new treatment of determiners in TAGs which is based on the study of a few hundred French and English determiners which lead to the following observations:

- determiners are a more open class than is usually thought
- complex and frozen determiners ("a bunch of", "three liters of"..) have to be taken into account
- an NP may include more than one determiner.

Sofar, determiners have been considered substituted into NP initial trees headed by nouns (or compounds). We propose instead to have the determiners adjoined onto the root node N of the noun and its domain of locality thus extended. The difference now between NP and N is simply a feature <Det>=+ (corresponding roughly to NP) and <Det>=- (corresponding roughly to N). In the English morphological lexicon, plural forms ('flowers') are not marked for <Det>, since determiners are optional for them, whereas singular ones ('flower') are usually marked <Det>=- (at their bottom). In both lexicons names are marked <Det> =+. This is an example of a syntactic feature present in the morphological lexicons. If all N-initial trees are marked in advance <Det>=+ at their top, an obligatory adjunction constraint will result for forms such as 'flower'.The main advantages of this representation are as follows:

1. Complex determiners (such as 'a bunch of' or 'the majority of') can be handled in the same way as simple ones ('the', 'a') while being assigned an internal structure which is that of regular NPs ('a whole bunch of...). It is required that the noun be dominated by a PP node with determiners such as "all of N" or "a bunch of N" (as shown by the accusative a bunch of them all of them). Adjunction is the only way to achieve this result since the N node can also be an interior node (as in idioms with frozen object but free determiners).

2. Determiners can be made optional without assigning two different elementary trees to the head noun: I like butter/this butter; flowers/these flowers. In English, singular and plural forms of nouns will thus have the same structure (although different features).

3. Combinations of determiners (such as 'la plupart de ce type de gens') are easier to represent, especially the fact that some features (number, definiteness) of the whole NP may change depending on which determiner is finally adjoined.

4. Numerals and some other modifiers can be represented with only one structure yielding a phrase which can behave both as N or NP, for example 'three men' / 'the three men' or 'Je n'ai jamais lu semblable aventure'/ 'une semblable aventure'.

All nouns have only one maximal projection (elementary tree) whether they occur in an N or an NP context. In French, the top <det>=+ feature on the noun is dependent on the context: 'voir *sorcifflre' / 'une sorcifflre' vs. 'changer quelqu'un en sorcifflre' / '*une sorcifflre' (see: 'a witch' / 'change someone into a witch').

Syntactic properties of the wole NP can more easily be made dependent on the lexical value of the determiner. We thus present a feature system for distinguishing determiners on the basis of the syntactic properties of the NP they introduce (extractable or not, topicalizable or not). These features also serve to rule out some combinations of determiners.

Japanese Tree Adjoining Grammar and its Application to On-Line Help System NeoAssist

Kuniaki Uehara
Department of Systems Engineering
Faculty of Engineering
Kobe University
Rokkodai-cho, Nada
Kobe 657, Japan uehara@gradient.scitex.kobe-u.ac.jp

ne of the greatest obstacles faced when attempting to develop a text generation anism for a language like Japanese is the unpredictability caused by the rela- free word order and by the case assignment. It is, thus, necessary to develop rammatical formalism which gives an account of some linguistic phenomena iar to Japanese. This paper proposes the Japanese Tree Adjoining Grammar G for short) which has more powerful mechanism for treating the word order ion than that of the original Tree Adjoining Grammar (TAG for short).
rst of all, by using a set of linear precedence statements, we can define word variation in Japanese, there still remains a linguistic phenomenon which can e explained in the framework of TAG. For example, embedded sentences in ese do not normally carry any sign (i.e. *which*, *where* in English) to mark beginning. As a result, the beginning of a deeply embedded sentence can look y like the beginning of a simple top-level sentence. Furthermore, no other cannot be inserted between the embedded sentence and the antecedent. In to explain this linguistic phenomenon in JTAG, we will introduce the new precedence relationship '$\leq$'. The new relationship $x \leq y$ (x strongly precedes ntroduced so as to prohibit some words or phrases from moving into a phrase ase.
cond, Japanese postnominal suffixes, by themselves, do not always provide necessary information for case assignment. In other words, the postnominal of the same deep case interpretation changes depending on the aspectual es (stative, transitive, process, completive, momentary), voice, or volition. In to solve the problem of case assignment, we will extend the notion of an

elementary tree by introducing a set of feature-value pairs, so that JTAG is able to express control and feature constraints. Control constraint is used to deal with Equi-NP Deletion and Passive transformation. Feature constraint is used to constrain a feature of a node whose value is expected to be defined by a separate specification.

As a result, JTAG can formally deal with some linguistic phenomena often found in a typical Japanese text: passivization, topicalization, relative clauses, embedded sentences, etc. The framework of JTAG is now used as a text generation mechanism in an intelligent on-line help system NeoAssist. However, JTAG is still in its evolving stage, and it needs further refinement. For example, we could include in the framework of JTAG some semantic constraints such as 'a sentence can be transformed into the passive one, if the subject of the sentence is volitional'. Such a semantic constraint could be specified by using feature constraints described above. We have not yet explored what kind of features and their values should be prepared to express semantic constraints. We could also augment JTAG with the mechanism to deal with given and new information. This problem is closely related with the context of a sentence, we must develop the mechanism along with the selection mechanism of auxiliary trees. Such refinements and improvements will continue.

Coordination in TAG in the manner of CCG (Combinatory Category Grammars) : Fixed vs Flexible Phrase Structure

Aravind Joshi
Department of Computer and Information Science
R-555 Moore School
University of Philadelphia
220 South Street 33rd Street
Philadelphia, PA 19104-6389, USA
joshi@linc.cis.upenn.edu

So far there is no good account of the coordination phenomena in the natural language in the framework of TAG. The best account of coordination so far is provided by CCG. Lexicalized TAGs are very close to CCG except for the fact (and a very crucial fact) that the elementary trees of TAG (lexicalized TAG) do not have a curried representation. The categories in CCG are represented as curried functions. In my talk at the Dagstuhl workshop on TAG, I tried to show that this crucial difference can be exploited for constructing a CCG-like account for coordination in TAGs without - giving up the phrase structure defined in the set - of elementary trees. In CCG there is no fixed phrase structure, almost any contiguous sequence of lexical items (words) can be grouped together as a constituent, thus creating groupings which ordinarily will not be considered as constituents. There are a number of questions about my approach that need to be settled, in particular, it is necessary to investigate the power of the resulting system and to make save that no additional complexity is added while trying to get rid of the multiplicity of constituents in CCG.Interaction with the participants promised me a lot of new ideas about how to settle these questions.

Bisher erschienene und geplante Titel:

W. Gentzsch, W.J. Paul (editors):
Architecture and Perfonnance, Dagstuhl-Seminar-Report; 1, 18.-20.6.1990; (9025)

K. Harbusch, W. Wahlster (editors):
Tree Adjoining Grammars, 1st. International Worshop on TAGs: Fonnal Theory and Application, Dagstuhl-Seminar-Report; 2, 15.-17.8.1990 (9033)

Ch. Hankin, R. Wilhelm (editors):
Functional Languages: Optimization for Parallelism, Dagstuhl-Seminar-Report; 3, 3.-7.9.1990 (9036)

H. Alt, E. Welzl (editors):
Algorithmic Geometry, Dagstuhl-Seminar-Report; 4, 8.-12.10.1990 (9041)

J. Berstel , J.E. Pin, W. Thomas (editors):
Automata Theory and Applications in Logic and Complexity, Dagstuhl-Seminar-Report; 5, 14.-18.1.1991 (9103)

B. Becker, Ch. Meinel (editors):
Entwerfen, Prüfen, Testen, Dagstuhl-Seminar-Report; 6, 18.-22.2.1991 (9108)

J. P. Finance, S. Jähnichen, J. Loeckx, M. Wirsing (editors):
Logical Theory for Program Construction, Dagstuhl-Seminar-Report; 7, 25.2.-1.3.1991 (9109)

E. W. Mayr, F. Meyer auf der Heide (editors):
Parallel and Distributed Algorithms, Dagstuhl-Seminar-Report; 8, 4.-8.3.1991 (9110)

M. Broy, P. Deussen, E.-R. Olderog, W.P. de Roever (editors):
Concurrent Systems: Semantics, Specification, and Synthesis, Dagstuhl-Seminar-Report; 9, 11.-15.3.1991 (9111)

K. Apt, K. Indermark, M. Rodriguez-Artalejo (editors):
Integration of Functional and Logic Programming, Dagstuhl-Seminar-Report; 10, 18.-22.3.1991 (9112)

E. Novak, J. Traub, H. Wozniakowski (editors):
Algorithms and Complexity for Continuous Problems, Dagstuhl-Seminar-Report; 11, 15-19.4.1991 (9116)

B. Nebel, C. Peltason, K. v. Luck (editors):
Terminological Logics, Dagstuhl-Seminar-Report; 12, 6.5.-18.5.1991 (9119)

R. Giegerich, S. Graham (editors):
Code Generation - Concepts, Tools, Techniques, Dagstuhl-Seminar-Report; 13, 20.-24.5.1991 (9121)

M. Karpinski, M. Luby, U. Vazirani (editors):
Randomized Algorithms, Dagstuhl-Seminar-Report; 14, 10.-14.6.1991 (9124)

J. Ch. Freytag, D. Maier, G. Vossen (editors):
Query Processing in Object-Oriented, Complex-Object and Nested Relation Databases, Dagstuhl-Seminar-Report; 15, 17.-21.6.1991 (9125)

M. Droste, Y. Gurevich (editors):
Semantics of Programming Languages and Model Theory, Dagstuhl-Seminar-Report; 16, 24.-28.6.1991 (9126)

G. Farin, H. Hagen, H. Noltemeier (editors):
Geometric Modelling, Dagstuhl-Seminar-Report; 17, 1.-5.7.1991 (9127)

A. Karshmer, J. Nehmer (editors):
Operating Systems of the 1990s, Dagstuhl-Seminar-Report; 18, 8.-12.7.1991 (9128)

H. Hagen, H. Müller, G.M. Nielson (editors):
Scientific Visualization, Dagstuhl-Seminar-Report; 19, 26.8.-30.8.91 (9135)

T. Lengauer, R. Möhring, B. Preas (editors):
Theory and Practice of Physical Design of VLSI Systems, Dagstuhl-Seminar-Report; 20, 2.9.-6.9.91 (9136)

F. Bancilhon, P. Lockemann, D. Tsichritzis (editors):
Directions of Future Database Research, Dagstuhl-Seminar-Report; 21, 9.9.-13.9.91 (9137)

H. Alt , B. Chazelle, E. Welzl (editors):
Computational Geometry, Dagstuhl-Seminar-Report; 22, 07.10.-11.10.91 (9141)

F.J. Brandenburg , J. Berstel, D. Wotschke (editors):
Trends and Applications in Formal Language Theory, Dagstuhl-Seminar-Report; 23,14.10.-18.10.91 (9142)

H. Comon , H. Ganzinger, C. Kirchner, H. Kirchner, J.-L. Lassez , G. Smolka (editors):
Theorem Proving and Logic Programming with Constraints, Dagstuhl-Seminar-Report; 24, 21.10.-25.10.91 (9143)

H. Noltemeier, T. Ottmann, D. Wood (editors):
Data Structures, Dagstuhl-Seminar-Report; 25, 4.11.-8.11.91 (9145)

A. Dress, M. Karpinski, M. Singer(editors):
Efficient Interpolation Algorithms, Dagstuhl-Seminar-Report; 26, 2.-6.12.91 (9149)

B. Buchberger, J. Davenport, F. Schwarz (editors):
Algorithms of Computeralgebra, Dagstuhl-Seminar-Report; 27, 16..-20.12.91 (9151)