Final Report for the ESF Science Meeting

# Workshop on Tree Automata

June 7–9, 2006, Bonn, Germany

AutoMathA Project

**Mikołaj Bojańczyk**
Warsaw University, Poland
bojan@mimuw.edu.pl

**Christof Löding**
RWTH Aachen, Germany
loeding@informatik.rwth-aachen.de

**Sophie Tison**
Université des Sciences et Technologies de Lille, France
tison@lifl.fr

# 1 Summary

As planned, the "Workshop on Tree Automata" was held from June 7–9 in the "Bonn-Aachen International Center for Information Technology" in Bonn, Germany. The total number of participants was 31, consisting of all the 14 invited persons listed in the application, the 3 organizers, and young researchers from the groups of the invited persons.
The talks (5 tutorials of one hour, 12 talks of half an hour, and one open problems session of one hour) and discussions focused on the use of tree-automata techniques for handling semi-structured documents. XML documents are naturally modeled by trees and, in this context, tree automata are used for

specifying and validating a document: an automaton can be viewed as an abstract model for describing a language, and/or as an algorithm for type-checking a document.

querying: node selecting – like in XPath – is a relevant task in information extraction and automata are a good tool for performing this task.

transforming: tree-transducers are expressive formalisms for reasoning about tree transformations, like XSLT-like ones.

For all these aspects, links with logics are fundamental and have been systematically exhibited.

# 2 Description of scientific content

## 2.1 From ranked trees to unranked ones

A priori, XML documents are modeled as unranked trees – the number of successors of a node is unbounded – whereas classical tree-automata are defined for ranked trees. Of course, encoding an unranked tree by a ranked one can be done easily in several ways; another approach is to define tree-automata handling this directly with two kinds of recursion – vertical and horizontal – e.g. representing horizontal recursion by a finite word automaton or finite regular expression.
Christof Loeding surveyed these different approaches, exhibiting the consequences of the choice of the encoding for complexity. In line with this talk, Wim Martens compared these different approaches w.r.t. the minimization problem, showing that the step-wise automata yield the most succinct representation.
Frank Neven discussed also this point by comparing several schema languages for XML including DTDs, XML Schema, and Relax NG. He showed

how modeling them in terms of automata and grammars helps in understanding properties of those languages. The challenge here is to exhibit classes which correspond to most real-life documents and which have good algorithmic properties.

Igor Walukiewicz presented a new algebraic framework for recognizing unranked tree languages. He argued that – in certain respects – unranked trees are actually simpler to describe algebraically than ranked ones. The new framework allows characterizations for logics like EF, FOL, PDL, Chain Logic, CTL*.

**Future directions.** Probably the most important future direction in this field is increasing interaction with the applied XML community. Clearly, this interaction can be beneficial for both sides. In particular, the talk given by Frank Neven shows that the XML community has developed classes of tree languages that were not known to people working in formal language and automata theory. We need to establish how these classes relate to our established formalisms.

There are also many interesting theoretical problems for unranked trees that cannot be solved by encoding into binary trees. This happens when one considers tree automata that can test if the sibling subtrees are equal: over ranked trees, emptiness for such automata is decidable (Bogaert and Tison), over unranked trees it is not. It would be a good idea to identify restrictions on the equality tests that would give decidable emptiness for unranked trees. Ongoing work by Christof Loeding and Wong Karianto is promising; also worth noting is the connection with languages over infinite alphabets that were surveyed by Anca Muscholl. Generally, equality tests are important for XML reasoning, since they are one of the principal ways of comparing data.

## 2.2 Ordered or not?

For semi-structured data, the order can be irrelevant, particularly from a database point of view. This aspect has been discussed by Jean-Marc Talbot and, from a security protocols point of view by Hubert Comon-Lundh. The unified framework of tree automata with arithmetical constraints has been presented; the link with logics has been exhibited, as instantiating this framework by well-known classes of arithmetical constraints leads to the notions of PMSO-definable, CMSO-definable, MSO-definable set of trees. The link with fragments of TQL logic has also been exhibited. Here tree automata can be seen as an algorithmic toolbox providing algorithms for deciding membership or satisfiability. The approach based on tree automata on congruence classes has been discussed by Hubert Comon-Lundh.

## 2.3 Data tree languages

Most work in the workshop considered trees labeled by a finite fixed alphabet. Certain problems related to XML documents cannot be modeled this way: it may be necessary to describe a tree whose nodes are labeled by an infinite alphabet, in order to take into account arbitrary data such as text (leaf nodes), attribute values, references. Automata and logics with infinite alphabets have been surveyed by Anca Muscholl. Due to the infinite alphabet, interesting logics and automata classes are rarely decidable. The talk emphasized recent work exhibiting what restrictions lead to decidable logics with data; most notably first-order logic with two variables.

**Future directions.** Data trees are a new object and almost any work here is future work. Currently, the natural open questions are: what is the appropriate automaton model for data trees, and what is the type of tree navigation that can be used in a logic with decidable emptiness.

## 2.4 Towards a sequential model of tree automata

Usual tree models are branching – evaluation of subtrees can be done in parallel– whereas validating and visiting document is "naturally" modeled as a sequential process. Defining a sequential model for tree automata is not new: tree-walking automata have been defined in early 1970's by Aho and Ullman, but semi-structured data has given rise to new interest in such models. Mikolaj Bojanczyk surveyed results on tree-walking automata; he emphasized how long-standing open questions concerning their expressive power have been recently answered. Pebble automata – tree-walking automata with some pebbles that can be used to mark some nodes – have been also discussed. Thomas Schwentick showed that adding pebbles creates an infinite hierarchy, and this hierarchy does not cover all recognizable languages.

**Future directions.** Some of the most interesting open questions in this field concern complementation. Perhaps the most important one is: are languages recognized by tree-walking automata closed under complementation? Moreover, similar questions can be posed for pebble automata. This is an important area of future research, especially since it is connected with the expressive power of logics with transitive closure.

## 2.5 Querying and transforming

Regular queries are a class of queries which is specially relevant for XML query languages. Alex Berlea presented an algorithm based on tree automata which allows answering queries while scanning the input.
Tree transducers are an expressive formalism for modeling tree transformations. An important problem for tree transducers is the type-checking

problem: given a tree transducer and two tree languages (input and output), decide if every tree in the input language is transformed into a tree in the output language. Type-checking algorithms can then adopt two points of view: a backward one – compute the inverse image of the output type and check if it contains the input type – or forward one – compute the direct image of the input type and check if it is included in the output one. Helmut Seidl presented new type checking algorithms using the forward approach for stay macro-tree transducers – which are useful for modeling XSLT transformations – and generalized his approach to stay macro forest transducers. Thomas Wilke described a "backward approach" for a tree automaton and transducer model which can handle with infinite alphabets.

**Future directions.** A whole session – by Sebastian Maneth – was dedicated to open problems and future directions in querying and transforming. Future research ranges from basic and long standing theoretical open problems – given a regular tree language $L$ and a tree homomorphism $h$, decide if the image language $h(L)$ is regular – to more practically oriented questions, for instance finding compact representation of trees that still allow efficient querying.

## 2.6 Logics over trees

Links with logics have been systematically exhibited but several talks were devoted solely to these links. Wolfgang Thomas surveyed connections of definability in monadic second-order logic and tree automata. He discussed several subclasses of the regular tree languages induced by fragments of MSO, with an emphasis on chain and antichain logic. Luc Segoufin gave an effective characterization of regular tree-languages which are FO-definable in first-order logic (with the successor relation, but without the descendant order). This recent joint work with M. Benedikt can be extended with modular quantifiers. H. J. Hoogeboom showed why first-order logic with $k$-ary deterministic transitive closure has the same power as two-way $k$-head deterministic automata with a finite set of nested pebbles. In particular, for $k = 1$, this logic corresponds exactly to the pebble automata described by Thomas Schwentick.

**Future directions.** There is a large number of unsolved questions remaining in this field. One of the principal questions is finding an algorithm for the problem: "can a given regular tree language be defined in first-order logic with the descendant relation (FOL)?" This question has attracted considerable attention for more than two decades, although no algorithm has been heretofore presented. The new results on first-order logic with the successor relation, and the algebraic approach presented by Igor Walukiewicz, give new hope for solving this question. Similar questions can be asked regarding chain logic and several temporal logics.

There are also interesting open problems concerning logics with transitive closure. The most interesting ones are related to complementation: is positive transitive closure (where the use of negation is restricted) equally expressive as unrestricted transitive closure? These results are closely related with open problems concerning complementation of sequential automata.

### 2.7 Learning, probabilistic models

How discover a DTD from documents? How to learn a node selecting query from annotated examples? These questions are crucial in information extraction. The first one has been discussed by Frank Neven, who presented an efficient approach for learning DTDs using "single occurrence automata". The second question has been answered for MSO-definable $n$-ary queries by Joachim Niehren, who represented $n$-ary queries by deterministic node-selecting tree transducers: this class can be learned from polynomial and data and allows efficient enumeration of answers. This approach has been successfully implemented in "Squirrel", a tool for wrapper induction in Web information extraction.
Information extraction also motivates considering probabilistic models for trees. Different models have been studied and compared by Rémi Gilleron, who presented Stochastic Tree Automata and Conditional Random fields for unranked terms.

**Future directions.** The Squirrel tool is a very impressive example of what can happen when tree automata theory is applied to the practical problem of generating queries by user-provided examples. The tool is a Firefox extension, which can be easily downloaded from the web and installed by even an unexperienced user. On the one hand, future work should involve more widespread diffusion of this tool – perhaps involving feedback from the users – and, on the other hand, further exploring the scientific questions involved. One direction is finding techniques that deal with the data in a document, and not just its structure (this is closely related to the work on infinite alphabets).

### 2.8 Analysis of protocols

Besides XML applications, use of tree automata techniques for analysis of cryptographic protocols have been emphasized by Hubert Comon-Lundh and Thomas Wilke.

**Future directions.** A future direction mentioned by Thomas Wilke is establishing the complexity of the iterated preimage problem of Tree Automata with Anonymous Constants; this problem is used in analyzing cryptographic protocols.

## 3 Impact on future directions

We summarize the impact on future directions separately for each subject; this is done in the previous section, along with each subsection.
It should be mentioned that there has been a lot of positive feedback, and one of the participants (Hubert Comon-Lundh) has shown interest in organizing a similar event next year.
Furthermore, it is planned to use the tutorial “Automata for Unranked Trees” as a basis for a new chapter of the electronic book *Tree Automata Techniques and Applications* (see `www.grappa.univ-lille3.fr/tata`).

## 4 Final program

Wednesday (June 7)

| | |
|---|---|
| 13:00 - 14:15 | Reception and Opening |
| 14:15 - 15:15 | Christof Loeding<br>*Automata for Unranked Trees (tutorial)* |
| 15:15 - 15:45 | Coffee |
| 15:45 - 16:45 | Wofgang Thomas<br>*Fundamentals on Logics over Trees (tutorial)* |
| 17:00 - 18:00 | Luc Segoufin<br>*Regular tree languages definable in FO and FO+MOD*<br>Igor Walukiewicz<br>*Unranked Tree Algebra* |

Thursday (June 8)

| | |
|---|---|
| 9:00 - 10:00 | Frank Neven<br>*Tree Automata and XML (tutorial)* |
| 10:00 - 10:30 | Helmut Seidl<br>*Type-Checking Macro Tree Transducers in Polynomial Time* |
| 10:30 - 11:00 | Coffee |
| 11:00 - 12:00 | Sebastian Maneth<br>*Open Problems Session* |
| 12:00 - 12:30 | Alex Berlea<br>*Online Evaluation of Regular Tree Queries* |
| 14:00 - 15:00 | Anca Muscholl<br>*Logics and automata with infinite alphabets (tutorial)* |
| 15:00 - 15:30 | Coffee |
| 15:30 - 17:00 | Wim Martens<br>*Minimization Problem for Deterministic Unranked Tree Automata*<br>Joachim Niehren<br>*Learning Tree Automata: Squirrel*<br>Rémi Gilleron<br>*On probabilistic models for trees* |
| 19:30 - | Dinner |

Friday (June 9)

| | |
|---|---|
| 9:00 - 10:00 | Mikolaj Bojanczyk |
| | *Tree-Walking Automata (tutorial)* |
| 10:00 - 10:30 | Hendrik Jan Hoogeboom |
| | *Nested Pebbles and Transitive Closure* |
| 10:30 - 11:00 | Coffee |
| 11:00 - 12:00 | Thomas Schwentick |
| | *Pebble Automata* |
| | Thomas Wilke |
| | *Tree Automata and Tree Transducers for Analyzing Cryptographic Protocols* |
| 13:30 - 14:30 | Hubert Comon-Lundh |
| | *Examples of Applications of Tree Automata to Security Protocols* |
| | Jean Marc Talbot |
| | *TQL and Tree Automata (Unordered Unbounded Case)* |
| 14:30 | Closing and coffee |

The program together with electronic slides of most of the talks can be found on `www.mimuw.edu.pl/~automat/bonn/program.html`