Finding Your Way in a Forest: On Different Types of Trees and Their Properties

Igor Walukiewicz

Finding your way in a forest:
on different types of trees and their properties

Igor Walukiewicz[*]

CNRS LaBRI
Université de Bordeaux
351, Cours de la Libération
33405 Talence, France

We see trees in almost any part of computer science. Traditionally, ranked trees, that are nothing else but terms, captured most attention, although exceptions could have been found in graph theory or linguistics [9]. Recently unranked trees are a subject of renewed interest, mainly because of the developement of XML [22]. It is also quite comon nowadays to see trees with infinite paths, especailly in the context of verification. We will omit this aspect, as for the questions we want to discuss finite trees are sufficiently interesting. We prefer instead to make distinction between ordered and unordered trees, i.e., distinguish situations when syblings are ordered or not. Thus we will deal with four types of trees depending on two parameters: ranked/unranked, and ordered/unordered.

There are even more formalisms to describe tree properties than there are tree types. Here, the main reference point for us will be monadic second-order logic which captures recognizable sets of trees. This logic has a binary predicate interpreted as a descendant relation in a tree and a monadic predicate for every possible node label. If models are ordered trees then the logic may also have another binary predicate interpreted as the sybling order. Important logics are obtained by restricting the range of quantification: when quantifying only over elements we obtain first-order logic (FOL), when quantifying over chians (sets where every pair of nodes is in a descendant relation) we get chain logic, finally antichain logic is obtained when quatifiers range over sets where no two elements are in the descendant relation. Appart from these classical logics we have important variants of modal and temporal logics over trees: CTL, CTL*, PDL, the μ-calculus.

Given this multitude of formalisms, the first question one can ask is to compare their expressive power, i.e., establish if all properties expressible in a logic A can be also expressed in a logic B. We know the answers to this kind of questions for the formalism listed above. Figure 1 presents the case

---

of ranked ordered trees. Some of the presented inclusions are nontrivial. For insatance, the inclusions of the MSOL in the μ-calculus [18] or FOL in CTL* [14] show that one can obtain the same expressive power using very different means. For other types of trees the picture is not the same. For example, if we consider ordered unranked trees, all equalities on the left change to strict inclusions, and chain logic becomes incomparable with antichaing logic.
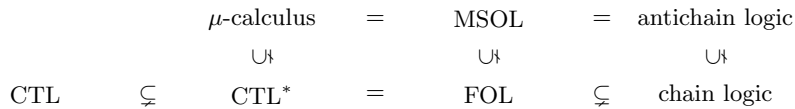
$$
\begin{array}{ccccccc}
 & & \mu\text{-calculus} & = & \text{MSOL} & = & \text{antichain logic} \\
 & & \cup\!\!| & & \cup\!\!| & & \cup\!\!| \\
\text{CTL} & \subsetneq & \text{CTL}^* & = & \text{FOL} & \subsetneq & \text{chain logic}
\end{array}
$$

Fig. 1. Relations between logics over ordered binary trees

While diagrams as that on Figure 1 are important, they are far from giving a complete explanation of the expressive power of the logics in question. Consider the following situation. Separating FOL from MSOL over ordered binary trees is easy. We know that over words FOL cannot express counting modulo properties [20]. For example, "even length" is not first order definable. A short argument shows that the language of trees with the leftmost path of even length is not FOL expressible. This observation though, does not tell much about what properties are expressible in FOL. Take the language of ordered binary trees "the depths of all leaves are even". Somehow surprisingly, it turns out that this language is FOL-expressible [19]. Indeed it is a major open question to find an algorithm deciding if a given regular tree language is experssible in FOL. At present decidable characterisations are known for very few fragments of MSOL [1, 7, 4, 8].

Apart from the mathematical curiosity, there is a growing number of reasons for looking closer at tree formalisms. In the context of XML, the aspect of data (infnite set of labels with some operations on them) is important. It is rather difficult to come with a decidable nontrivial formalism [6, 16]. Understanding well expressive power in the case without data can help substantially.

Another reason is the study of order invariance. A property is order invariant if does not distingush two trees that differ only in the order of syblings. It is natural to ask if such a property can be expressed without referring to this order. Over unranked trees, order invariant properties are exactly those expressible in MSOL extended with counting modulo

quantifiers [11]. The language "the depths of all leaves are even" described above cannot be defined in FOL without a sybling order. So the situation is much less clear for FOL. Curiously enough if we restrict to FOL[succ] where we allow only successor relation in place of descendant relation then all order invariant FOL[succ] properites can be expressed in FOL[succ] [2].

Finally, there is a question of automata and grammars for trees. In the literature one can find many proposals of differen automata on trees. Even for ordered binary trees we have for example several versions of tree-walking automata [12, 5, 13]. For other types of trees the number of variants is even bigger [10]. The similar situation is with regular expressions for trees [21, 15, 17, 3]. Better understanding of logical formalisms is indenspensable to classify and clarify all these notions.

In this talk we will survey some recent results in the field. We will start with a unifying presentation of formalisms discussed above. For this a small detour to algebra will be useful [8]. We will present dependencies between different formalisms, and known decidability results. Some less common questions as order invariance, or existance of a finite base will be also considered. The talk will present joint work with Mikolaj Bojańczyk.

References

1. M. Benedikt and L. Segoufin. Regular languages definable in FO. In *STACS'05*, volume 3404 of *LNCS*, pages 327 – 339, 2005. See the corrected version on the authors web page.
2. M. Benedikt and L. Segoufin. Towards a characterization of order-invariant queries over tame structures. In *CSL'05*, volume 3634 of *LNCS*, pages 276–291, 2005.
3. M. Bojanczyk. Forest expressions. In *CSL'07*, volume 4646 of *LNCS*, pages 146–160, 2007.
4. M. Bojanczyk. Two-way unary temporal logic over trees. In *LICS'07*, pages 121–130, 2007.
5. M. Bojanczyk and T. Colcombet. Tree-walking automata do not recognize all regular languages. In *STOC'05*, pages 234–243. ACM, 2005.
6. M. Bojanczyk, C. David, A. Muscholl, T. Schwentick, and L. Segoufin. Two-variable logic on data trees and XML reasoning. In *PODS'06*, pages 10–19. ACM, 2006.
7. M. Bojanczyk and I. Walukiewicz. Characterizing EF and EX tree logics. *Theoretical Computer Science*, 358(2-3):255–272, 2006.
8. M. Bojanczyk and I. Walukiewicz. Forest algebras. In J. Flum, E. Grädel, and T. Wilke, editors, *Logic and Automata*, volume 2 of *Texts in Logic and Games*, pages 107–132. Amsterdam University Press, 2007.
9. B. Carpenter. *The Logic of Typed Future Structures*. Cambridge University Press, 1992.
10. H. Comon, M. Dauchet, R. Gilleron, F. J. andD. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications, 2002. Available on http://www.grappa.univ-lille3.fr/tata/.

11. B. Courcelle. The monadic second-order logic of graphs V: On closing the gap between definability and recognizability. *Theor. Comput. Sci.*, 80(2):153–202, 1991.
12. J. Engelfriet and H. J. Hoogeboom. Tree–walking pebble automata. In J. K. et al., editor, *Jewels are forever*, pages 72–83. Springer, 1999.
13. J. Engelfriet, H. J. Hoogeboom, and B. Samwel. XML transformation by tree-walking transducers with invisible pebbles. In *PODS'07*, pages 63–72. ACM, 2007.
14. T. Hafer and W. Thomas. Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree. In *14th Internat. Coll. on Automata, Languages and Programming (ICALP'87)*, volume 267 of *LNCS*, pages 269–279, 1987.
15. U. Heuter. First-order properties of trees, star-free expressions, and aperiodicity. In *STACS'88*, volume 294 of *LNCS*, pages 136–148, 1988.
16. M. Jurdzinski and R. Lazic. Alternation-free modal mu-calculus for data trees. In *LICS'07*, pages 131–140. IEEE Computer Society Press, 2007.
17. W. Martens, F. Neven, T. Schwentick, and G. J. Bex. Expressiveness and complexity of XML schema. *ACM Trans. Database Syst.*, 31(3):770–813, 2006.
18. D. Niwiński. Fixed points vs. infinite generation. In *LICS '88*, pages 402–409, 1988.
19. A. Potthoff. First-order logic on finite trees. In *Theory and Practice of Software Development*, volume 915 of *LNCS*, pages 125–139, 1995.
20. M. P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8:190–194, 1965.
21. W. Thomas. Logical aspects in the study of tree languages. In *Colloquium on Trees and Algebra in Programming (ICALP'84)*, pages 31–50, 1984.
22. V. Vianu. A web odyssey: From CODD to XML. In *PODS'01*. ACM, 2001.