The Power of Deterministic Alternating Tree-Walking Automata[1]

Loránd Muzamel

The concept of a *tree-walking automaton (twa)* was introduced in [1] for modeling the syntax-directed translation from strings to strings. Recently twa are rather used in XML theory as sequential finite state recognizers for tree languages. A twa A, obeying its state-behaviour, walks on the edges of the input tree s and accepts s if the (only) accepting state q_{yes} is accessed. Every tree language recognized by a twa is regular, although it was an open problem for more than 30 years whether twa can be determinized or whether twa can recognize all regular tree languages. The answer for these two questions were provided in [2] and [3] saying that (1) twa cannot be determinized and (2) twa do not recognize all regular tree languages. Hence $dTWA \subset TWA \subset REG$, where $dTWA$ and TWA denote the tree language classes recognized by deterministic twa and twa, respectively, and REG is the class of regular tree languages.

The concept of alternation for various kinds of tree automata was introduced in [7] as a natural extension of nondeterminism. Alternation for twa is considered in [6] and in [4] (without formal definition in the later paper), although they define semantically different computation models. The tree language classes recognized by the two models agree in the nondeterministic case, but do not agree in the deterministic one.

We give a formal definition of the *alternating tree-walking automaton (atwa)* of [4]. A computation of an atwa A on an input tree s starts in the initial state with the reading head at the root node of s. Depending on the applicable rules it generates new parallel computations (such that each has its own copy of s with the current position of the reading head). A accepts s if all the computations spawned from the initial configuration terminate in the accepting state q_{yes}. We denote the tree language class recognized by a (deterministic atwa) atwa by ($dATWA$) $ATWA$. One can prove that atwa recognize exactly the class of regular tree languages, hence $ATWA = REG$, however, it is still an open problem whether the inclusion $dATWA \subseteq REG$ is strict or not.

Roughly speaking, an atwa A is circular if there is an input tree s such that one of the computations of A on s gets into an infinite loop. Otherwise A is noncircular. We denote the class of tree languages recognized by noncircular deterministic twa (atwa) by $dTWA_{nc}$ ($dATWA_{nc}$).

We investigate the recognizing power of deterministic noncircular atwa and prove that they are more powerful than deterministic twa, but less powerful than atwa. Thus we can write the strict inclusions $dTWA \subset dATWA_{nc} \subset ATWA$. Since $dATWA_{nc} \subseteq dATWA$, we also obtain that $dTWA \subset dATWA$.

Finally we raise an open question. In [5] it was proved that circularity does not give extra power for deterministic twa, i.e. $dTWA = dTWA_{nc}$. This result suggests that possibly $dATWA_{nc} = dATWA$. If it does, then the strict inclusions $dTWA \subset dATWA \subset ATWA$ holds.

References

[1] A.V. Aho and J.D. Ullman. Translations on a context–free grammar. *Inform. Control*, 19:439–475, 1971.

[2] M. Bojanczyk and T. Colcombet. Tree-walking automata cannot be determinized. In *Proceedings of the 31st International Conference on Automata, Languages, and Programming*, pages 246–256. Springer-Verlag, 2004.

[3] M. Bojanczyk and T. Colcombet. Tree-walking automata do not recognize all regular languages. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 234–243, New York, NY, USA, 2005. ACM Press.

[1]Research of the author was partially supported by German Research Foundation (DFG) under GrantGK 334/3 during his stay in the period February-April 2005 at TU Dresden

[4] J. Engelfriet and H.J. Hoogeboom. Automata with nested pebbles capture first-order logic with transitive closure. Technical Report 05-02, Leiden University, The Netherlands, April 2005.

[5] A. Muscholl, M. Samualides, and L. Segoufin. Complementing deterministic tree-walking automata. *To appear in IPL*, 2005.

[6] T. Milo, D. Suciu, and V. Vianu. Typechecking for XML transformers. *J. of Comp. Syst. Sci.*, 66:66–97, 2003.

[7] G. Slutzki. Alternating tree automata. *Lect. Notes Comput. Sci.,CAAP*, 159:391–404, 1983.