An automaton model for forest algebras

Antoine Delignat-Lavaud

Department of Computer Science, Boston College
Advisor: Howard Straubing

June 1 - August 15, 2010

Context and state of the art

Regular tree languages and finite machines on trees have been studied for over half a century and the field has gained a lot of popularity in the past decade because of new applications, like XML schema languages. Although the theories of regular languages and regular tree languages are very similar, the case of trees is missing an important feature: the algebraic theory of recognizability. Since Schützenberger and McNaughton showed that star-freeness is characterized by aperiodicity of the syntactic monoid and FO-definability, semigroup theory has played a central role in the study of regular languages, epitomized by Eilenberg's theorem that states that there is a bijection between varieties of monoids and varieties of regular languages.

There have been several attemps to find a suitable algebraic structure to use as a syntactic object for regular tree languages, but none of them has led to results as significant as in the word case. In 2007, Walukiewicz and Bojańczyk proposed a new candidate for forests (unranked trees minus the root) called *forest algebras* and it was used to obtain a characterization of languages definable in EF, and some necessary conditions for more expressive logics like CTL.

The studied problem

The first goal of this internship was to propose an automaton model for forest algebras, i.e. such that the transition algebra of the automaton would be a forest algebra recognizing the same language as the automaton. The second goal was to implement this automaton model in a tool that could be used to test the properties of the syntactic forest algebra of a given language.

Our contribution

We propose a model of forest automata and show that there is a unique (up to isomorphism) minimal automaton for a given forest language and its transition forest algebra is the syntactic forest algebra of the language. We also show that for an automaton $\mathcal{A}$, there is an algorithm that computes the minimal automaton in $\mathcal{O}(|\mathcal{A}| \log |\mathcal{A}|)$. To our knowledge, this is the most efficient minimization algorithm for forest automata. Furthermore, it can be easily adapted to other models of automata on unranked trees. We also show how to determinize the non-deterministic variant of this automaton model without creating all the possible subset states of the subset construction, and how to trim an automaton of its non-reachable states.

All those features were implemented in a GAP package that can also draw forest automata and representations of the horizontal monoid with the action of the letters and insertions. It is also able to draw the Cayley graph and Green's relation of the two monoids of a forest algebra, and test membership to some varieties.

We also obtain two side results on forest algebras: a bound on the size of the vertical monoid and the fact that there is a canonical representation of a syntactic forest algebra as a subset of a transformation monoid such that the action is function application.

Relevance of our results

Although our automaton model was designed for theoretical purposes, it turned out to be very efficient in practice. One of the important difference between unranked trees and forests is the fact that you must accept trees based on the vertical state of the automaton, forcing a dinstinction between the horizontal automaton of each letter. When accepting by horizontal state, it is necessary to accept forests, otherwise the syntactic object is not unique.

Because of this difference, our model is simpler on the horizontal level by a factor of at least $|A|$, a very significant fact from a computational point of view because the number of horizontal states is in substance the dimension of the transformation semigroup where the horizontal monoid lives. Furthermore, minimization becomes a much more natural operation: we only discovered the loglinear algorithm we used in the implementation of our GAP package beat all the published algorithms when we compared our model with hedge automata.

Conclusion and further research

The results of this research are only a starting point for an algebraic theory of regular forest languages. Our GAP package can be useful to test algebraic properties of the syntactic forest algebra on examples, but for now we do not know exactly which properties to look for in a forest algebra. Of course, we can use our experience in the word case to test properties of the horizontal and vertical monoid, but we already know that it is not sufficient even in the case of FO-definability. A recent result by Segoufin and Benedikt states that a tree language is FO-definable if and only if it is vertically aperiodic and closed under an operation called guarded swaps. It would be a big step forward to give a fully algebraic characterization.