Conference track «Mathematical logic, algebra and number theory»

On the equivalence checking problem for deterministic top-down tree automata

Научный руководитель – Захаров Владимир Анатольевич

Deng Zhibo
Student (master)
Университет МГУ-ППИ в Шэньчжэне, Шэньчжэнь, Китай
E-mail: dzb2291273643@gmail.com

A *top-down finite tree automaton* [1] (FTA) is a tuple $\mathcal{A} = (Q, \mathcal{F}, \Delta)$, where Q is a finite set of control states, $\mathcal{F}$ is a finite nonempty set of functional symbols, and Δ is a set of transition rules of the following type:

$$\delta(q, f) : q(f(x_1, \ldots, x_n)) \to f(q_1(x_1), \ldots, q_n(x_n)) \tag{1}$$

where $f \in \mathcal{F}_n$, and $q, q_1, \ldots, q_n \in Q$. An FTA $\mathcal{A}$ is called *deterministic* (DFTA) if there are no two rules in Δ with the same left-hand side.

FTAs operate on terms — finite trees whose nodes are marked with symbols in $\mathcal{F}$. An FTA $\mathcal{A}$ *accepts* a term t in a state $q \in Q$ when it starts its run at the root of the tree in a state q and finally applies terminating rules $q(c)$ *toc*, where $c \in \mathcal{F}_0$ at all leaves of t. A set of all terms accepted by $\mathcal{A}$ in a state q is a tree language $L(\mathcal{A}, q)$. We study the equivalence checking problem for deterministic FTAs: given a DFTA $\mathcal{A}$ and a pair of states q', q'' check if $L(\mathcal{A}, q') = L(\mathcal{A}, q'')$.

Our algorithm checks the equivalence of two states q' and q'' in $\mathcal{A}$ by constructing a system of equations modeling the equality of tree languages $L(\mathcal{A}, q') = L(\mathcal{A}, q'')$ [2,3]. To this end we associate a variable X_q with each state $q \in Q$ of $\mathcal{A}$, and a term $t_\delta = f(X_{q_1}, \ldots, X_{q_k})$ with every transition rule $\delta(q, f)$ of the form (1). The system of equations $\mathcal{E}_0$ required for our purpose is as follows:

$$\mathcal{E}_0 = \{X_q = \sum_{\delta \in \Delta(\mathcal{A}, q)} t_\delta : q \in Q\} \cup \{X_{q'} = X_{q''}\}.$$

The algorithm iteratively solves the system of equations $\mathcal{E}_0$; each iteration i consists of four steps:

1) Termination: if $\mathcal{E}_i$ has no equations of the form $X_{q'} = X_{q''}$ then the algorithm terminates.
2) Substitution: if $\mathcal{E}_i$ has an equation of the form $X_{q'} = X_{q''}$ then remove the equation and in the case $q' \neq q''$ replace all occurrences of $X_{q'}$ with $X_{q''}$.
3) Conflict: if two equations with the same left-hand side do not have the same functional symbols in their right-hand sides then a conflict is detected.
4) Restoration: For each pair of equations $X_q = f(X_{q_1}, \ldots, X_{q_k})$ and $X_q = f(X_{p_1}, \ldots, X_{p_k})$, add equations $X_{q_1} = X_{p_1}, \ldots, X_{q_k} = X_{p_k}$, and remove one of the equations.

Repeat the above steps until termination (output True) or conflict detection (output False).

Теорема 1. *The equivalence checking algorithm defined above always terminates and correctly identifies the equivalence of q' and q'' in $O(n^2)$ time, where n is the number of states of $\mathcal{A}$.*

References

1) Tree Automata Techniques and Applications / Comon H., Dauchet M., Gilleron R., et al. // http://www.grappa.univ-lille3.fr/tata. 2008.

2) Zakharov V.À. Equivalence checking of prefix-free transducers and deterministic two-tape automata // Proceedings of the 13th International Conference Language and Automata Theory and Applications. Lecture Notes in Computer Science. 2019. Vol. 11417. P. 146–158.

3) Zakharov V.À. Efficient Equivalence Checking Technique for Some Classes of Finite-State Machines // Automatic Control and Computer Sciences. 2021. Vol. 55, Iss. 7. P. 670–701.