Journal of Automata, Languages and Combinatorics **22** (2017) 4, 243 – 269

BOTTOM-UP QUOTIENTS FOR TREE LANGUAGES

Jean-Marc Champarnaud Ludovic Mignot Nadia Ouali-Sebti
Djelloul Ziadi

Département d'informatique, Université de Rouen Normandie
76801 Saint-Étienne du Rouvray Cedex, France
{Jean-Marc.Champarnaud,Ludovic.Mignot}@univ-rouen.fr
{Nadia.Ouali,Djelloul.Ziadi}@univ-rouen.fr

ABSTRACT

We extend the notion of tree language quotients to bottom-up quotients. Instead of computing the residual of a tree language from top to bottom and producing a list of tree languages, we compute a set of k-ary trees, where k is an arbitrary integer. We define the quotient formula for different combinations of tree languages: union, symbol products, compositions, iterated symbol products and iterated composition. These computations lead to the definition of the bottom-up quotient tree automaton, that turns out to be the minimal deterministic tree automaton associated with a regular tree language in the case of the 0-ary trees. Furthermore, we also define a syntactic test to determine whether a tree belongs to a given tree language, that allows us to define a new membership decision algorithm, without constructing the whole automaton.

Keywords: tree automaton, tree expression, expression derivative, minimal tree automaton

1. Introduction

Tree languages are used in numerous domains of applications, e. g., representation of XML documents. Regular tree languages are recognized by finite tree automata, well-studied objects leading to efficient decision problems. Among them, we can exhibit the membership test, that is to determine whether a given tree belongs to a language. Tree languages, that are potentially infinite, can be finitely described by regular tree expressions. Consequently it is an important subject of research to convert an expression into an equivalent automaton. In the case of words (which can be seen as trees with unary symbols) this is an active subject for more than fifty years: One of the first conversion methods is the computation of the position automaton [5] in linear size and quadratic time w.r.t the number of occurrences of symbols in the expression. Three years later, Brzozowski proposed another construction, the derivative automaton [2], that is deterministic and then exponential sized automaton. This construction is based on the operation of expression derivation, implementing the computation of the language quotient over expressions. Slightly modifying this method, by replacing expressions by sets of expressions, Antimirov constructed the