Extended Versions of Regular Expressions

Zuzana Beníčková and Dominika Regéciová

November 3, 2017

Abstract

Regular expressions are a tool for describing a language L over alphabet V, with a finite number of finitely structured elements. There exist in fact a countably infinite number of equivalent regular expressions for any RE. For any *language description* D, if some sentence in *L(D)* can be described in more than one way, then the language is *ambiguous.* This can be troublesome for purpose of having a unique meaning of sentences (e.g. in cases of programming languages). Regular expressions can be ambiguous as well and although they are easier to read and understand, the problems remain. Can any *RE* over V be transformed into an equivalent *unambiguous RE* over V? We will discuss this question using *definable characteristics* of REs.

Regular expressions used in XML Schemas, DTD and XSD are required to be *deterministic* (unambiguous), but can we effectively test if RE e is deterministic? The regular expression is unambiguous if and only if its representation using *Glushkov automaton* is deterministic. However, this test has quadratic time complexity. As was shown in the paper by Groz, Maneth, and Staworko, this can be done in $O(|e|)$ time, by using a decomposition of e's parse tree – for each symbol they build a skeleton based on lowest common ancestors. They are also discussing matching RE against given input, where REs have extended definition with option for choice (? mark) and numeric occurrences. The question if the matching can be done in linear time is still open, but they presented an efficient algorithm for matching words against DRE, and linear time algorithms for matching against *k-occurrence* (each symbol from V can occur at most k time in e, *k-ORE* for short), **-free* and *bounded plus-depth expressions* (the nesting depth of alternating union and concatenation symbols is bounded).