発表概要

バックトラックによる正規表現マッチングの時間計算量解析

中川 みなみ[1,a)]　南出 靖彦[2,b)]

2016年1月14日発表

正規表現マッチングは文字列を操作するウェブプログラムなど，様々な場面で用いられており，その実装の多くはバックトラックに基づいている．そのため，正規表現マッチングにかかる時間が文字列の長さに関して線形でないことがあり，最悪の場合指数関数時間となる．本発表では正規表現マッチングの時間計算量を判定する手法を提案する．対象の正規表現から先読み付き木トランスデューサを構成する．その先読み付き木トランスデューサから準同型写像や有限遷移系を構成し，それらを用いた遷移過程の中に反復補題が成り立つ遷移過程が存在しているか調べることで計算量を判定することができる．この判定には，Aho と Ullman による木トランスデューサの増加率の判定法を先読み付き木トランスデューサに拡張したものを利用する．先行研究では，Engelfriet と Maneth のマクロ木トランスデューサの増加率の判定法を用い，正規表現マッチングの計算時間が入力文字列に対して線形であるかどうかを判定する手法が提案された．本発表で提案する手法は，時間計算量が線形であるかどうかの判定だけでなく $O(n^2), O(n^3), \ldots$ になることも判定できる．この提案手法を OCaml によって実装し，既存の PHP プログラムで使用されている正規表現を対象に実験を行った．実験の結果，393 個中入力文字列の長さに対して 338 個が線形，44 個が 2 乗，6 個が 3 乗の計算量であると判定された．

Analyzing Time Complexity of Regular Expression Matching Based on Backtracking

Minami Nakagawa[1,a)]　Yasuhiko Minamide[2,b)]

Presented: January 14, 2016

Regular expression matching is used in various situations such as web programming. Most of its implementations are based on backtracking. Thus, its execution time may not be linear with respect to the length of an input string. In the worst case, it takes exponential time. In previous research, they proposed a method checking whether the matching of a given regular expression runs in linear time. They translated a regular expression to a tree transducer with regular lookahead and applied the result of Engelfriet and Maneth on proliferation rate of macro tree transducers. In this presentation, we propose a new method of analyzing time complexity of regular matching. This method can check a regular expression matching runs in $O(n), O(n^2), O(n^3), \ldots$ with respect to the length of input. Instead of the result of Engelfriet and Maneth, we extend the result of Aho and Ullman about proliferation rate of a tree transducer to a tree transducer with lookahead. We obtain a set of homomorphisms from a transducer and then construct a transition system to check whether or not there exist a transition satisfying the condition of the pumping lemma. We implemented this method by OCaml and conducted experiments analyzing execution time of regular expression matching. We applied out method to regular expressions used on existing PHP programs. Our experiments showed that the time complexity of 338 regular expressions are linear, 44 ones are quadratic, and 6 ones are cubic.

[1] 筑波大学システム情報工学研究科コンピュータサイエンス専攻
Department of Computer Science, University of Tsukuba, Tsukuba, Ibaraki 305–8573, Japan
[2] 東京工業大学数理・計算科学専攻
Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Meguro, Tokyo 152–8552, Japan
[a)] nakagawa@score.cs.tsukuba.ac.jp
[b)] minamide@is.titech.ac.jp