Randomization in Parallel Stringology

S. Muthukrishnan[1]

Bell Labs, 700 Mountain Avenue, Murray Hill, NJ 07974

In this abstract, we provide an overview of our survey of randomized techniques for exploiting the parallelism in string matching problems. Broadly, the study of string matching falls into two categories: *standard stringology* and *non-standard stringology*.

Standard Stringology concerns the study of various exact matching problems. The fundamental problem here is the *basic string matching problem* where given a *pattern* string $p \in \Sigma^m$ and a *text* string $t \in \Sigma^n$, the problem is to find all occurrences i in t where the pattern occurs, that is, $t[i \cdots i+m-1] = p[1 \cdots m]$. Other well-studied problems within standard stringology include multiple pattern matching, dictionary matching, text indexing, etc.

Non-standard Stringology concerns the study of various inexact matching problems. A fundamental problem here is *string matching with wildcards* where the problem, as before, is to find all occurrences of the pattern within the text; however, some positions in the text and the pattern are marked as "wild-cards"(denoted ϕ), that is, these positions match whatever be the symbol in the string they align against. Formally, given p occurs in t at i if for all $1 \leq j \leq m$, $t[i+j-1] = p[j]$ if $p[j] \neq \phi$ *and* $t[i+j-1] \neq \phi$. Another important problem here is the *k-mismatches problem* in which the goal is to determine all those positions in the text where the pattern occurs with at most k mismatches amongst the aligned symbols. Other problems here include subset matching, range matching etc.

Stringology includes study of many other problems involving

- generalization of alignments – as in the Longest Common Subsequences problem, where the occurrence of p in t at i may depend on the match between $t[i+j']$ and $p[j]$ for j, $1 \leq j \leq m$, not necessarily equal to $j'-1$.
- generalization of pattern descriptions – matching regular expressions as opposed to strings,
- generalization of strings to other objects such as trees, multi-dimensional arrays, etc.

In our survey, we do not consider the generalizations above. We restrict ourselves to presenting a set of fundamental randomized techniques that turn out to broadly applicable to many problems within standard and non-standard stringology.

We adopt the CRCW PRAM - Concurrent Read, Concurrent Write Parallel Random Access Machine model. Parallel string algorithms rely on many basic operations including prefix sum, list ranking, sparse array compaction, sorting under various input assumptions, Euler tour techniques, integer hashing etc.

The technical results we cover are as follows. We will present the randomized fingerprints due to Karp and Rabin, and its extensions. We will apply these fingerprints to solve

- Standard string matching,
- Digitized string matching,
- Dictionary pattern matching, and
- Suffix tree construction.

All these algorithms are of *Monte Carlo* type, that is, their running times are guaranteed in the worst case and they may commit errors, although with provably small probability. If the goal is to produce *Las Vegas* type algorithms, that is, ones that never err but with running times that hold only with high probability, two approaches may be followed: (1) Check the correctness of the output of Monte Carlo algorithms efficiently so they may be repeatedly rerun until there are no errors, or (2) Design alternate randomized fingerprints for strings that are Las Vegas type. In approach (1) above, efficient checking procedures have been designed for standard string matching, suffix tree construction, and dictionary prefix matching; we will survey these results. In category (2), there has been some progress but open issues exist.

For problems in nonstandard stringology, all known efficient algorithms reduce them to computing convolutions. There are three such reductions: alphabet-based, clique-cover based, and character-shifting based. We will review each of these approaches, and compare them.

We conclude with

- string matching results on other conventional parallel computing models such as meshes, VLSI circuit models and Exclusive/Concurrent Read and Exclusive Write PRAMs. We will also investigate the difficulty of string matching on recently proposed PRAM variants such as those based on queuing, and on external memory models with parallel disks.
- a few novel string matching problems where randomization and parallelism may prove useful.
- a few interesting results in randomized parallel string matching not covered by the main focus of this survey.