# How to Exploit Periodicity

**Paweł Gawrychowski**

Institute of Computer Science, University of Wrocław, Poland
gawry@cs.uni.wroc.pl

--- **Abstract** ---

Periodicity is a fundamental combinatorial property of strings. We say that $p$ is a period of a string $s[1..n]$ when $s[i] = s[i + p]$ for every $i$ such that both $s[i]$ and $s[i + p]$ are defined. While this notion is interesting on its own, it can be often used as a tool for designing efficient algorithms. At a high level, such algorithms often operate differently depending on whether a given string does or does not have a small period, where small usually means smaller than half of its length (or, say, quarter). In other words, we design an algorithm that is efficient if the given string is repetitive, and another algorithm that is efficient if the given string is non-repetitive, in every case carefully exploiting either the periodicity or the fact that input looks sufficiently "random", and then choose the appropriate algorithm depending on the input. Of course, in some cases, one needs to proceed in a more complex manner, for example by classifying the whole string look at its substrings and process each of them differently depending on its structure.

I will survey results, mostly connected to different version of pattern matching, that are based on this paradigm. This will include the recent generalization of periodicity that can be applied in approximate pattern matching, and some examples of how the notion of periodicity can be applied to design a better data structure.