Kernelization

(Invited Talk)

Fedor V. Fomin

Department of Informatics, Univeristy of Bergen, 5020 Bergen, Norway
fomin@ii.uib.no

Preprocessing (data reduction or kernelization) as a strategy of coping with hard problems is universally used in almost every implementation. The history of preprocessing, like applying reduction rules simplifying truth functions, can be traced back to the 1950's [6]. A natural question in this regard is how to measure the quality of preprocessing rules proposed for a specific problem. For a long time the mathematical analysis of polynomial time preprocessing algorithms was neglected. The basic reason for this anomaly was that if we start with an instance I of an NP-hard problem and can show that in polynomial time we can replace this with an equivalent instance I' with $|I'| < |I|$ then that would imply P=NP in classical complexity.

The situation changed drastically with advent of Parameterized Complexity [3]. The philosophy of parameterized complexity is that beyond overall input size, key secondary measurements fundamentally affect the computational complexity of problems and govern the opportunities for designing efficient algorithms. Combining tools from parameterized complexity and classical complexity it has become possible to derive upper and lower bounds on the sizes of reduced instances, or so called *kernels*. In parameterized complexity each problem instance comes with a parameter k and the parameterized problem is said to admit a *polynomial kernel* if there is a polynomial time algorithm (the degree of polynomial is independent of k), called a *kernelization* algorithm, that reduces the input instance down to an instance with size bounded by a polynomial $p(k)$ in k, while preserving the answer. This reduced instance is called a $p(k)$ *kernel* for the problem. If $p(k) = O(k)$, then we call it a linear kernel and if$p(k) = k^{O1)}$, then a polynomial kernel.

The classical example goes back to the work of Nemhauser and Trotter [5], who gave a polynomial time algorithm that for a given graph G and integer k, reduces G to a graph K on $2k$ vertices (and thus of size $O(k^2)$) such that G has a vertex cover of size k if and only if K does. In other words, there is an $O(k^2)$ kernel for k-Vertex Cover. Other examples of kernels include a linear kernel for k-Dominating Set on planar graphs [1] and an $O(k^2)$ kernel for k-Feedback Vertex Set [7].

In this talk we give an overview of recent results and techniques for obtaining linear and polynomial kernels on planar graphs and more generally, on graphs excluding some fixed graph as a minor. The talk is based on joint works with Hans Bodlaender, Daniel Lokshtanov, Elko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos [2,4].

F. Ablayev and E.W. Mayr (Eds.): CSR 2010, LNCS 6072, pp. 107–108, 2010.

References

1. Alber, J., Fellows, M.R., Niedermeier, R.: Polynomial-time data reduction for dominating set. J. ACM 51, 363–384 (2004)
2. Bodlaender, H., Fomin, F.V., Lokshtanov, D., Penninkx, E., Saurabh, S., Thilikos, D.M.: (meta) Kernelization. In: Proceedings of the 50th Annual Symposium on Foundations of Computer Science (FOCS 2009), pp. 629–638. IEEE, Los Alamitos (2009)
3. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, Heidelberg (1998)
4. Fomin, F.V., Lokshtanov, D., Saurabh, S., Thilikos, D.M.: Bidimensionality and kernels. In: Proceedings of the 21th ACM-SIAM Symposium on Discrete Algorithms (SODA 2010), pp. 503–510 (2010)
5. Nemhauser, G.L., Trotter, L.E.: Vertex packings: Structural properties and algorithms. Math. Program. 8, 232–248 (1975)
6. Quine, W.V.: The problem of simplifying truth functions. Amer. Math. Monthly 59, 521–531 (1952)
7. Thomassé, S.: A quadratic kernel for feedback vertex set. In: Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms (SODA 2009), pp. 115–119 (2009)