Sébastien Tavenas, LAMA, Université de Savoie Mont-Blanc

Structural results for Arithmetic formulas

Proving explicit lower bounds on the size of arithmetic formulas remains a central open problem in algebraic complexity theory. Recent advances — such as lower bounds for constant-depth arithmetic formulas — suggest a promising strategy: reduce general formulas to a more structured class (e.g., homogeneous formulas with bounded depth) with only moderate size blow-up, and then establish strong lower bounds for this restricted class. In this talk, we will study the cost of operations such as parallelization at depth $O(\log d)$ or the homogenization of arithmetic formulas.

Classical depth-reduction results offer valuable tools for this approach. The work of Brent, Kuck, and Maruyama shows that any arithmetic formula of size s can be transformed into one of depth $O(\log s)$ with only a polynomial increase in size. Similarly, the famous result by Valiant, Skyum, Berkowitz, and Rackoff demonstrates that any arithmetic circuit computing a degree-d polynomial can be converted to one of depth $O(\log d)$, again with polynomial blow-up. This raises a natural question: Can we similarly reduce the depth of arithmetic formulas to $O(\log d)$ efficiently? We will see that this is indeed possible for *homogeneous* formulas. In fact, the result extends even further — to a broader class we call *quasi-homogeneous* formulas, whose syntactic degree is only polynomially bounded in d.

This brings us to the next key question: what is the cost of homogenizing formulas? A standard folklore argument shows that any formula F computing a homogeneous degree d polynomial can be homogenized with a size increase of at most $d^{O(\log s)}$. We will show that the homogenization overhead can be reduced *superpolynomially* when $d = s^{o(1)}$. Previously, such an improvement was only known in the more restricted case $d = (\log s)^{1+o(1)}$ [Raz]. Further, we can also get rid of the condition on d at the expense of getting a *quasi-homogenization* result.

Joint work with Fournier, Limaye, Malod and Srinivasan.