Handout for the REU Student Presentation 2013
Entropy Compression Method

presented by Vojta Tuma

In combinatorics, one usually deals with a family of objects without too much restrictions, and tries to find some regular patterns/structure among them. For instance, take the family of all graphs – what can be said about all graphs in general? On the other hand, once you have a graph with, say, perfect matching, or with a Hamiltonian cycle, you can work with such graph in much more sophisticated way.

Our combinatorial setting is natural numbers, that is, the set $\mathbb{N} = \{1, 2, 3, \ldots\}$. We consider all possible two-colourings of $\mathbb{N}$ with the colours red and green (without any restrictions). The set of all such colourings is a very wild set, you can find a huge number of both structured and random-like behaviour in it. Thus it might be a little surprising that the following "regular" theorem holds.

Theorem 1 (Van der Waerden, 1927)**.** *For every k there exists n such that for every 2-colouring of $1..n$ one can find a monochromatic arithmetic progression of length k in it.*

A monochromatic arithmetic progression is a set of the form $a, a + d, a + 2d, \ldots, a + (k - 1)d$, such that all the numbers are red, or all are green. The theorem says that there cannot be complete chaos – no matter how "wild" a colouring of $\mathbb{N}$ you take, you can find arbitrarily long monochromatic arithmetic progressions in it (in some initial segment of it).

This theorem is very important – for instance, it is related to the famous theorem of Green and Tao regarding arithmetic progression in primes, for which they were awarded the Fields medal.

Let us denote by $W(k)$ the smallest n such that $1..n$ always contains a monochromatic arithmetic progression (MAP). Thanks to the theorem, this number is well-defined. A good combinatorial question is "How big is this number? How fast does it grow compared to other functions?" In this talk we deal with lower bounds, that is, we try to find some function $f(k)$ such that $W(k) \geq f(k)$ and $f(k)$ is some simple (quickly evaluable) function. Obtaining lower bounds in combinatorics is usually done in two ways – either constructively, that is, explicit construction of some object which does not satisfy the investigated property, or non-constructively. A typical example of non-constructive approach is the use of probabilistic method (which has been successfully used in this case too). We present a quite novel approach here – the entropy compression method.

The basic idea is simple. Consider strings over the alphabet $\{0, 1\}$ (or over *red* and *green*). If one has the set of all strings of length n_1, the set of all strings of length n_2, and an injective mapping from the former to the latter, what does it mean? Clearly, it means that $n_2 \geq n_1$, otherwise there are not enough elements in the latter set.

How to use the idea to obtain a lower bound on $W(k)$? Set $n = W(k)$, and consider the set S of all strings over $\{0, 1\}$ of length n. We will construct some

injective mapping from S to some other set P, and the size of P will bound our n. First of all, every element of S contains a MAP of length k (by the definition of S and Van der Waerden's theorem). We do the following "compression" procedure for every element $s \in S$:

1. look at what position the MAP in s starts, let that number be p,
2. look at what difference that MAP has (the difference between two consecutive numbers of it), let that number be d,
3. look at the colour of that MAP, let that be c,
4. erase the MAP from s (that is, replace s by a shorter string, which has the elements corresponding to the MAP removed),
5. prepend to s the number p written in binary, the number d written in binary, and the number c (which is either 0 or 1).

The last step gave us a new binary string. If we possibly prepend p and d with some zeroes, this procedure always outputs a binary string of length

$$n - k + \lceil \log_2 n \rceil \cdot 2 + 1.$$

Note that this procedure is injective, that is, we can reconstruct the original string after we have applied the compression procedure to it. Thus the corresponding mapping must be injective. This gives the following chain of inequalities:

$$\begin{aligned} n - k + \lceil \log_2 n \rceil \cdot 2 + 1 &\geq n \\ \lceil \log_2 n \rceil \cdot 2 &\geq k - 1 \\ n^2 \cdot 2^{\delta} &\geq 2^{k-1}, \end{aligned}$$

where $\delta \in [0, 1]$ (because of the ceils). Note that this gives us a bound on $W(k) = n$, actually a quite good one.

One can use more sophisticated techniques to obtain better bounds. For instance note that our approach (that is, our encoding scheme) somehow assumes that the number of possible APs is n^2, but the actual number can be bounded from above by n^2/k, which gives a better bound (by $\sqrt{k}$). Even more rapid techniques of compressing can be employed. Also, the results can be straightforwardly generalised to strings over bigger alphabet (that is, colourings with more colours).

Our work on this REU is concerned with applications of this method to other combinatorial problems.