ORCO – Graphs and Discrete Structures
November 30, 2022 – Lecture 9

1 Universal sequences

You might know *Gray codes*, a way to enumerate all sequences of n bits by simpling changing a single bit at each step. Here we will also consider enumerating sequences of n-bits (or equivalently subsets of $[n]$), but with a different property.

A sequence $A = a_1 a_2 \ldots a_s$ of elements $a_i \in [n]$ is *n-universal* if for any subset $S \subseteq [n]$, we can find the elements of S as a subsequence in A (i.e. the elements appear consecutively somewhere in the sequence A). For instance $A = 123413241$ is 4-universal.

The following is a classical problem in information storage and retrieval.

Question *What is the minimum size of an n-universal sequence?*

All subsets of fixed size (say $n/2$) must start at a different location in A, so $|A| \geq \binom{n}{\lfloor n/2 \rfloor} \approx \sqrt{\frac{2}{\pi n}} \cdot 2^n$. On the other hand it is easy to construct such a sequence by appending all subsets of $[n]$: the length of the corresponding sequence is at most $n \cdot 2^n$. The following result shows that we can gain a factor linear in n compared to this trivial bound.

Theorem 1 (Lipski, 1978)**.** *There is an n-universal sequence of size at most $\frac{4}{\pi} \cdot 2^n$.*

Proof. We only consider the case where n is even for simplicity. We let $S = 1, 2, \ldots, n/2$ and $T = n/2 + 1, \ldots, n$. By Theorem 3 in Lecture 8, S has a partition into at most $\binom{n/2}{n/4} \approx \sqrt{\frac{4}{\pi n}} \cdot 2^{n/2}$ symmetric chains $C_1, \ldots, C_r$ and similarly T has a partition into the same number of symmetric chains $D_1, \ldots, D_r$. Each chain C_i is obtained from some minimum set S_i by adding some elements $x_1, x_2, \ldots, x_h$ one by one to S_i. To this chain we associate the sequence starting with the elements of S_i (in any order), followed by $x_1, x_2, \ldots, x_h$ in this order. Let us call this sequence A_i. Similarly, we associate fo each chain D_i a sequence as above, except that we consider the elements in reverse order (we start with the element x_h that was last added to the chain, and end with the elements of the minimum set S_i of the chain, in any order). Let us call this sequence B_i.

We are now ready to define our n-universal sequence. We simply concatenate all sequences $B_i A_j$, for all $1 \leq i, j \leq r$. For instance we can take:

$$A = B_1 A_1 B_1 A_2 \dots B_1 A_r B_2 A_1 \dots B_r A_1 B_r A_2 \dots B_r A_r.$$

The size of A is at most $n \cdot \binom{n/2}{n/4}^2 \approx \frac{4}{\pi} 2^n$, as desired. To see that A is n-universal, observe that every subset $S' \subseteq S$ appears as a prefix of some sequence A_i, and every subset $T' \subseteq T$ appears as a suffix of some sequence B_j, so the elements of $S' \cup T'$ appear consecutively in $B_j A_i$ (and any subset of $[n]$ can be written as $S' \cup T'$ for some $S' \subseteq S$ and $T' \subseteq T$). $\square$

2 De Bruijn sequences

In this section we consider *cyclic sequences*. A cyclic sequence of elements of $\{0, 1\}$ is a *de Bruijn sequence of order n* if it contains all sequences of $\{0, 1\}^n$ as a subsequence exactly once. Note that such a cyclic sequence has size precisely 2^n.

To show that these cyclic sequences exist for every n, we create a graph G_n whose vertices are all the elements of $\{0, 1\}^{n-1}$, and such that for any $(x_1, \dots, x_{n-1}) \in \{0, 1\}^{n-1}$ and any $y \in \{0, 1\}$, we add an arc from $(x_1, \dots, x_{n-1})$ to $(x_2, \dots, x_{n-1}, y)$ (the resulting directed graph has some loops, for instance an arc from $(0, 0 \dots, 0)$ to itself, labelled 0). Note that all vertices of G_n have in-degree and out-degree 2, so G_n is Eulerian. Take any Eulerian tour of G, and consider the cyclic sequence consisting of the labels of the arcs along the tour. Note that since G_n is Eulerian, the sequence contains exactly 2^n elements. Moreover, n-bit words are in bijection with arcs of G_n, and by the definition of G_n each n-bit word can be found as a subsequence in the cyclic sequence.

3 Independence number

Given a graph G, an *independent set* (or *stable set*) of G is a set of pairwise non-adjacent vertices of G. The *independence number* (or *stability number*) of G, denoted by $\alpha(G)$, is the size of the largest independent set in G.

The following is a very useful tool to find lower bounds on the chromatic number of a graph.

Lemma 2. *For any graph G on n vertices, $\chi(G) \geq \frac{n}{\alpha(G)}$.*

Proof. A coloring of a graph G with k colors is a partition of the vertices of G into k independent set. Taking the largest color class, it follows that any graph G on n vertices contains an independent set of size $n/\chi(G)$. Thus $\alpha(G) \geq n/\chi(G)$ and $\chi(G) \geq n/\alpha(G)$, as desired. □

In the remainder of this section we prove a lower bound on the independence number of graphs. The result will be interpreted differently next week, in the context of *extremal combinatorics.*

Theorem 3. *For any graph $G = (V, E)$, $\alpha(G) \geq \sum_{v \in V} \frac{1}{d(v)+1} \geq \frac{n^2}{2m+n}$.*

Proof. We will actually provide a randomized algorithm producing an independent set of the required size in expectation. Consider an order $v_1, \dots, v_n$ on the vertices of G, chosen uniformly at random among all such orders, and apply Algorithm 1 below.

Algorithm 1 Find a large independent set S in G

```
Set S = ∅
for i = 1 to n do
  if v_i has no neighbor in S then
    add v_i to S
  end if
end for
return S
```

Given the order $v_1, \dots, v_n$, let T be the set of vertices v_i such that v_i has no neighbor v_j with $j < i$ (i.e. all the neighbors of v_i are after v_i in the order). Clearly T is contained in S (the set returned by the algorithm above), and in order to find a lower bound on $|S|$, it is thus enough to find a lower bound on $|T|$.

For each vertex v_i, $\mathbb{P}(v_i \in T) = \frac{1}{d(v_i)+1}$, since each vertex among v_i and its $d(v_i)$ neighbors has the same probability to be before all the others in the order. It follow that $\mathbb{E}(|S|) \geq \mathbb{E}(|T|) = \sum_{i=1}^{n} \frac{1}{d(v_i)+1}$. Since the set S returned by Algorithm 1 is clearly an independent set, the algorithm returns an independent set of size at least $\sum_{v \in V} \frac{1}{d(v)+1}$ in average (and thus an independent set of this size exists in the graph).

To prove the last part of the inequality, observe that by convexity of the function $x \mapsto \frac{1}{x+1}$, $\sum_{v\in V} \frac{1}{d(v)+1}$ is minimized when all the vertices have the same degree, namely $2m/n$. Hence, $\sum_{v\in V} \frac{1}{d(v)+1} \geq n \cdot \frac{1}{\frac{2m}{n}+1} = \frac{n^2}{2m+n}$. $\square$