Ad hoc method for independent sets

https://blog.carolin-zoebelein.de/2020/05/adhocindependentsets.html
Wed 20 May 2020 in Math, Carolin Zöbelein

Today, we want to determine indpendent sets of an arbitrary undirected graph. Which can be done by a trivial ad hoc method, described as follows.

Algorithm

We have given an arbitrary undireced graph $G = (V, E)$ with $\deg(u_i) \geq 1$ for all $u_i \in V$. The set $V_0 \subseteq V$ of vertices with $\deg(u_i) = 0$ will not be considered. Since, we are interested in the independent sets of G, the set V_0 is a trivial case which, and it's subsets, can be simply added to any final solution of independent sets. Our following considerations will work for every arbitrary kind of graph, so we don't have to make any other further assumption regarding possible special kinds of graphs respectively sub-graphs, at the moment.

We start our independent set generation with the assumption that all edges $(u_i, u_j) \in E$ are removed from our graph G. For this trivial case, we would have the one largest independet set easy given by $S_0 := \{u_1, u_2, u_3, u_4, \ldots, u_{|V|-1}, u_{|V|}\}$.

Now, we take an arbitrary edge $e_1 := (u_1, u_2)$ from the set of the belonging edges E of G. From the definition of an independent set follows that u_1 and u_2 can not be element of the same independent set of the same time. Since $u_1 \in S_0 \wedge u_2 \in S_0$ is given, we have to split S_0 into the two new independet sets $S_0 = \{u_1, u_3, u_4, \ldots, u_{|V|-1}, u_{|V|}\}$ and $S_1 = \{u_2, u_3, u_4, \ldots, u_{|V|-1}, u_{|V|}\}$.

In the next step, we take edge e_2. Here, we have now to differ between three cases. The two vertices of e_2 both lies completely in S_0 or both lies in S_1 or in S_0 and S_1, at the same time. Which means our possible outcome for $e_2 := (u_1, u_3)$ would be $S_0 \setminus \{u_1\}$, $S_0 \setminus \{u_3\}$ and S_1, or for $e_2 := (u_2, u_3)$ we get S_0, $S_1 \setminus \{u_2\}$ and $S_1 \setminus \{u_3\}$, or finally for $e_2 := (u_3, u_4)$ we get $S_0 \setminus \{u_3\}$, $S_0 \setminus \{u_4\}$, $S_1 \setminus \{u_3\}$ and $S_1 \setminus \{u_4\}$.

So, we see that we can generate all of our independent sets by a simple, naive way of taking an edge of E and looking if this edge is an element of an already generated independet set S_i. If yes then split this set into two new ones, by copy S_i to a second set S_j and removing the first vertex of e from S_i and removing the second vertex of e from S_j, else nothing has to be done.

By doing this for all edges e of E, we finally get all independent sets of G.

Complexity

The complexity of this algorithm is easy to determine. Assume we have given the worst case scenario of a complete graph G (a graph in which every pair of distinct vertices is connected by an unique edge) with $|V|$ vertices and edges whose number is given by $|E| = \binom{|V|}{2} = \frac{1}{2}|V|^2 - \frac{1}{2}|V|$. To determine all independent sets, we have to do our algorithm for $|E|$ steps, in which we will have to check for this edge in every already existing independent set and finally split it into two. Hence, in the $|E|$'th step, we have to check $2^{|E|}$ independent sets for the current edge. For the complexity we get with this $\mathcal{O}\left(2^{|V|^2}\right)$.

Outlook

All of this seems not to look very exciting, but we will see that we can derive nice things from it, in future blog posts.