Lecture 12: Vertex colouring · 1MA020

Vilhelm Agdur[1]

[1] vilhelm.agdur@math.uu.se

20 November 2023

We study the notion of a vertex colouring of a graph, and prove some results about the chromatic number of a graph. We prove Brooks' theorem, the five-colour theorem for planar graphs, and the three-colour theorem for outerplanar graphs. We use the latter to prove the art gallery theorem.

Introductory results

We already saw the notion of colouring a graph in an earlier lecture, but we proved nothing about it other than some very basic bounds. Let us restate the definition, and then move on to proving some more interesting things.

Definition 1. A *k-colouring* of a graph $G = (V, E)$ is a function $c : V \to [k]$, where we think of the numbers $1, 2, \dots, k$ as colours, such that no two adjacent vertices are sent to the same colour.[2] The *chromatic number* of a graph G, denoted $\chi(G)$, is the least integer k such that G has a k-colouring.

[2] On rare occasions, we will want to think about colourings that do not necessarily fulfill this condition. Then we will call those colourings *improper*, and the ones with the property will be *proper colourings*.

In order to upper bound the chromatic number of a graph, the most obvious approach is of course to construct a colouring of it. The most straightforward way to do this is greedily:

Definition 2. Let $G = (V, E)$ be a graph with an ordering on the vertices, say $v_1 < v_2 < \ldots < v_n$. On the colours, we use the obvious ordering $1 < 2 < \ldots$. The *greedy colouring algorithm* starts with an empty colouring, and then at each time-step, it colours the first uncoloured vertex (according to the ordering on the vertices) with the first colour that has not been used on any of its neighbours.

If we do nothing special in the choice of ordering of vertices, we don't get any very strong bound from this, of course. We do, however, get the following general bound:

Lemma 3. *For any graph G with maximum degree Δ, we have*

$$\chi(G) \leq \Delta + 1.$$

Proof. Pick an arbitrary ordering on the vertices, and run the greedy colouring algorithm. For any vertex, when it is about to be coloured, it has at most Δ neighbours that already have colours, and so one of the first $\Delta + 1$ colours must be unused. □

If we know something about the graph, we can be a bit more clever in our choice of ordering, and get a better bound. To do this, however, we first need to define the breadth-first ordering of the vertices of a graph.

Definition 4. Let $G = (V, E)$ be a graph. *Breadth-first search* started at a vertex v proceeds as follows:

1. Initialize the *currently visited vertex* to v, the list of *discovered vertices* D to be empty, and the list of *visited vertices* W to be empty.
2. Add all the neighbours of the currently visited vertex to the end of D, in an arbitrary order.
3. Add the currently visited vertex to the end of W.
4. If D is not empty, remove the first element of D, and let it be the new currently visited vertex. Otherwise, return W.

This algorithm will return a list of the vertices of the connected component containing v. The order in which these vertices appear is called a *breadth-first ordering* of the vertices, started at v.

We could also modify the algorithm slightly, to make each vertex remember which vertex was the currently visited one when it was added to the list of discovered vertices. Then we get a *breadth-first spanning tree* rooted at v by having each vertex consider its parent as the vertex that discovered it.

Lemma 5. *Suppose G is a graph with maximum degree Δ, whose vertices do not all have the same degree. Then*

$$\chi(G) \leq \Delta.$$

Proof. Since not all vertices have the same degree, there must exist a vertex v with $d_v < \Delta$. Now order the vertices of G according to the *inverse* of the breadth-first order on the vertices started at v,[3] and run the greedy colouring algorithm with this ordering.

The crucial property we need from the ordering is that every vertex except v will, when visited by the greedy colouring algorithm, have at least one neighbour that has not yet been coloured. So it has at most $\Delta - 1$ coloured neighbours, and so there is always at least one colour available among the first Δ colours.

When we finally get to v, it has degree less than Δ, so it too has fewer than Δ coloured neighbours, and can pick a colour from among the first Δ. □

[3] By assumption G is connected, so the breadth-first search will find every vertex.

We can also make use of our previous work on connectivity to break down the problem of finding a colouring of a graph into finding a colouring of its two-connected blocks.

Lemma 6. *Suppose G is some graph which has at least one two-connected block, and let $B_1, B_2, \ldots, B_r$ be its collection of two-connected blocks. Then*

$$\chi(G) = \max_{i \in [r]} \chi(B_r).$$

If G has no two-connected blocks, $\chi(G) = 2$ if G has at least one edge, and $\chi(G) = 1$ if it has no edges.[4]

Proof. That the chromatic number of G is lower bounded by the max over the blocks is trivial. To see the other direction, consider the block graph of G, which we saw in our previous lecture is a forest.

Colour each block of G, using at most $\max_{i \in [r]} \chi(B_r)$ colours.[5] Then, because any two blocks only intersect in at most one vertex, we can permute the colours in each block so that they agree on the cutvertices, and this can be done in such a way that we get a colouring on the entire graph because the block graph is a tree.[6] Clearly permuting the colours does not add new colours, so our global colouring with also use only $\max_{i \in [r]} \chi(B_r)$ colours. □

[4]

Exercise 1. Prove this part of the theorem statement.

[5] If the block is two-connected, this is trivially possible. Otherwise it can be coloured with at most two-colours, since it must be an isolated vertex or a K_2.

[6]

Exercise 2. Write out the details of this argument – how do you actually algorithmically do this?

Brooks' theorem

When we first introduced the notion of a colouring of a graph, we found an example where Lemma 3 is actually sharp: For the complete graph, we have $\chi(K_n) = n = \Delta(K_n) + 1$. There is also another example, namely the cycle graphs of odd length, which have chromatic number three and maximum degree two.

Are there any other cases where this inequality is sharp? It turns out that these are in fact the only connected examples, which is the content of Brooks' theorem.

Theorem 7 (Brooks, 1941)**.** *Let G be a finite connected graph with maximum degree Δ that is not complete nor a cycle of odd length. Then*

$$\chi(G) \leq \Delta.$$

Proof. If G is not regular, then the theorem follows from Lemma 5, so we can assume that all vertices of G have degree k. Now, obviously, the only 1-regular connected graph is K_1, and we have assumed G is not complete.

For $k = 2$, the only connected 2-regular graphs are the cycles. We have assumed G is not a cycle of odd length, and we know the cycles of even length have chromatic number 2.

So we may assume $k \geq 3$. We may additionally, by Lemma 6, assume that G is at least two-connected.So we now divide into two cases:

Suppose G is in fact also three-connected. Then, since G is not complete, we can find three vertices a, v, b such that $a \sim v, b \sim v \in E$, but a and b are non-adjacent. Since G is three-connected, the graph $G[V \setminus \{a, b\}]$ is connected.

We can now colour G in a way similar to what we did for Lemma 5. First, we let a and b both take the colour 1, which is possible since they are non-adjacent. Then we run the greedy colouring algorithm on the rest of the vertices using the inverse of a breadth-first search order started at v.

Again, each vertex other than v will have fewer than k already-coloured neighbours when it is visited,[7] so they can be coloured using only k colours, as before. Now, v will have all of its k neighbours coloured – however, two of those neighbours, a and b, have the same colour, so its neighbours are using fewer than k *distinct* colours, so there is one left over for v.

So for the other case, suppose that in fact $\kappa(G) = 2$. Now, if we could find a triple of vertices a, v, b with the same properties as in the three-connected case, that argument would of course work again – so we need a new argument for why such a triple exists.

So let a be an arbitrary vertex of G. If $G[V \setminus \{a\}]$ is still 2-connected, we can pick x as any neighbour of a and b as any neighbour of x other than a, and these will have the required properties.

Now, if $G[V \setminus \{a\}]$ is not two-connected, we claim that its block graph must be a tree with at most k leaves, and each of these leaves is a block B_i which contains a non-cutvertex b_i which is adjacent to a in G.

If this were not the case, there would be some block B_ℓ such that adding a back in did not create a cycle intersecting B_ℓ in at least two vertices, and so B_ℓ would still be a separate block in G. But G is two-connected, so this is impossible.

Now we pick a as the central vertex and b_1 and b_2, say, as the other two vertices. To see that this is a triple of vertices with the right properties, we need to see that b_1 and b_2 are non-adjacent. Now, we know that if they were adjacent they would be in the same block, but we also also know they are not cutvertices and so belong to only one block, and by assumption they are in the distinct blocks B_1 and B_2. So they are not adjacent, and we have found a triple with the right properties, finishing the proof. □

[7] Even taking into account that a and b already have a colour – think about why this is true.

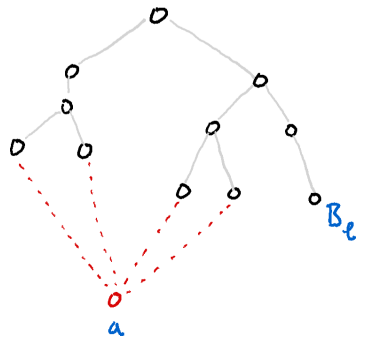

Figure 1: A sketch of the block graph of $G[V \setminus \{a\}]$ and the vertex a drawn in in red, with dotted edges connecting it to blocks it is in as a non-cutvertex, and the putative but impossible leaf B_ℓ that does not contain a. Notice how the addition of a will create cycles forcing everything except B_ℓ to become a single block. Also notice how a being the cutvertex in B_ℓ wouldn't help – that would create a dotted line from a to the parent of B_ℓ, which still does not create a cycle with B_ℓ in it.

Exercises