Notes on Brooks' Theorem

Rich Schwartz

March 18, 2016

Let G be a connected graph. Let k denote the maximum degree of G and let $\chi(G)$ denote the chromatic number of G. Brooks' Theorem is that $\chi(G) \leq k$ as long as G is not a complete graph or an odd cycle. My proof is mostly the same as what is in §5.1 in West's book, but the end of my proof is different.

Case 1: If $k = 2$ then G is either a path or an even cycle. This case is trivial. So, assume that $k \geq 3$.

Case 2: Suppose that G is not k-regular. This means that some vertex v of G has degree less than k. Let T be a spanning tree of G, and orient T so that all directed paths end at v. Choose any ordering of the vertices of G so that the labels increase along directed paths of T. So, every vertex of G except v is incident to a vertex of G which has a larger label.

Now do the greedy algorithm. At each step of the algorithm, except the last one, the current vertex is incident to at most $k-1$ vertices which have already been colored. Hence, there is some color in $\{1, ..., k\}$ which is available for use. Since v is incident to at most $k-1$ vertices, the same goes for v. So, the greedy algorithm with the given ordering produces a proper k-coloring.

Figure 1 shows an example in which $k = 3$. The vertex colors are red, blue, magenta.

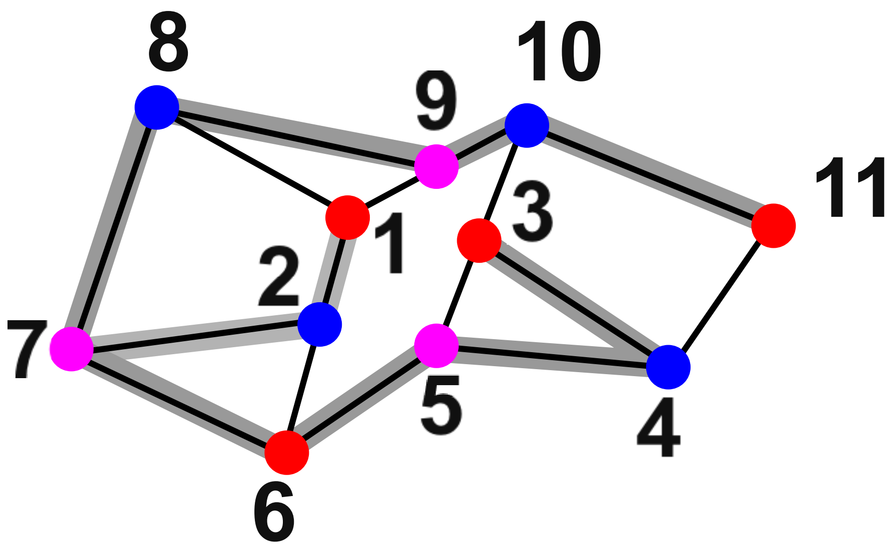

Figure 1: Ordering the vertices using a spanning tree.

Case 3: Suppose that G is k-regular but not 2-connected. So, there is some cut vertex v of G. That is $G - v$ is a finite union $G'_1 \cup \ldots \cup G'_m$ of connected graphs. Let G_j be the subgraph of G obtained by adding v to G'_j and also all the edges of G which connect v to vertices in G'_j. We rename this extra vertex v_j. So, $v_1, \ldots, v_m$ are really all the same vertex, but it is useful to give them different names. By construction, v_j has degree less than k in each graph G_j. So, by Case 2 or induction on k, we see that G_j has a proper k-coloring. Permuting the colors if necessary, we can guarantee this proper coloring assigns 1 to v_j. But then all the separate proper colorings fit together to give a proper coloring of G.

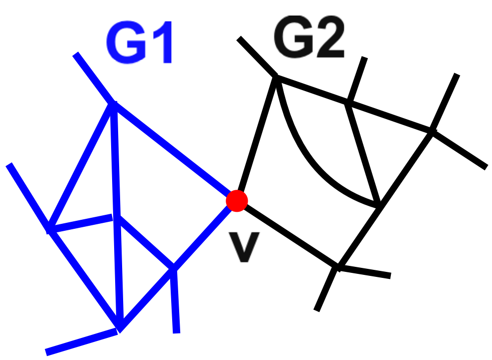

Figure 2: A cut vertex spanning tree.

Case 4: Suppose that G is k-regular and 2-connected. Suppose also that we can find two vertices v and w, which are not adjacent, so that $G - \{v, w\}$ is disconnected. Then we draw the same kind of picture as in Figure 2, except that rather than a single vertex v there are two vertices v and w. Consider the corresponding graphs $G_1, ..., G_m$. As in Case 3, we have the vertices $v_1, w_1, ..., v_m, w_m$. Since G is 2-connected, each v_j has an edge connecting to a vertex in G_j. The same goes for w_j. Hence, the degree of v_j and the degree of w_j are both less than k. This means that we can properly k-color each G_j.

Suppose that, for each j, we have either $\deg(v_j) < k-1$ or $\deg(w_j) < k-1$. Then we can choose our coloring of G_j so that v_j and w_j get different colors. Permuting the colors, we can arrange that v_j gets color 1 and w_j gets color 2. But then we can piece the separate colorings together to get a proper k-coloring of G.

After re-numbering the components of $G - \{v, w\}$, we arrive at a single exceptional case:

- There are just 2 graphs G_1 and G_2.
- $\deg(v_1) = \deg(w_1) = k - 1$ and the coloring gives the same color to v_1 and w_1.

in this case v_2 and w_2 have degree 1 in G_2, and we can adjust the colors so that the coloring of G_2 gives the same color to v_2 and w_2. But now we can permute the colors so that both colorings give the same color to v and w. This gives us a proper k-coloring of G.

Case 4: In the remaining case, G is k-regular and there are two vertices v_1 and v_2, two units apart, such that $G - \{v_1, v_2\}$ is connected. Let x be a vertex incident to both v_1 and v_2. Now we repeat Case 2, but with a twist. Let T be a spanning tree of $G - \{v_1, v_2\}$ which is directed towards x. We label the vertices of T using the numbers $3, 4, 5, ...$ so that the labels increase along any directed path towards x.

Now we do the greedy algorithm. The algorithm assigns the same color to both x_1 and x_2, because these are not adjacent. The algorithm then works exactly as in Case 2 except when we get to the last step. It looks like we might get into trouble because we have already colored the k vertices incident to x. However, we have used the first color twice. Hence, there is still an available color in $\{1, ..., k\}$ which we can use for x. This gives us the desired k-coloring of G.