Linear implementation of steps needed to color one edge in the algorithmic proof of Vizing's theorem

Besides $neighbor(v,f) = \begin{cases} w & \text{if } c(\{v,w\}) = f \\ 0 & \text{otherwise} \end{cases}$ $\forall v \in V(G)$, $\forall f \in 1, \Delta(G)+1$

we use another data structure called missingColor at defined by $missingColor at(v) = f_v$ s.th. $neighbor(v, f_v) = 0$, $\forall v \in V(G)$

Initialization of missingColor at at the very beginning (no edges colored yet) as $missingColor at(v) = 1$ $\forall v \in V(G)$.

Update of the missingColor at

Then during the repeat loop to construct the sequence.

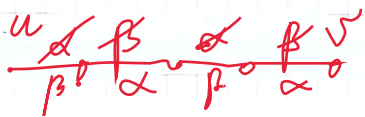
we set $b_i := missingColor at(y_i)$ and check whether $neighbor(x, b_i) = 0$. If not we set $y_{i+1} = neighbor(x, b_i)$ and go on with $b_{i+1} = missingColor at(y_{i+1})$ and so on.

When for some k $neighbor(x, b_k) = 0$ we change color of (x, y_k) from b_{k-1} to b_k and update $missingColor at(y_k) := b_{k-1}$, then change color of $\{x, y_{k-1}\}$ from b_{k-2} to b_{k-1} and update $missingColor at(y_{k-1}) = b_{k-2}$ and so on until $missingColor at(y_2) = b_1$.

Then we finally color $\{x,y\}$ by b.

At this point we actualize $missingColor at(x)$ by going through the row corresponding to x in $neighbor(x, f)$ and setting $missingColor at(x) = f_x$, where f_x is the first row s.th. $neighbor(x, f_x) = 0$. We get $missingColor at(y)$ similarly.

The update of missingColor at during the swapping colors along a path is simple: if the path is in $H(\alpha, \beta)$ and has endpoints u and v we just flip $missingColor at(\cdot)$ from α to β or from β to α for u and v, respectively



One can easily check that maintaining the correct values of missingColorAt( ) takes $O(n)$ value assignments per coloring of each v_i if done as described above.

Clearly we need to save also the sequences $y_1, \dots, y_k$ and $b_1, \dots, b_k$ for each edge $\{x,y\}$ (uncolored yet) but this can be done also in $O(n)$ time because the length of each of these sequences is bounded by the nr of vertices in the graph.

Finally observe that by using missingColorAt(.) it takes just a constant time to access a missing color at every vertex $y_1, \dots, y_k$ amounting at $O(n)$ time per each edge $\{x,y\}$ to be colored.

And answers the questions which remained open at the lecture of April 27.