[to be embellished!]

Constructing the proof of Vizing's Theorem

by Josyula R. Rao & Edsger W. Dijkstra

We shall construct the proof of the following theorem, due to V. G. Vizing.

Theorem For an undirected graph, without autoloops and without multiple edges, at whose vertices at most N edges meet, N+1 colours suffice for an edge colouring such that no two adjacent edges have the same colour. (End of Theorem.)

In the following "edge colouring" is short for an edge colouring with at most N+1 colours such that no two adjacent edges have the same colour. We shall demonstrate the existence of an edge colouring by designing a colouring algorithm.

The colouring algorithm starts with the vertices of the graph and adds its edges one at a time. At each step it constructs an edge colouring for the graph with the newly added edge; if necessary, such a step modifies colours of some older edges.

The step starts with a graph with an edge colouring, to which an uncoloured edge has been added. The fact that the number of colours exceeds the maximum number of edges meeting at any vertex can (and will) be exploited by assigning to each vertex its "originally free colour", i.e. one of the colours not carried by any of the edges meeting at that vertex. (In case of freedom, e.g. at the endpoints of the uncoloured edge, a choice is made; the assignment of the originally free colours remains constant all through the step.)

The existence of the originally free colours enables us to give the uncoloured edge such a colour c that at most one colour conflict is introduced, say at its endpoint X ; if so, this conflict can be resolved by uncolouring the other edge of colour c that is incident on X . This suggests the following invariant for our colouring step:

P: There is no colour conflict, there is at most 1 uncoloured edge, and, if present, the uncoloured edge is incident on X .

At the start of the step P is established by identifying X with one of the endpoints of the uncoloured edge. Once the originally free colours have been assigned as well, a first approximation of the rest of the step is the following program

```
{P} do there exists an uncoloured edge →
      |[ let the uncoloured edge be XY
       ; let the originally free colour of Y be y
       ; give XY colour y
       ; if no colour conflict at X → skip
         [] colour conflict at X →
              |[ let XZ have colour y with Z ≠ Y
               ; uncolour XZ ]|
         fi {P}
      ]|
   od {there is a complete edge colouring}
```

Let the originally free colour of X be x. We note that as long as each y differs from x, the repetition maintains that no edge incident on X has colour x. Therefore, an execution of the repeatable statement with $x = y$ would introduce no conflict at X and no uncoloured edge would remain. Hence we strengthen the invariant with

Q: If there exists an uncoloured edge, no edge incident on X has colour x.

If the above repetition terminates, it does the job, but we cannot prove that it terminates because it may fail to do so. To correct the program, we should provide an alternative to "give XY colour y". From

Q we deduce that "give XY colour x" introduces no conflict at X. Would it introduce a conflict at Y? Only if another edge, YW say, has colour x. Note that no edge incident on Y has colour y — the reason being that of all edges incident on Y, only XY may have changed its colour —. The conflict at Y can hence be removed by giving colour y to YW, but this at the risk of creating a conflict at W. Can the conflict be pushed further until it disappears?

To answer this question we consider paths along which colours x and y alternate, more precisely, we call X and one of its neighbours Y "linked" if and only if there exists from Y to X a path that differs from edge XY and whose colouring satisfies the grammar $(xy)^*$. If X and Y are not linked, the conflict can be pushed until it disappears by inverting the colours on the alternating x-y path emanating from Y.

We incorporate this alternative in the second version of our program. In anticipation of the termination argument we introduce a variable C of type bag of colours.

```
{P ∧ Q} C := ∅
; do there exists an uncoloured edge →
    |[ let the uncoloured edge be XY
    ; let the originally free colour of Y be y
    ; if X and Y linked →
          give XY colour y
         ; if no colour conflict at X → skip
          [] colour conflict at X →
               |[ let XZ have colour y with Z ≠ Y
               ; uncolour XZ ; C := C + {y} ]|
         fi
    [] X and Y not linked →
          invert colours on x-y path emanating from Y
         ; give XY colour x
    fi
   ]|
od {there is a complete edge colouring}
```

The repeatable statement either colours all edges or adds a colour to bag C . Since the number of colours is finite, it suffices to show that each colour occurs at most once in C , and for the purpose of termination we can restrict our attention to those executions of the repeatable statement that add a colour to C.

Let us consider the execution of the repeatable statement in which uncoloured edge XU is coloured with U's originally free colour

u and u is added to C for the first time. Since X and U were linked, there was a path $(xu)^*$ from U to X. Hence, with the exception of U, all vertices of the path have an originally free colour $\neq u$, and, as long as XU remains coloured u, the vertices reachable from X via (the prefix of) a path $(ux)^*$ are the ones on the path and hence have an originally free colour different from u.

Edge XU does not lose its color u before an uncoloured edge XV is encountered with originally free colour v of V $= u$, an encounter which is necessary for the addition of a second u to C. This addition, however, does not occur because X and V are not linked: V is <u>not</u> reachable from X via (the prefix of) a path $(ux)^*$, because vertices thus reachable have, in contrast to V, an originally free colour $\neq u$. And this completes our construction of the proof.

* * *

The above has been written in response to a request by Robert E. Tarjan to EWD: "I include a proof that is neither clear nor elegant, in the hope that you will rise to the challenge and find the right proof. The theorem to be proved is Vizing's theorem". The above proof contains the same ingredients as the published proofs we have seen: they avoid mathematical induction over N and, more importantly, are independent of the structure of the graph. Our improvements seem the following.

(i) We have avoided a lot of notational immodesty such as $\mu_{\beta r_j}[a,z]$ – Claude Berge – and $P_{s_0,t}(x_k)$ – Béla Bollobás – .

(ii) By approaching the problem as a programming task and designing the program in an orderly fashion we could provide almost complete heuristics.

(iii) Most of the case analyses of the published proofs have been avoided; moreover we have avoided the reductiones ad absurdum.

We thank the members of the Austin Tuesday Afternoon Club, who joined us for two afternoons in our explorations.

Austin, 21 February 1990

Josyula R. Rao
prof. dr. Edsger W. Dijkstra
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712-1188
USA