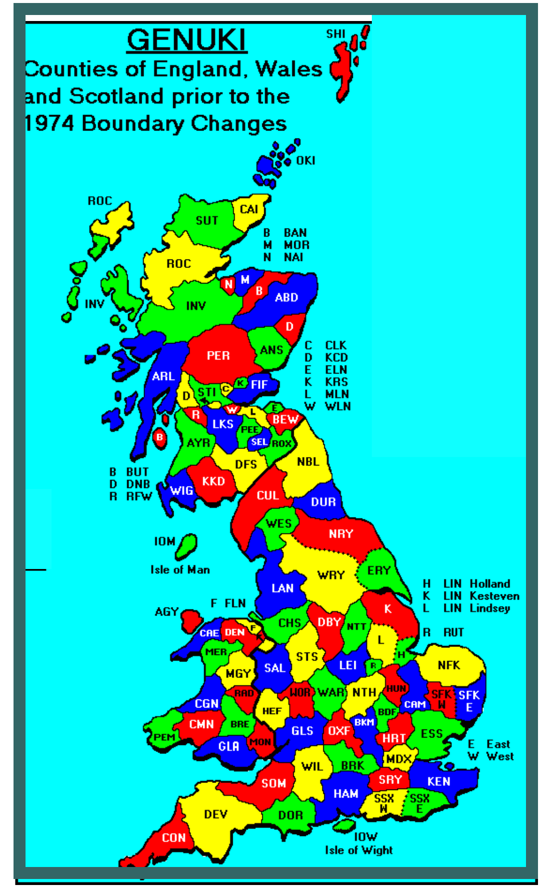
Verifying the Four Colour Theorem

Discrete Mathematics *Fall 2023*

Huazhong University of Science and Technology
School of Automation and Artificial Intelligence

170 years of history…

- 1852 Conjecture (*Guthrie* → *DeMorgan*)
- 1878 Publication (*Cayley*)
- 1879 First proof (*Kempe*)
- 1880 Second proof (*Tait*)
- 1890 Rebuttal (*Heawood*)
- 1891 Second rebuttal (*Petersen*)
- 1913 Reducibility, connexity (*Birkhoff*)
- 1922 Up to 25 regions (*Franklin*)
- 1969 Discharging (*Heesch*)
- 1976 Computer proof (*Appel & Haken*)
- 1995 Streamlining (*Robertson & al.*)
- 2004 Self checking proof (*Gonthier*)



So what about it ?

- It shows software can be as reliable as math.
- It's been done by applying computer science to mathematics.
- The art of computer proving is maturing.

Outline

- The Four Colour Theorem
  - what it says
  - how it's proved
- Computer proofs
  - how it's done

The Theorem

open and connected

disjoint subsets of R x R

Every simple planar map can be colored with only four colors

∃good covering map

with at most four regions

adjacent regions covered with different colors

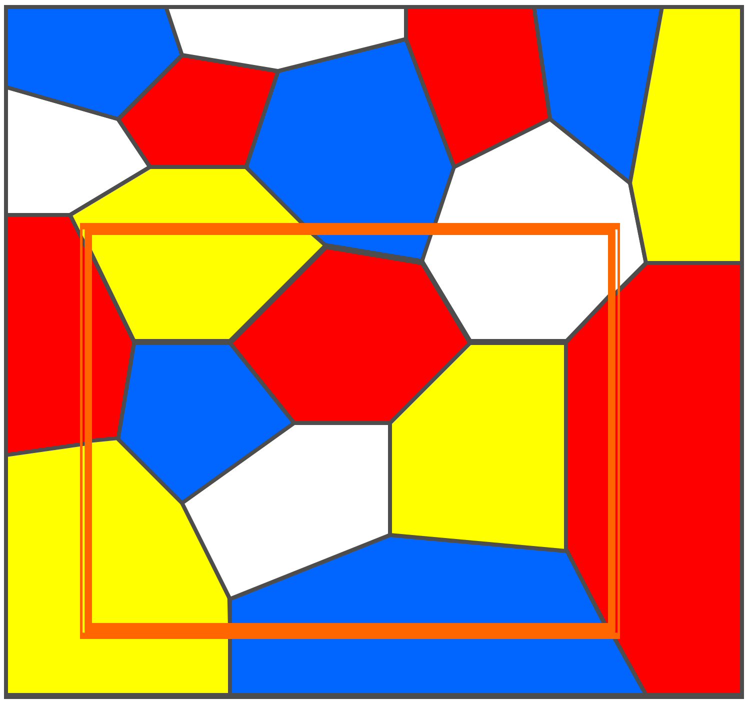
have a common border point that is not a corner

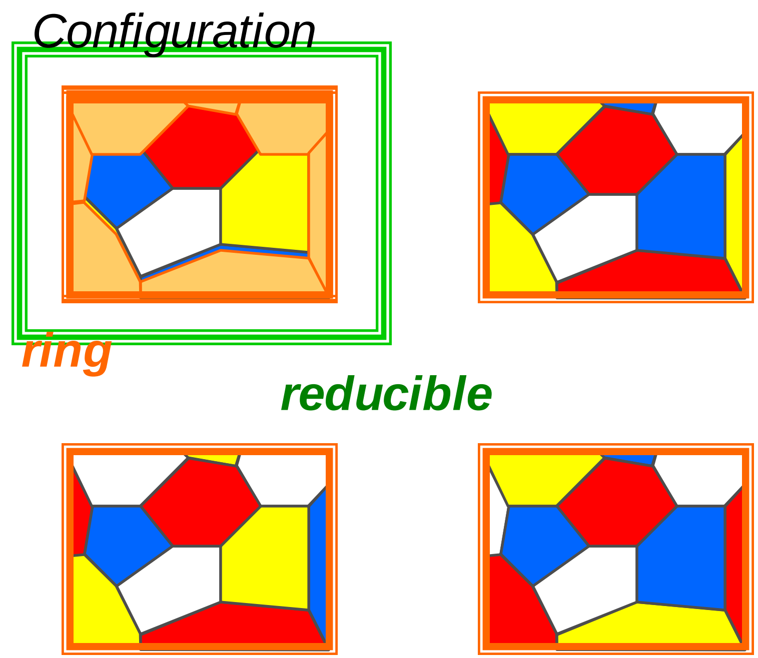
touches more than two regions

Outline

- The Four Colour Theorem
  - what it says
  - **how it's proved**
- Computer proofs
  - how it's done

Colouring by induction





4-CT's proof (by kempe)   proof by induction

① node = 1 , straightforward.

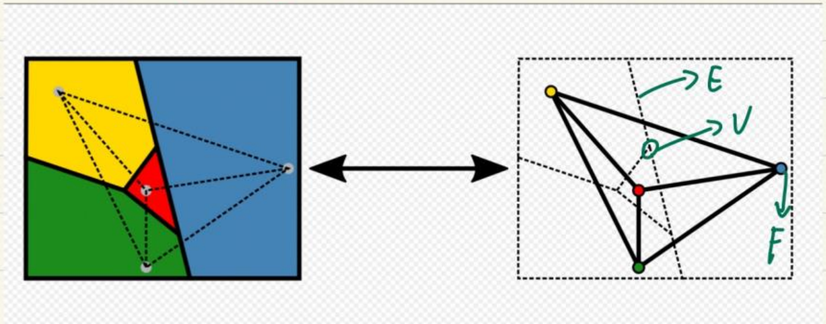
② suppose node = n 时成立, 下证 node = n+1 的情况:

1. 引理: $\forall$ planar graph, 存在度小于等于5的节点   反证法:

V顶点数, E为边数, F为区域数

①. $2E \geq 3F$ 每条边隔开2个区域
每个区域至少由3条边围成

②. $2E \geq 6V$ (假设每个顶点有6条边)



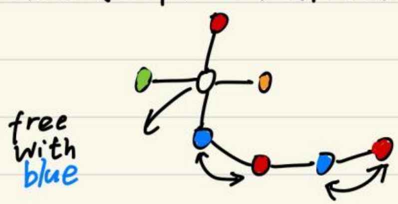
$\Rightarrow V+F \leq \frac{1}{3}E + \frac{2}{3}E = E$

欧拉公式: $V+F = E+2$

证明: n个节点时成立 4-CT, 下证 n+1个结点的情况, 令 degree $n \leq 5$

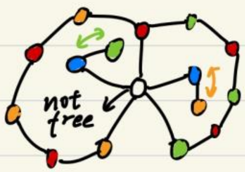
case 1: degree = 4

if. Red and Blue 之间不存在 Chain
即不存在左图这种情况, 那么非常简单. 交换颜色即可



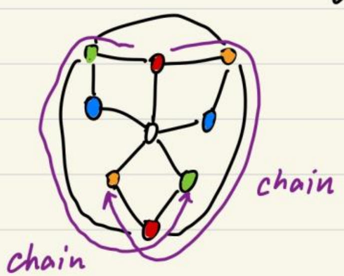
if Red and Blue 之间存在 Chain
那么 Yellow and green 之间一定不存在 chain
因为是平面图, 不可交叉

Case 2. degree = 5



Kempe thought in this case, the blue neighbor can be free.

Heawood's Rebuttal



exchange后 发生冲突!

Progress in verification

1976 A & H

2 books
1000's of
figures

?

IBM 370
reducibility

1995 RSST

35-page
paper

?

C program
reducibility
unavoidability

2005 MSR

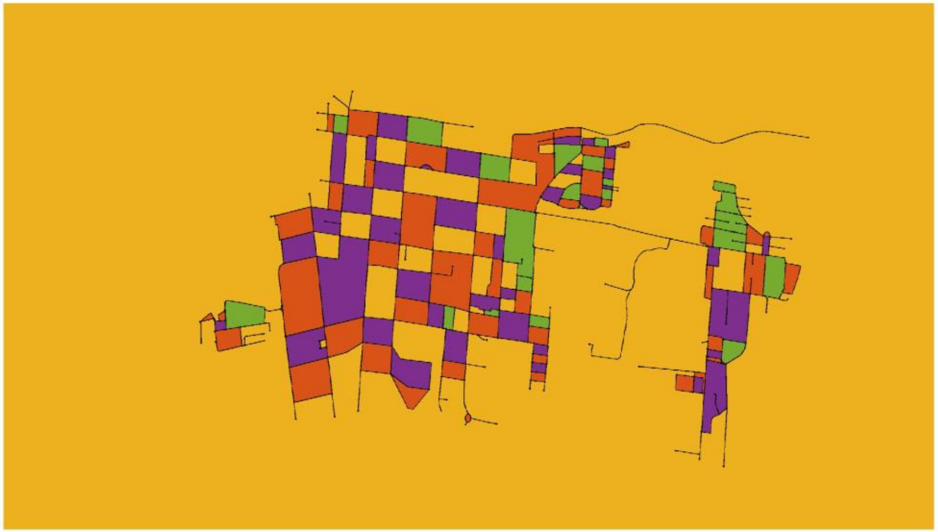
35 lines of
definitions

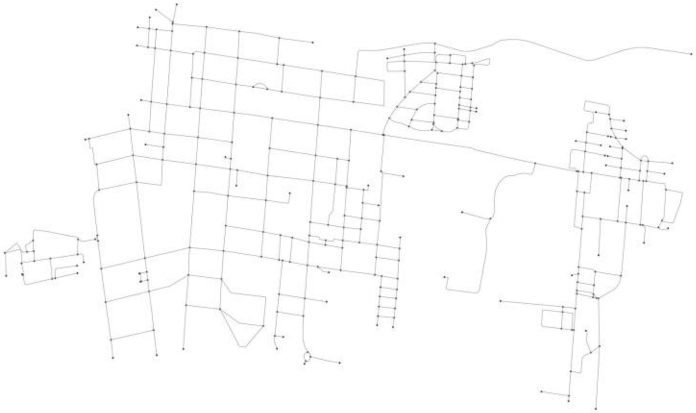
Gallina
reducibility
unavoidability
graph theory
topology
data structures
…

Outline

- The Four Colour Theorem
  - what it says
  - how it's proved
- Computer proofs
  - **how it's done**

华科路网4着色





问题陈述

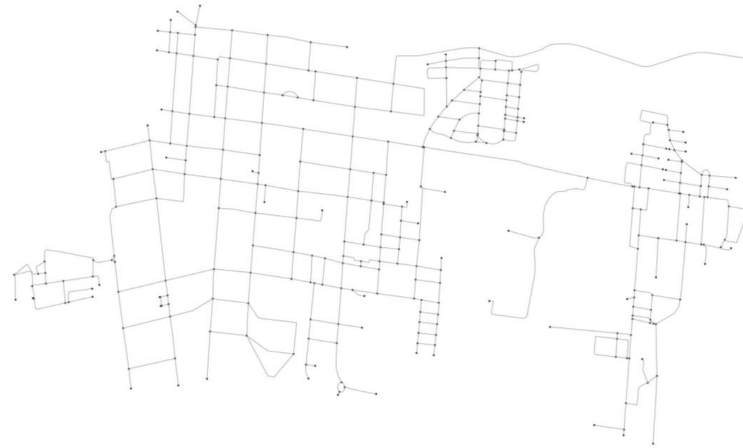
四色定理：任何平面图都**是4-可着色的**

华科路网可看作平面图

华科路网是**4-可着色的**

构造邻接矩阵

目标：得到地图各个面（对偶图的点）之间的邻接关系



方式：

1.通过数据作出华科路网二值图像

2.Matlab函数bwlabel 分割连通区域

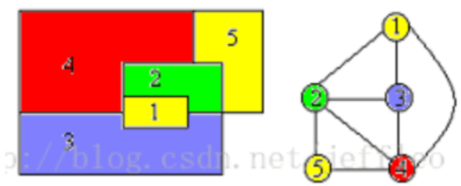
3.遍历所有边，找到所有面相邻关系，完成邻接矩阵

128x128 double

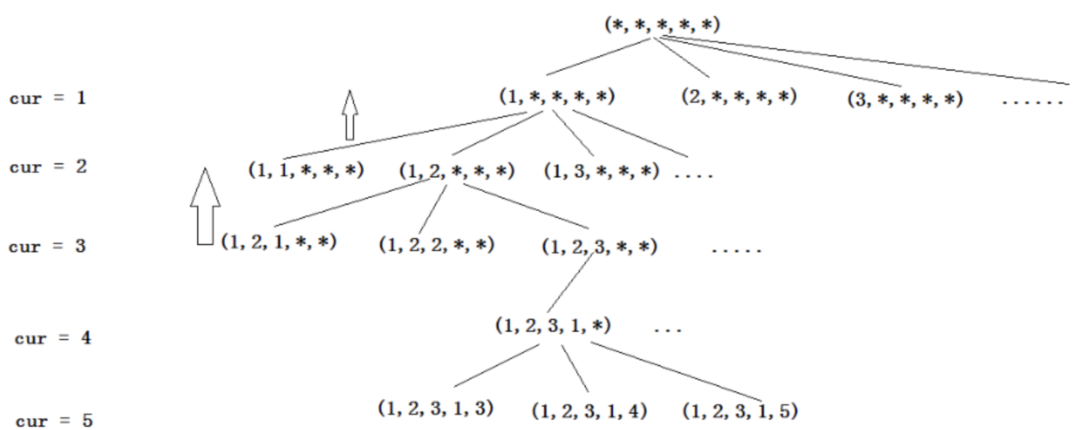
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	1	1	1	1	1	1	1	1	1	1	1
2	1	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	1	0	0	0	0	0	0	0	0	0
4	1	0	1	0	1	1	1	0	0	0	0	0	0
5	1	0	0	1	0	0	1	0	0	0	0	0	0
6	1	0	0	1	0	0	1	0	0	0	0	0	0
7	1	0	0	1	1	1	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0	0	1	0	0
9	1	0	0	0	0	0	0	0	0	1	1	1	0
10	1	0	0	0	0	0	0	0	1	0	0	0	0
11	1	0	0	0	0	0	0	1	1	0	0	0	0
12	1	0	0	0	0	0	0	0	1	0	0	0	1
13	1	0	0	0	0	0	0	0	0	0	0	1	0
14	1	0	0	0	0	0	0	1	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	1	0	0
16	0	0	0	0	0	0	0	0	1	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	1	0
18	1	0	0	0	0	0	0	0	0	0	0	1	0
19	1	0	0	0	0	0	0	0	0	0	0	0	0
20	1	0	0	0	0	0	0	0	0	0	0	0	0
21	1	0	0	0	0	0	0	0	0	0	0	0	0
22	1	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0
26	1	0	0	0	0	0	0	0	0	0	0	0	0
27	1	0	0	0	0	0	0	0	0	0	0	0	0
28	1	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0
37	1	0	0	0	0	0	0	0	0	0	0	0	0
38	1	0	0	0	0	0	0	0	0	0	0	0	0
39	1	0	0	0	0	0	0	0	0	0	0	0	0
40	1	0	0	0	0	0	0	0	0	0	0	0	0

回溯法染色其优化

回溯法（探索与回溯法）是一种选优搜索法，又称为试探法，按选优条件向前搜索，以达到目标。但当探索到某一步时，发现原**先**选择并不优或达不到目标，就退回一步重新选择，这种走不通**就退回再走的技**术为回溯法，而满足回溯条件**的某个**状态**的点称**为“**回溯点**”。



对于上面这图，颜色数量为4，顶点数为5，求得的**解答树**如下：



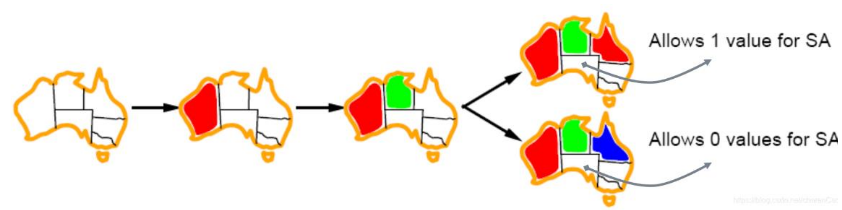
回溯法染色其优化

回溯法如果只是无信息地盲目搜索，在最坏情况下需要达到指数级别的时间复杂度，这就是一个灾难，根本无法求解。注意到，**根据**约束条件我们可以得到一定的启发式信息，利用这些启发式**信息**进行启发式的回溯搜索可以大大提高速度。而启发式算法一**般都会采取剪枝的策略**，这样就可以减少空间搜索树的分支，从**而提高搜索速率**。

思路：**尽可能早的**发现矛盾，从而不会在矛盾的路上走远

优化1：**怎么**选择节点
（1）.优先选择可选颜色少的节点
（2）.优先选择度大的节点

优化2：选定节点后怎么选择颜色
选择留给相邻节点颜色更多的节点



优化3：**提前**发现矛盾
当相邻节点只有**1种**颜色可选，马上选择，不断迭代