Exercices
(session 1)

Exercice 1

Weak Consensus Algorithm

Algorithm of process p **with input value** v

begin

write (p,v)

snapshot

let **View** = $((p_1, v_{p1}), \ldots, (p_k, v_{pk}))$ */*the view of p*/*

write (p,**View**)

snapshot

let **W** = $((p_1, \mathbf{View}_{p1}), \ldots, (p_m, \mathbf{View}_{pm}))$ */*the meta-view of p*/*

let **View*** = $\bigcap_{i=1,\ldots,m} \mathbf{View}_{pi}$ */*smallest view in meta-view*/*

if for every $i \in [1,n]$ such that $v_i \in$ **View***, $\mathbf{View}_i \in \mathbf{W}$ holds

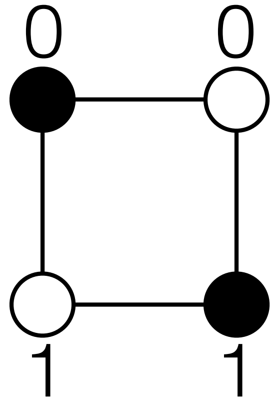
then decide smallest value in **View***

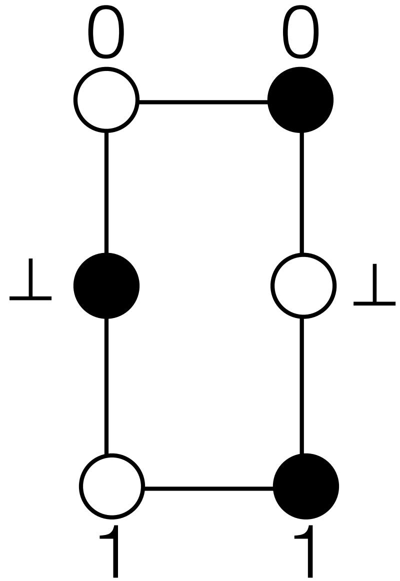
else decide $\perp$

end

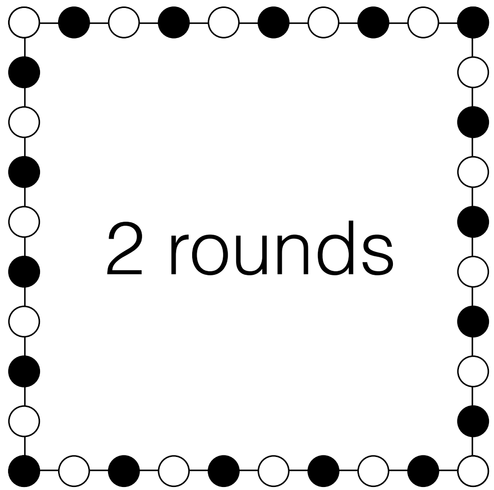
Input/Output Complexes of Weak Consensus

Input Complex





Output Complex

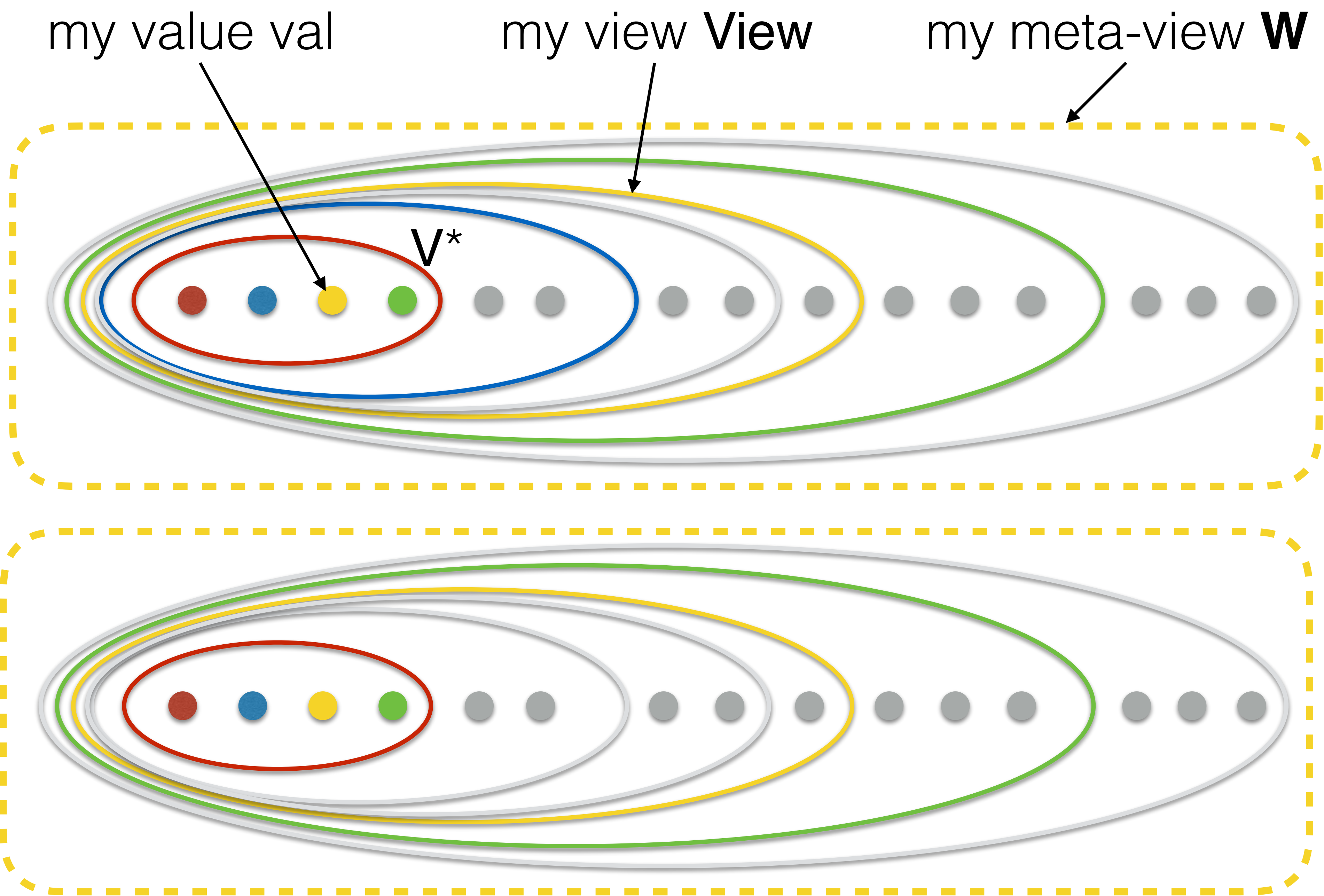


Questions

1. In the case of 2 processes, what is the protocol complexe P_1 after one write/snapshot instruction, and what is the protocol complexe P_2 after two write/snapshot instructions?

2. Explain what one can map P_2 to the output complex, but not P_1 (for ID-oblivious algorithms).

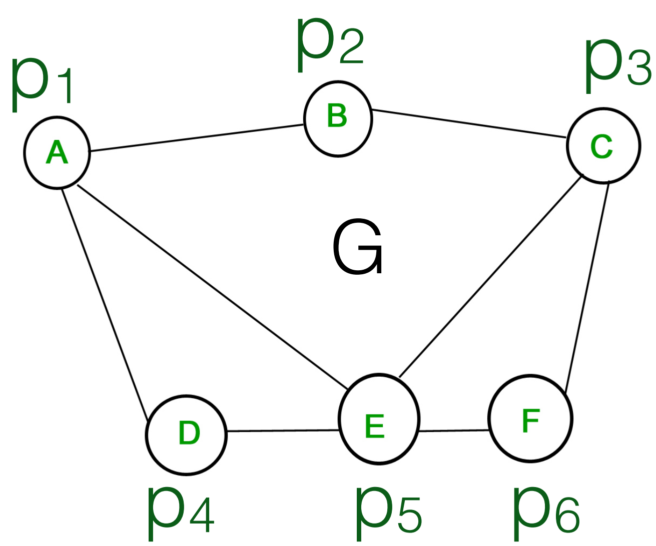
Intuition



Exercice 2

Setting

- Processes occupy nodes of a graph G
- Synchronous model
- Communication by messages
- No failures



Domination Number

A dominating set in $G=(V,E)$ is a set $D \subseteq V$ such that every node not in D has a neighbor in D.

Definition G has dominating number d if the minimum size of a dominating set in G is d.

Questions

1. Assume $m = n$ possible input values. Show that k-set agreement in G requires at least r rounds where r is the smallest integer such that G^r has dominating number $\leq k$.

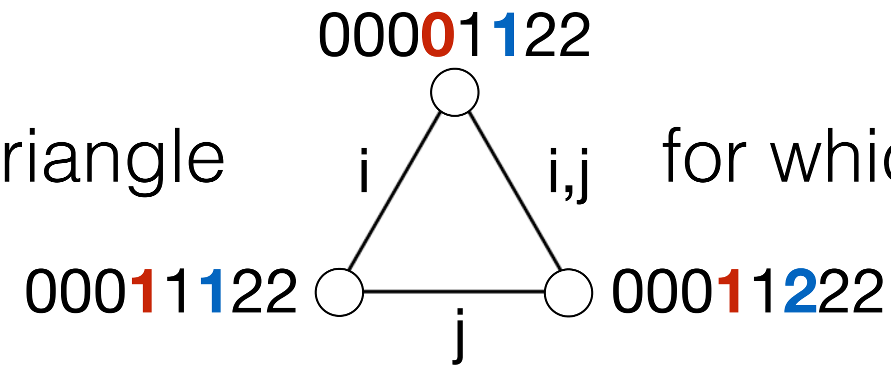
2. Assume $m = 3$ possible input values. Show that 2-set agreement in G requires at least r rounds where r is the smallest integer such that G^r has dominating number ≤ 2.

m=3 and k=2

Input configuration: $v_1 v_2 \ldots v_n$ with $v_i \in \{0,1,2\}$

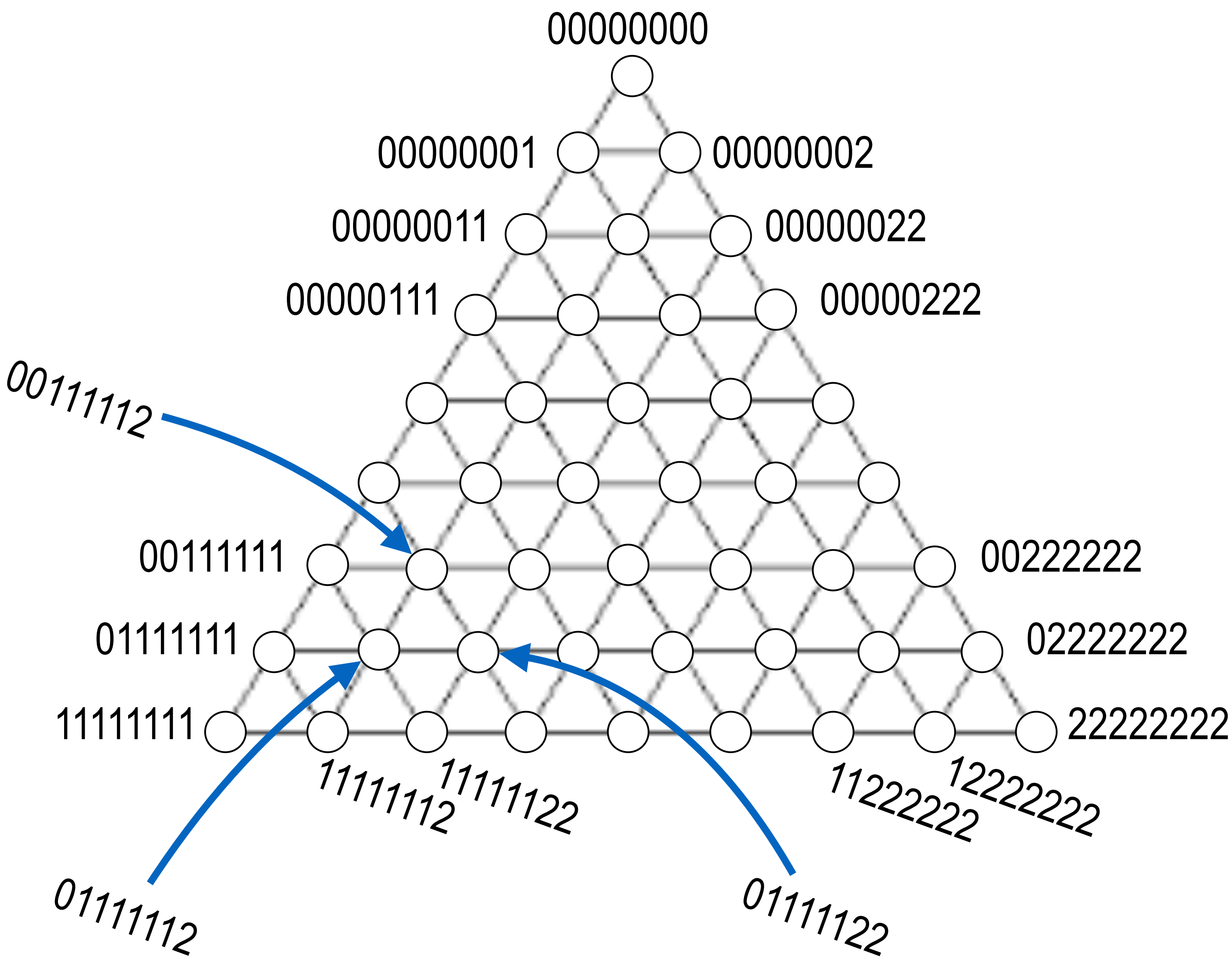
For every i,j, there exists process q that is not dominated by $\{p_i,p_j\}$.

➥ triangle



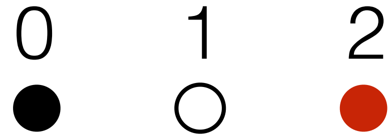
for which q does not see the difference

These triangles can be glued together

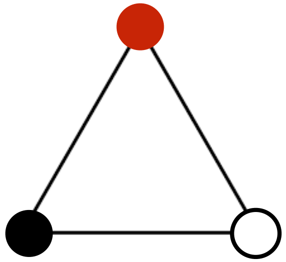

00000000
00000001
00000002
00000011
00000022
00000111
00000222
00111112
00111111
00222222
01111111
02222222
11111111
22222222
11111112
11111122
11222222
12222222
01111112
01111122

Assume existence of an algorithm.

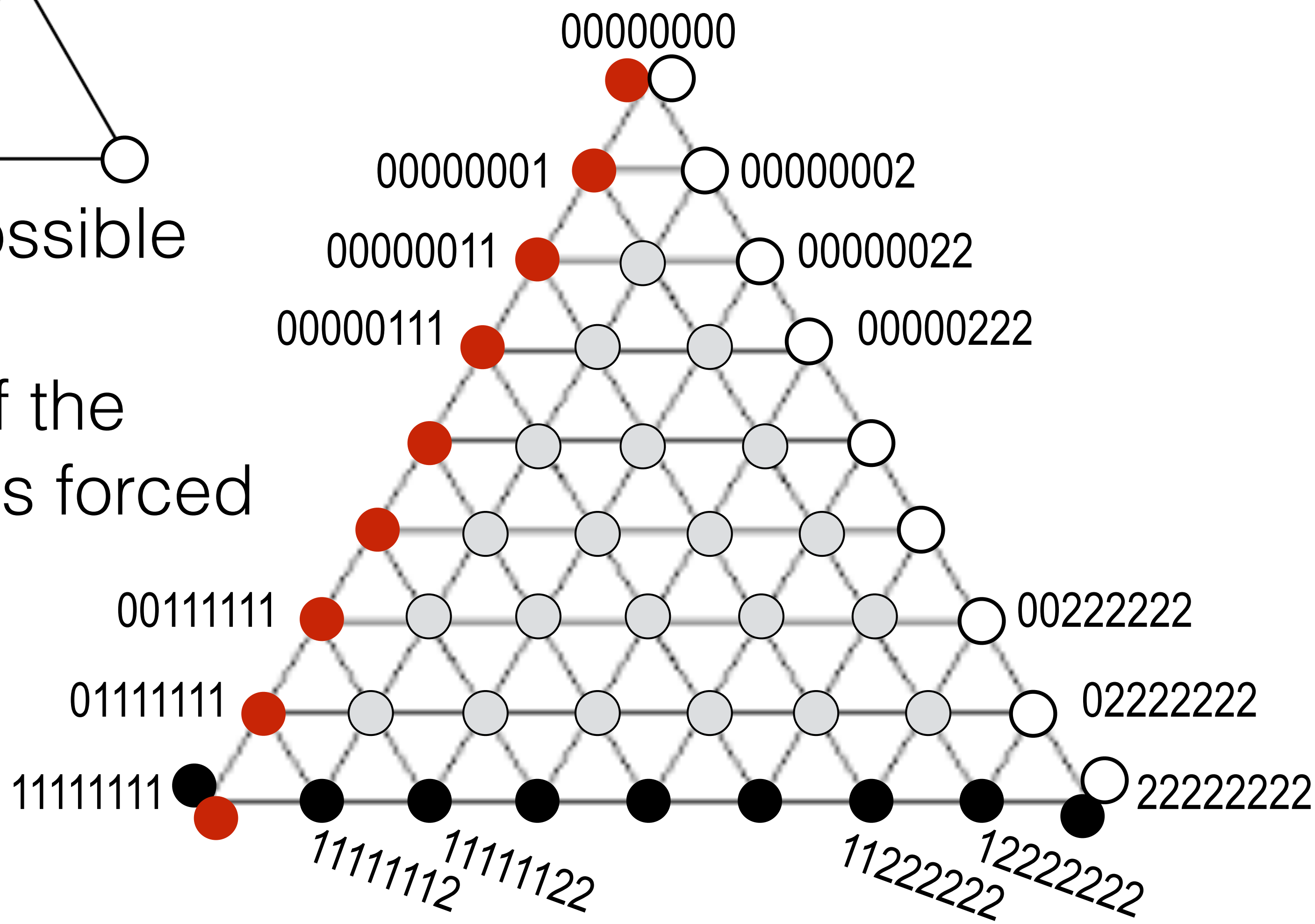
➠ Color each node by the discarded color:



➠ Remark:



Impossible



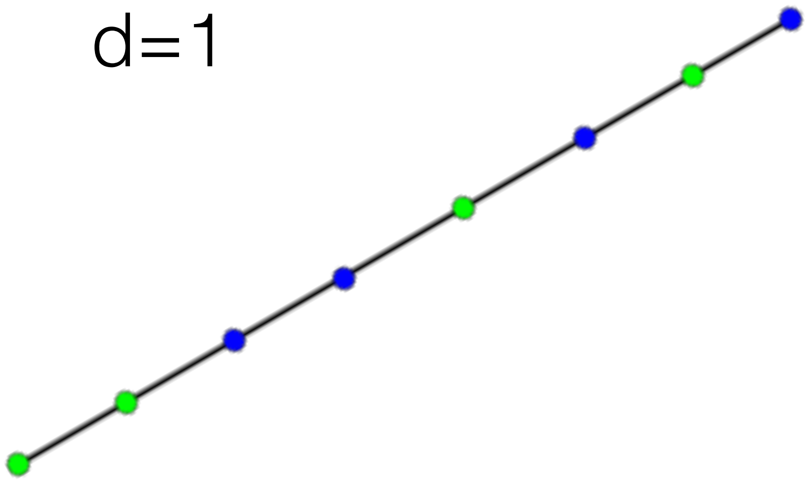
The coloring of the border nodes is forced

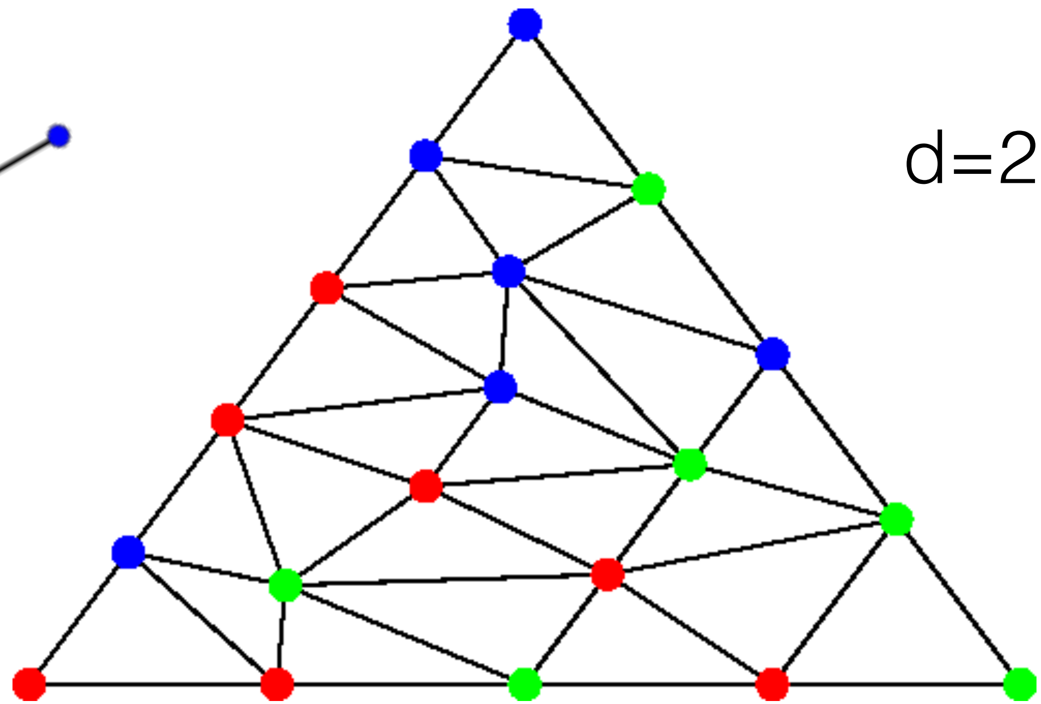
Claim: There must exist a triangle with the 3 colors.

Sperner’s Lemma

Lemma Every Sperner coloring of a triangulation of an d-dimensional simplex contains a cell colored with a complete set of colors.

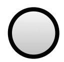
d=1

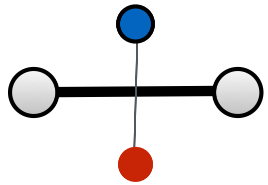


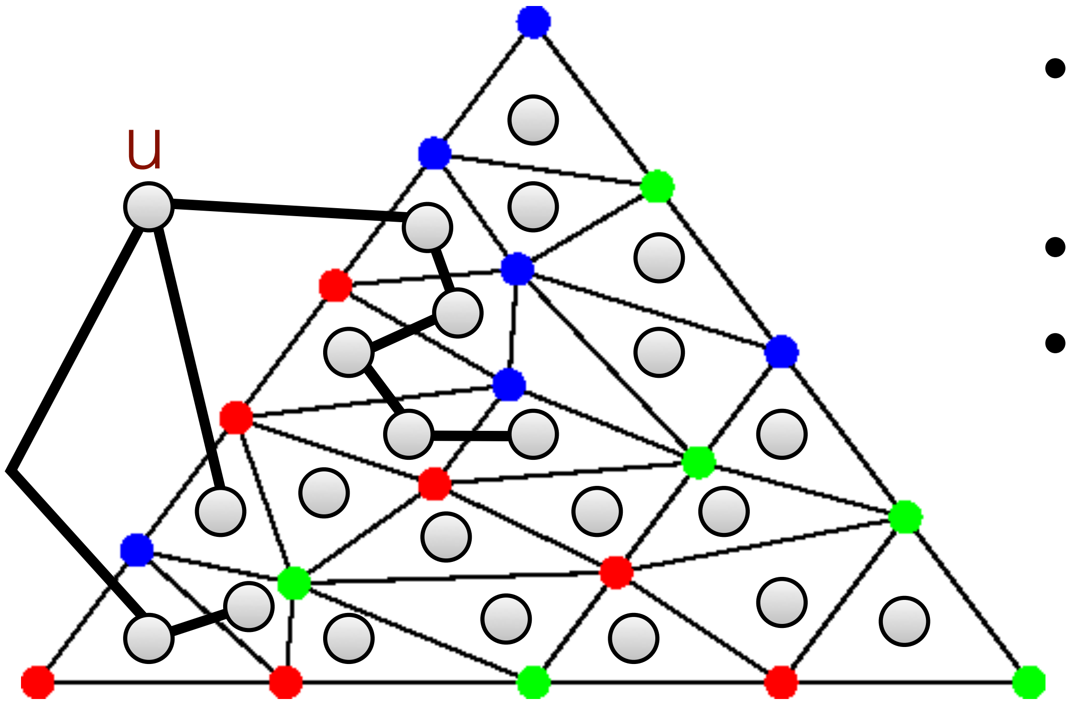


d=2

Proof sketch

V(G) = {  }

E(G) = {  }



- By induction on d, deg(u) is odd
- $\sum_{v \in V(G)} \deg(v) = 2\,|E(G)|$
- triangles with 1 or 2 colors induce nodes with even degrees (0 or 2)

⇩

odd number of
3-colored triangles

❑

Exercice 3

Renaming

- n processes start with unique names taken from a large name space [1,N]
- they must decide new unique names from a name space [1,m] as small as possible.
- **Theorem** Renaming with $m = 2n-1$ names is possible wait-free

Renaming Algorithm

```
Algorithm for process pᵢ with initial name xᵢ
begin
    y ← 1     /* pᵢ will try to rename itself y */
    stop ← false
    while stop = false do
        write(xᵢ,y) in pᵢ's register (erasing old value)
        S ← snapshot all registers
        let S = {(xⱼ,yⱼ) : j ∈ J} for some J ⊆ {1,…,n}
        if ∄ j ∈ J∖{i}  such that y = yⱼ then
            newname ← y    /* pᵢ adopts y as new name */
            stop ← true
        else
            r ← rank of xᵢ in {xⱼ, j ∈ J}
            y ← rth integer not in {yⱼ, j ∈ J∖{i} }
end
```

Questions

1. Show that no two processes decide on the same new name
2. Show that the new names are in the range [1,2n-1]
3. Show that every correct process terminates