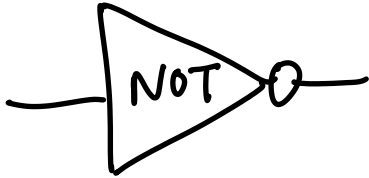


Making circuits out of stuff

KYE, 4/19/22.

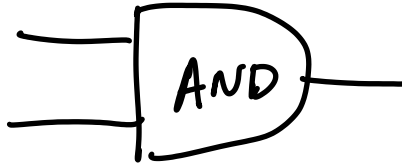
CIRCUITS

$\text{NOT } x = 1$ iff:
 $x = 0$



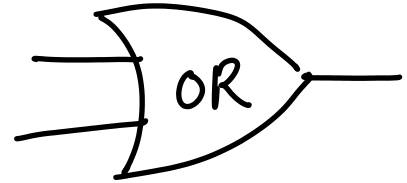
x	NOT x
0	1
1	0

$x \text{ AND } y = 1$ iff:
 x, y both 1



x	y	x AND y
0	0	0
0	1	0
1	0	0
1	1	1

$x \text{ OR } y = 1$ iff:
at least one of $x, y = 1$



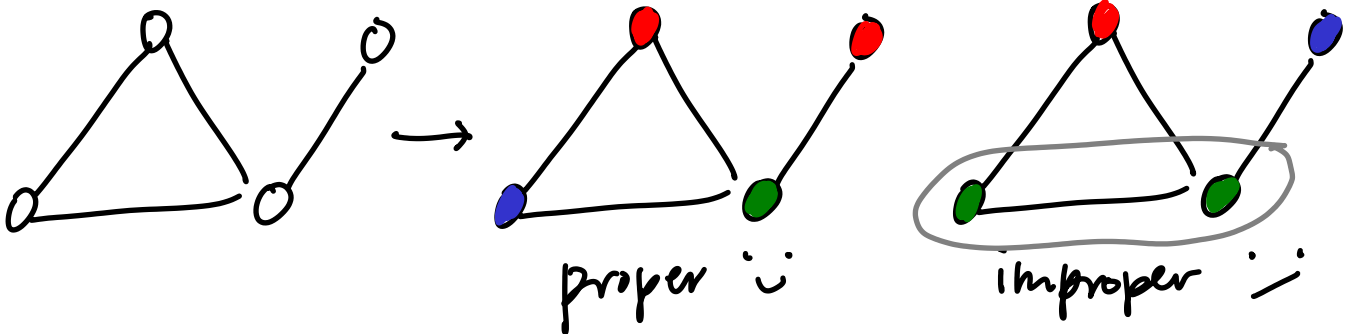
x	y	x OR y
0	0	0
0	1	1
1	0	1
1	1	1

GRAPH 3-COLORINGS

Coloring: a map
 $\kappa: \text{vertices} \rightarrow \{0, 1, 2\}$

κ is proper iff:

$\forall \text{ neighbors } (u, v), \kappa(u) \neq \kappa(v).$

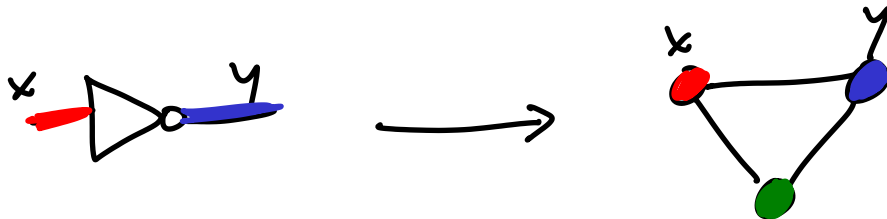
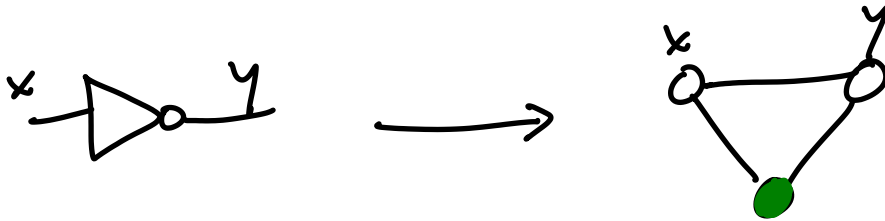


QUESTION .

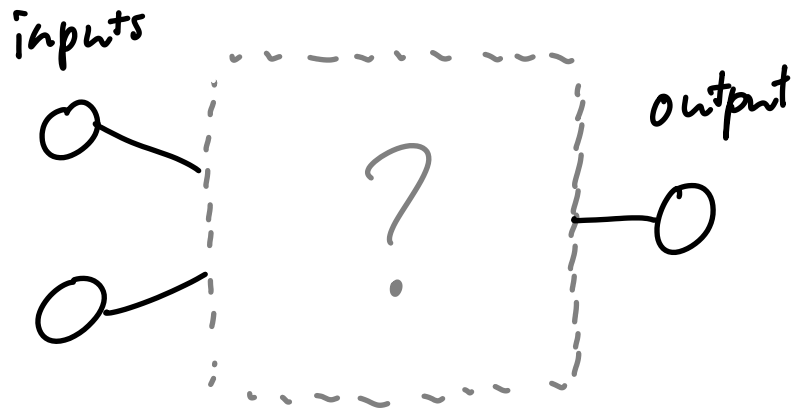
Can we emulate circuits
using graph colorings?

EXAMPLE: NOT gate.

● = 0 ● = 1 (● is a "helper" color)



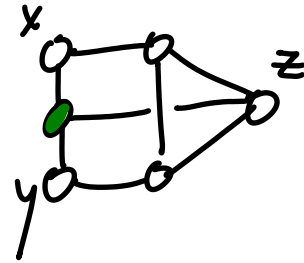
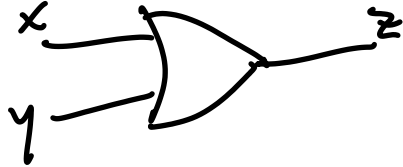
GOAL: using $\text{blue circle} = 0$ and $\text{red circle} = 1$, construct graphs whose inputs force outputs according to boolean rules



Can we make an OR gate?

Goal: get z 's color to always be x OR y

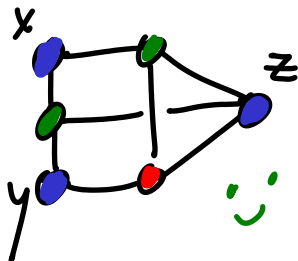
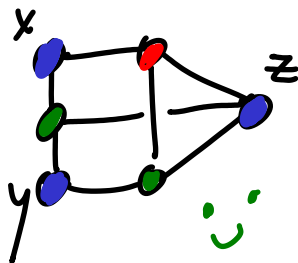
Last time I said:



WRONG! WRONG! WRONG! Why?

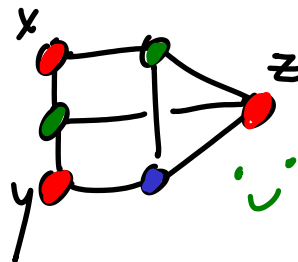
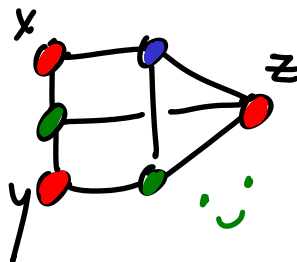
$x=0$

$y=0$



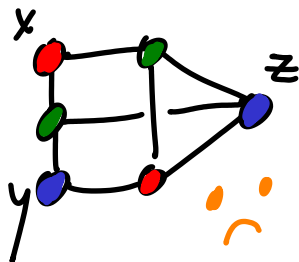
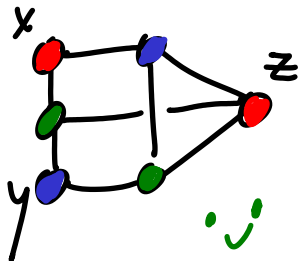
$x=1$

$y=1$



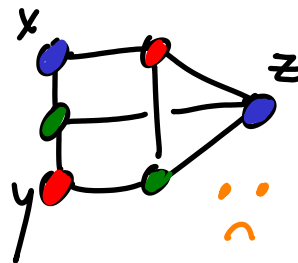
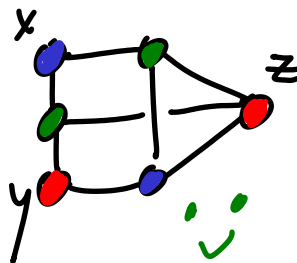
$x=1$

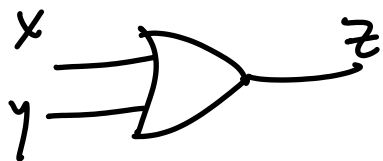
$y=0$



$x=0$

$y=1$

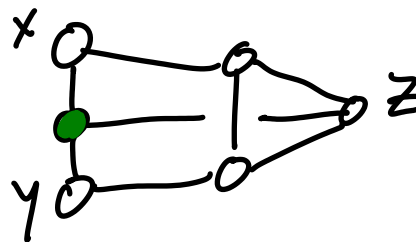




x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

must be 1 iff $x \text{ OR } y$

"fake OR" gate



x	y	z
0	0	0
0	1	0/1
1	0	0/1
1	1	1

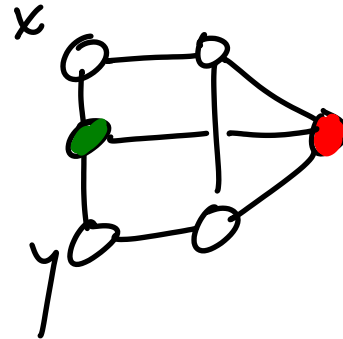
can be 1 iff $x \text{ OR } y$

Alternate idea: use multiple fake-OR
gates to constrain possible relations
between input & output.

What does this do?

x	y	
0	0	0
0	1	0/1
1	0	0/1
1	1	1

impossible!

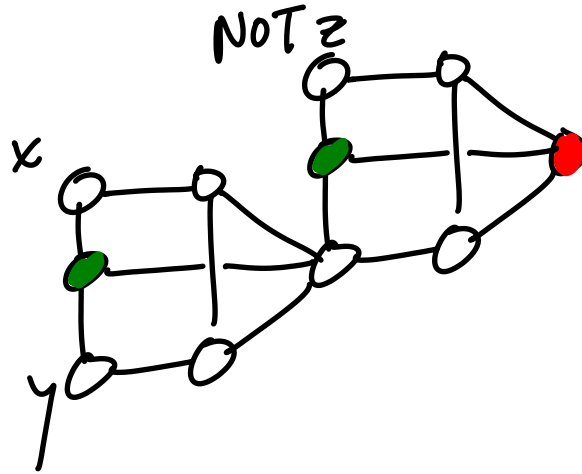


colorable iff
 $x \text{ OR } y$

Also: $x=0 \Rightarrow y=1$

$y=0 \Rightarrow x=1$

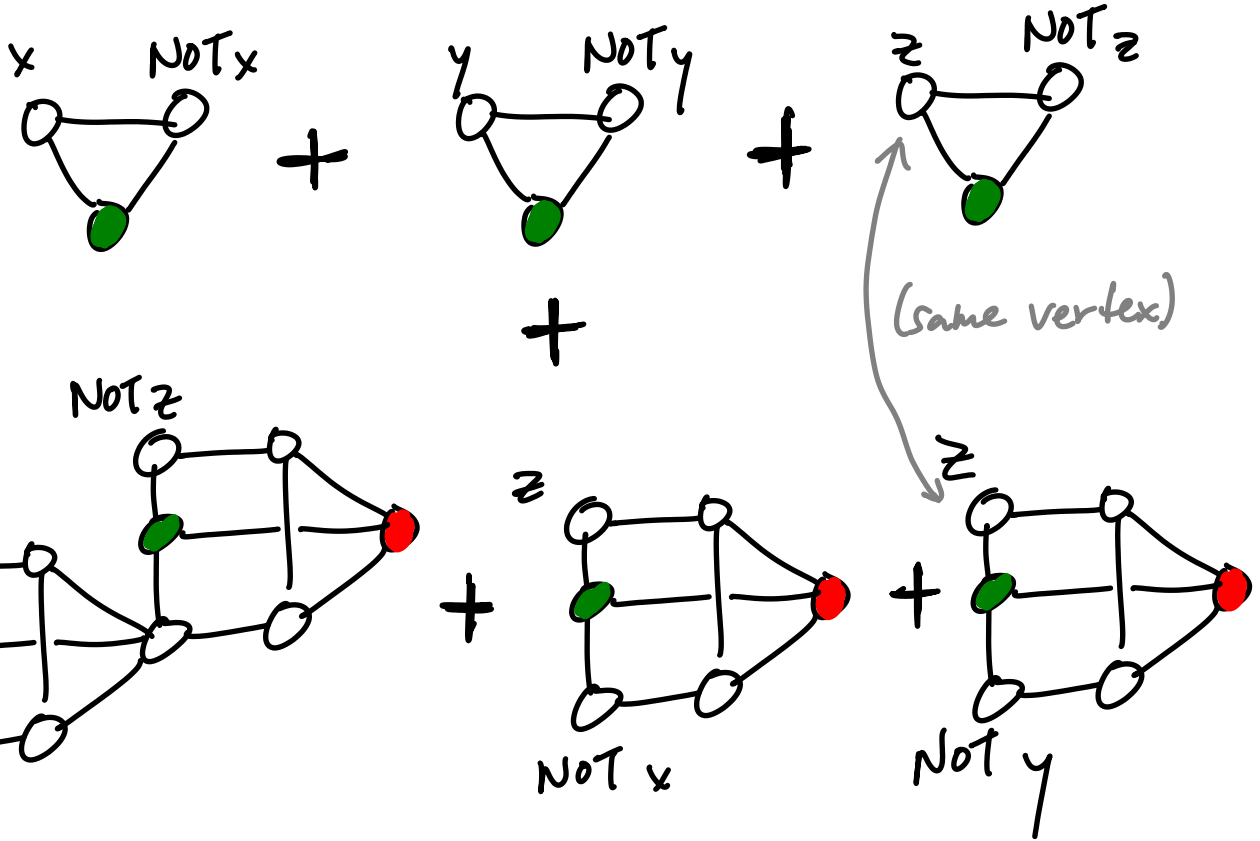
What about this?

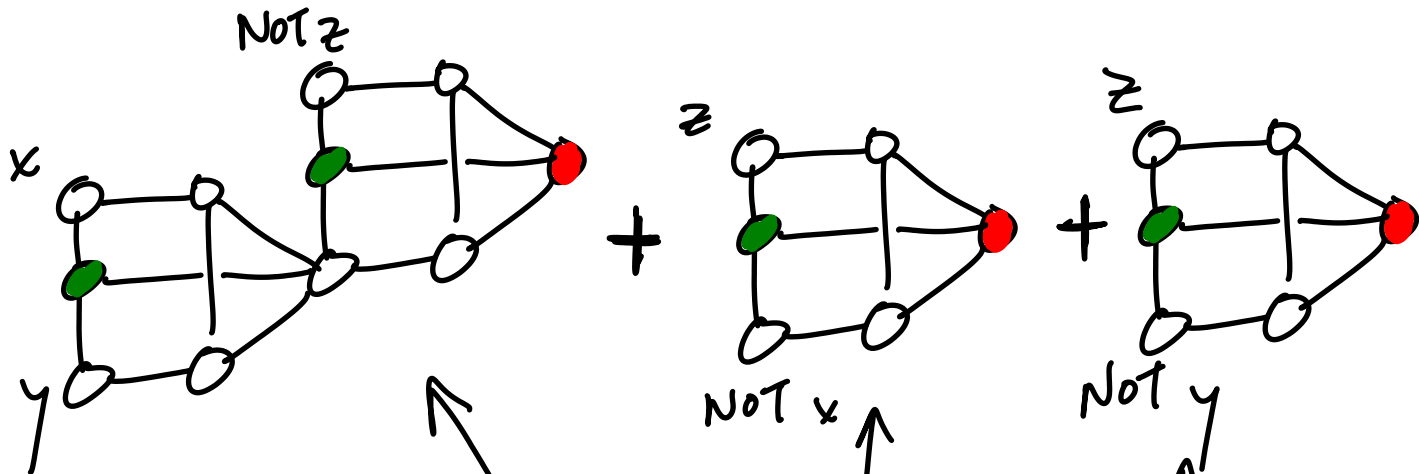


colorable iff $(\text{NOT } z) \text{ OR } (x \text{ OR } y)$

$$z = 1 \Rightarrow x \text{ OR } y = 1$$

The OR-gate redemption

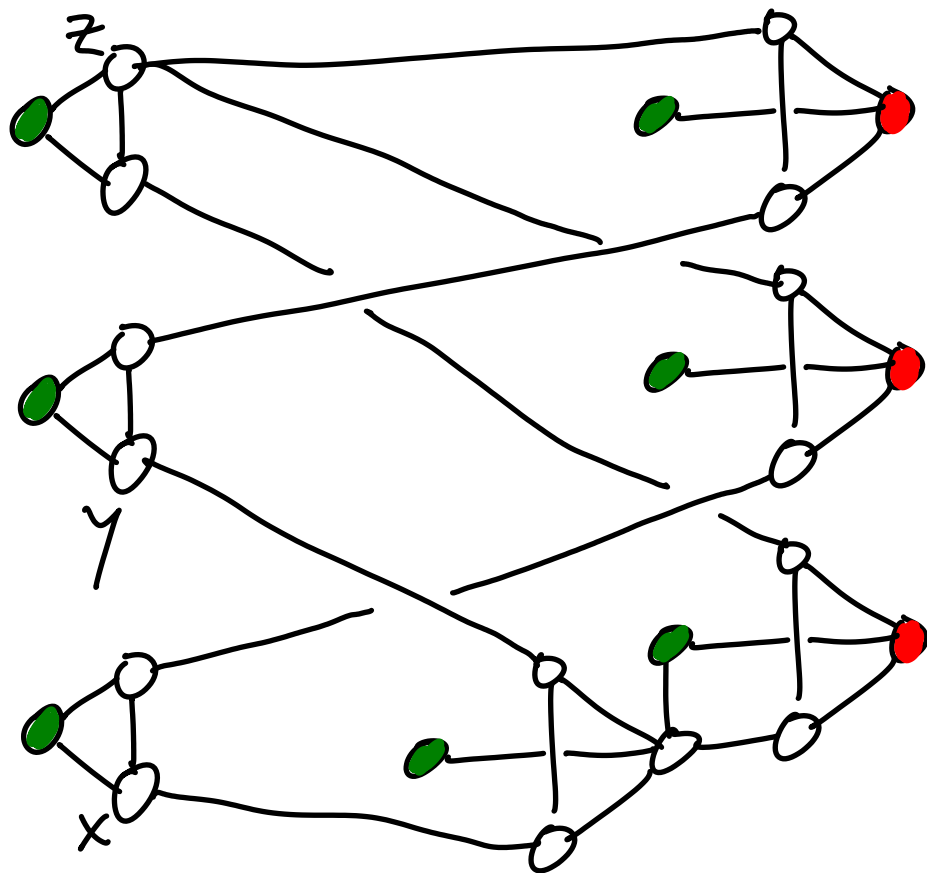


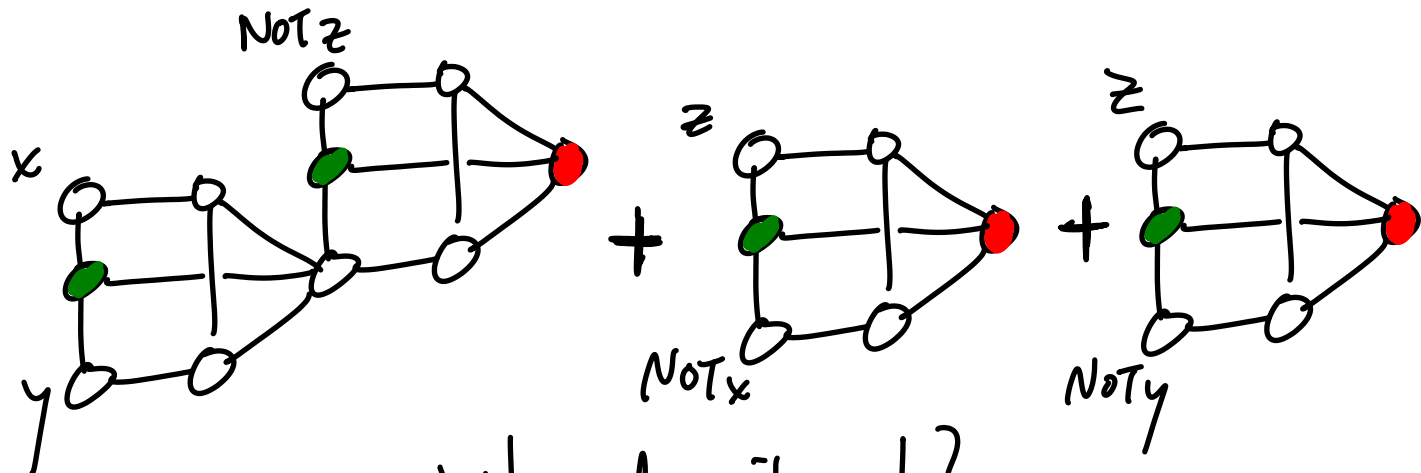


If both $x, y = 0$: forces $z = 0$

If $x = 1$: forces $z = 1$

If $y = 1$: forces $z = 1$

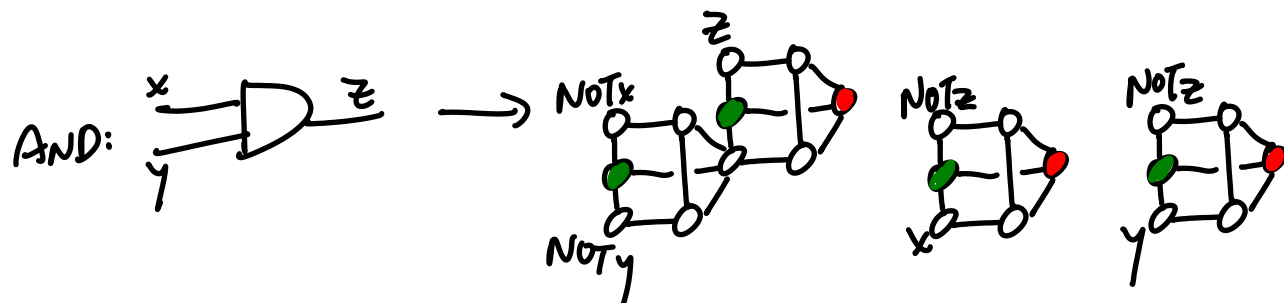
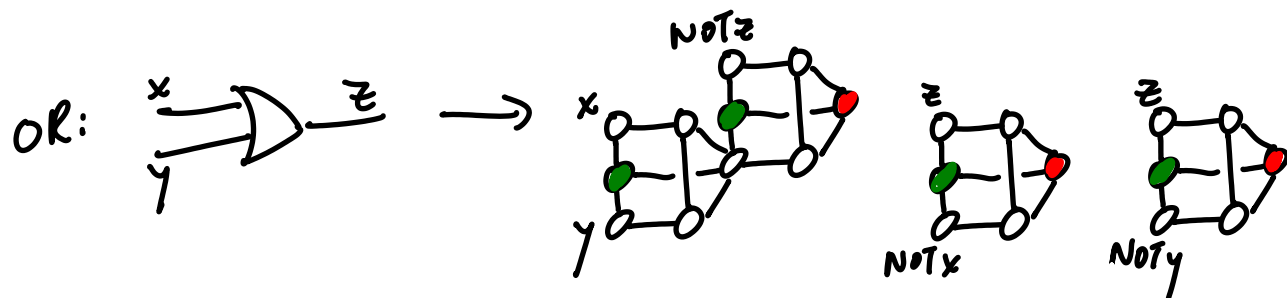
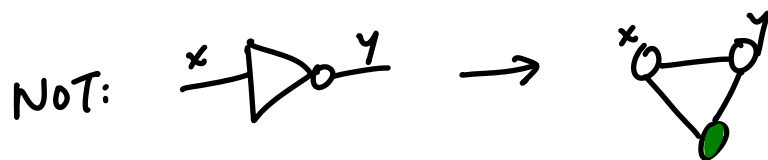




Why does it work?

$$z = x \text{ OR } y \iff \begin{aligned} &(\text{Not } z \text{ OR } x \text{ OR } y) \\ &\text{AND } (z \text{ OR } \text{Not } x) \\ &\text{AND } (z \text{ OR } \text{Not } y) \end{aligned}$$

which you can derive by FOIL'ing!
(which means you could do it too :D)



CIRCUIT & GRAPH GAMES

Circuit game:

two players A, B take turns assigning inputs.

A wins if final output = 1, B wins if output = 0.

Graph coloring game:

two players A, B take turns coloring vertices, following properness constraints.

A wins if B has no proper moves, vice versa.

If all turns played, A wins.

PUZZLES & GAMES

k-turn circuit games: Σ_k P-complete



k-turn graph coloring games: also Σ_k P-complete!