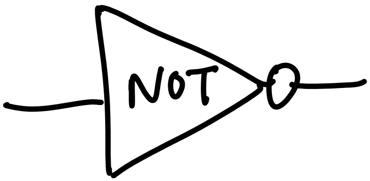
# CIRCUITS

NOT $x = 1$ iff:

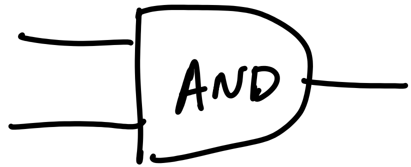
$x = 0$




| $x$ | NOT $x$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

$x$ AND $y = 1$ iff:

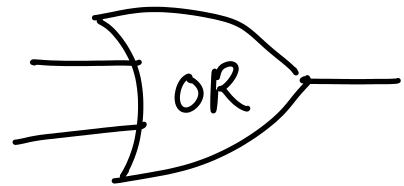
$x, y$ <u>both</u> $1$




| $x$ | $y$ | $x$ AND $y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$x$ 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 |

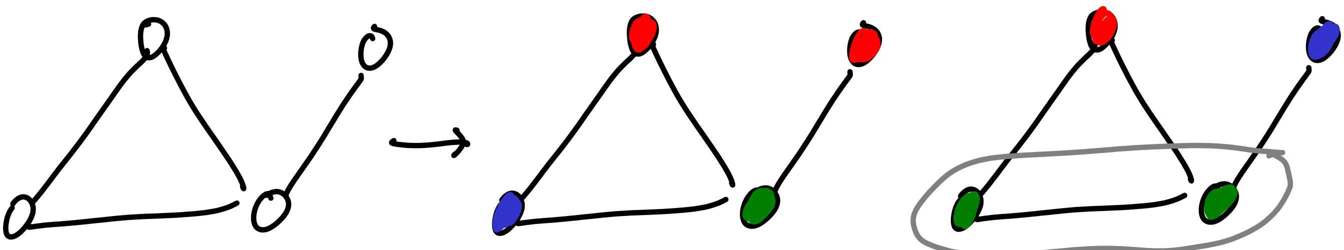
# GRAPH 3-COLORINGS

**Coloring**: a map

$\kappa$: vertices $\to \{0, 1, 2\}$

$\kappa$ is **proper** iff:

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



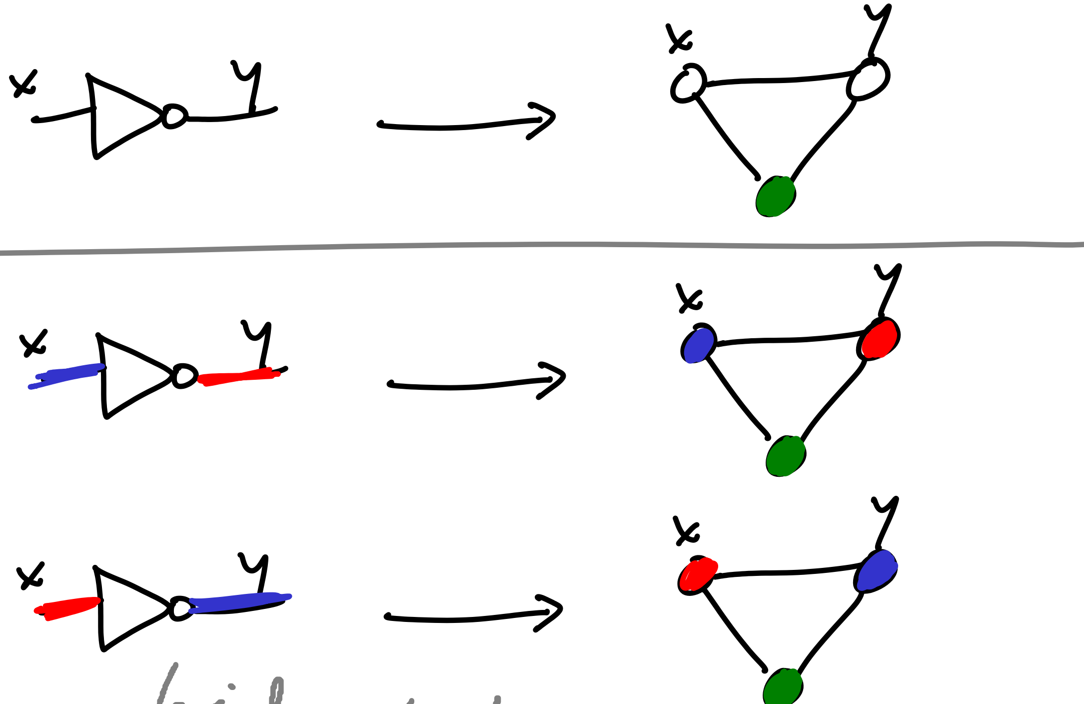
proper :) improper :(

# QUESTION.

Can we emulate circuits using graph colorings?

# EXAMPLE: NOT gate.

 = 0  = 1 ( is a "helper" color)




(aside: △ always 3 col)

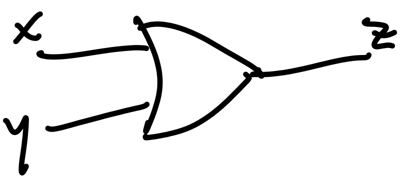
todo State

goal precisely.

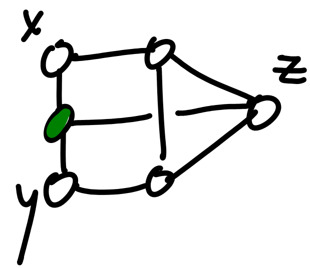
# Can we make an OR gate?

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

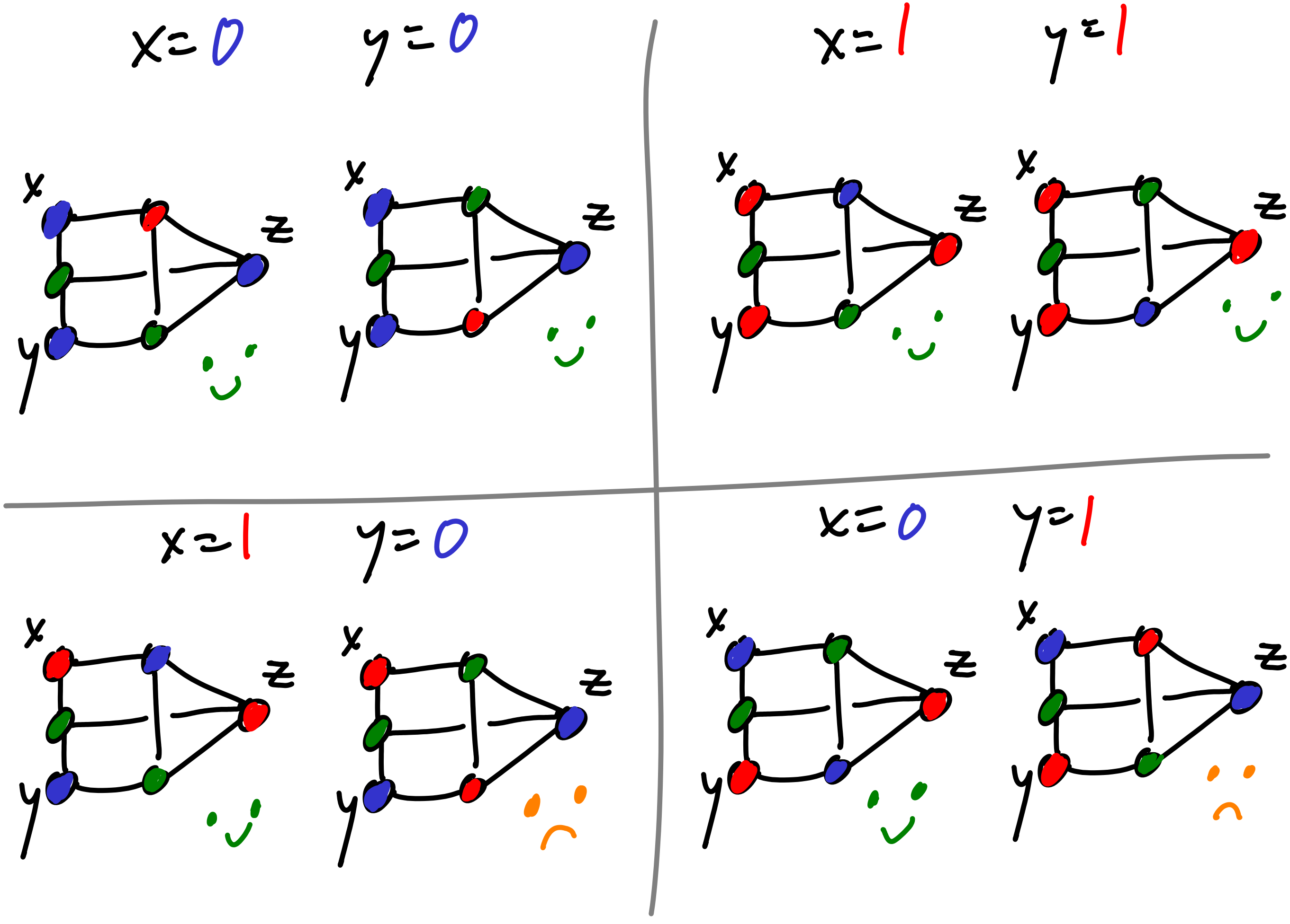
Last time I said:



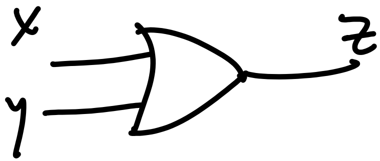

→

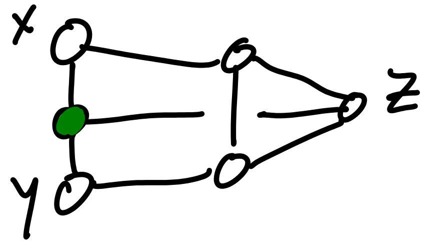



WRONG! WRONG! WRONG! Why?



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 |

| x | y | z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0/1 |
| 1 | 0 | 0/1 |
| 1 | 1 | 1 |

TODO correct

there is a

fix!

use that to "constrain" relation btwn

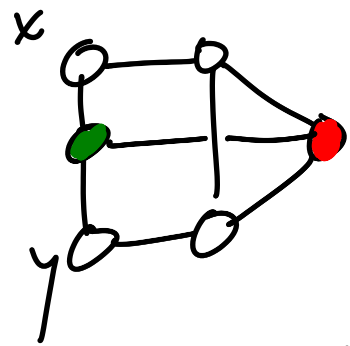
x y z.

show examples

# What does this do?

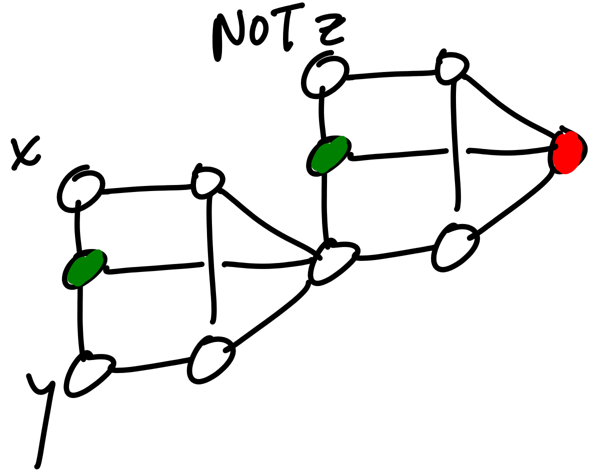
| x | y | |
|---|---|---|
| ~~0~~ | ~~0~~ | ~~0~~ |
| 0 | 1 | 0/1 |
| 1 | 0 | 0/1 |
| 1 | 1 | 1 |

impossible



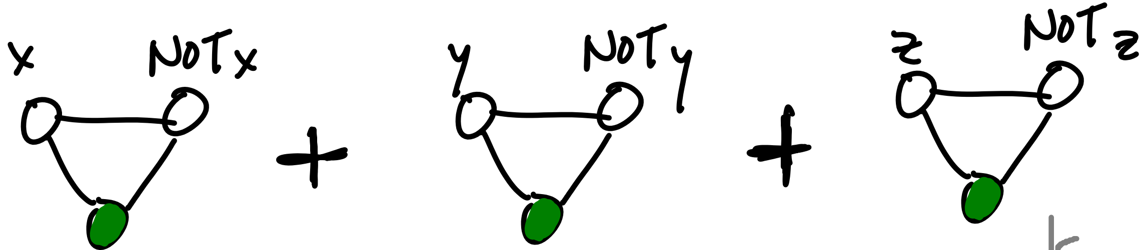

$\Longrightarrow$ colorable iff x OR y

# What about this?

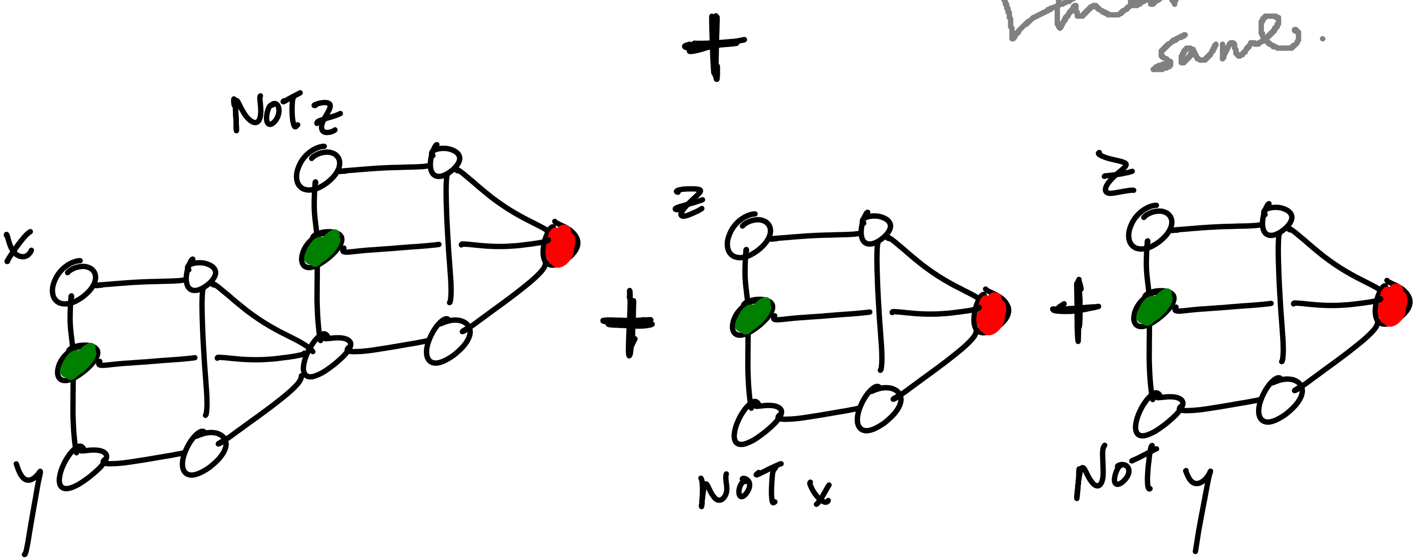


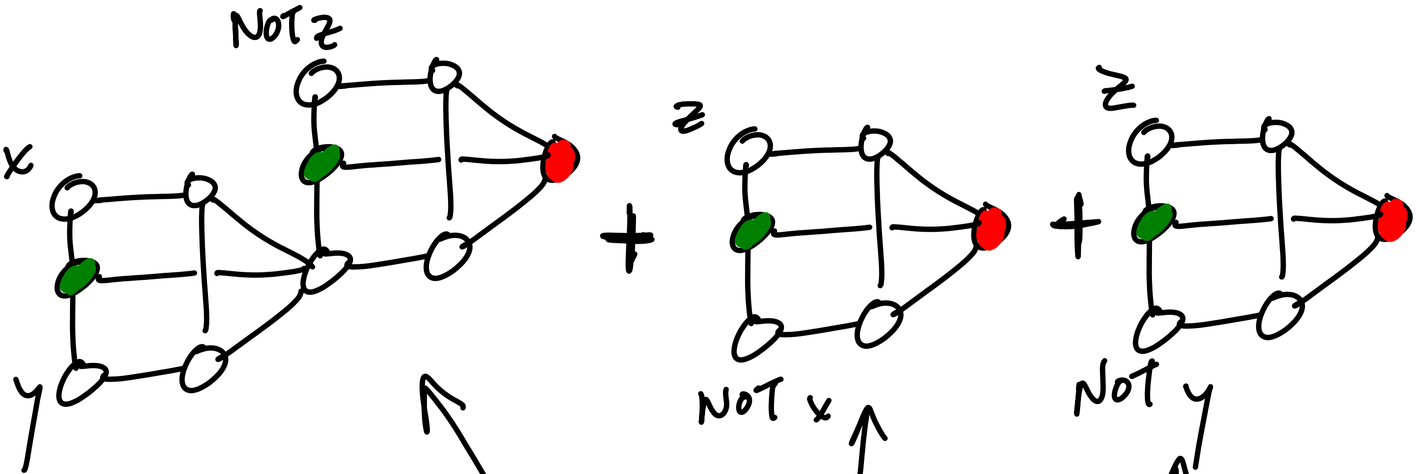

$\Rightarrow$ colorable iff (NOT z) OR (x OR y)

# The OR-gate redemption




remark
same.

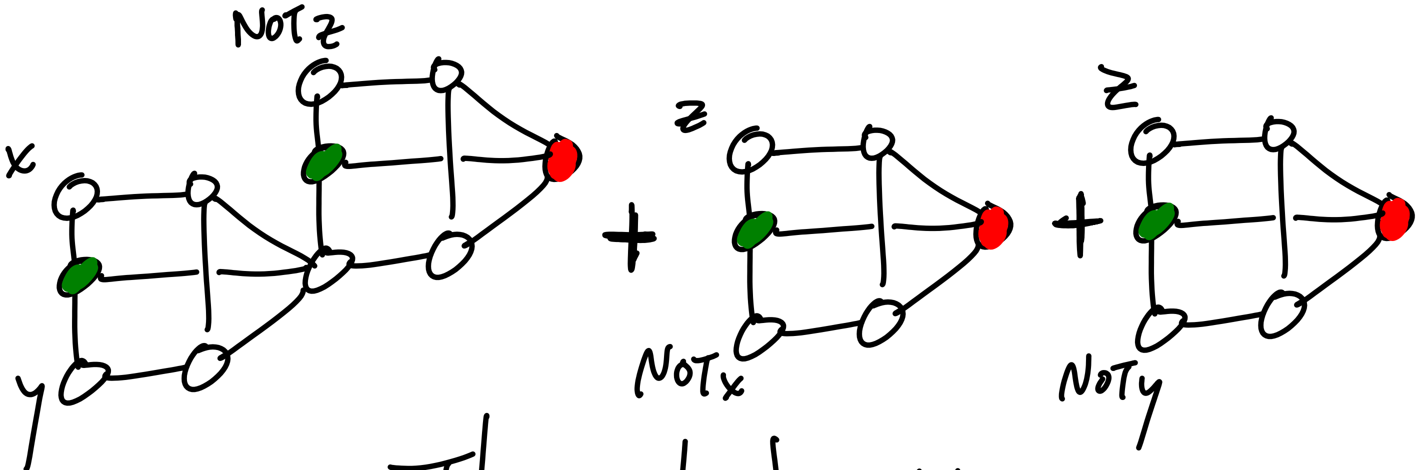





If both $x, y = 0$: forces $z = 0$

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

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



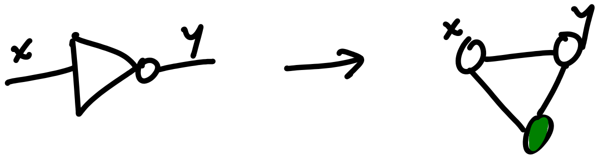

This works because:

$$z = x \text{ OR } y \iff \begin{array}{l} (\text{NOT } z \text{ OR } x \text{ OR } y) \\ \text{AND } (z \text{ OR NOT } x) \\ \text{AND } (z \text{ OR NOT } y) \end{array}$$

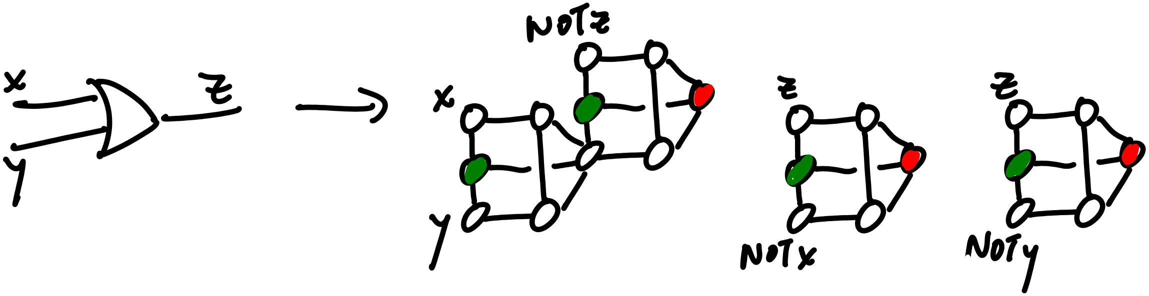
which you can derive by FOIL'ing!

(which means you could do it too :D)

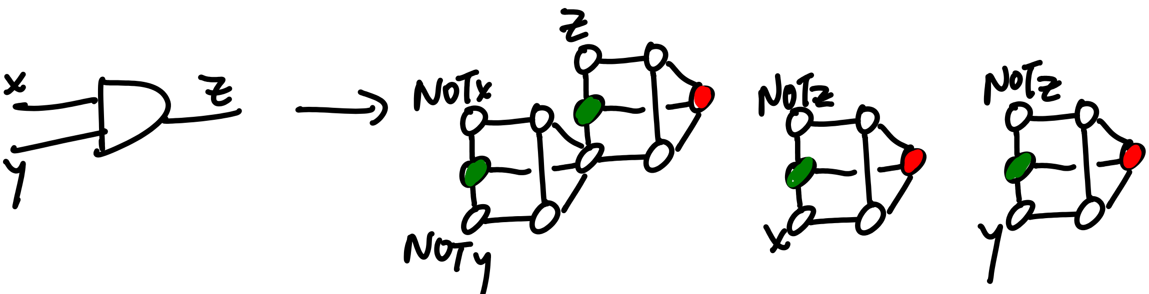
NOT:




OR:




AND:



# 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.