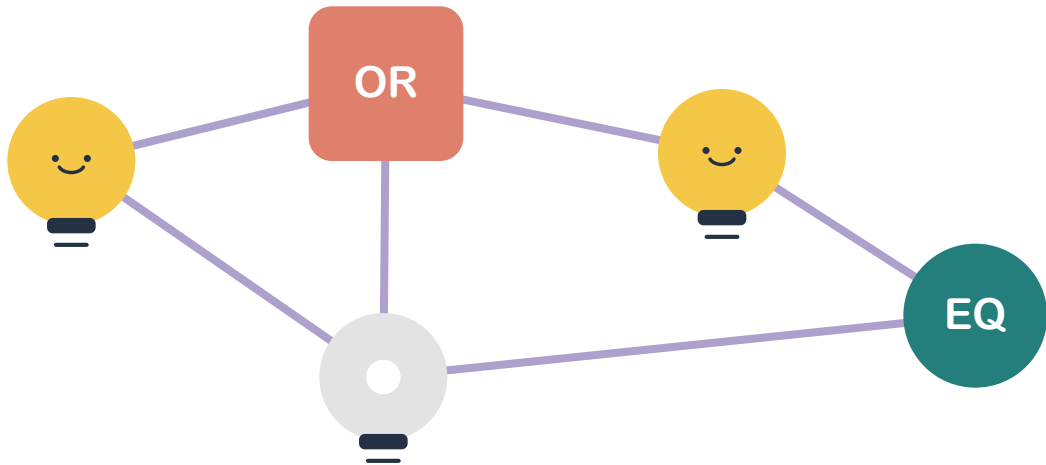


COUNTING PROBLEMS

for babies



A little book about #P, Holant, and dichotomies.



This is a light.

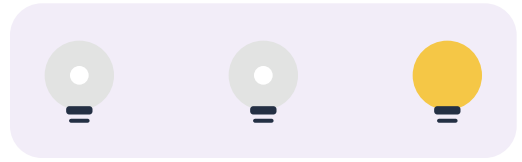
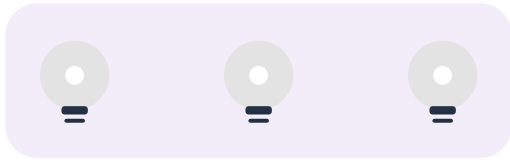


0

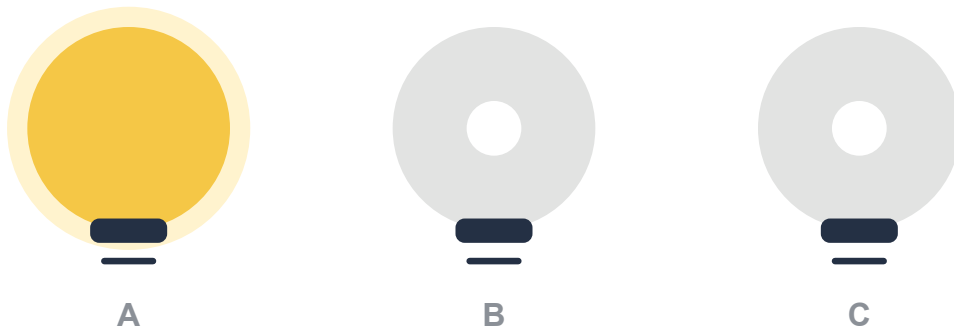


1

It can be off.
It can be on.

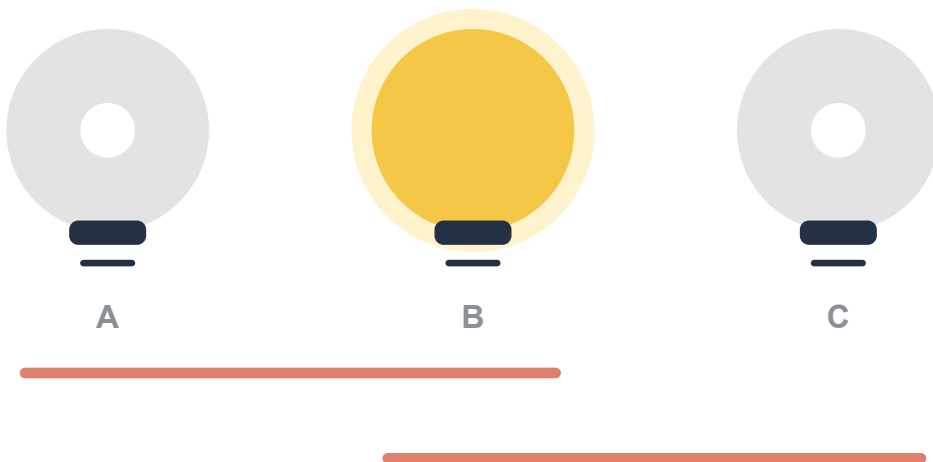


Three lights have eight possible settings.

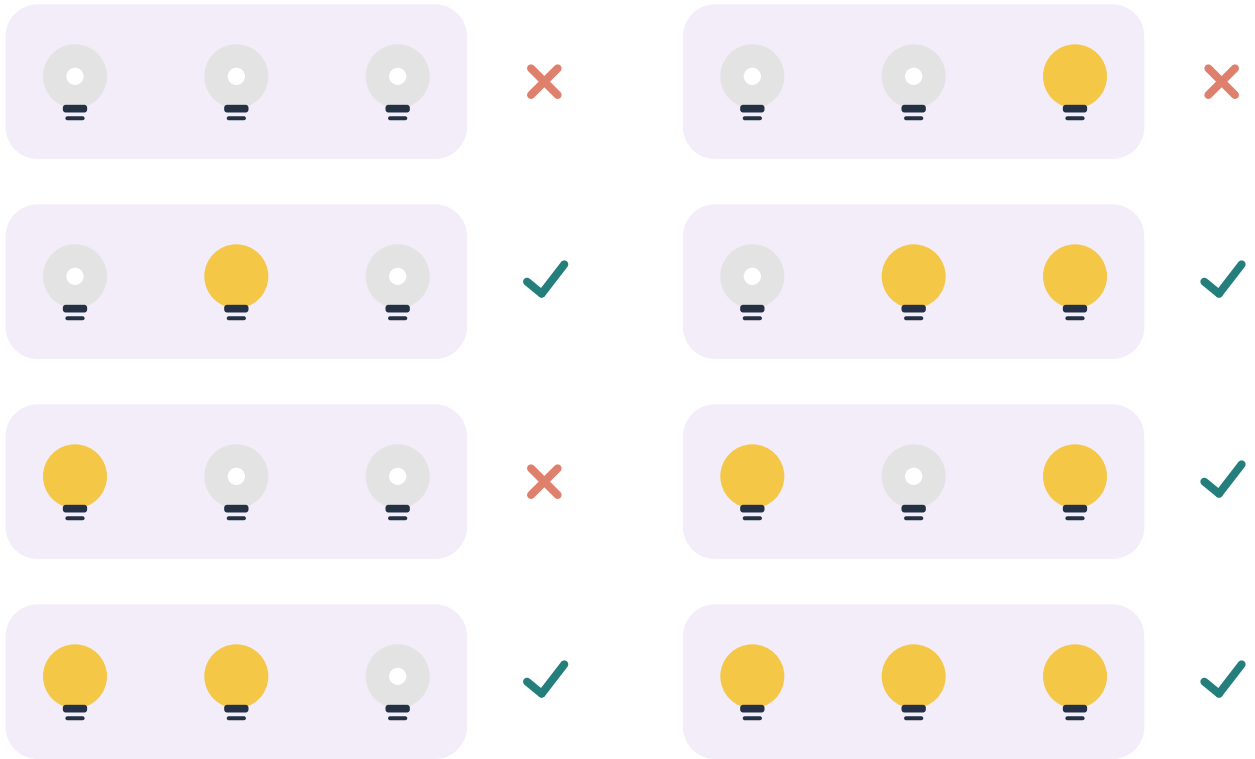


OR

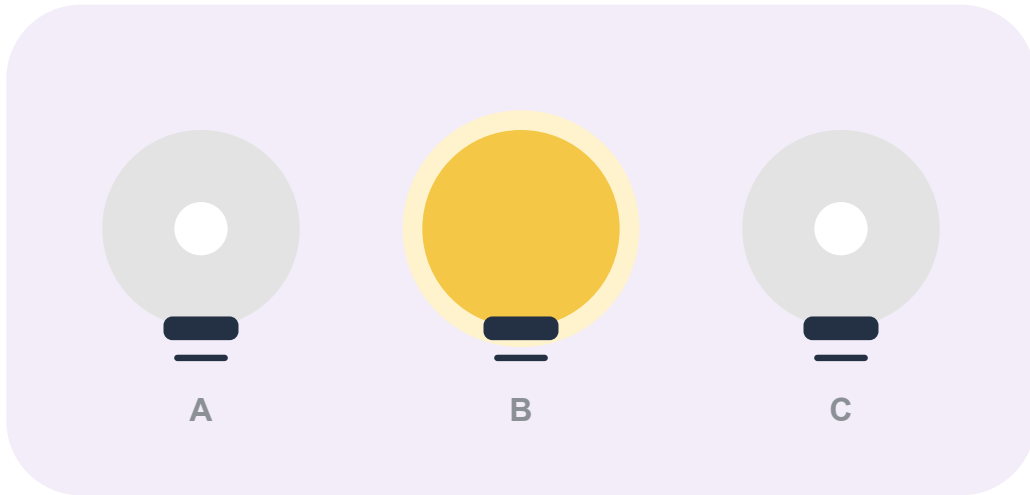
**Here is a rule:
A or B must be on.**



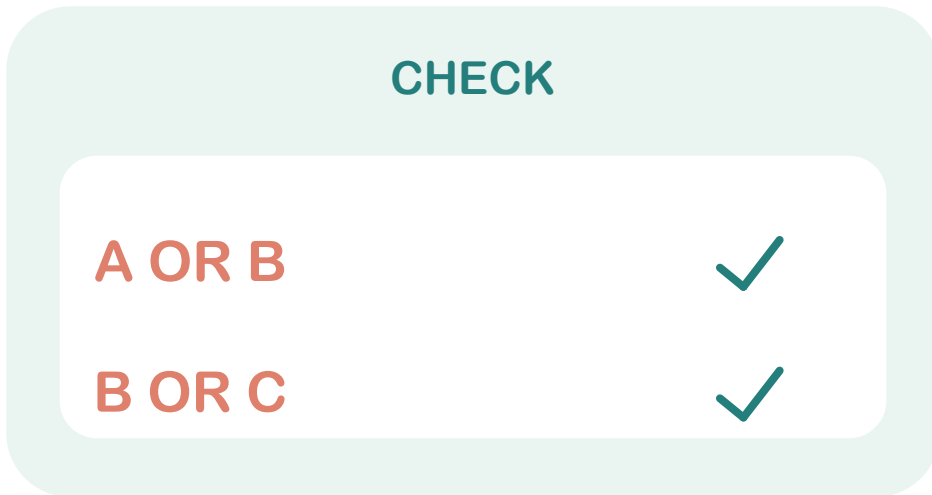
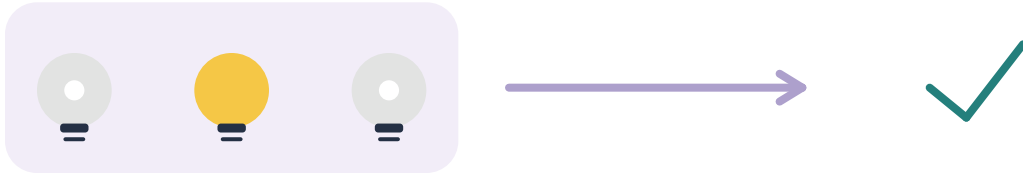
**Here is another:
B or C must be on.**



Five settings obey both rules.



This is one possible answer.



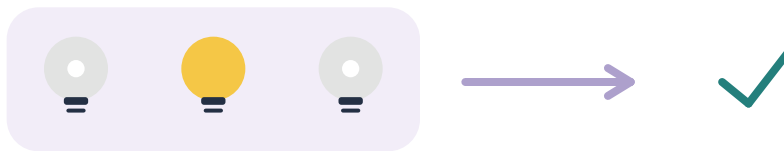
We can check it quickly.



Now count all the answers that pass.

#P

“sharp P”



Counting short answers with quick checks
gives a class called #P.



COUNT



CHECK

e.g. n^2 steps

means “how many?”
P is for polynomial-time checks.

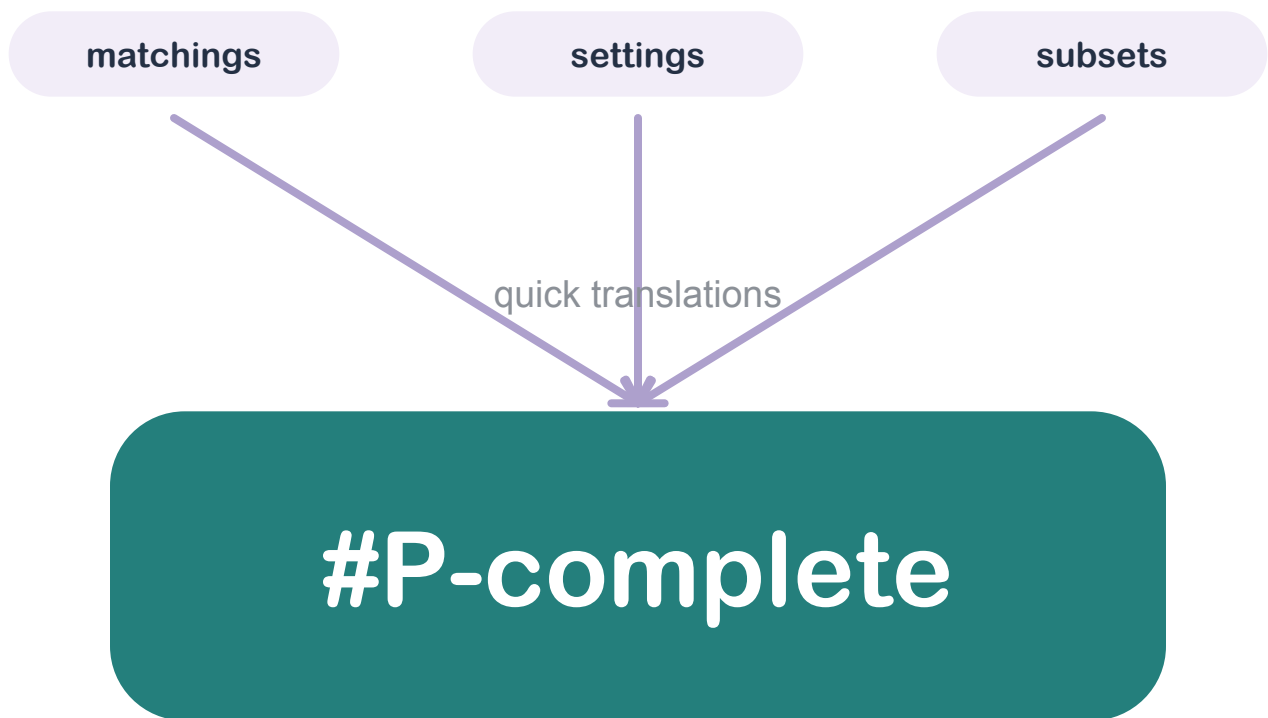
CHECK



COUNT



**A quick check does not tell us
how to count quickly.**



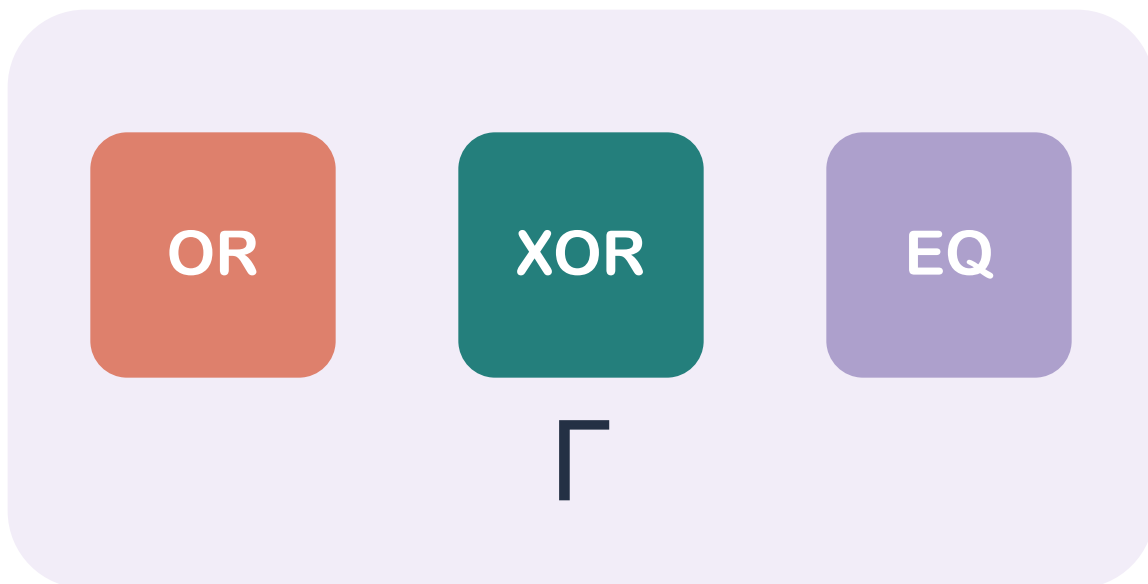
**Some #P problems can count for all the others.
We call them #P-complete.**



$$2^n$$

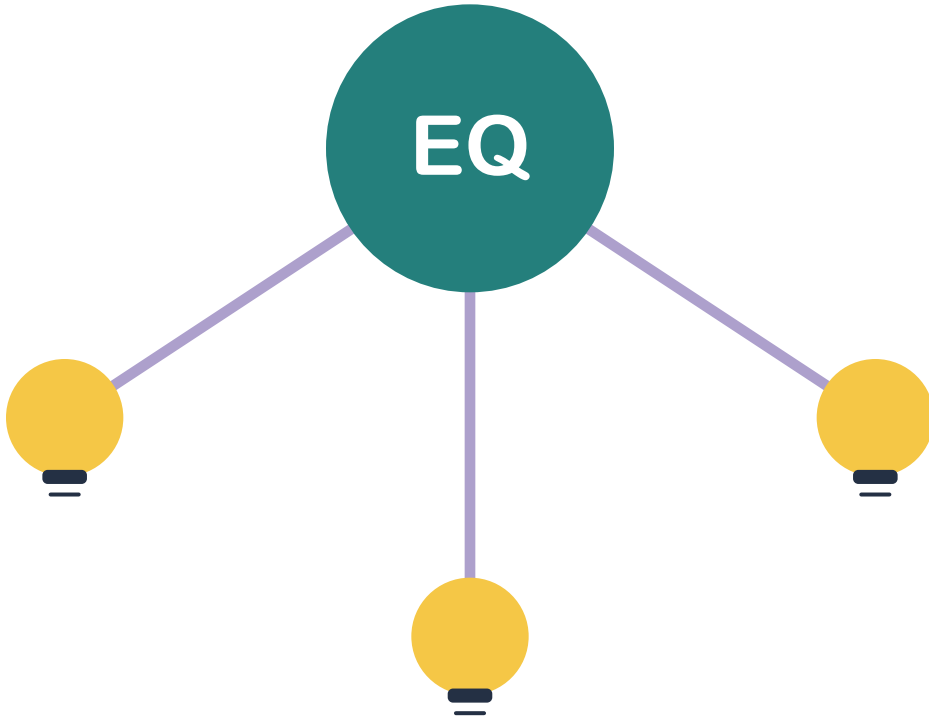
n independent lights

Many answers can still be easy to count.

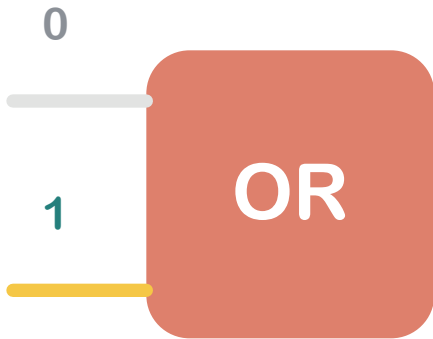


**Choose a fixed box of rules.
Counting their solutions is called #CSP.**

000 or 111

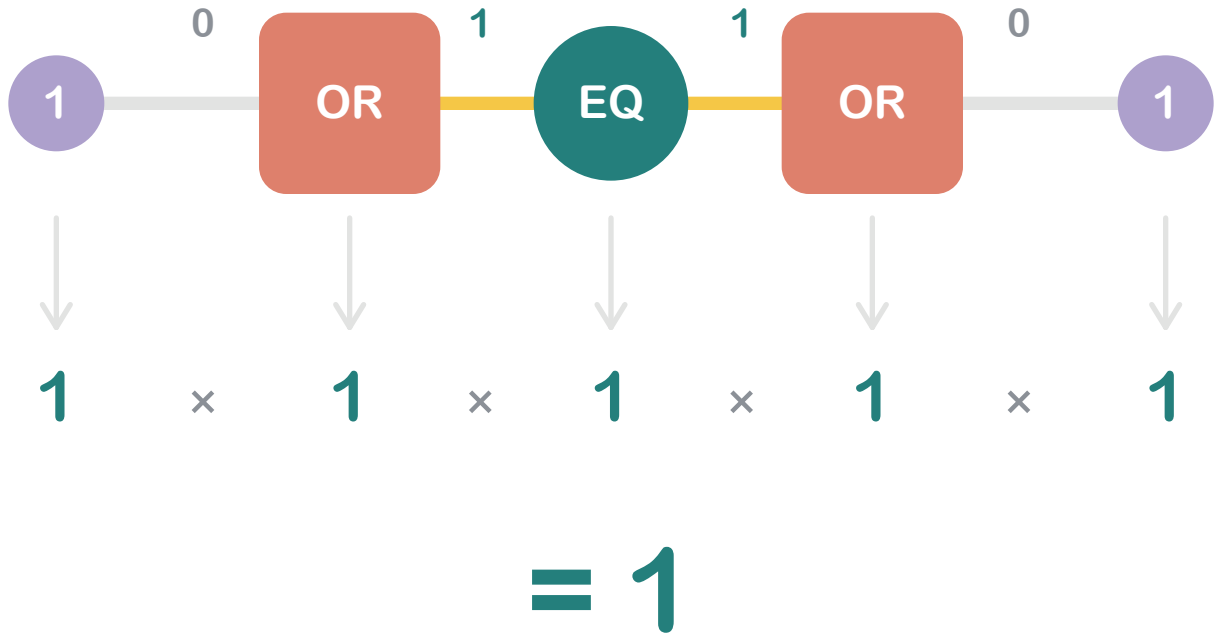


**This block says:
all these choices must agree.**



signature		
00	→	0
01	→	1
10	→	1
11	→	1

**Each edge carries a choice.
Each block gives a number.**

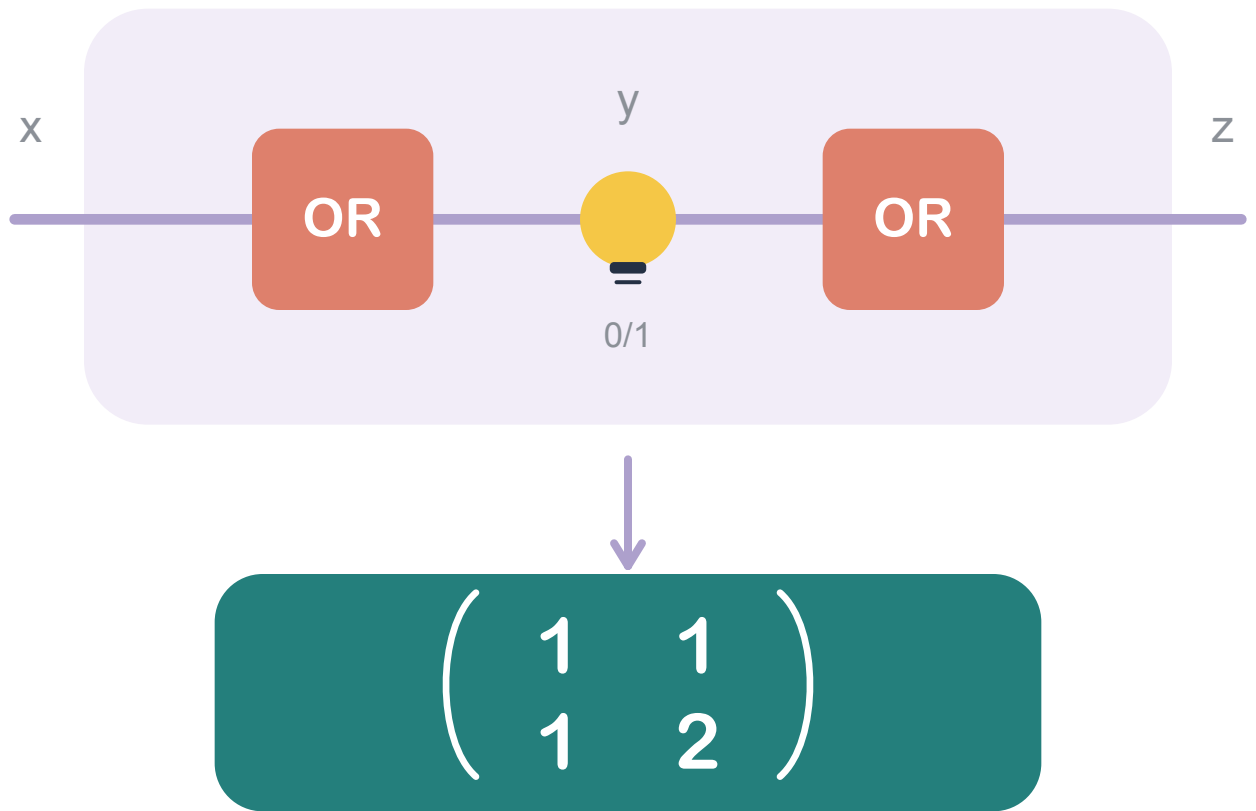


For one choice, multiply the numbers.

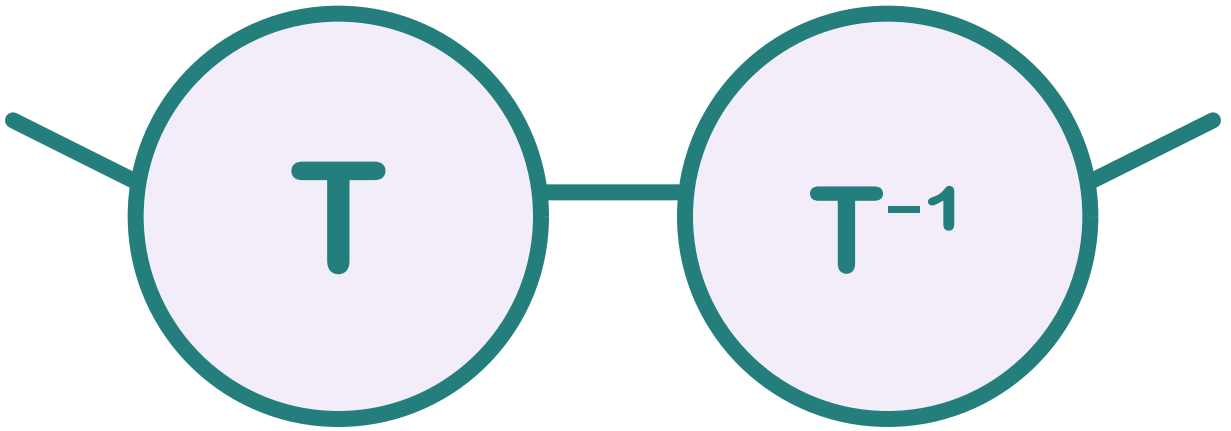
			→	0	
			→	+	0
			→	+	1
			→	+	1
			→	+	0
			→	+	1
			→	+	1
			→	+	1

$\Bigg] = 5$

For all choices, add the results.
This number is the Holant.

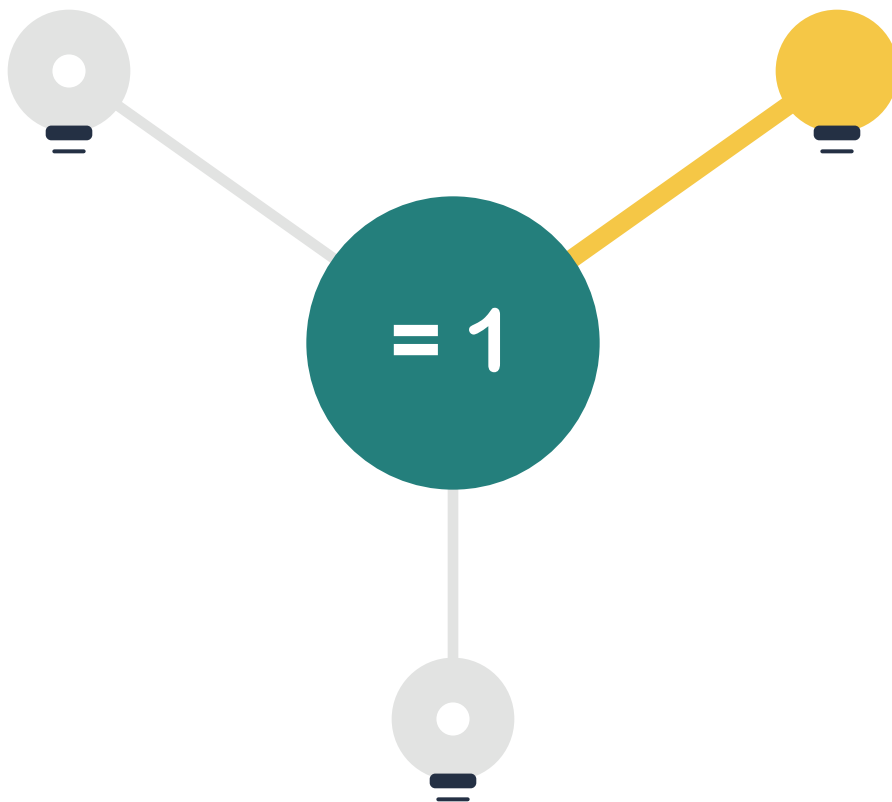


A little network can act like one big block.

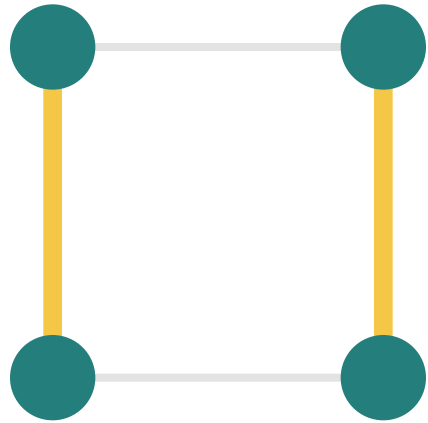
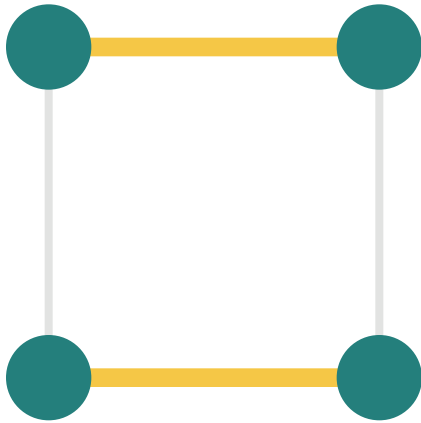


$$5 = 5$$

**A change of basis can keep
the total unchanged.**

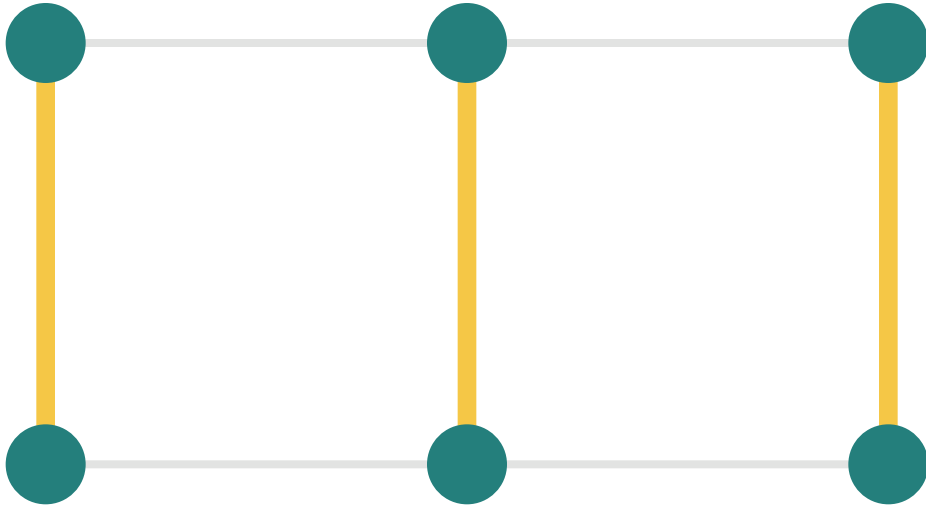


This block asks for exactly one bright edge.



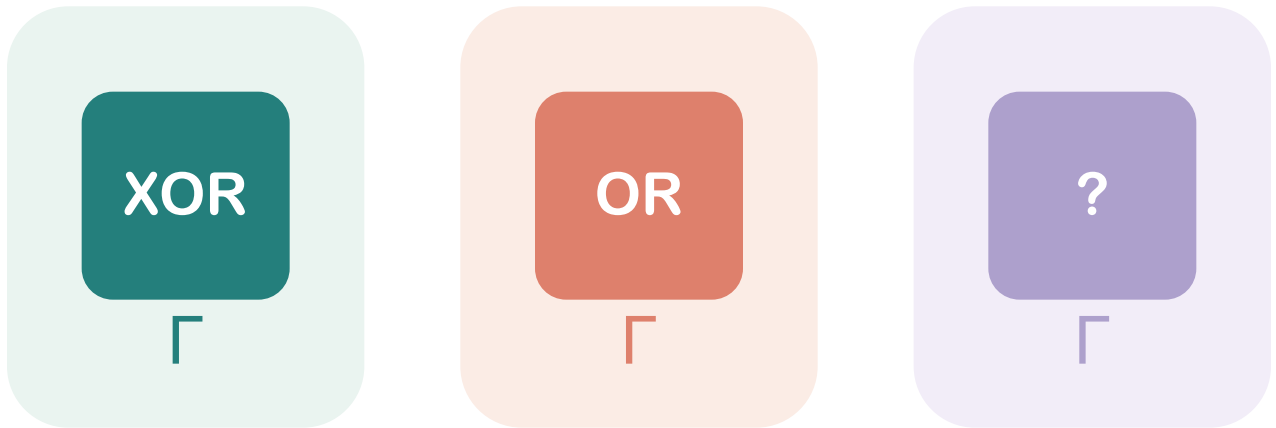
2

**Together, these blocks count
perfect matchings.**



FKT

**On planar graphs,
we can count these quickly.**



Which boxes of rules make counting easy?

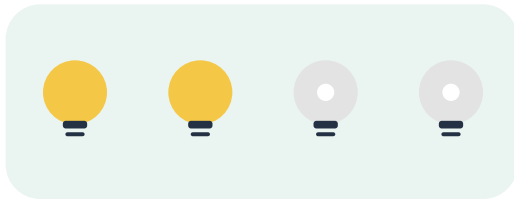
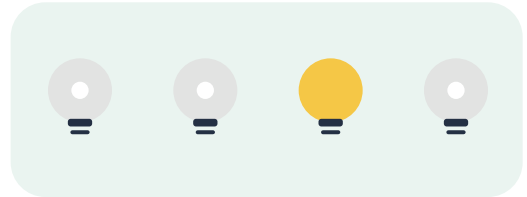
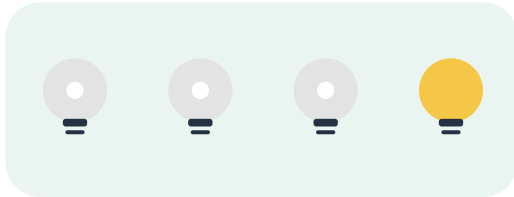
$$A \oplus B = 0$$

even

$$B \oplus C \oplus D = 1$$

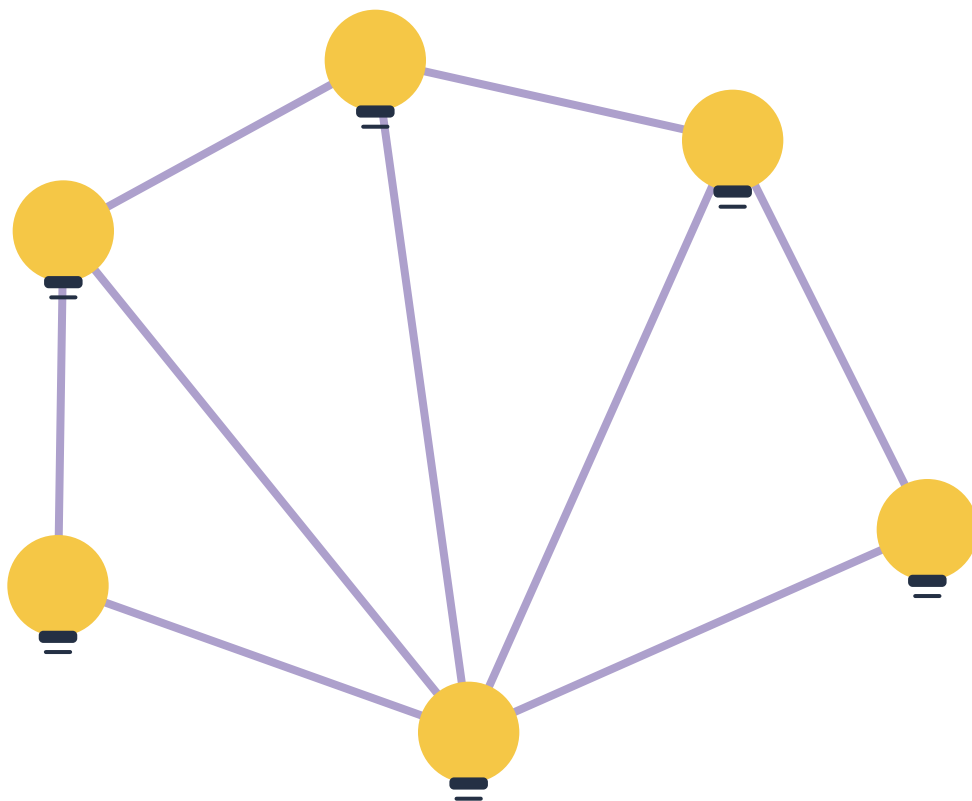
odd

Some rules are systems of parity equations.
We call them affine.

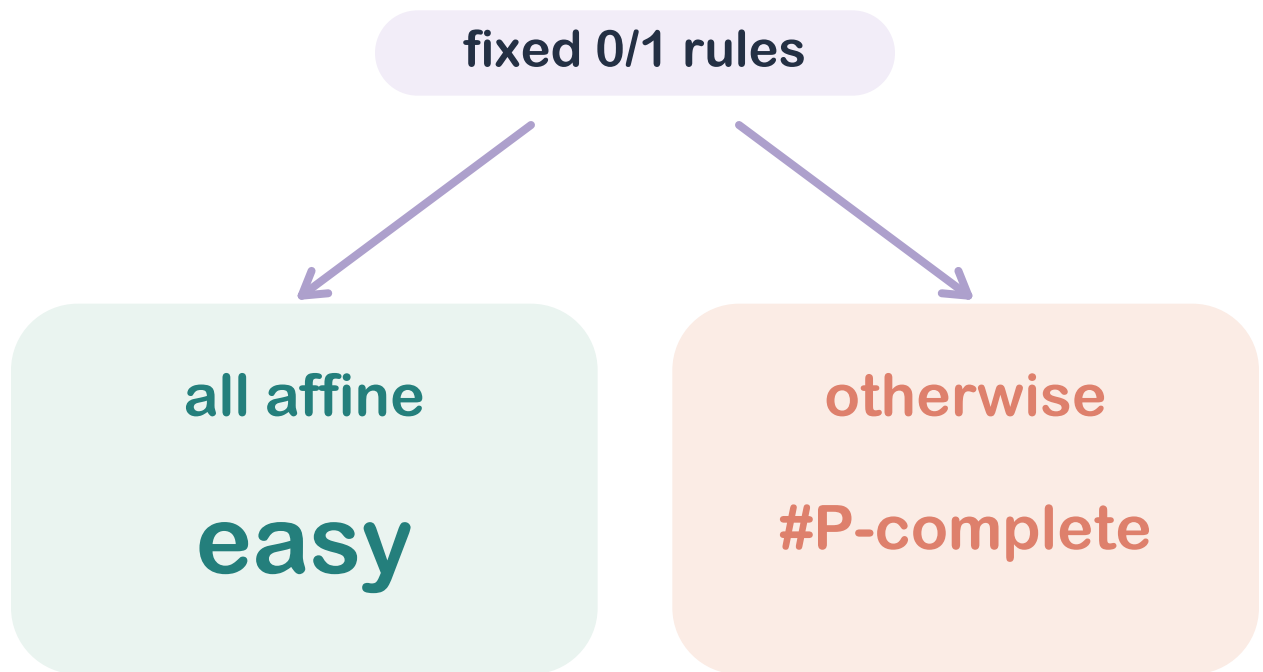


$$2^2 = 4$$

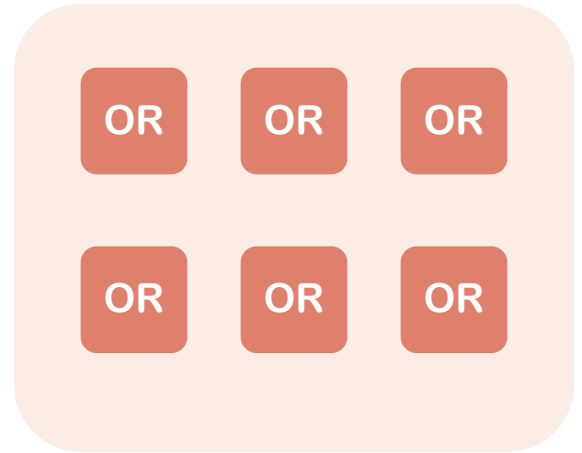
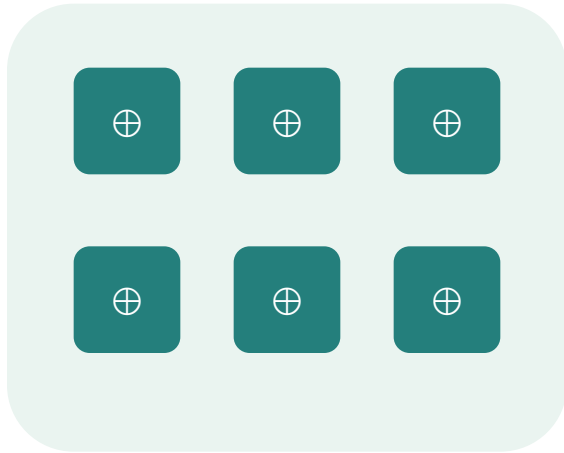
The equations leave some choices free.
We can count those.



**An OR rule looks simple.
Counting with OR rules is #P-complete.**

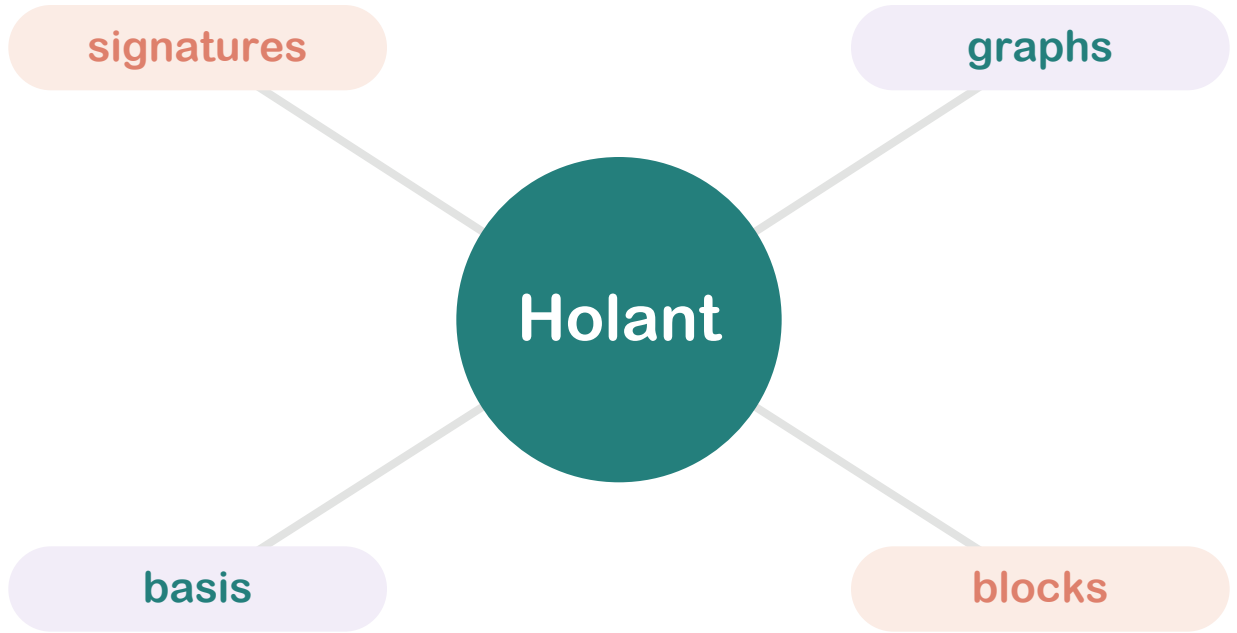


**For Boolean #CSP,
there is a complete classification.**

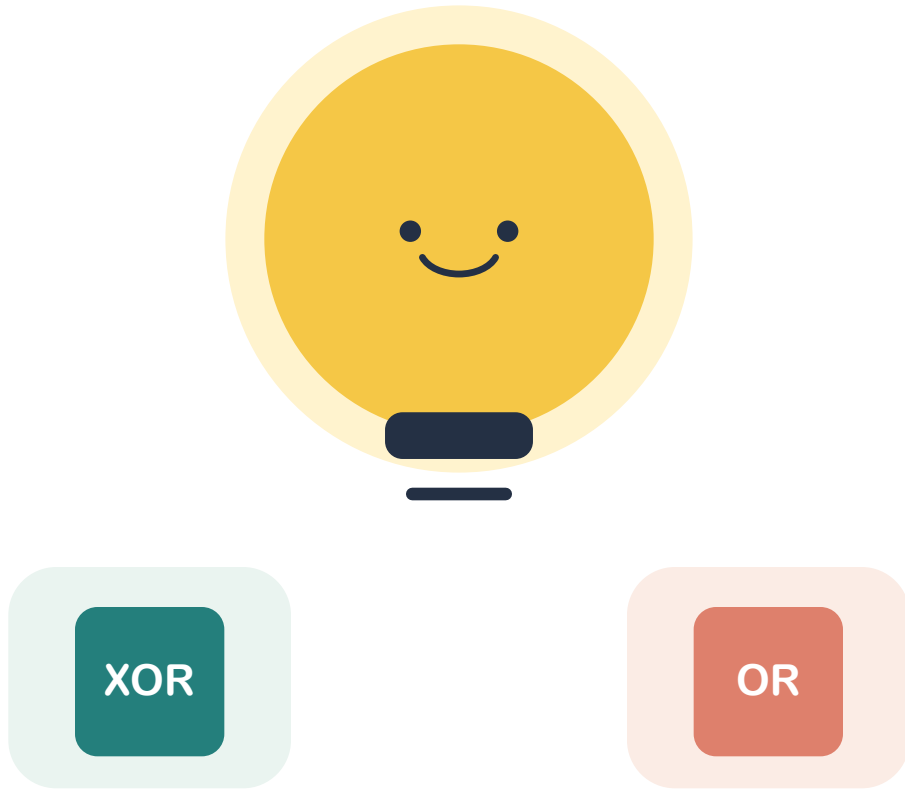


dichotomy

**Every rule box is classified.
This is a dichotomy.**



**Holant has its own dichotomies.
The signatures and allowed graphs matter.**



**Now you can ask:
which rules make counting easy?**



Good night, little counter.

What #P is counting

A count, not a yes/no answer

For an input x , choose a polynomial p and a deterministic polynomial-time verifier V . The value $f(x)$ is the number of binary strings y of length $p(|x|)$ for which $V(x,y)=1$. The class of functions obtained this way is #P, pronounced “sharp P”. Short answers means polynomially many bits; quick checks means polynomial time in the input size.

Why the P is there

The formal machine definition counts accepting computation paths of a nondeterministic polynomial-time machine. The short-answer/verifier definition is equivalent. The book uses the latter: choose a setting, check the rules, count the settings that pass. Page 12 gives n^2 only as an example of a polynomial time bound.

Checking, deciding, counting

Checking asks whether a given candidate passes. The associated NP decision question asks whether at least one candidate passes. #P asks how many pass. A fast verifier is not, by itself, a fast counting algorithm. #P also contains easy counting functions: n unconstrained Boolean variables have 2^n assignments, whose binary answer has only $n+1$ bits.

Complete, and what that does not prove

A #P-complete function belongs to #P and can be used to compute every #P function by polynomial-time reductions. The notes use polynomial-time Turing reductions; #SAT also has parsimonious completeness. Hardness concerns a family of inputs, not each small drawing. It is not an unconditional proof that no polynomial-time algorithm exists.

What the blocks are doing

Holant

In a graph $G=(V,E)$, each edge receives 0 or 1. A vertex v has a signature f_v that reads the incident edge values. Multiply all vertex values for one edge assignment, then sum over every edge assignment. This is $\text{Holant}(G; f)$. With 0/1 signatures, it counts accepted edge assignments; with weights, it is a sum of products and need not be a nonnegative integer.

From #CSP to Holant

A #CSP variable may occur in many constraints. Split these occurrences into edges and attach an equality signature EQ_d to require all d copies to agree. This preserves the count. An unused variable contributes a separate factor of 2. The endpoint “1” blocks on page 19 mean the unary signature $[1,1]$.

The same five answers

The example $(A \text{ OR } B) \text{ AND } (B \text{ OR } C)$ has five solutions: 010, 011, 101, 110, 111. EQ_2 ties the copies of B . Summing over B gives the binary signature h , with matrix:

$$H = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$$

Rows x and columns z are ordered 0,1. The symmetric notation $[1,1,2]$ gives the values at Hamming weights 0,1,2.

A change of basis

On a bipartite Holant graph, let r be a row on the left and c a column on the right. For invertible T , replace r by rT and c by $T^{-1}c$:

$$(rT)(T^{-1}c) = r(TT^{-1})c = rc$$

Each edge contraction is preserved. For degree d , apply the corresponding d -fold tensor power to the signature. New weights may be negative or complex; this need not give a bijection between solutions.

Where the dichotomy applies

A precise rule-box theorem

Fix a finite set Γ of Boolean 0/1 relations. An instance of $\#CSP(\Gamma)$ is any finite collection of constraints using these relations. Variables may occur arbitrarily often, including repeatedly inside one constraint. There is no planarity restriction. The problem asks for the exact number of satisfying assignments.

The Creignou-Hermann dichotomy

If every relation in Γ is affine, meaning a solution set of linear equations over $GF(2)$, $\#CSP(\Gamma)$ is in FP. Gaussian elimination gives 0 solutions when inconsistent and 2^{n-r} otherwise; r is the rank. If any relation is not affine, $\#CSP(\Gamma)$ is $\#P$ -complete. The book's two parity equations have four solutions. This is the counting theorem, not Schaefer's decision theorem.

Why OR and matching belong here

With only binary OR constraints, all-ones always satisfies every instance, yet counting assignments is $\#P$ -complete. EXACT-ONE at every graph vertex counts perfect matchings. FKT counts them in polynomial time on planar graphs; the count on general bipartite graphs is $\#P$ -complete.

Holant needs a richer classification

Holant requires its own classification: signatures, graph restrictions, and available functions all matter. Some planar tractable families go beyond holographic reductions to FKT. Complex-weighted Holant need not belong to $\#P$; its hardness is stated as $\#P$ -hard. We discuss exact computation with effectively represented fixed algebraic weights.

Sources and picture-book design

Chenghua Liu and Boning Meng

Authors; with AI assistance from GPT-6 Astra

Renan Gross (2021)

Stochastic Calculus for Babies

Chris Ferrie

Baby University / Quantum Physics for Babies

Creignou and Hermann (1996)

Complexity of Generalized Satisfiability Counting Problems

Cai, Lu, and Xia (2011)

Computational Complexity of Holant Problems

Guo and Lu (2017)

On the Complexity of Holant Problems

Cai, Fu, Guo, and Williams (2015)

A Holant Dichotomy: Is the FKT Algorithm Universal?

Valiant (1979)

The Complexity of Computing the Permanent

Renan Gross and Chris Ferrie inspired the sparse picture-book format. Text and diagrams are original. Formal qualifications and references are collected in this appendix. Click a source title to open it.