2 ways to represent variables in 3SAT:

1) Dual-rail logic:
   - variable gadget forces exclusive OR of 2 "semi-wires" (true & false)
   - semiwire connects to clauses a variable active only when chosen
     (e.g. Nintendo, pushing blocks, Phutball — most 3SAT reductions we've seen)

2) Binary logic:
   - wire gadget has 2 (types of) solutions
   - split gadget to make copies of wire
     (e.g. flat-foldable crease patterns)
   - not gadget (for 3SAT, not 1-in-3/NAE 3SAT)
   - terminator gadget (for ending unused wires without constraints e.g. Circuit SAT inputs)

- Circuit SAT needs true terminator to constrain output = true

\{ in both cases, may need
- turn gadget to route (semi)wires
- crossover gadget to cross (semi)wires
- shift gadget to adjust/fix parity/mod-k spacing
Akari/Light Up: [Nikoli 2001]
- given square grid with some obstacles
- some obstacles have a number
  → how many (0-4) edge-adjacent lights
- light illuminates like rook, up to obstacles
- goal: place lights in blanks so that
  - black space lit
  - no lights light each other
  - satisfy numbers

NP-complete by reduction from Circuit SAT: [McPhail 2005]
- wire, turn gadgets
- split/negation gadget
  → split & negation gadgets (via terminators)
- OR/XNOR gate
- crossover gadget: just XORs!
Minesweeper: given square grid of numbers & unknowns & possibly mines

Consistency: does there exist a solution?
- e.g. see whether mine at \(x\) is consistent with (consistent) info so far; if not, play \(x\)
  \(\rightarrow\) special case of interest

NP-complete by reduction from Circuit SAT

- wire, terminator
- split/NOT/turn
- phase changer (shift by 2) via 2 NOTs
- AND
- crossover gadget: just use NANDs!

[Kaye 2000]

[Goldschläger 1977]
Winning: can I force a win? (no guessing) i.e. figure out all squares? [Hearn 2006]

Inference: can I figure out any squares? [Scott, Stege, van Rooij 2011]

$\in \text{CoNP}$: proof of NO = 2 differing solutions

CoNP-complete by reduction from Circuit UNSAT:

\[ \forall x_1, x_2, \ldots, x_n \text{ s.t. } f(x) \equiv \forall x_1, x_2, \ldots, x_n : \neg f(x) \]

- wire, turn, terminator
- NOT, OR, shifter
- split
- crossover: just use NORs!

- special care to ensure equal # mines in all cases (# mines part of puzzle) & ports aligned (middle of 3)

- unsatisfiable $\iff$ output forced to be F

Planar Circuit SAT: given noncrossing circuit

- only NAND or - only NOR

[new?]
Candy Crush / Bejeweled
- given square grid of colors (among 6
- move = swap two edge-adjacent squares
- whenever 3 equal colors in a row/column:
  3 squares disappear & columns fall
  > “pop”
NP-complete to get p points with k moves
by reduction from 3SAT
... in model where pops happen
  sequentially bottom to top

reward

huge points

clause

positive & negative wires

connectors

clause

reward

clause

reward

clause

clause

# clauses

var. \quad \neg \quad \text{var.}

- claim: worse to trigger
  wire (even x & \neg x) directly
- only use 5 colors

[Walsh 2014]
NP-complete with simultaneous pops by reduction from 1-in-3SAT
- works for many goals:
  - p points in k moves
  - p points
  - pop p gems
  - p moves
  - pop a specific gem