6.890	Lecture 6	Sept. 23, 2014

2 ways to represent variables in 3SAT:

① Dual-rail logic:
- variable gadget forces exclusive OR of 2 "semiwires" (true & false)
- semiwire connects to clauses ∋ variable
  active only when chosen

(e.g. Nintendo, pushing blocks, Phutball — most 3SAT reductions we've seen)

② Binary logic: (not just Circuit SAT)
- 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
  → special case of interest

NP-complete by reduction from Circuit SAT [Kaye 2000]

- wire, terminator
- split/NOT/turn
- phase changer (shift by 2) via 2 NOTs
- AND
- crossover gadget: just use NANDs! [Goldschlager 1977]

Planar Circuit SAT: given noncrossing circuit
- only NAND or - only NOR
- or any functionally complete gate set [McColl - IEEE Trans. Computers 1981]

(& splitters for fan-out)

<u>Winning</u>: can I force a win? (no guessing)
i.e. figure out all squares?
[Hearn 2006]

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

<u>$\in$ CoNP</u>: proof of NO = 2 differing solutions

<u>CoNP-complete</u> by reduction from
<u>Circuit UNSAT</u>: $\neg \exists x_1, x_2, \dots, x_n$ s.t. $f(\vec{x})$
$\equiv \forall x_1, x_2, \dots, x_n : \neg f(\vec{x})$

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

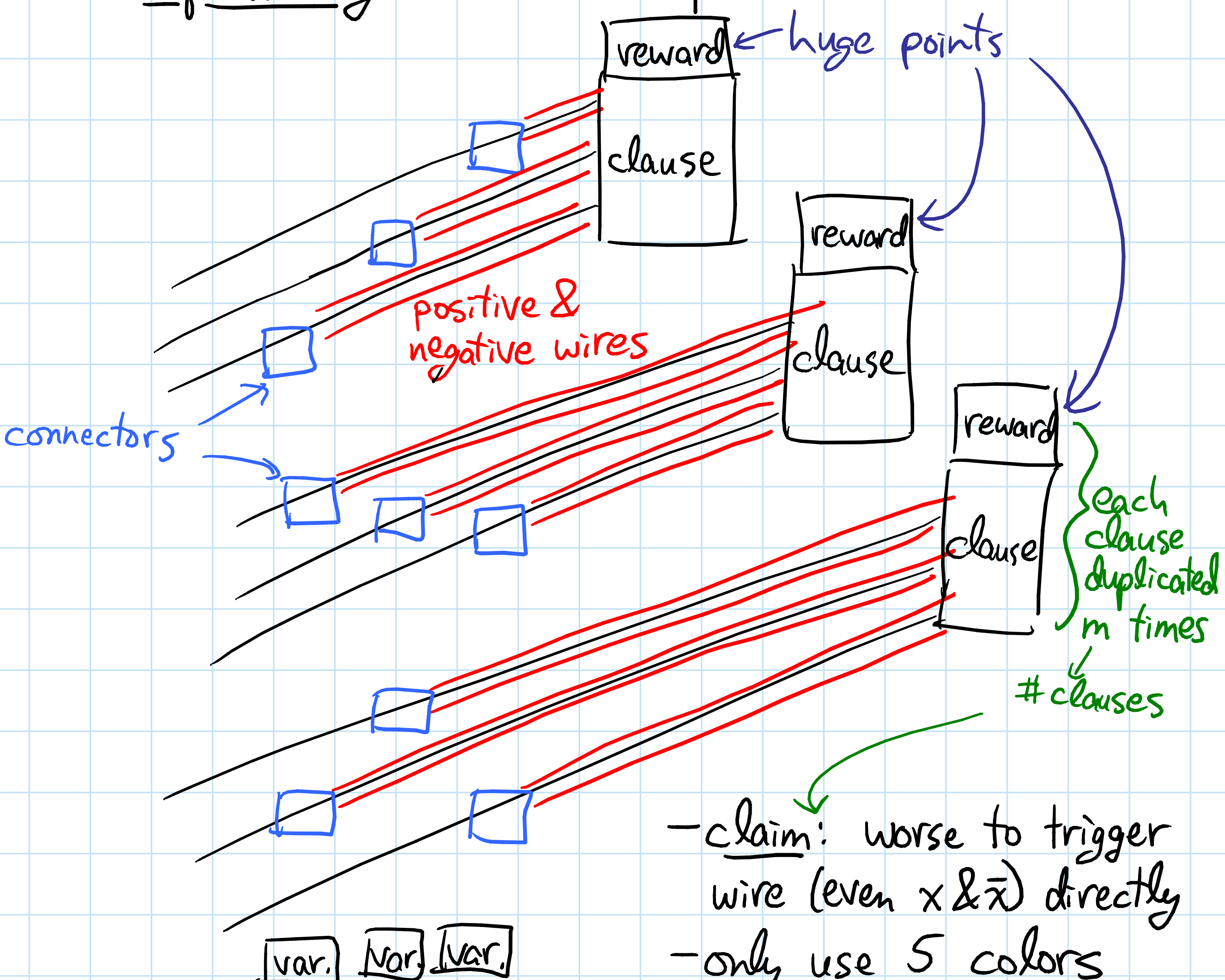
- special care to ensure equal # mines in all cases (# mines part of puzzle)
& ports aligned (middle of 3)

- unsatisfiable $\Leftrightarrow$ output forced to be F

Candy Crush / Bejeweled (perfect information)

- 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 [Walsh 2014]
... in model where pops happen sequentially bottom to top



- claim: worse to trigger wire (even x & $\bar{x}$) directly
- only use 5 colors

<u>NP-complete</u> with simultaneous pops by reduction from 1-in-3SAT

[Gualà, Leucci, Natale 2014]

- works for many goals:
  - p points in k moves
  - p points
  - pop p gems
  - p moves
  - pop a specific gem