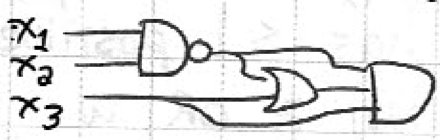
3-SAT & NP-hardness

The most important NP-complete (logic) problem family!

SAT = Satisfiability: [Cook 1971; Levin 1973]

- given a Boolean formula (AND, OR, NOT) over n variables $x_1, x_2, \ldots, x_n$
- can you set x_i's to make formula true?

Circuit SAT: formula expressed as circuit of gates (allows re-use)



CNF SAT: formula = AND of clauses [Cook 1971]
↓
Conjunctive Normal Form
clause = OR of literals
literal $\in \{x_i, \text{NOT } x_i\}$

- can view as bipartite graph: variables vs. clauses, positive/negative edges

3SAT: clause = OR of 3 literals [Cook 1971]
i.e. clause degrees $= 3$

3SAT-5: each variable occurs in ≤ 5 clauses
[Feige - JACM 1998; perhaps earlier?]

Monotone 3SAT: [Gold - I&C 1978]
each clause all positive or all negative

Beware polynomial-time variants!

P	Q	P→Q
T	T	T
T	F	F
F	T	T
F	F	T

2SAT: clause = OR of 2 literals
- polynomial
- x OR y ≡ NOT x ⇒ y (≡ NOT y ⇒ x)
- guess x_i, follow all implication chains to check OR

BUT...

Max 2SAT: set variables to maximize # true clauses
- NP-complete [Garey, Johnson, Stockmeyer 1976]

Horn SAT: (generalization of 2-SAT) each clause has ≤ 1 positive literal
- NOT x OR NOT y OR NOT z OR w

≡ NOT (x AND y AND z) OR w (Morgan's thm)

≡ (x AND y AND z) ⇒ w [Horn 1951]

⇒ polynomial like 2SAT (every time that you assign a variable you should not get a contradiction)

Dual-Horn SAT: each clause has ≤ 1 negative literal

↳ "weakly positive satisfiability" [Schaefer 1978]
- negate all variables → Horn SAT

⇒ polynomial

Also note that $p \to q \equiv \sim q \to \sim p$

DNF SAT: formula = OR of clauses

clause = AND of literals

↓ Disjunctive Normal Form

⇒ satisfiable ⇔ ≥ 1 clause (of not form $x_i \wedge \bar{x}_i \wedge \ldots$)

Alternative clauses for 3SAT:

1-in-3SAT = exactly-1 3SAT [Schaefer 1978]
- clause = exactly 1 of 3 literals is true ($\Rightarrow$ 2 false ~ TFF, FTF, FFT)

omitted by Schaefer
"Monotone" 1-in-3SAT: no negations – all literals positive

BUT...

"Monotone" not-exactly-1 3SAT: [Schaefer 1978]
- clause = 0, 2, or 3 variables are true
(for each shift i,j,k) i.e. $x_i \Rightarrow (x_j \text{ OR } x_k)$ → Dual Horn
- also require x_1 = TRUE (else set all x_i = FALSE)
- polynomial

NAE 3SAT = not-all-equal 3SAT [Schaefer 1978]
- clause = 3 literals not all the same value
(forbid FFF & TTT $\Rightarrow$ 1 or 2 true, 2 or 1 false ~ whereas 3SAT forbids just FFF)
- nice symmetry between TRUE & FALSE

omitted by Schaefer
"Monotone" NAE 3SAT: no negations – all literals positive

"Monotone" NAE-3SAT is NP-complete as well.

The most important ones to remember: 3-SAT, 1-in-3SAT & NAE-3SAT

Schaefer's Dichotomy Theorem: (Universal Theorem) [Schaefer - STOC 1978]

- formula = AND of clauses
- general clause (type) $\stackrel{\text{def}}{=}$ relation on variables (with implicit truth table)
  - assume in CNF

    $\Rightarrow$ AND of subclauses

$\Rightarrow$ SAT is polynomial if either:

OR – setting all variables true or all variables false satisfies all relations,

OR – subclauses are all Horn or all Dual Horn,

OR – relations are all 2-CNF (subclause sizes ≤ 2, i.e., 2-SAT case),

OR – every relation can be expressed as a system of linear equations over $\mathbb{Z}_2$:

$$x_i \oplus x_j \oplus x_k \oplus x_\ell = 0 \text{ or } 1$$

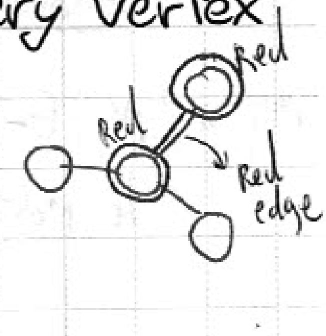
$\hookrightarrow$ XOR $\quad\quad \hookrightarrow$ Gaussian elimination

& otherwise, SAT is NP-complete!

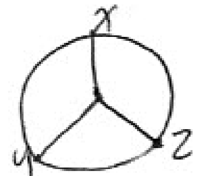
2-colorable perfect matching: [Schaefer 1978]

- given a planar 3-regular graph
- 2-color the vertices such that every vertex has exactly 1 same-colored neighbor
- special case of 2-in-4 SAT

(planarity & 3-regular left as exercise)
We just say the general graph case

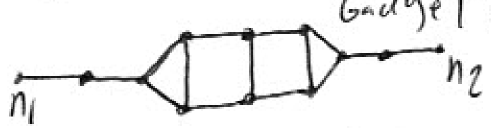


Thm [Schaefer '78] There is a reduction from monotone NAE 3-SAT to 2-colorable perfect matching: (5)

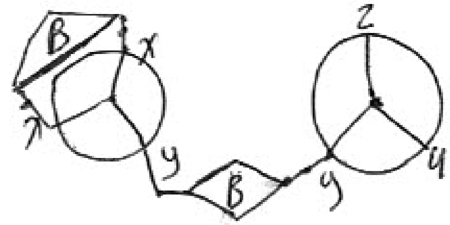


for each clause we put third gadget

$A = R(x,x,y) \wedge R(y,z,u)$



$n_1 = n_2$ [copies values]



Some fun games can be proved to be NP-complete via reduction from 3-SAT, e.g. Push-1 [Hoffman 2000]

You may see Erik's class for high-level ideas

An early Push-1 game: Sokoban

You can prove hardness of other games as well such as:

- Super Mario Bros.
- Legend of Zelda
- Metroid
- Donkey Kong country
- Pokemon
- etc

→ Note that is the bipartite graph between clauses and <u>variables</u> is planar; the problem is called planar CNF (type 1)

→ If the bipartite graph [between clauses and <u>literals</u>] plus all edges $(x_i, \bar{x}_i)$ form a planar graph, the problem is called planar CNF (type 2)

→ Both versions are NP-complete by a reduction from 3-SAT (by uncrossing the edge-crosses

→ very useful to prove NP-hardness of problems on planar graphs and geometric plane graphs

Cryptarithms / alphametics [Madachi 1979]

- given formula $x+y=z$ with each number written in base b & encoded with "letters" by unknown bijection between $\{0, 1, \ldots, b-1\}$ & letters
- goal: feasible? / recover bijection
- strongly NP-complete [Eppstein 1987]

Examples: (mod 10)
$$\begin{array}{r} 9567 \\ +\,1085 \\ \hline 10652 \end{array} \quad \text{can be represented as:} \quad \begin{array}{r} abcd \\ +\,efgb \\ \hline efcbh \end{array}$$

OR
$$\begin{array}{r} SEND \\ +\,MORE \\ \hline MONEY \end{array}$$

$\boxed{\text{We need to have base } f(n) \text{ to be interesting}}$

rightmost three columns

$$\begin{array}{r} 0P0 \\ 0P0 \\ \hline 110 \end{array}$$

here letters 0 and 1 are forced to be 0 and 1 for any mod

Reduction from 3SAT:

- variable gadget:
  - $b_i = 2a_i$
  - $v_i = 2b_i + C \qquad C = \text{carry}(y_i + y_i) \in \{0, 1\}$
    $= 4a_i + C \equiv C \pmod 4$
  - $d_i = 2c_i + C$
  - $e_i = d_i + 1 + C$
    $= 2c_i + 1 + 2C$
  - $\overline{v_i} = d_i + e_i$
    $= 4c_i + 1 + 3C$
    $\equiv 3C + 1 \equiv 1 - C \pmod 4$

for v_i & $\overline{v_i}$
$$\begin{array}{l} d_i\,0\,1\,y_i\,0\,c_i\,y_i\,0\,b_i\,y_i\,0\,a_i\,0 \\ e_i\,0\,d_i\,y_i\,0\,c_i\,y_i\,0\,b_i\,y_i\,0\,a_i\,0 \\ \hline \overline{v_i}\,0\,e_i\,z_i\,0\,d_i\,z_i\,0\,v_i\,z_i\,0\,b_i\,0 \end{array}$$

- clause gadget:
  - $g_i = 2f_i$
  - $h_i = 2g_i + \{0, 1\}$
    $= 4f_i + \{0, 1\}$
  - $t_i = h_i + 1 + \{0, 1\}$
    $= 4f_i + 1 + \{0, 1, 2\}$
    $= 4f_i + \{1, 2, 3\}$
  - $v_a + v_b + v_c = t_i \equiv \{1, 2, 3\} \pmod 4$

Cryptarithm Rules:
- each letter represents a unique digit
- often numbers must not start with zero
- often the solution is unique

for clause $(v_a \vee v_b \vee v_c)$ {use u_{ab} for all clauses with v_a and v_b}

$$\begin{array}{l} u_{ab}\;0\;v_a\;0\;1\;r_i\;0\;g_i\;w_i\;0\;f_i\;0 \\ v_c\;0\;v_b\;0\;h_i\;r_i\;0\;g_i\;w_i\;0\;f_i\;0 \\ \hline t_i\;0\;u_{ab}\;0\;t_i\;s_i\;0\;h_i\;x_i\;0\;g_i\;0 \end{array}$$

The reduction is good for NP-completeness for any mod multiple of 4, but still we need a solution for the puzzle from a satisfying solution, e.g. uniqueness issues ⑦

Simplified reduction from 1-in-3SAT:

- variable gadget: just v_i, no $\overline{v_i}$ (monotone)
- clause gadget:
  - $g_i = 2f_i$
  - $h_i = 2g_i = 4f_i$
  - $t_i = h_i + 1 = 4f_i + 1$
  - $v_a + v_b + v_c = t_i = 4f_i + 1 \equiv 1 \pmod 4$

(*) They proved for any k there is a set of k numbers all between $1..k^3$ such that their sum in triples are distinct (we can use powers of 2 but base would be in $O(4^n)$ and not good for strong NP-hardness)

3SAT solvable ⇒ cryptarithm solvable:

one class for each variable { - distinguish $a_i, b_i, c_i, d_i, \ldots$ by value mod 128

- e.g. $v_i \equiv 8 \pmod{128}$ if true
  $\equiv 9 \pmod{128}$ if false
  possible choices for each
  $a_i \equiv \{2, 34, 66, 98\} \pmod{128}$
  $b_i = \{4, 68\}$
- set $\lfloor v_i / 128 \rfloor$ & $\lfloor \overline{v_i} / 128 \rfloor \in [0, (2n)^3]$ for v_i and $\overline{v_i}$
  such that we have distinct sums of triples for all clause
  [Bose & Chowla 1959](*) (see above)
- easy proof of polynomial range: (based on fusion trees)
  - if $< i$ set by induction, v_i must avoid
    $v_j + v_k - v_l - v_m - v_p \sim < (2n)^5$ choices
  - $\Rightarrow (2n)^5$ suffices

$\Rightarrow$ strongly NP-hard

The final result would be in base $(2n)^3 \cdot 3 \cdot 128 = 3072n^3$ [see Eppstein '87 (revised in 2000) for all details.]