Bounded team private-information games:

NEXPTIME-complete [Peterson, Reif, Azhar-C&M 2001]
- Dependency QBF (DQBF):
  \( \forall X_1 : \forall X_2 : \exists Y_1(X_1) : \exists Y_2(X_2) : \text{CNF formula} \)
  - black player only sees \( X_1 \)
  - white player 1 only sees \( X_1 \) variables
  - white player 2 only sees \( X_2 \) variables
- can white force a win? (satisfied formula)
- only one round! (multiple rounds don't help)
- \( \in \) NEXPTIME: guess \( Y_1 \) \& \( Y_2 \) \& \( X_2 \) (exponential)

- Bounded Team Private Constraint Logic (TPCL)
  - with 3 players & planar graph
  - moves must be known legal with visible information
  - \( \in \) NEXPTIME: guess strategy for all possible visible information (exp.\# states)
- reduction from DQBF
- first black sets all vars. (white twiddles thumbs)
- chosen activates \( \implies \) long chain (black threat)
- white players set their vars.
- chosens \( \implies \) unlock all \( \implies \) formula activation
- white wins (just in time) if formula satisfied
Unbounded team private-information games:
undecidable \cite{HearnDemaine}
(based on work by Peterson & Reif – FOCS 1979)

Team Computation Game:
- instance = space-$k$ algorithm/Turing machine
  (memory/tape initially blank)
- black move = run alg./machine for $k$ more steps;
  output (if any) determines winner;
  else set $x_1, x_2 \in \{A, B\}^*$
- white $i$ sees only $x_i$ & can set only $m_i$
- white $i$ move = set $m_i$
- does white have a forced win?

- reduction from Halting problem: does this Turing machine ever terminate?
- build $O(1)$-space algorithm to check white players play valid computation history $\Rightarrow$ halt of the form $\#$ state$_0\#\text{state}_1\# \cdots\#\text{halt}\text{state}$
- in fact each white player must have in mind 2 pointers $A$ & $B$ into common history
- $x_i = A$ asks for character at $A$ & advance $A$
- but white players have no idea of other's $A$/$B$
- alg. maintains whether 1's $x_1$ state = 2's $x_2$ state
  (identical from $\#$ with $(x_1, x_2)$ moves since)
then if \((x_1, x_2)\) moves until 1 reports \(\geq 1 x_1\) ahead one
and if \((x_1, x_2)\) moves then continue,
then check this 1 state valid transition from 2's & vice versa with \(1 \rightarrow 2 \Rightarrow O(1)\) space!
white strategies must work for all possible black moves \(\Rightarrow\) valid computation history

Team Formula Game:
- black sets \(X\) such that \(F(X, X', Y_1, Y_2)\) (else lose)
- black wins if \(G(x)\) \(^\uparrow F \Rightarrow \neg F\)
- black sets \(X'\) such that \(F'(X, X')\) (else lose)
- white 1 sets \(Y_1\), seeing only \(Y_1\) & \(x_1 \in X\)
- white 2 sets \(Y_2\), seeing only \(Y_2\) & \(x_2 \in X\)
- standard reduction from Team Computation Game

(Unbounded) TPCL with 3 players, planar graph
Parallelism & P-completeness:

  "Limits to Parallel Computation: P-Completeness Theory"

\[ \text{NC} \quad (\text{Nick's Class, after Nick Pippinger}) \]
= problems solvable in \( \log^{O(1)} n \) time
using \( n^{O(1)} \) processors (PRAM)
i.e. circuit of size \( n^{O(1)} \) & depth \( \log^{O(1)} n \)
- e.g. sorting: compare all pairs, \( \{0(lg n)\} \) time on \( 0(n^2) \) proc.
  compute rank = sum of '<'s via binary tree

\[ \text{P-hard} = \text{all problems } \in \text{NC can be reduced} \]
via NC algorithm to your problem \text{karp-style reduction}

\[ \Rightarrow \in \text{NC if } \text{NC } \neq P \]
\[ \text{P-complete} = \in \text{P} + \text{P-hard} \]
Base P-complete problems:

**Generic Machine Simulation Problem:**
given a sequential algorithm & time bound, written in unary, does it say YES within?  
\( \text{to make } \emptyset \text{ P} \sim \text{ else EXPTIME-complete} \)

**Circuit Value Problem (CVP):** [Ladner - SIGACT 1975]
given an (acyclic) Boolean circuit & input bits, is the output TRUE?

- **NAND CVP:** just NAND gates
- **NOR CVP:** just NOR gates
- **Monotone CVP:** just AND & OR gates
- **Alternating monotone CVP:** (AMCVP)
  - input \( \rightarrow \) output path alternates AND/OR, starting & ending with OR
- **Fanin-2, fanout-2 AMCVP:** (AM2CVP)
  - all gates have in & out degree 2
  - allow outputs other than one of interest
- **Synchronous AM2CVP:** (SAM2CVP)
  - all inputs to each gate have same depth

**Planar CVP:** planar circuit  [Goldschläger - SIGACT 1977]
- use NAND crossover
- but: planar monotone \( \not\in \text{ NC} \) [Yang - FOCS 1991]
- start & end with ORs
- reduce fan out to \( \leq 2 \) (also fanin to \( \leq 2 \))
- make AND & OR alternate
- fanin 1 \( \rightarrow \) fanin 2
  (preserving alternation & start with OR)
- fanout 1 \( \rightarrow \) fanout 2
  by duplicating circuit \( x \rightarrow x \& x' \)
  & combining extra outputs
  (preserving alternation & end with OR)
- synchronization: \( n = \# \text{gates} \)
  - \( n/2 \) copies of circuit
  - \( i \)th copy = levels \( 2i \) & \( 2i+1 \)
    inputs & ANDs ORs
  - OR takes inputs from \( i \)th copy,
    sends outputs to \((i+1)\)st copy
    (determining ANDs by alternation)
  - AND in 0th copy become \( 0 \) input
    \( \Rightarrow \) level 0 = inputs
  - inputs fed to \( i \)th copy by input gadget
  - output in \( n/2 \) copy
**Bounded DCL:**
- edges are active (just flipped) or inactive
- vertex active if its active incoming edges have total weight \( \geq 2 \)
- round = reverse unreversed edges pointing to active vertices
  (& these are the new active edges)

- P-complete for AND, SPLIT, OR graphs 
  (but not necessarily planar)
- reduction from Monotone CVP
- even easier from SAM2CVP
Lexically first maximal independent set:
- as found by greedy algorithm: \( \Rightarrow \epsilon P \)
  \[ S = \emptyset \]
  \[ \text{for } v = 1, 2, \ldots, |V| : \]
  \[ \text{if } v \text{ not adjacent to } S : \]
  \[ S = S \cup \{ v \} \]
- decision question: is \( v \in S ? \)
- \( \Pi \)-hard: [Greenlaw, Hoover, Ruzzo - Book 1995]
  - reduction from NOR CVP
  - number gates & inputs in topological order
  - drop edge orientations
  - add vertex \( \emptyset \) connected to all \( \emptyset \) inputs
  \[ \Rightarrow v \in S \iff v = \emptyset \text{ or gate } v \text{ outputs true} \]

- computing whether size \( \leq k \) also \( \Pi \)-complete:
  - reduction from previous problem
  - connect \( v \) to \( n + 1 \) new vertices, set \( k = n \)
  \[ \Rightarrow \text{size } \leq n \iff v \in S \]

- gap-producing reduction: \( n + 1 \rightarrow n^c \)
  \[ \Rightarrow n^{1-\varepsilon} \text{-gap problem is } \Pi \)-complete
  \[ \Rightarrow n^{1-\varepsilon} \text{-approximation is } \Pi \)-complete

- Game of Life: cell \((x,y)\) alive at unary time \(t\)?
- 1D cellular automata
- acyclic Generalized Geography
- is point \(p\) on \(k\)th convex hull of point set?
- multiset ranking: given \(k\) lists, is \(x\) the \(k\)th smallest in the union?
- \(a \mod b_1 \mod b_2 \cdots \mod b_n = 0\)?
- first fit decreasing bin packing \(\frac{\not{\square}}{\square}\) strongly P-complete
- LP with coefficients 0 & 1
- max flow \(-\) has fully RNC approx. scheme

OPEN:

- are two numbers relatively prime?
- \(a^b \mod c\)
- feasibility of LP with \(\leq 2\) variables per inequality
- maximum edge-weighted matching
  - pseudo RNC algorithm
- bounded-degree graph isomorphism