| 6.890 | Lecture 20 | Nov. 13, 2014 |
|---|---|---|

↳ actually 11

# Bounded team private-information games:

NEXPTIME-complete [Peterson, Reif, Azhar — C&M 2001] [Peterson & Reif — FOCS 1979]

- Dependency QBF (DQBF):

$$\underbrace{\forall X_1 : \forall X_2}_{\text{black player}} : \underbrace{\exists Y_1(X_1)}_{\substack{\text{white 1} \\ \text{only sees } X_1}} : \underbrace{\exists Y_2(X_2)}_{\substack{\text{white player 2} \\ \text{only sees } X_2 \text{ variables}}} : \text{CNF formula}$$

  - can white force a win? (satisfied formula)
  - only one round! (multiple rounds don't help)
  - $\in$ NEXPTIME: guess $Y_1$ $\underline{\forall X_1}$ & $Y_2$ $\underline{\forall X_2}$
    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 → long chain (black threat)
  - white players set their vars.
  - chosens → unlock all → formula activation
  - white wins (just in time) if formula satisfied

# Unbounded team private-information games: undecidable

[Hearn & Demaine]

(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 → halt of the form # state$_0$ # state$_1$ # ⋯ # haltstate
- 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 $(x_1, \overline{x_2})$ moves until 1 reports #, $\rightarrow$ 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→2 ↳ $O(1)$ space!
- white strategies must work for all possible black moves ⇒ valid computation history

- Team Formula Game:
  - black sets $X$ such that $F(X, X', Y_1, Y_2)$ (else lose)
  - black wins if $G(x)$ ↕ $F \Rightarrow \neg F'$
  - 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:

- book by Greenlaw, Hoover, Ruzzo [Oxford 1995]
  "Limits to Parallel Computation: P-Completeness Theory"

**NC** (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,
  compute rank = sum of <'s
  via binary tree

  } $O(\lg n)$ time on $O(n^2)$ proc.

**P-hard** = all problems $\in$ NC can be reduced via NC algorithm to your problem
Karp-style reduction
$\Rightarrow \notin NC$ if $NC \neq P$

**P-complete** = $\in P$ + P-hard

# Base P-complete problems:

## Generic Machine Simulation Problem:
given a sequential algorithm & time bound $t$ written in unary, does it say YES within $t$?
to make $\in P$ ~ else EXPTIME-complete

## Circuit Value Problem (CVP): [Ladner - SIGACT 1975]
given an (acyclic) Boolean circuit & input bits, is the output TRUE?
0 & 1

NAND CVP: just NAND gates
NOR CVP: just NOR gates
Monotone CVP: just AND & OR gates
Alternating monotone CVP: (AMCVP)
input → 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 [Goldschlager - SIGACT 1977]
- use NAND crossover
- but: planar monotone $\in$ NC [Yang - FOCS 1991]

**Reductions:** [Greenlaw, Hoover, Ruzzo - book 1995]

- start & end with ORs
- reduce fan out to $\le 2$ (also fanin to $\le 2$)
- make AND & OR alternate
- fanin 1 $\to$ fanin 2
  (preserving alternation & start with OR)
- fanout 1 $\to$ fanout 2
  by duplicating circuit $x \to x$ & $x'$
  & combining extra outputs
  (preserving alternation & end with OR)
- synchronization: $n = \#$gates
  - $n/2$ copies of circuit
  - $i$th copy = levels $\underbrace{2i}_{\text{inputs \& ANDs}}$ & $\underbrace{2i+1}_{\text{ORs}}$
    - 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: [Hearn & Demaine]

- 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: $\Rightarrow \in P$

  $S = \emptyset$
  for $v = 1, 2, \ldots, |V|$:
  if $v$ not adjacent to $S$:
  $S = S \cup \{v\}$
- decision question: is $v \in S$?
- P-hard: [Greenlaw, Hoover, Ruzzo - book 1995]
  - reduction from NOR CVP
  - number gates & inputs in topological order
  - drop edge orientations ↳ ($\in NC$)
  - add vertex $\emptyset$ connected to all $\emptyset$ inputs

  $\Rightarrow v \in S \Leftrightarrow v = \emptyset$ or gate $v$ outputs true

- computing whether size $\leq k$ also P-complete:
  - reduction from previous problem
  - connect $v$ to $n+1$ new vertices, set $k = n$

  $\Rightarrow$ size $\leq n \Leftrightarrow v \in S$

- gap-producing reduction: $n+1 \rightarrow n^c$

$\Rightarrow n^{1-\varepsilon}$-gap problem is P-complete
$\Rightarrow n^{1-\varepsilon}$-approximation is P-complete

# More P-complete problems:

[Greenlaw, Hoover, Ruzzo - book 1995]

- 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?
- multilist ranking: given $k$ lists, is $x$ the $k$th smallest in the union?
- $a \bmod b_1 \bmod b_2 \cdots \bmod b_n = 0$?
- first fit decreasing bin packing
- LP with coefficients 0 & 1

$\}$ strongly P-complete

- max flow — has fully RNC approx. scheme

$\boxed{\text{OPEN}}$:

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