6.892	Class 7	Mar. 30, 2019

2-player games naturally make quantifiers alternate

NP → PSPACE

Recall: PH

set of variables

Σ_k $\exists X_1 : \forall X_2 : \cdots X_k : \varphi(X_1, \ldots, X_k)$

$\Pi_k = co\Sigma_k$ $\forall X_1 : \exists X_2 : \cdots X_k : \varphi(X_1, \ldots, X_k)$

moves by player 1

Mate-in-k $\in \Sigma_{2k-1}$ — can I force my win?

Lose-in-k $\in \Sigma_{2k}$ — can I force my loss?

2nd player mate-in-k $\in \Pi_{2k+1}$ — can they force their win?

Real variant of NP & PSPACE:

polynomials

- $\exists\mathbb{R}$: $\exists x_1 \in \mathbb{R} : \cdots : \exists x_n \in \mathbb{R} : P_1(x_1, \ldots, x_n) \geq 0$
  $\wedge \cdots \wedge P_m(x_1, \ldots, x_n) \geq 0$

  $\subseteq$ PSPACE [Canny 1988]

  e.g.: art gallery problem (k guards to see polygon)
  unit-disk graph recognition
  drawing k planar graphs on same vertices
  Nash equilibria of multiplayer games
  linkage flexibility are $\exists\mathbb{R}$-complete

- First-Order Theory of Reals: $\exists : \forall : \exists : \forall : \cdots$
  $\subseteq$ 2EXPTIME [Renegar 1989]
  - k alternations $\Rightarrow 2^{2^{O(k)}} \cdot n^{O(1)}$

Gadget general model: (for robot motion planning)

- locations (entrances/exits)
- states
- transitions $(l_1, s_1) \to (l_2, s_2)$ } s_1: $l_1 \xrightarrow{s_2} l_2$

transition graph — visual notation

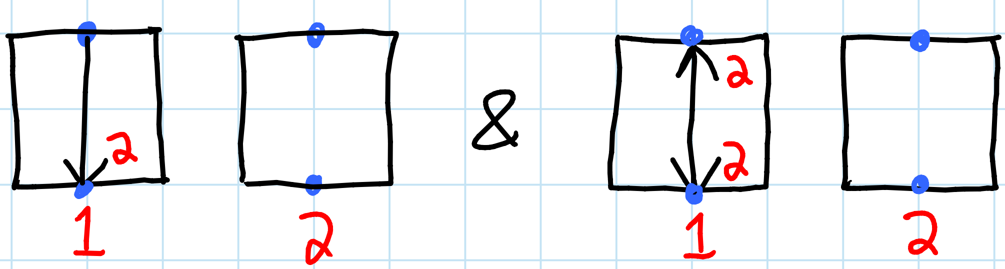
[Demaine, Grosof, Lynch, Rudoy – FUN 2018]
[Demaine, Hendrickson, Lynch – arXiv 2018]

Gadget types:

- k-tunnel = all transitions are along edges of perfect matching on locations
  (states can control traversability & directions)
- DAG = state-transition graph is acyclic
  ↳ possible transitions on states (merging all locations)

Characterization: 2-player motion planning with DAG gadgets is PSPACE-complete iff some gadget is nontrivial: has ≥1 transition
[Demaine, Hendrickson, Lynch – arXiv 2018]

Examples:



single-use 1-way   single-use 2-way