0-player games (simulations)
- polynomial # moves \( \rightarrow \) \(P\)
- polynomial space \( \rightarrow \) \(PSPACE\)
- infinite space \( \rightarrow \) undecidable

Conway's Game of Life: \[\text{[Conway 1970]}\]
- cellular automata
- live cell lives \(\iff\) exactly 2 or 3 live neighbors
- dead cell becomes live \(\iff\) exactly 3 live neighbors
- \(PSPACE\)-complete in finite board
- Turing machine simulation \[\text{[Paul Rendell 2000]}\]
  (pushdown automaton with 2 stacks)
- undecidable in infinite board (dead outside input)
- growing Turing machine \[\text{[Paul Rendell 2000]}\]
- \(2\)-counter machine = Minsky machine (exponential slowdown!)
  \[\text{[Conway, Berlekamp, Guy - Winning Ways 1982]}\]
- wire, terminator, turn, delay
- shift: many offsets cause glider destruction
- AND & OR gates
- kickback \(\rightarrow\) thinning \(\rightarrow\) crossover
- split & NOT \(\text{(complicated)}\)
- counter registers, test, create, push/pull
- precise glider positioning away from guns
- self-destruction via boomerangs
Deterministic Constraint Logic (DCL)
- edges can also be active or inactive
- vertex active if its active incoming edges' weight $\geq 2$
- in each round:
  - reverse inactive edges pointing to active vertices
  - reverse active edges pointing to inactive vertices
  - these are the new active edges

- PSPACE-complete even for planar AND/OR graphs
  - guarantee gadget inputs reverse at $t = 0 \mod 4$
  - quantifier gadgets use new "switch" & degree-2 vertices to control timing
- CNF formula uses AND', OR', split' gadgets which take inputs & return acknowledgments (fixes timing & "blow-back")
  - trick to guarantee first input of AND' activates before second (if they both do)
- remove degree-2 vertices
  - edge $\rightarrow$ 4-path & remove red-red vertices
  - remove blue-blue vertices
  - remove red-blue vertices (timing is OK)
- crossover gadget
Multiplayer games:
- typical question: given a game position, can next-player-to-move force a win?
- in worst case, other players collude against you, effectively acting as one player

2-player games:
- call players “white” & “black” (as in Chess, Go, …)
- polynomial # moves \(\in\) PSPACE:
  \(\exists\) move : \(\forall\) responses : \(\exists\) move : \(\forall\) responses : …
  (rules & I win, in 3CNF) \(\in\) Q3SAT

SAT games:

\[\text{Schaefer - JCSS 1978}\]
- QSAT is a 2-player game: \(G_w(CNF)\)
  - player 1 chooses \(x_1\), player 2 chooses \(x_2\), …
  - player 1 wins \(\iff\) formula satisfied

- impartial games: (both players have same moves)
  - on turn, player sets any unassigned variable

- partizan games: (different moves for players)
  - white variables & black variables (50/50%)
  - on turn, player sets unassigned var. of same color

- default game: player 1 wins \(\iff\) formula satisfied
- seek game: win if first to satisfy formula
- avoid game: lose if first to satisfy formula
- **PSPACE-complete:**
  - impartial game positive 11-SAT
  - impartial game positive 11-DNF SAT
  - partizan game CNF SAT
  - impartial/partizan avoid positive 2-DNF SAT
  - impartial/partizan seek positive 3-DNF SAT
  - impartial/partizan avoid positive CNF SAT
  - impartial/partizan seek positive CNF SAT

**Kayles:** (≈ indep. set)  
- (impartial) node Kayles:
  - on turn, player adds node to independent set
  - lose if can’t move
- (partizan) bipartite node Kayles:
  - white vs. black nodes is the bipartition

**Geography:** (generalization of word game) (≈ longest path)
- given (directed) graph & start node for token
- on turn, player moves token along (directed) edge
- node geography: can’t revisit nodes
  - directed PSPACE-complete [Lichtenstein & Sipser 1980]
  - undirected ∈ P [Fraenkel, Scheinerman, Ullman 1993]
- edge geography: can’t revisit edges
  - directed PSPACE-complete [Schaefer - JCSS 1978]
  - undirected PSPACE-complete [Fraenkel, Scheinerman, Ullman - TCS 1993]
Reversi/Othello:
- move = ○ ○ ○ ○ ○ & ⇒ ○ ○ ○ ○
  - reverse in between 1 & 8 directions

- PSPACE-complete [Iwata & Kasai - TCS 1994]
- polynomial # moves: move consumes board
- reduction from directed node geography
- in bipartite max-degree-3 graph
- rightward chains are threats by black:
- black takes α, then α', then corner, then all of bottom territory ⇒ win
- white wins if black can’t move
- degree-2 vertices: ➔
- degree-3 vertices: ➔ & ➔
  - if double visited
  - then white or black wins
  - black or white chooses
**ASIDE:**

**Bounded NCL:** NP-complete
- each edge can be reversed only once
- NP-complete for planar constraint graphs with **AND, SPLIT, OR, CHOICE** vertices
  - differ in initial edge orientations
  - planar via crossover
  - similar to proof of Constraint Graph Satisfaction

**Bounded 2-player Constraint Logic (2CL):**
- each edge is either white or black
- each edge can be reversed only once
- goal:
  - each player has target edge
  - player unable to move loses
- PSPACE-complete for planar constraint graphs with white **AND, SPLIT, OR, CHOICE** & VARIABLE vertex
- reduction from impartial game positive CNFSAT
- players take turns setting variables
- positive ⇒ white wants true, black wants false
- black can’t win (edge irreversible)
- white wins ⇔ formula satisfied
- crossover gadget (only use of CHOICE)
- can make OR protected using free edge
  - no constraint at degree-1 end