Battle Sheep is PSPACE-complete

Kyle Burke & Hirotaka Ono

Florida Southern College Nagoya University

Virtual CGT Seminar, November 17, 2025

Talk Plan

- Describe Battle Sheep
- Play Battle Sheep
- Plug Sprouts 2026
- PSPACE reduction overview
- Show some gadgets

Note: interruptible!

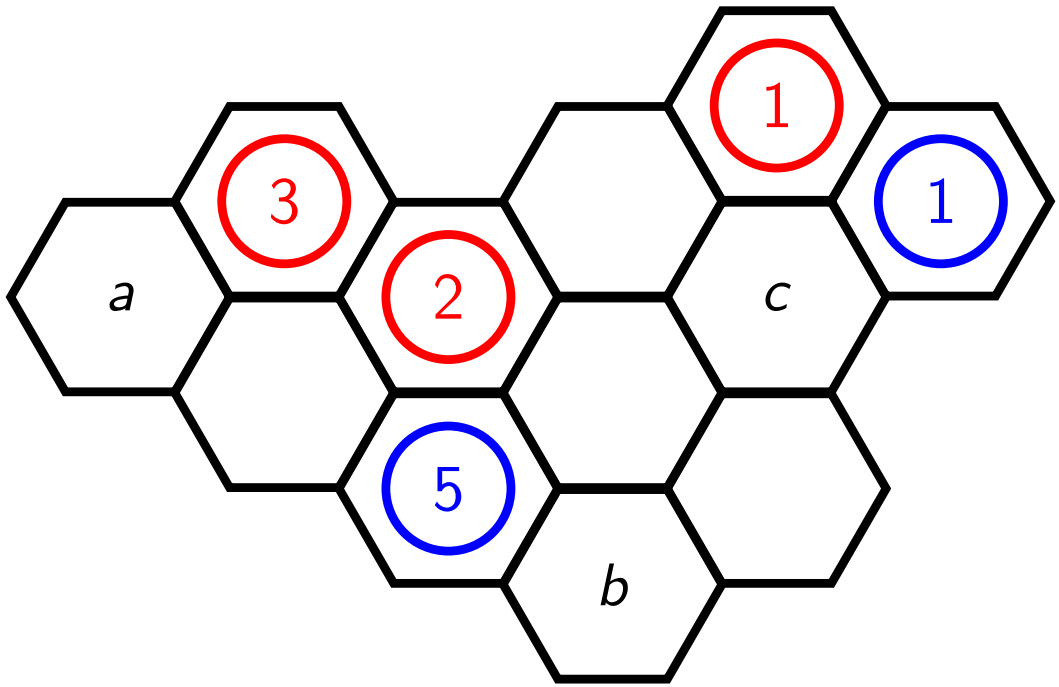
Battle Sheep

Actual published game!

Battle Sheep

- Start: sheep token stack along outer border.
- Game Turn:
  - Choose one of your stacks with more than one sheep. (Say k)
  - Choose some of those tokens (1 to $k-1$) to move as far as you can go in one direction.
  - Have to leave at least one token behind.
- If you can't move on your turn, you lose.
  - (Ignoring actual tiebreaking rule)

Battle Sheep



Battle Sheep

Let's play!
`https://kyleburke.info/DB/combGames/battleSheep.html`
(Recommendation: Width 6, Height 5.)

Battle Sheep

Battle Sheep is a Combinatorial Game!

- Two players ✓(actual game plays 2-4)
- Alternate Turns ✓
- No randomness ✓
- No hidden information ✓
- Normal Play: last person to play wins ✓

Use smallcaps: Battle Sheep

Sprouts 2026

We're playing Battle Sheep at Sprouts 2026

- Sprouts: Abstract games conference where undergrads present
- `https://kyleburke.info/sprouts/sprouts2026/`
- Sprouts 2026: April (Day TBD)
- Held online (Zoom)
- Fun for everyone:
  - Game tournament! 2026 Battle Sheep
  - Computer Tournament (JavaScript)
  - Lots of chatting.

PSPACE-completeness

- "Good" games have simple rules, but tough strategies.
- Not fun if players can always make the best move.
- Best Move $\approx$ Who can win
- Computational Complexity to the rescue!
  - Complexity Classes: categories of problems based on feasibility of running time.
  - P: Polymonial Time: problems that can be solved in time polynomial to the input size.
  - Problems in P are "tractible"; can be solved "efficiently".
  - $\mathrm{EXPTIME}$: problems that require exponential time to solve.
  - P vs NP: $\mathrm{P} \subseteq \mathrm{NP}$, but we don't know whether $\mathrm{P} = \mathrm{NP}$ or $\mathrm{P} \subsetneq \mathrm{NP}$.
  - Many people assume $\mathrm{P} \neq \mathrm{NP}$.

PSPACE-completeness

- PSPACE: computational problems solveable with polynomial-size memory/disk. (No time constraint.)
- $\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXPTIME}$
- Hardest problems in PSPACE (-complete) have no known polynomial-time algorithm. Includes periodic scheduling, deadlock detection, and Amazons, Hex, Konane.
- In PSPACE: Battle Sheep $\in$ PSPACE ✓
- Need to show: PSPACE-hard: every problem in PSPACE can be transformed into an instance of Battle Sheep.
- Inclusion and Hardness: Battle Sheep is PSPACE-complete

PSPACE-completeness

- Prove hardness by finding a reduction.
- f : another PSPACE-hard problem $\rightarrow$ Battle Sheep
  - Preserve winnability: 1st player wins $x \Leftrightarrow$ 1st player wins $f(x)$
  - $f(x)$ computable in time polynomial in size of x
- Then, we can efficiently solve Battle Sheep $\Leftrightarrow$ we can efficiently solve all of PSPACE.
  - If `battleSheepSolver` solves it in polynomial time, then...
  - `function pspaceSolver(A):`

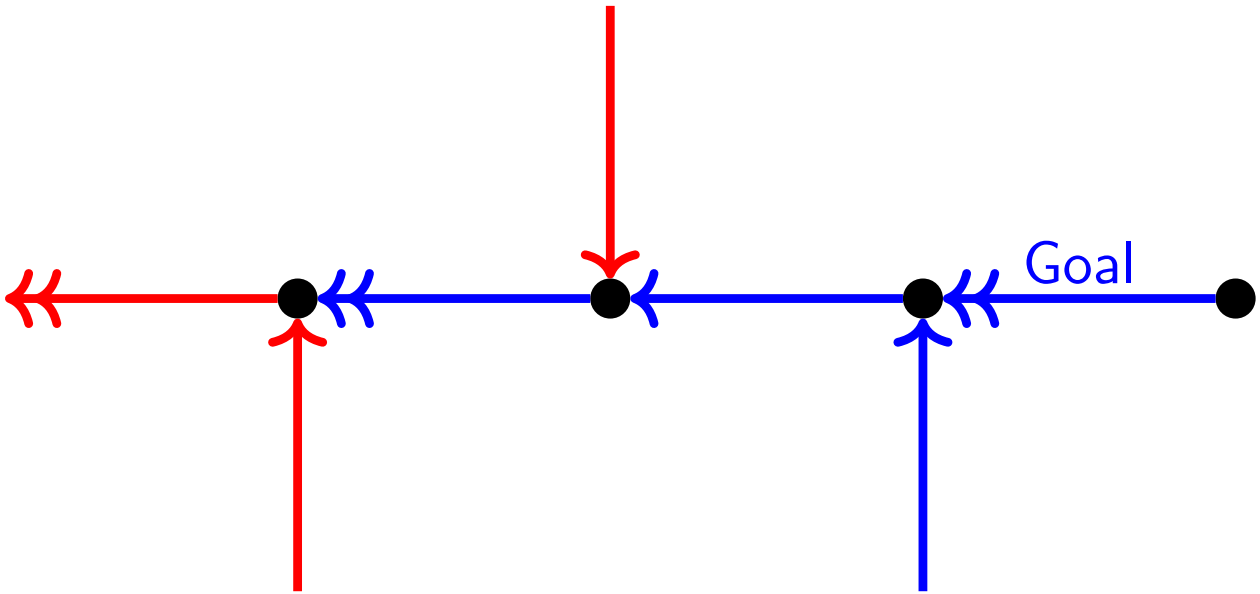
    ```
        bSheep = f(A)
        result = battleSheepSolver(bSheep)
        return result
    ```
  - Now `pspaceSolver` efficiently solves the original problem!
- A working reduction means Battle Sheep is PSPACE-hard.

B2CL

PSPACE-hard from Bounded Two-Player Constraint Logic (B2CL):

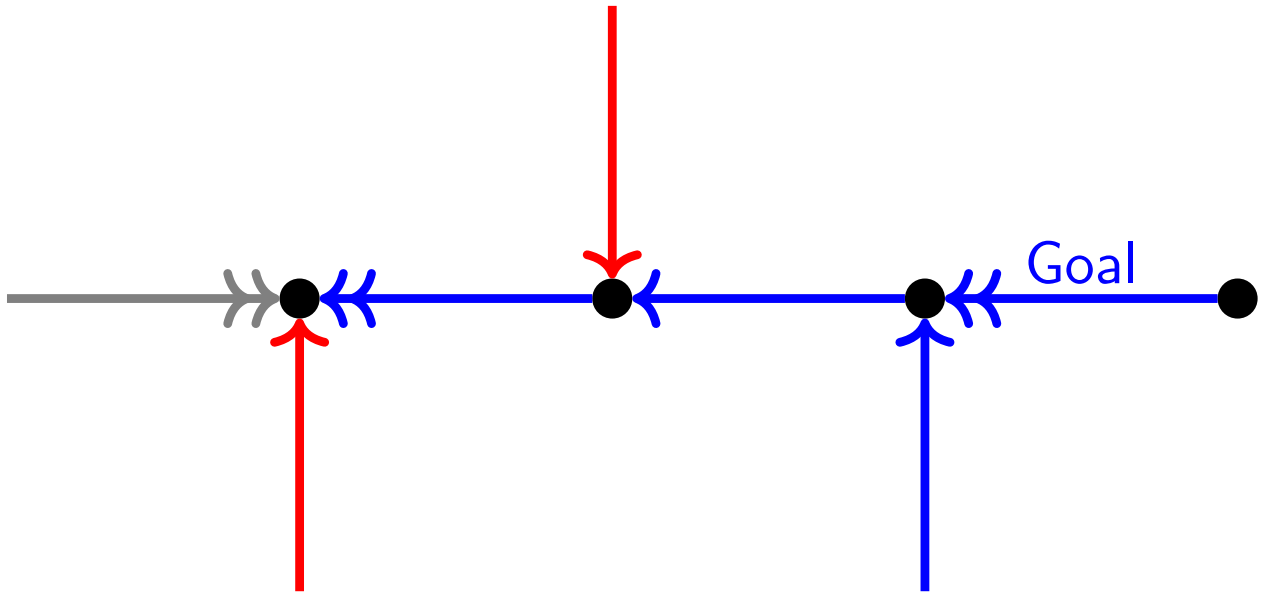
- B2CL is a game about flipping directed edges on graph.
- "Bounded": each edge can only be flipped once.
- "Two-Player": Blue and Red edges.
- Weight 1 and weight 2 edges.
- Each vertex must have in-weight of 2+.
- Win by flipping your goal edge.

B2CL



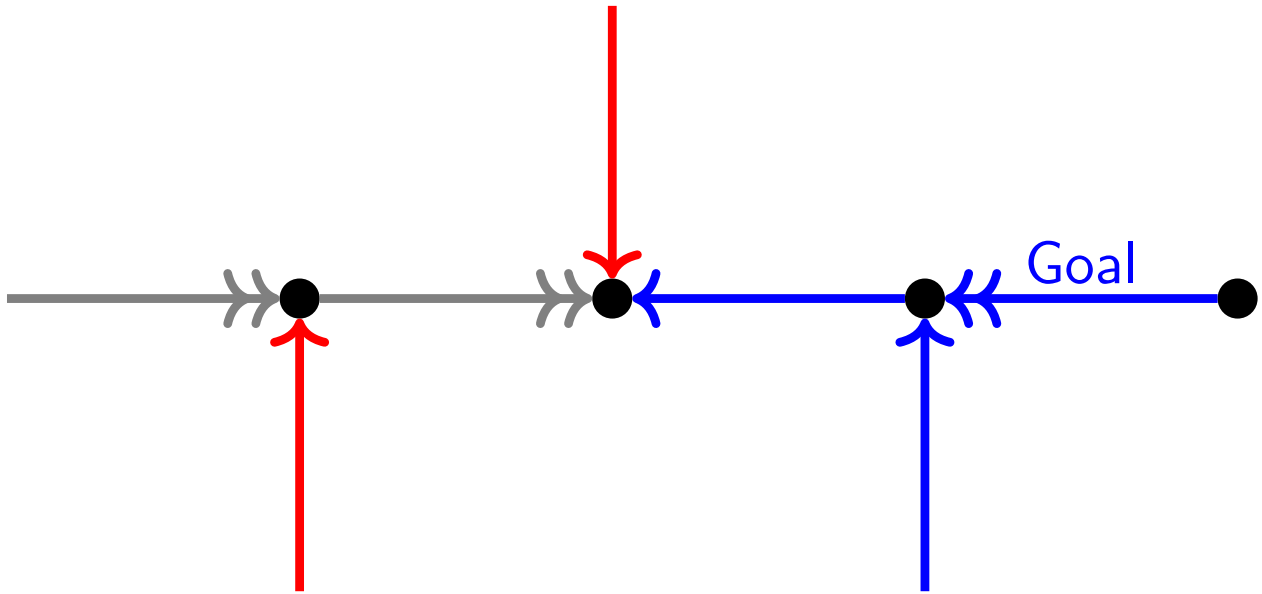
If Red flips left edge, Blue can win in three turns.

B2CL



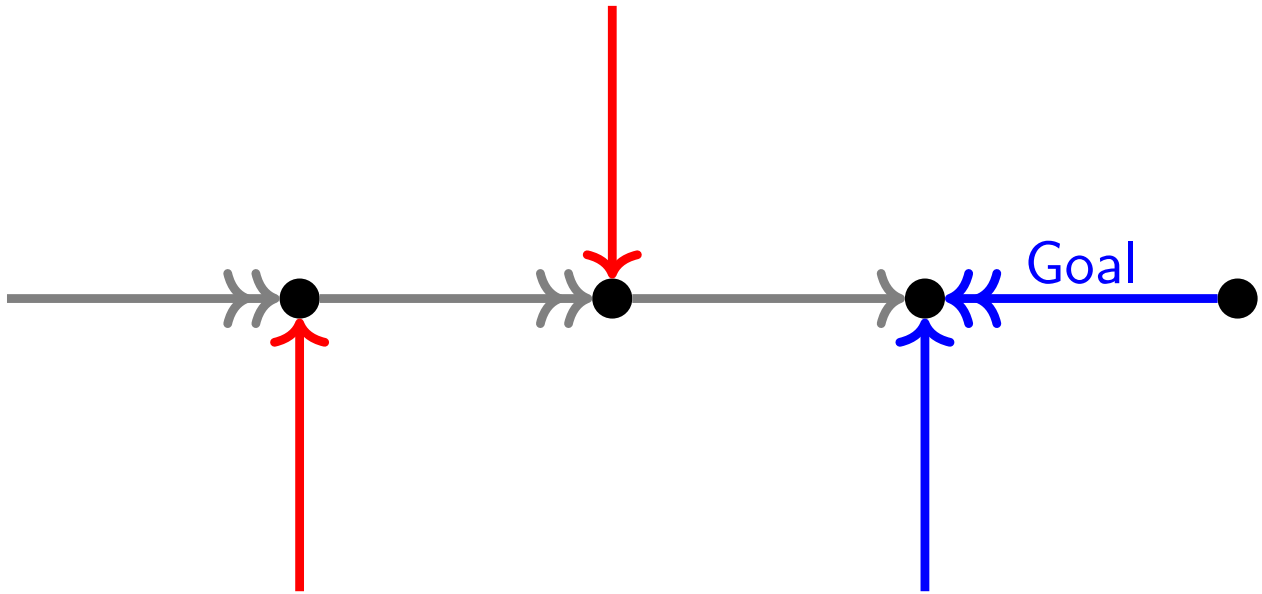
If Red flips left edge, Blue can win in three turns.

B2CL



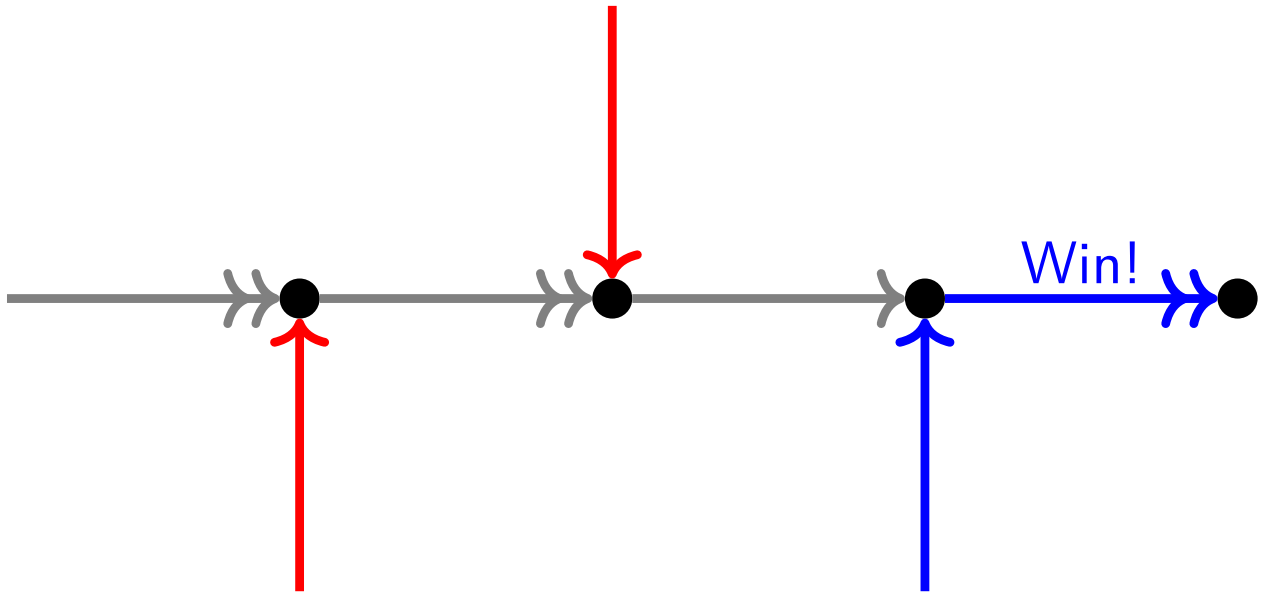
If Red flips left edge, Blue can win in three turns.

B2CL



If Red flips left edge, Blue can win in three turns.

B2CL



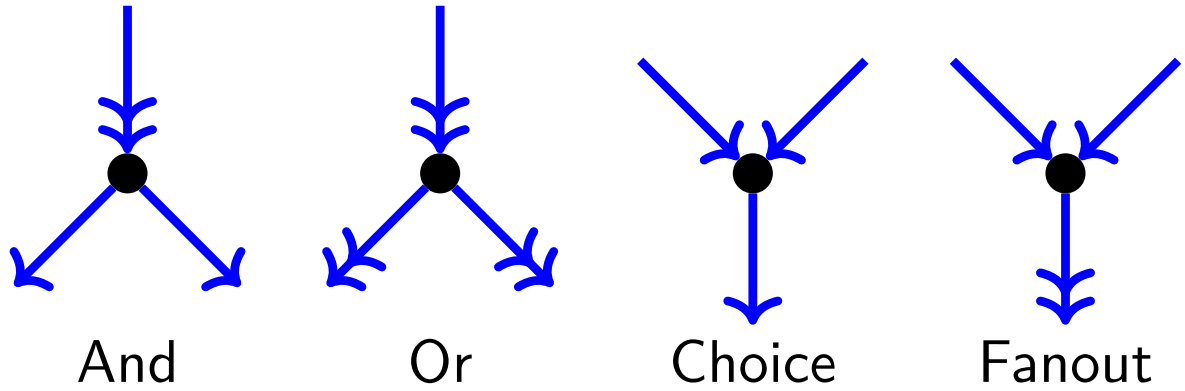
If Red flips left edge, Blue can win in three turns.

B2CL

Let's play a B2CL variant!
Note: no goals, normal play instead.
`https://kyleburke.info/DB/combGames/normalBoundedConstraintLogic.html`
(Recommend: 4 x 5)

B2CL

Amazingly, PSPACE-hard with only five vertex types:



Variable

Use "activate" term for positive/True signal.
(Via reduction from Positive CNF.)

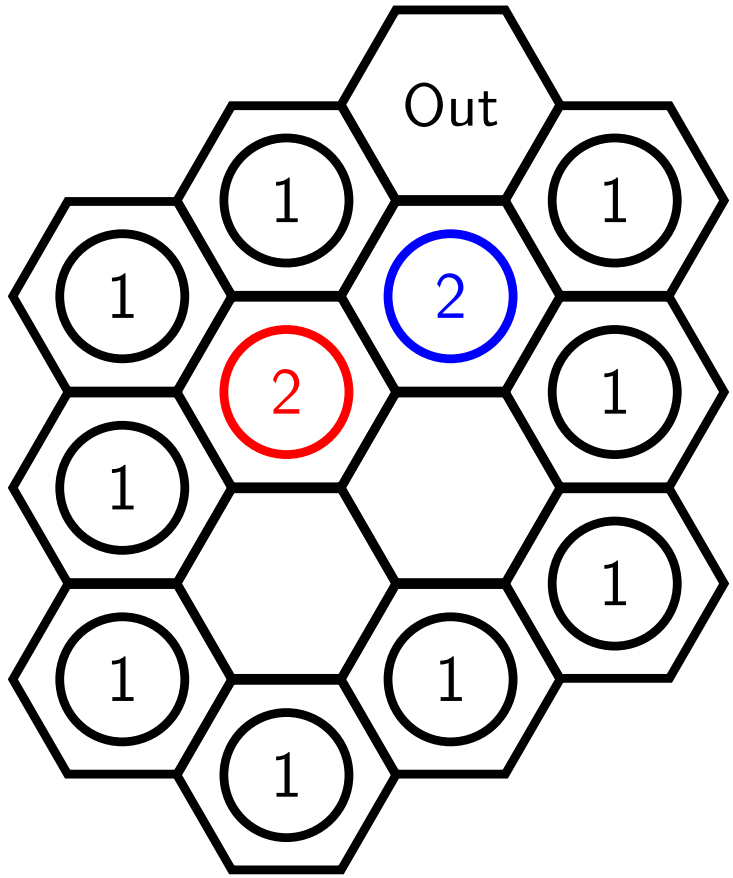
B2CL

Our gadgets include:

- Interface to represent the edges.
- Five from B2CL.
- More because our geometry is more structured:
  - Turns in "wires".
  - Goal gadget
  - Makeup gadget with time-taking Red moves.
- Note: to win, Blue must move to all single-token stacks.

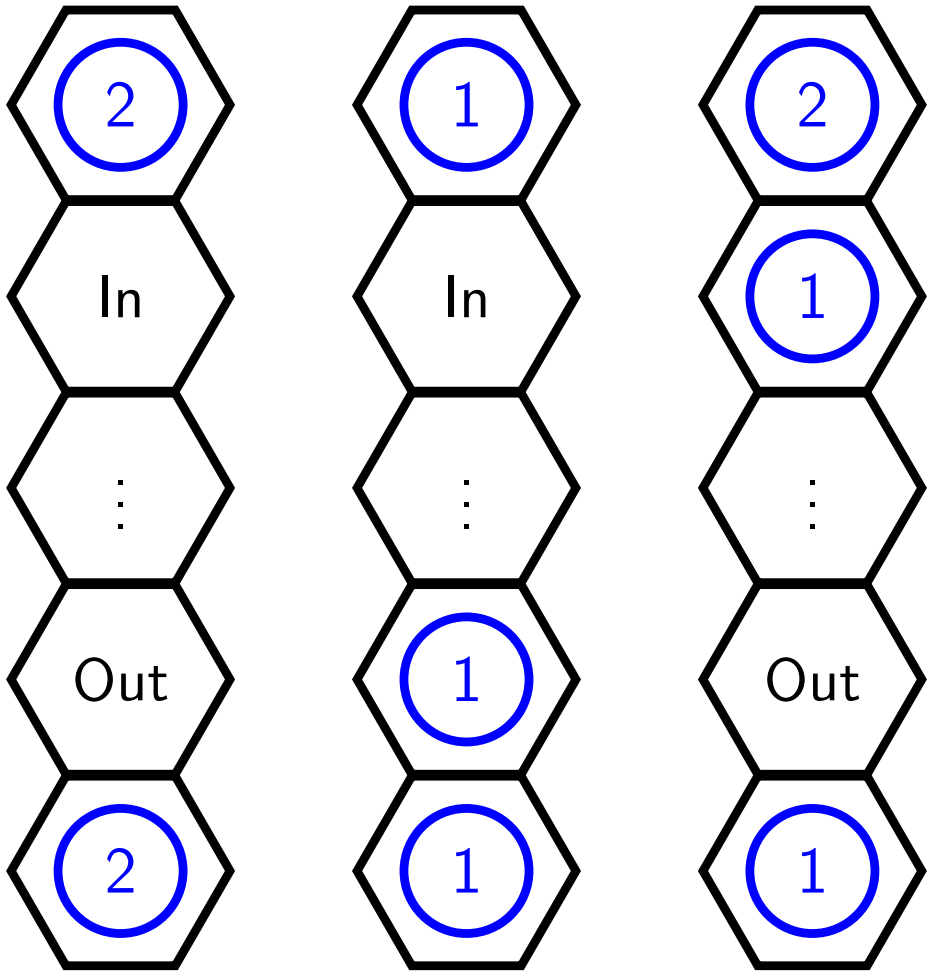
Reduction Gadgets

Variable:



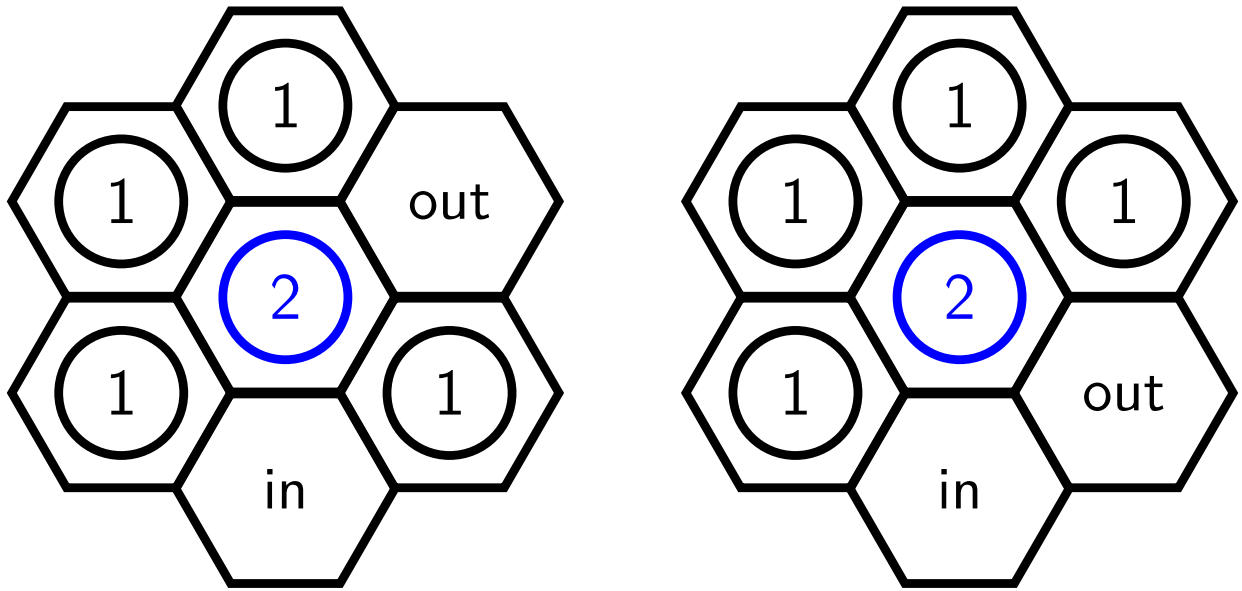
Reduction Gadgets

Interface between gadgets (Wire):



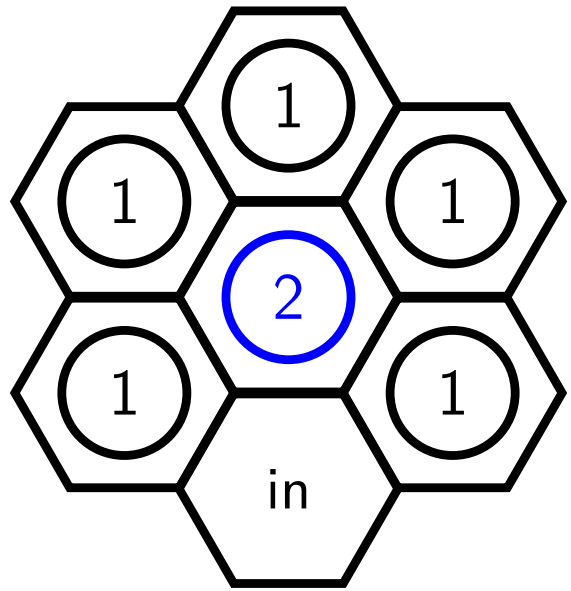
Reduction Gadgets

Wire Turns:



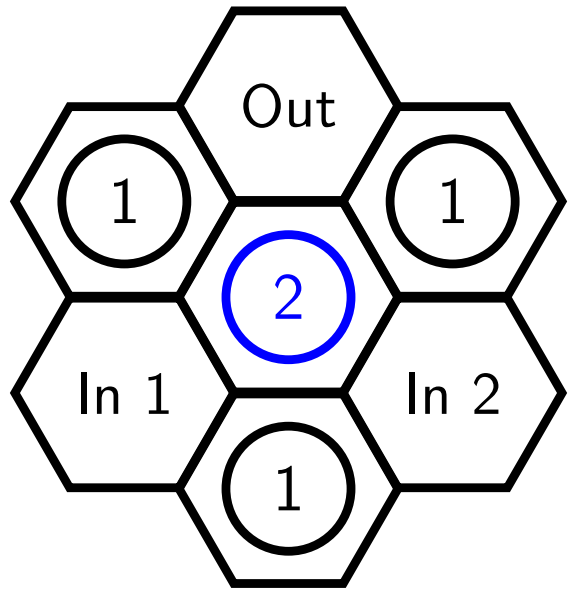
Reduction Gadgets

Goal:



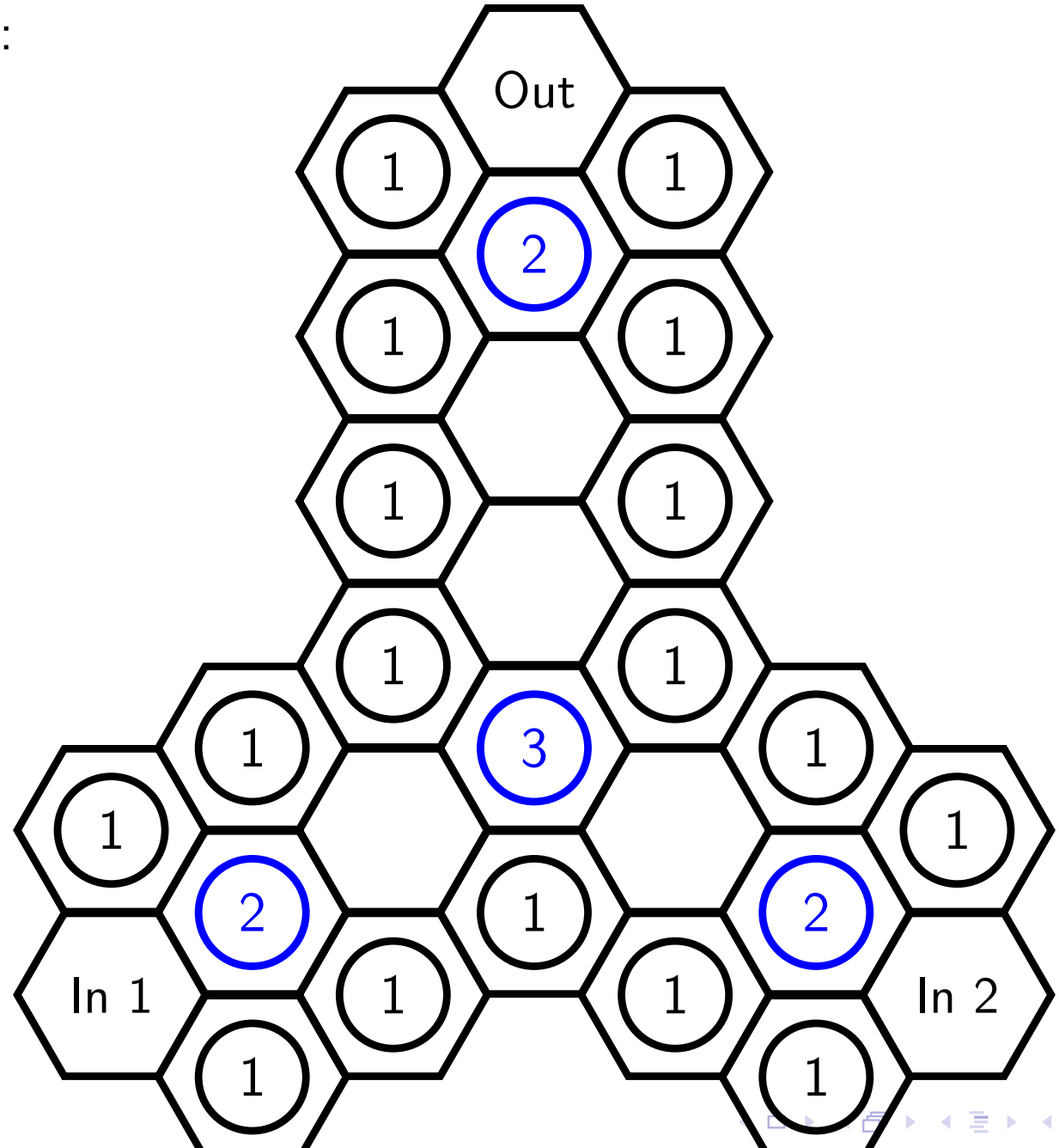
Reduction Gadgets

Or:



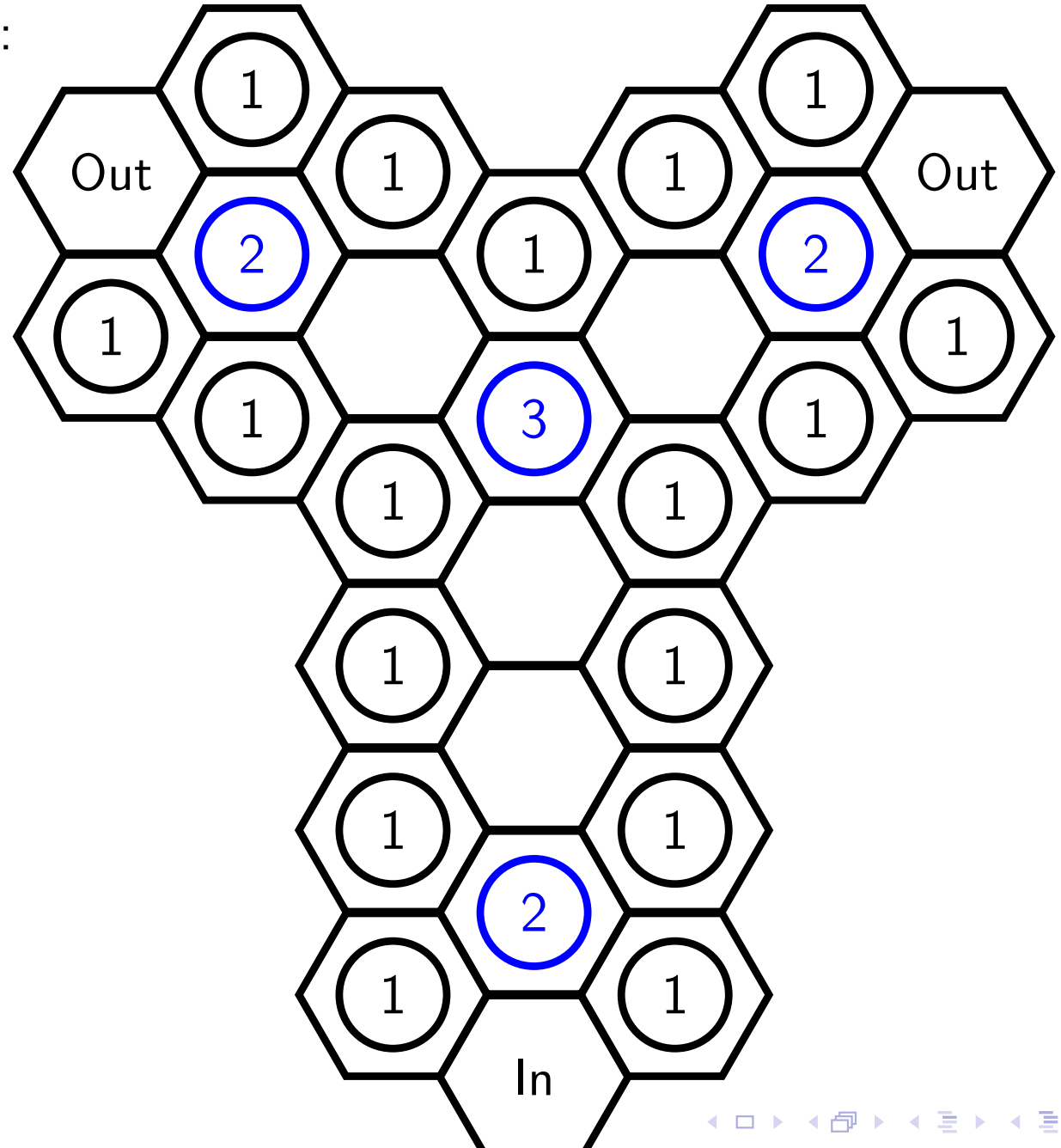
Reduction Gadgets

And:



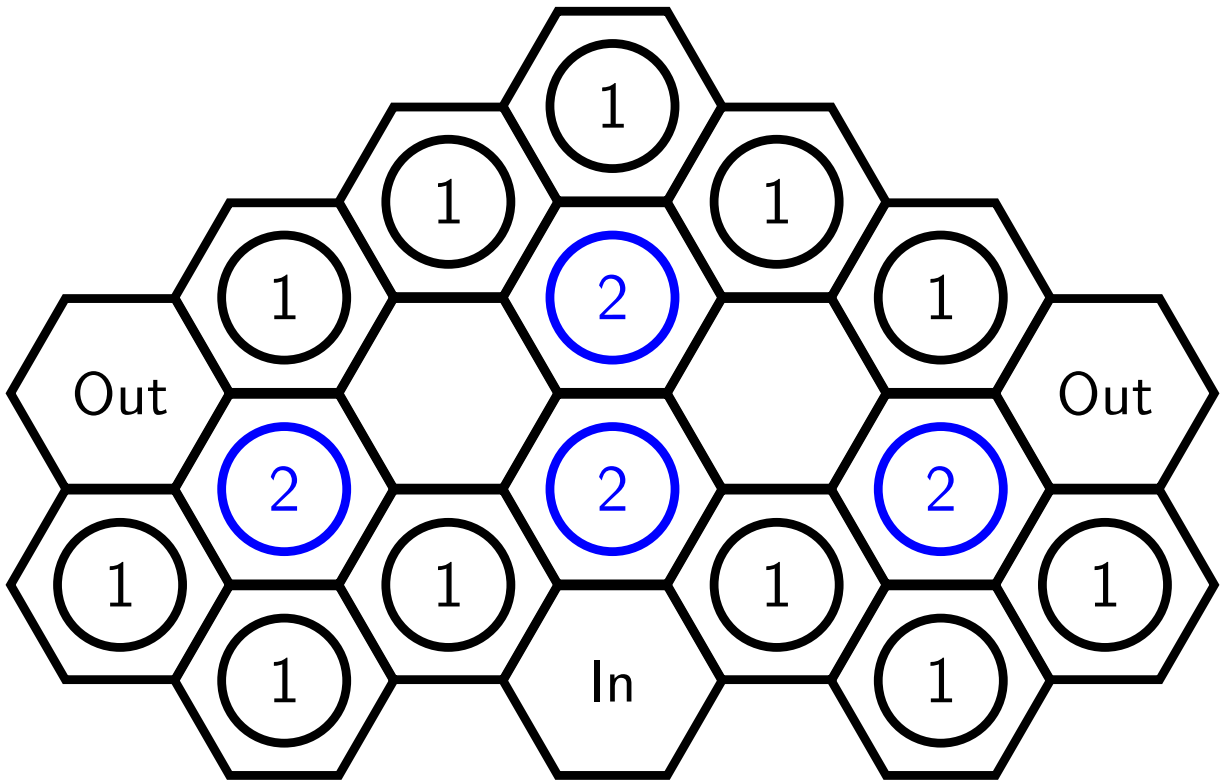
Reduction Gadgets

Fanout:



Reduction Gadgets

Choice:



Reduction Gadgets

- B2CL-needed gadgets:
  - Variable? ✓
  - Or? ✓
  - And? ✓
  - Fanout? ✓
  - Choice? ✓
- Other parts:
  - Wires? ✓
  - Turns? ✓
  - Goal? ✓
- Planarity? ✓B2CL gives a crossover gadget automatically!

The reduction works; Battle Sheep is PSPACE-complete.
(Likely) No polynomial-time solution for general positions.

Battle Sheep is PSPACE-complete

Conclusions!

- Battle Sheep is a fun game.
- Battle Sheep is PSPACE-complete: even more fun! (There are a lot of games like this.)
- You should attend Sprouts and play in the tournament! (Everyone)
- You should write a player for Battle Sheep for Sprouts! (Everyone)
- If you are an undergrad doing a project involving board games, you should give a talk at Sprouts!

Thank you!

Preprint: `http://arxiv.org/abs/2505.06414`

Sprouts: `https://kyleburke.info/sprouts/sprouts2026/`