Course Topics & Rules of Go (Informal)

CMPUT 355: Games, Puzzles, and Algorithms

Recap

- **Course information:** https://jrwright.info/gpaclass/
- **Evaluation:** All in-person (in-class quizzes and a final exam)
- **Topics:** Algorithmic issues and techniques for solving games and puzzles
  1. Evaluating group-based position rules
  2. Search with and without heuristics for single-player puzzles
  3. Minimax and alpha-beta search for two-player perfect-information alternating-move games
  4. Dynamic programming for minimax search
  5. **Domain-specific pruning for minimax search**
  6. **Monte Carlo tree search**
  7. **Noncooperative game theory for simultaneous-move games**

Lecture Outline

1. Course Topics cont.
2. Rules of Go
3. Demo!

After this lecture, you should be able to:

- Describe the basic overview of the (rest of the) course material
- Identify legal and illegal moves in Go

5. Domain-specific pruning

X	X	O
O	O	X
X		

Example games: Hex, tic-tac-toe, Nim

- Alpha-beta pruning is applicable to **any** game that can be represented as a search graph
- **Specific** games have specific structure
- Sometimes can look at a position and easily tell if it's a win/loss/tie
  - E.g., this tic-tac-toe position
  - Can tell if a Nim position is a loss just by doing some arithmetic!
- This can be used for pruning
  - If a state is provably a win, just mark it as a win, no need to actually search to the end

6. Monte Carlo tree search

Example game: Hex

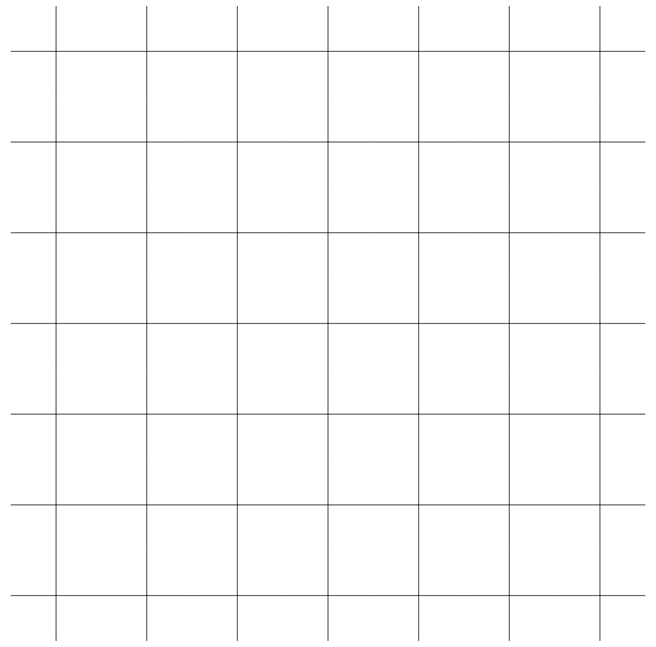
- A game is **solved** if some strategy is known to guarantee a win or draw for one player
  - i.e., must be able to prove that each move is a winning move no matter what the opponent does
- But a proof often requires searching the **whole state graph** (infeasible for Chess, Go, 11x11 Hex, etc.)
- **Question:** what can we do if we don't have time to search the whole tree?
- Monte Carlo tree search explores the state graph efficiently by using **simulations** ("roll-outs") to evaluate intermediate states
  - More promising subtrees are searched more intensively
- Doesn't necessarily **solve** the game, but often **plays** quite effectively

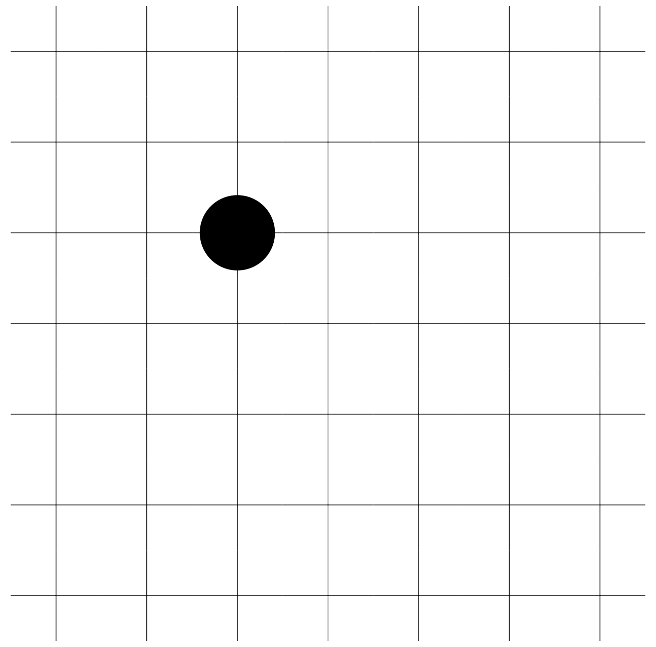
7. Noncooperative game theory

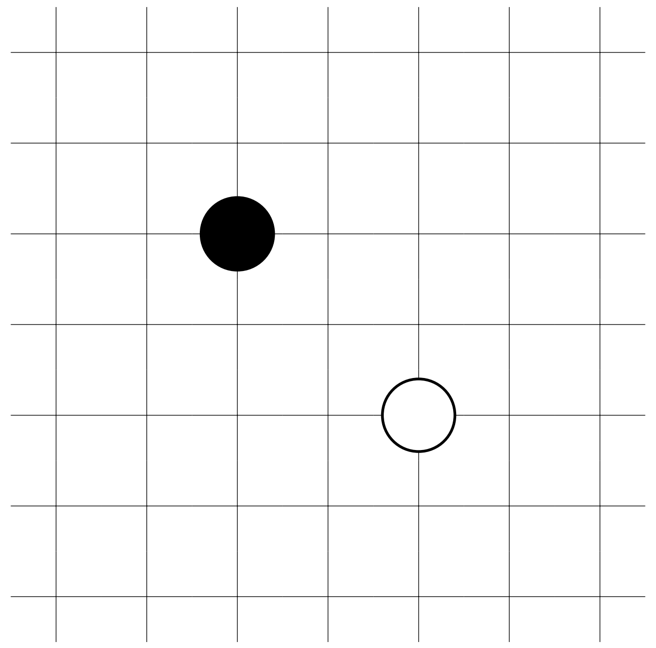
Example game: rock-paper-scissors

- Most of this course:
    - players choose actions sequentially
    - the state of the game is fully known by all players
    - one player's win is another player's loss
- Noncooperative game theory relaxes these assumptions
    - Players may have to move simultaneously (rock-paper-scissors)
    - Players may not know the current state of the game (Stratego, poker)
    - "Win-win" and "lose-lose" outcomes may exist (Prisoner's Dilemma)
- **Question:** What is even an optimal strategy for these games?

Game of Go





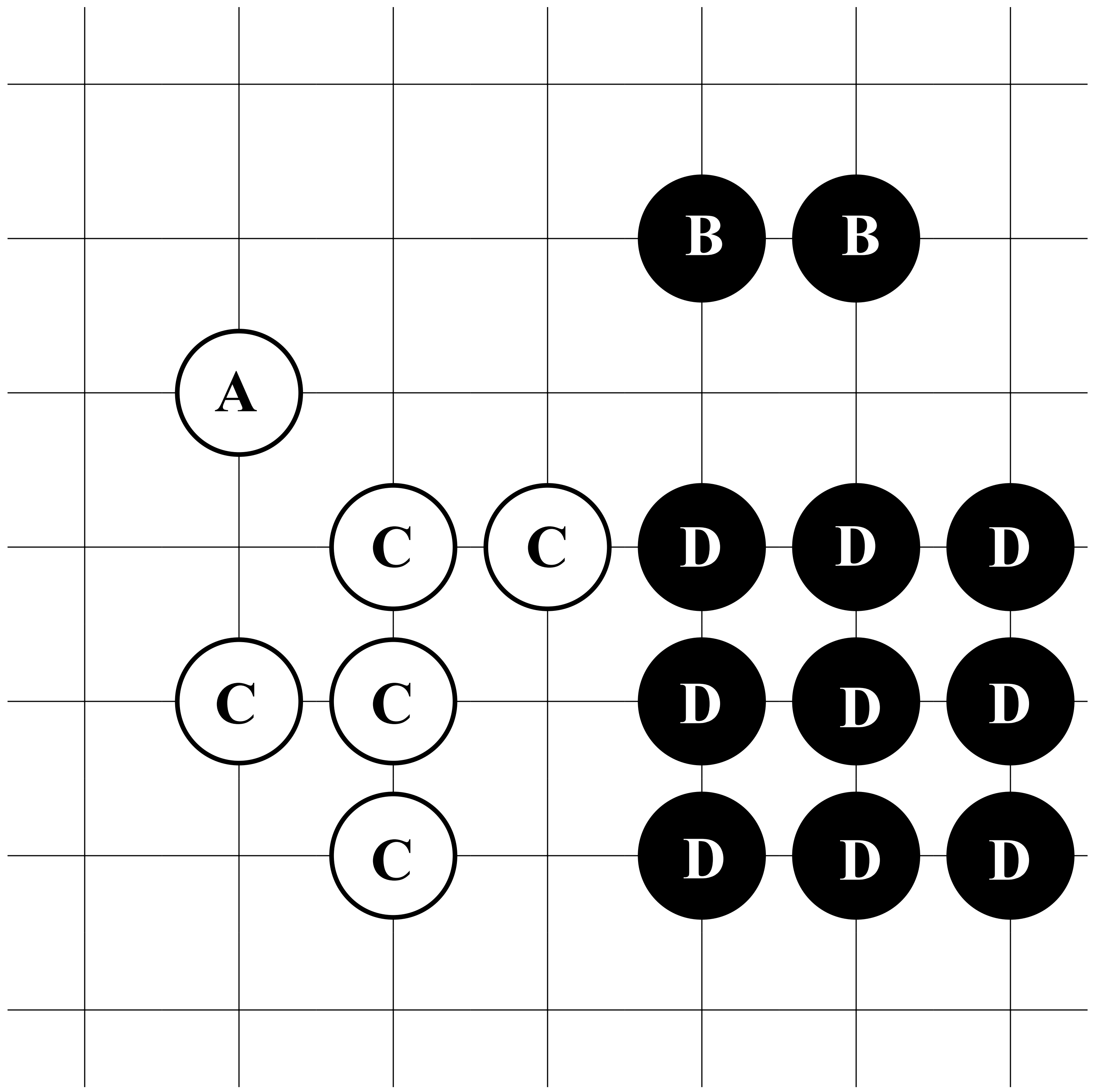


Basic rules:

- Start with an empty grid
- Two players (Black and White)
    - Black moves first
    - Players take alternate turns
- Each move: place a stone of own colour at an **empty intersection**

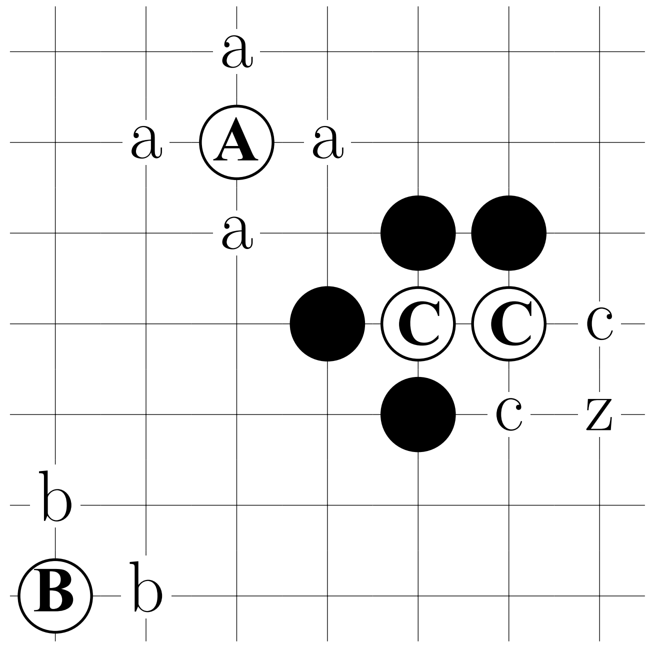
Blocks

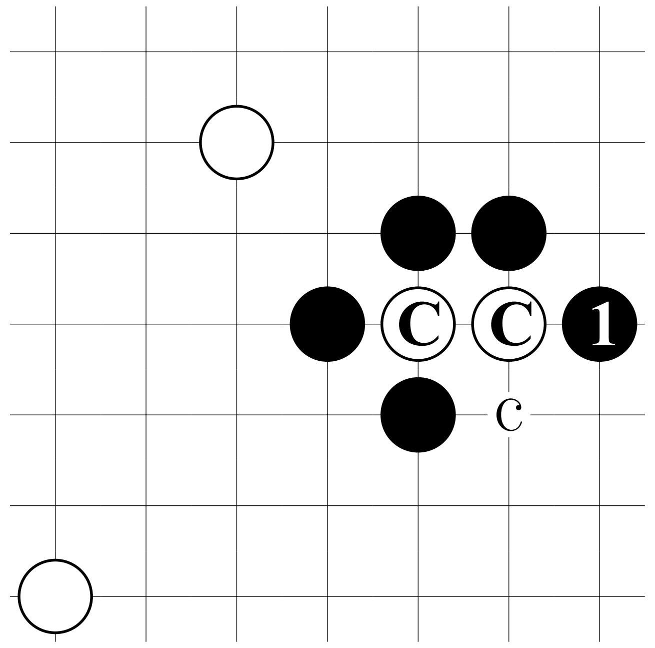
- Two stones of the same colour are **connected** if their intersections are joined by a line
  - B and D are not connected; **why?**
- A **block** is a connected group of stones
  - A is a single-stone block
  - B is a block consisting of two stones joined by a line
  - C is a block of 5 stones; D is a block of 9 stones
- A and C are not the same block; **why?**



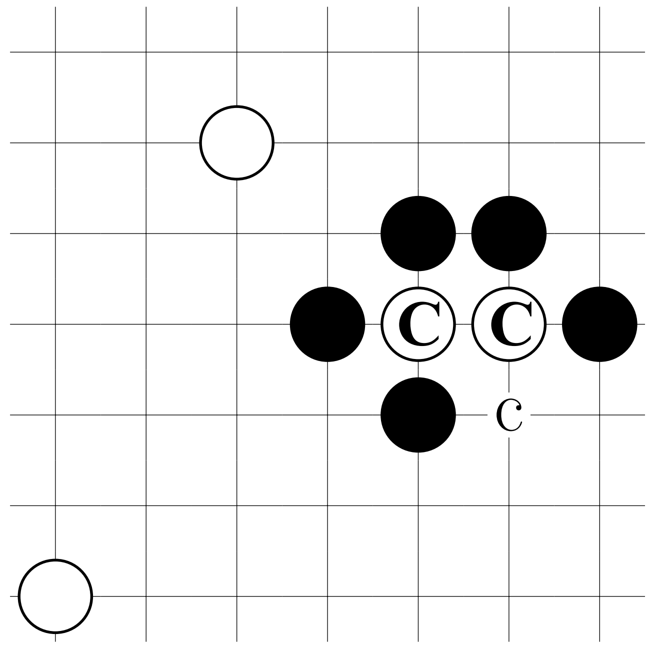
Liberties

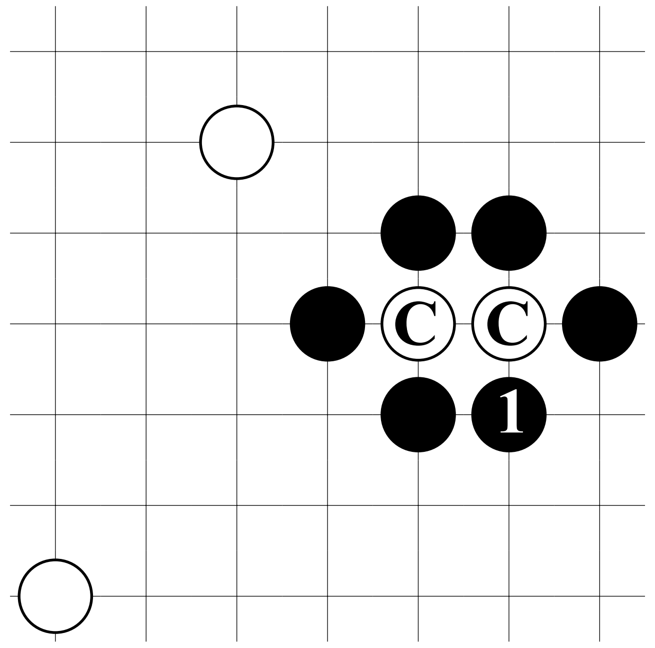
- An empty point is **adjacent** to a stone if it is joined by a line
- An empty point is **adjacent** to a **block** if is is joined by a line to *any* stone in the block
- Empty points adjacent to a block are called **liberties** of the block
  - Block A has four liberties marked $\mathbf{a}$
  - Block B has two liberties marked $\mathbf{b}$
  - Block C has two liberties marked $\mathbf{c}$
  - Point z is **not** a liberty of C; **why?**
- After Black plays 1, block C has only one liberty left
- A block that loses its last liberty is **captured**

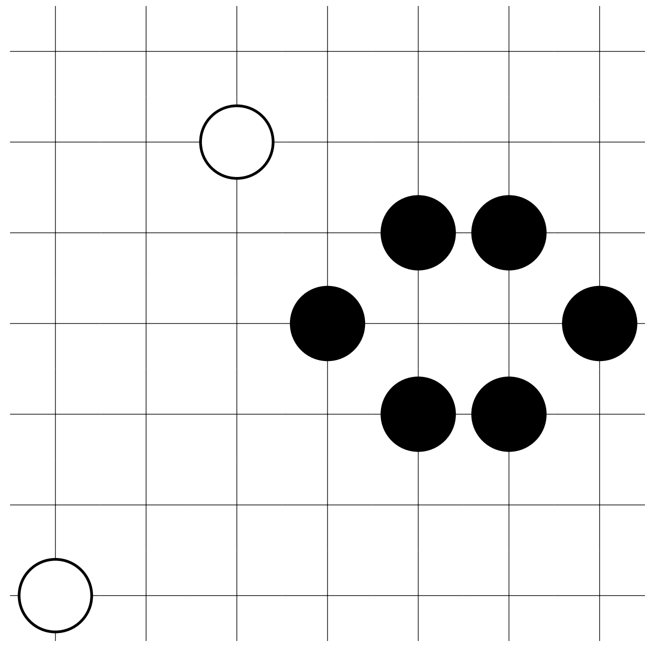




Capture

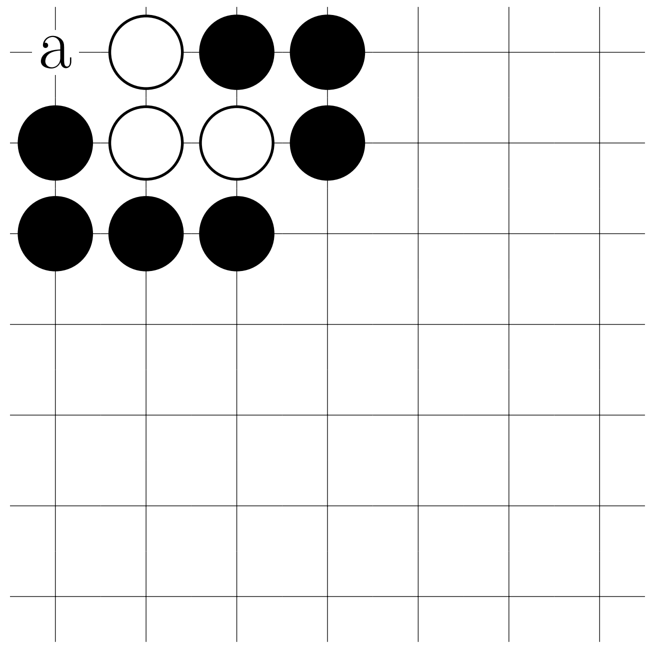


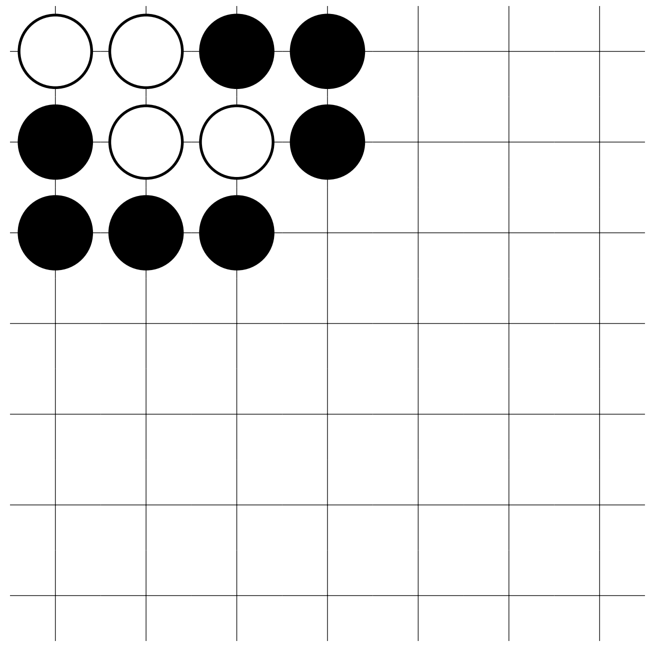


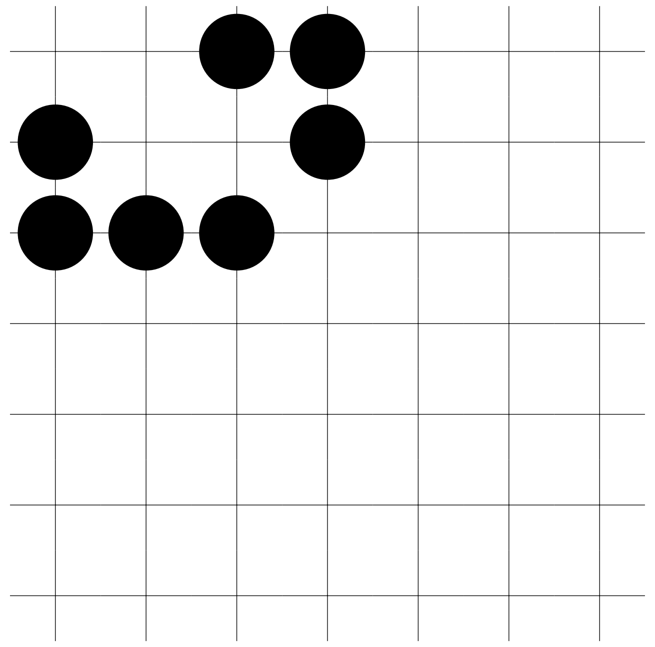


- Block C has only one liberty (at c)
- If Black plays there, the block loses its last liberty
- The block is then **captured**: removed from the board

Illegal Move: Suicide

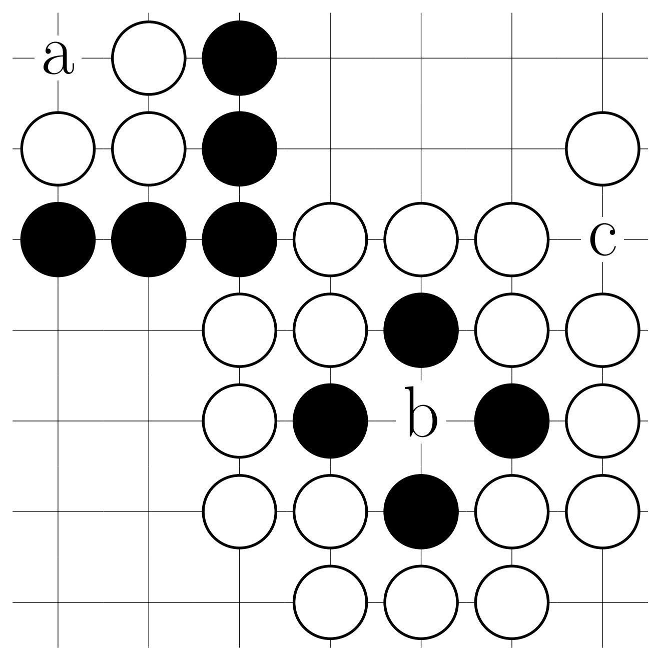




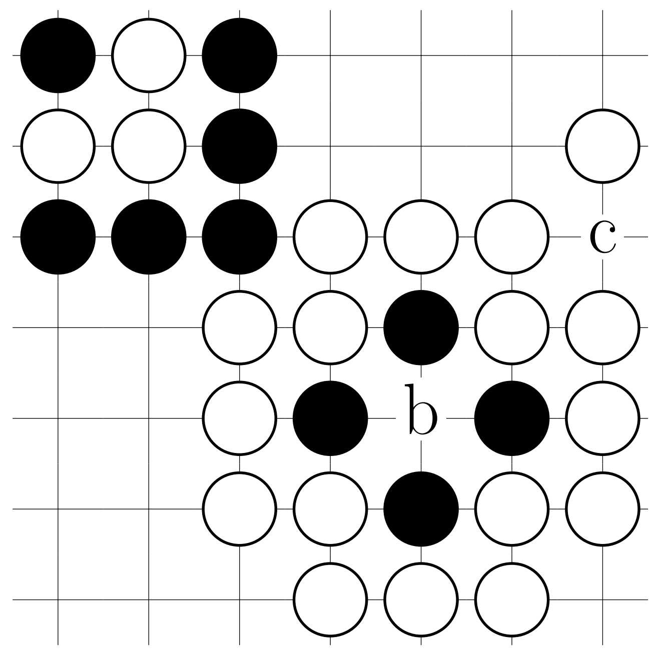


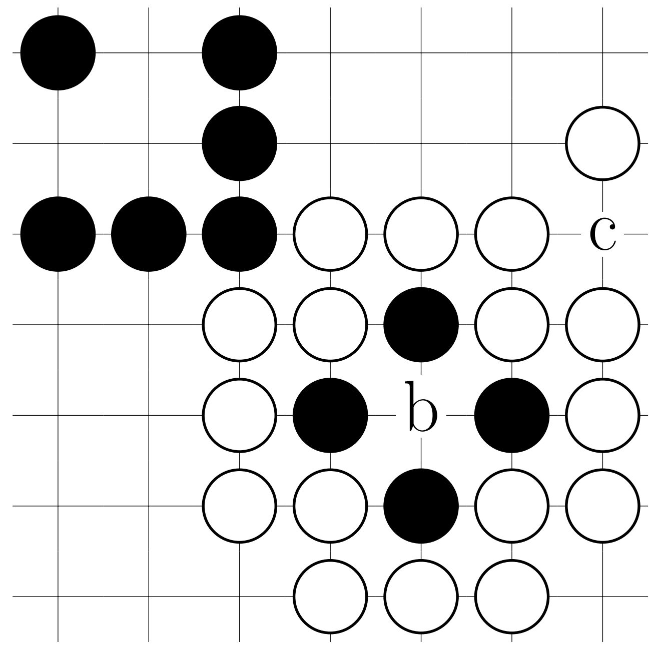
- Suppose it's White's turn to play
- Playing at a would be **suicide**
  - the new group would immediately have no liberties
- Suicide is **forbidden** in most Go variants
  - It is always forbidden in this course
- However, liberties of the new group are counted **after** any captures (see next slide)

Capture vs. Suicide

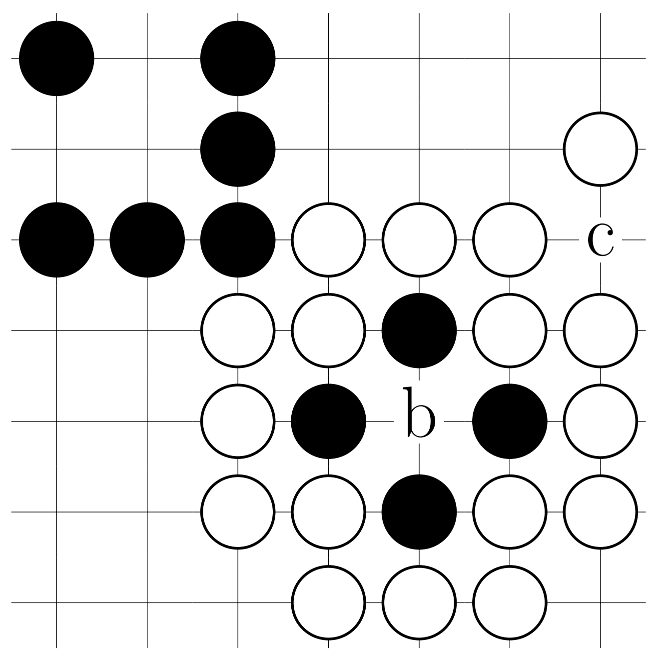


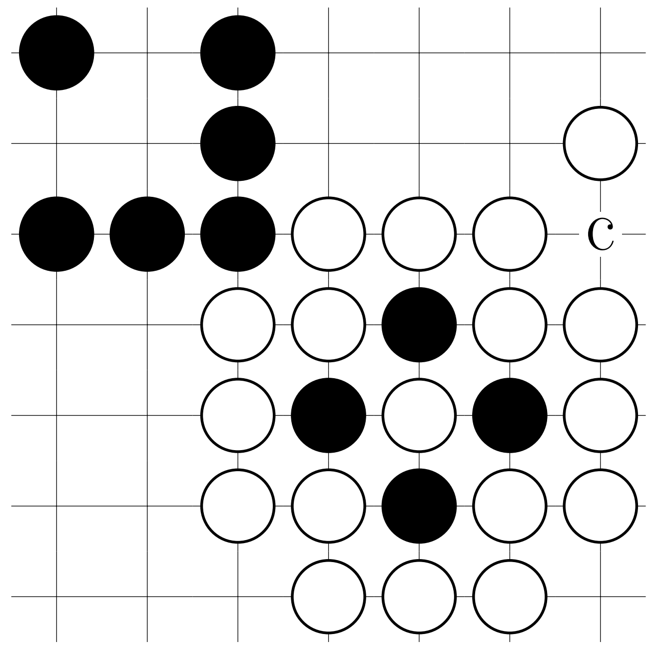
- Move a for Black looks like **suicide**
- However, move a **also** takes the last liberty of the White group; so it is also a **capture**
- Captures are processed **first**
- After the capture, the new stone has two liberties
- So this move is **legal**

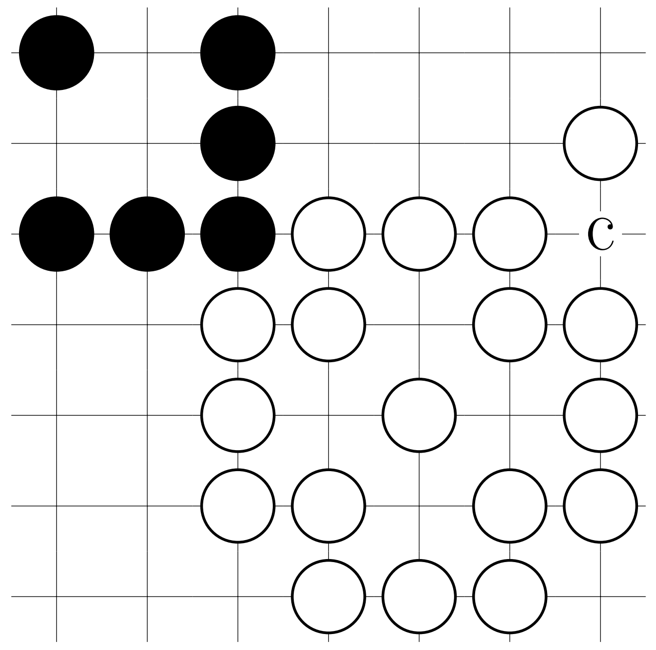




Capture vs. Suicide Examples



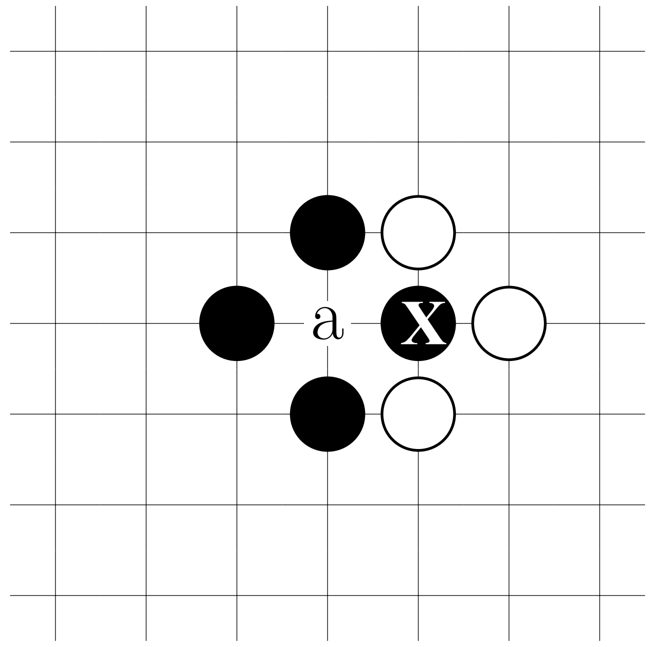


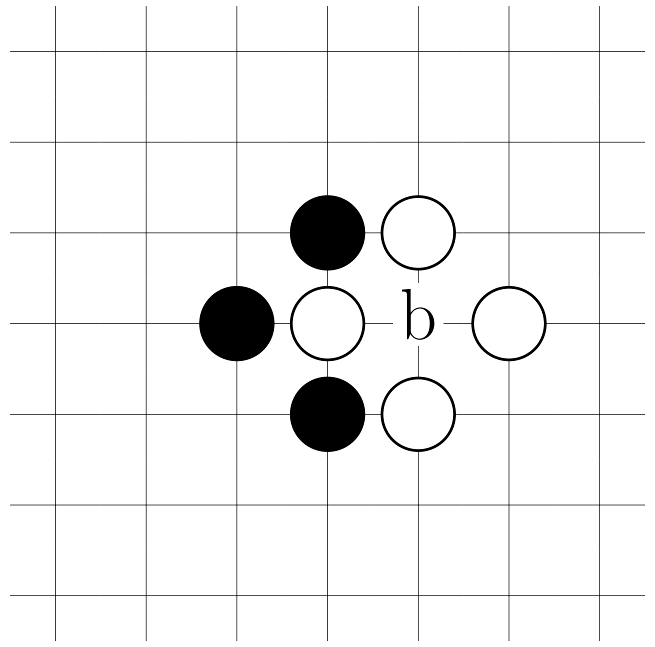


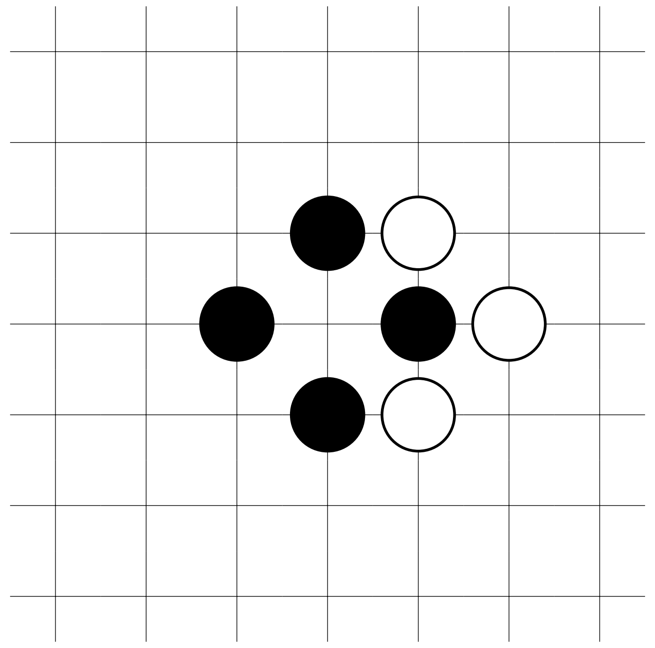
- **Question:** Is b a legal move for White? Why or why not?
- **Question:** Is c a legal move for Black? Why or why not?

Ko (Repetition)

- If White plays on a, it captures black stone $\mathbf{X}$
- If Black captures the new stone by playing on b, then we are back where we started!
- This repetition is called **Ko**
- Ko is **forbidden**
    - **Basic Ko:** A move that would return to **the** previous position is not a legal move
    - **Superko**: a move that would return to **any** previous position is not a legal move

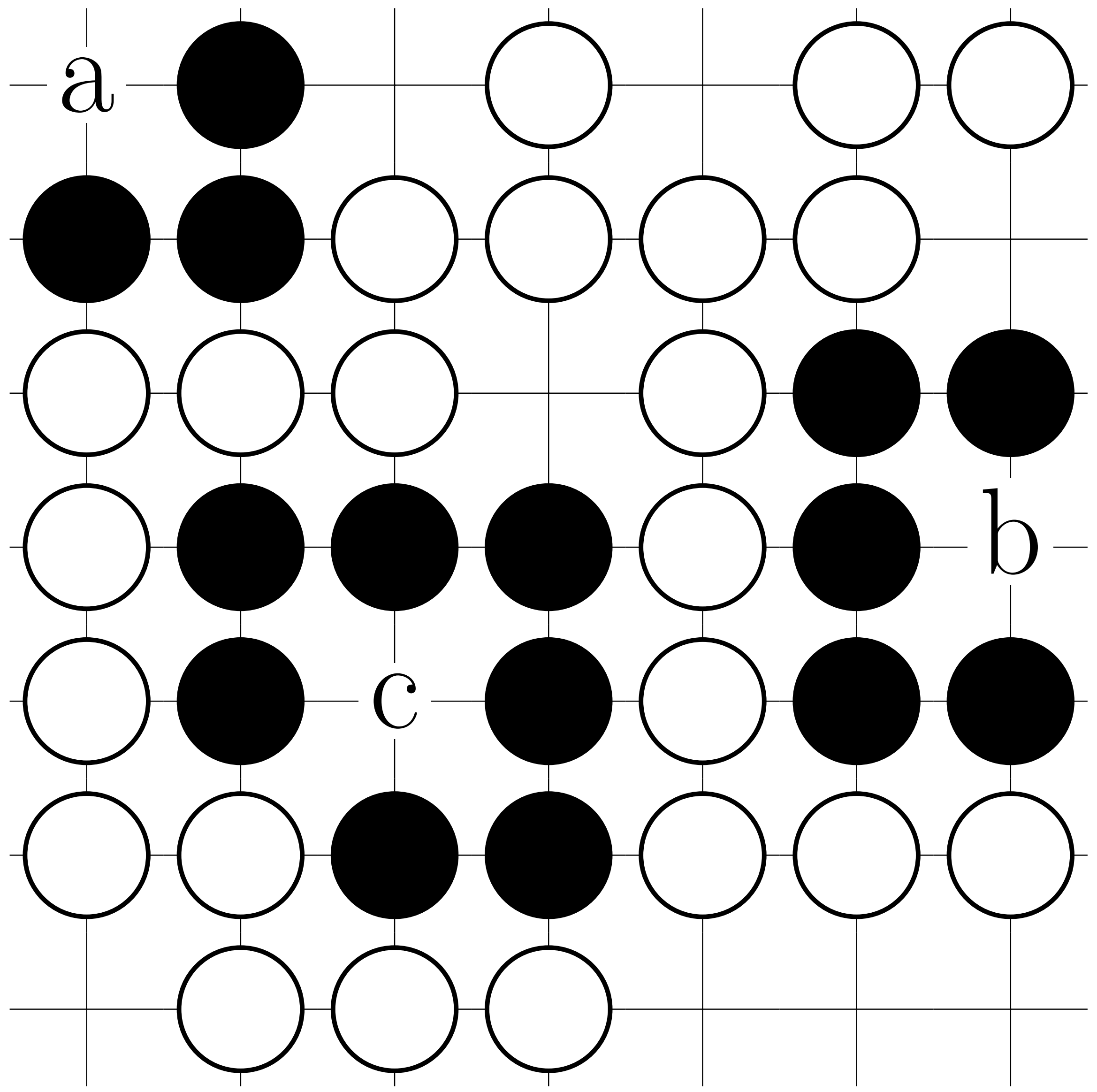






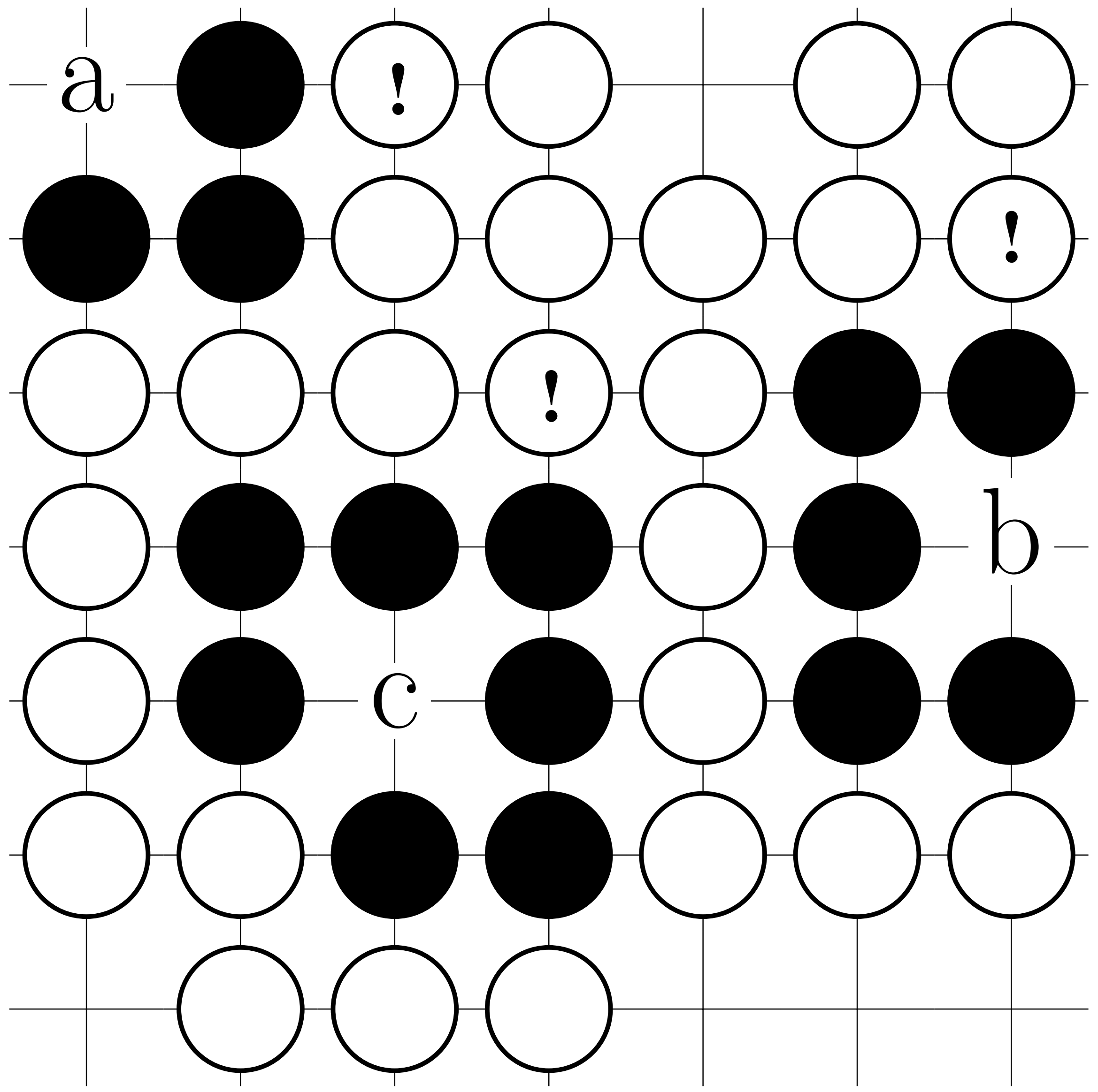
Simple Eyes

- An empty point that is surrounded by stones of a single colour is a (simple) **eye**
- a, b and c to the right are all eyes
- Illegal for White to play in them (suicide)
- Eyes are useful because they make a block more **safe**



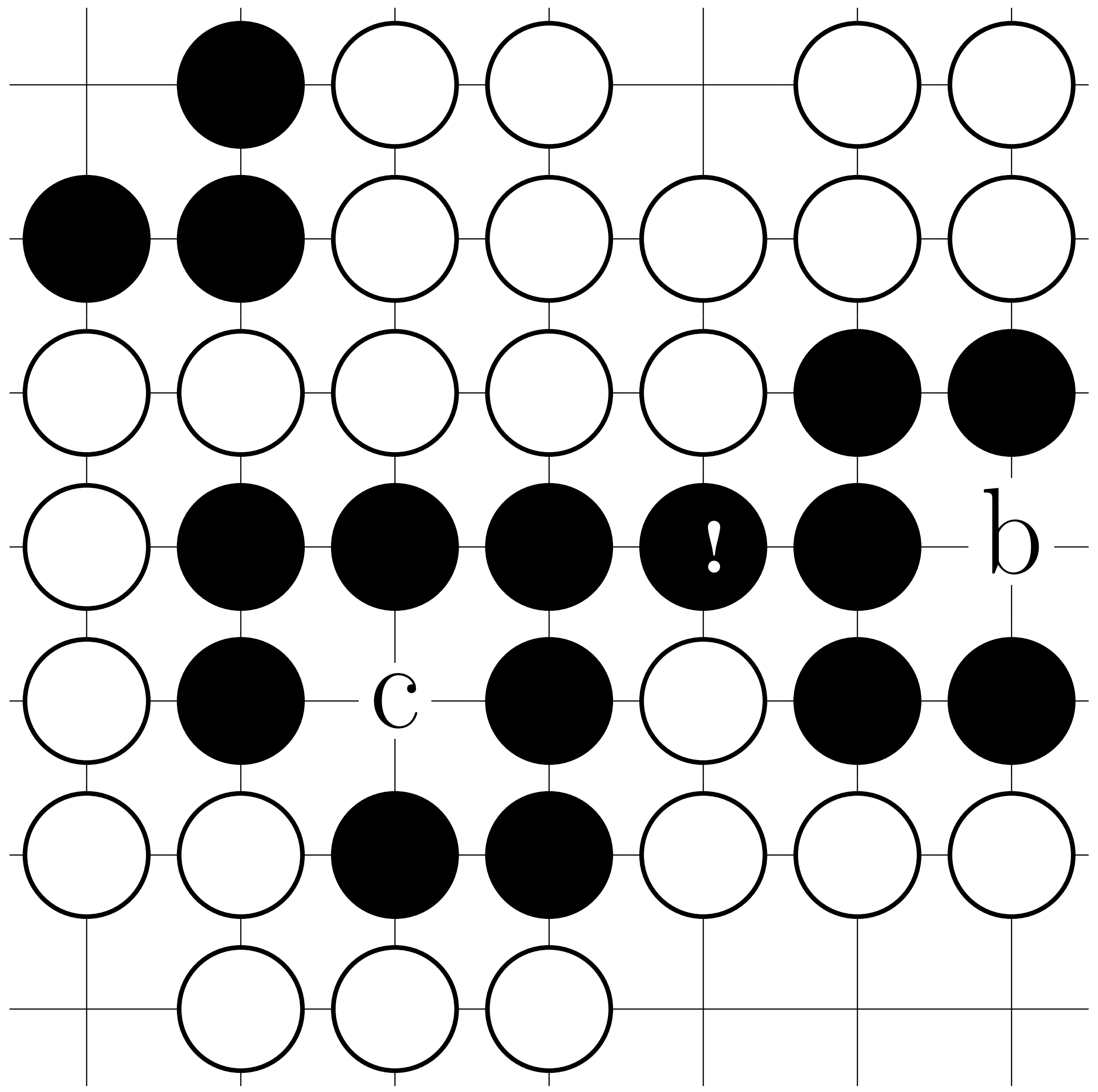
One Eye is Not Enough

- A single eye is not enough to guarantee that a block is safe
- Opponent can still play in an eye if it would cause a **capture**
- Slightly modified example to the right:
  White can legally play in any of $\mathbf{a}$, $\mathbf{b}$, or $\mathbf{c}$
  - The surrounding group is then captured



Groups with Two Eyes Are Safe

- If a single group has **two eyes**, then the group can **never be captured**
- In (again, modified) example at right:
Both b and c are suicide for White
- One eye always remains after filling in the other, so playing in an eye of a two-eyed block can't cause a capture



Legal Moves

At the beginning of a turn, a player either may:

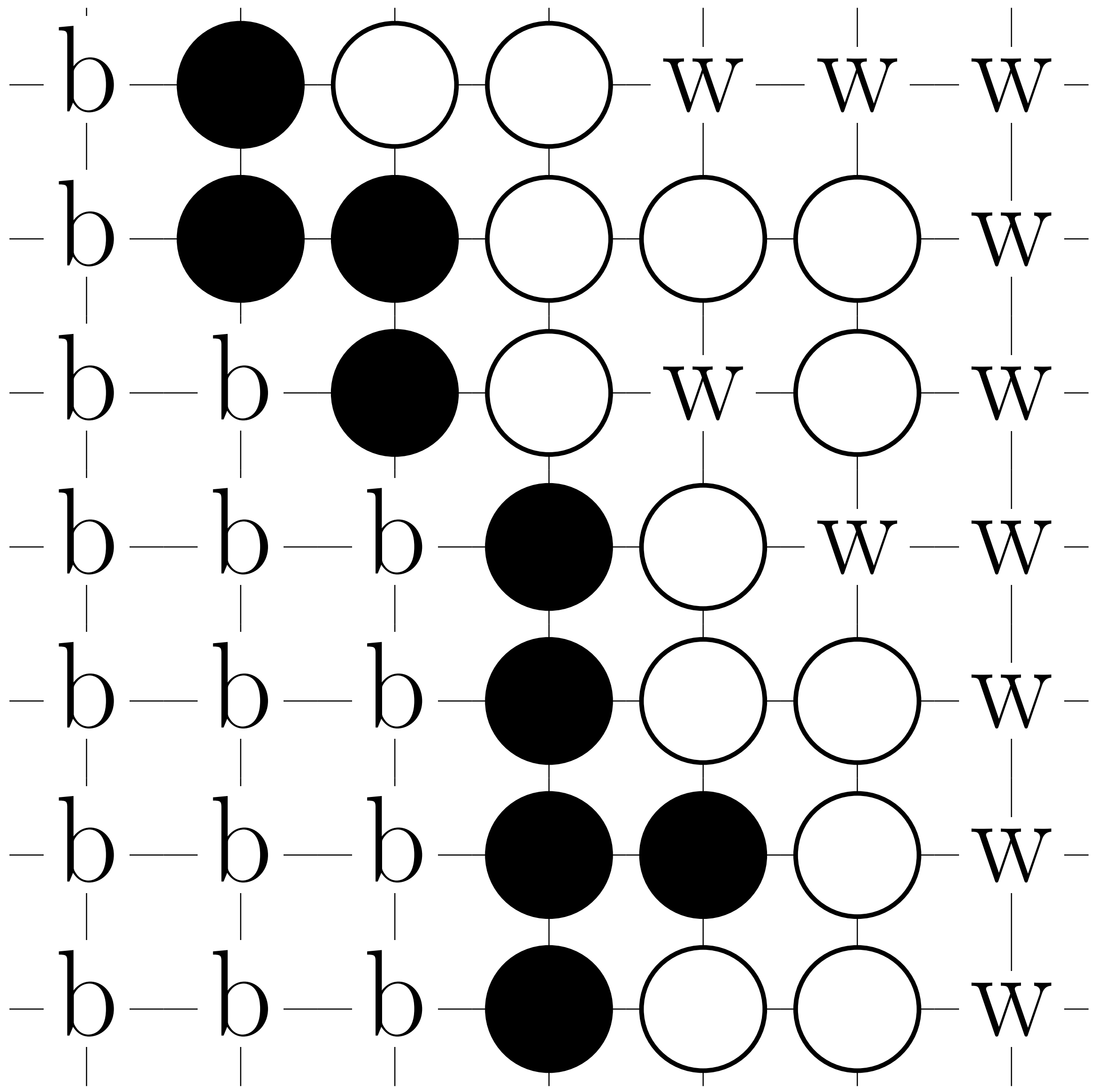
1. **Play** on an empty intersection, except:
    - points forbidden by Ko
    - points forbidden by suicide
2. **Pass**
    - No change to the board, opponent's turn to play
    - Generally there are better moves
    - Except when it's clear that the game is over
      (i.e., you expect opponent to pass also)

End of Game and Scoring

- The game ends with two consecutive **Pass** moves
- An empty point is Black **territory** if it is **surrounded** by Black stones
  - Likewise for White
- Each player gets:
  - 1 point for each intersection of **territory**
  - 1 point for each of their **own stones** on the board
  - 1 point for each of the opponent stones that were **captured**
  - White player gets a bonus called **komi**
    - often 4.5 or 7.5 points; the half-point guarantees **no ties**
- The player with the most points wins

Scoring Example

- Suppose komi of 4.5 and no stones have been captured
- Black has 9 **stones** on the board,
  16 points of **territory**,
  total points = 25
- White has 13 **stones** on the board,
  11 points of **territory**,
  4.5 **komi**,
  total points = 28.5
- White wins by 3.5 points
- **Question:** is it possible for an empty intersection to be territory of **neither player**?



Demo: Clone example code & try **go_helper.py**

1. Clone the repository:
   git clone https://github.com/jrwright/games-puzzles-algorithms.git
2. Run go_helper.py:
   cd games-puzzles-algorithms/go
   python3 go_helper.py
3. Try placing stones to demonstrate different rules:
   - capture
   - suicide detection
   - basic ko
   - superko
   - eyes