DM842

Computer Game Programming: AI

Lecture 9

Tactical Analysis Board Games

Christian Kudahl

Department of Mathematics & Computer Science
University of Southern Denmark

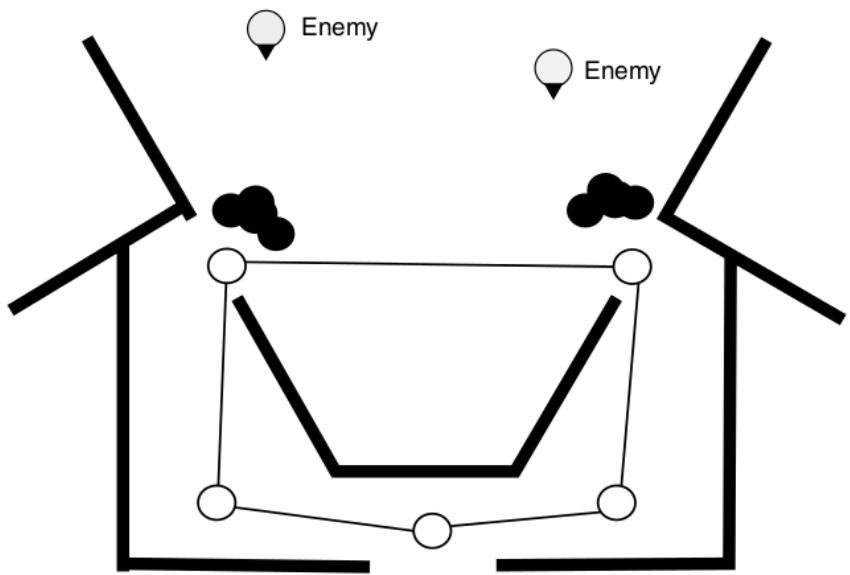
Outline

Outline

Outline

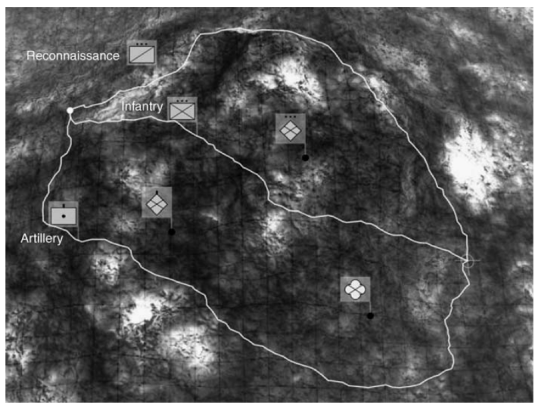
Tactical Pathfinding

- when characters move, taking account of their tactical surroundings, staying in cover, and avoiding enemy lines of fire and common ambush points
- same pathfinding algorithms as saw are used on the same kind of graph representation.
- cost function is extended to include tactical information $C = D + \sum_i w_i T_i$ where D is distance w_i is the weight for tactic i and T_i tactical quality for tactic i.
- problem: tactical information is stored on nodes
  $\rightsquigarrow$ average the tactical quality of each of the locations incident to the arc



Tactic Weights and Concern Blending

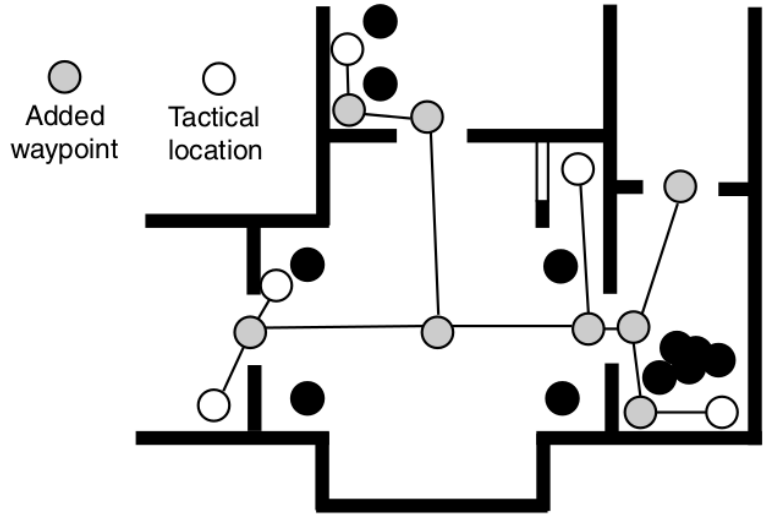
- If a tactic has a high positive weight, then locations with that tactical property will be avoided by the character. Conversely if large negative weight, they will be favoured.
- but! negative costs are not supported by normal pathfinding algorithms such as A*
- Performing the cost calculations when they are needed slows down pathfinding significantly.
  Is the advantage of better tactical routes for the characters outweighed by the extra time they need to plan the route in the first place?
- Weights have an impact on character personalization
  Tanks will likely have different weights than infantry.



- Issue: heuristic may become invalid (not admissible)
  Eg: Euclidean distance not anymore valid; maybe multiplying it by a factor?

- graph: nodes derived from influence maps for outdoor and tactical waypoints for indoor
  connections: generate them by running movement checks or line-of-sight checks between map locations or waypoints
  if grid based: adjacent locations are connected + gradient threshold considerations

- necessary to add additional waypoints at locations that do not have peculiar tactical properties. Superimpose the tactical waypoints onto a regular pathfinding graph.



Outline

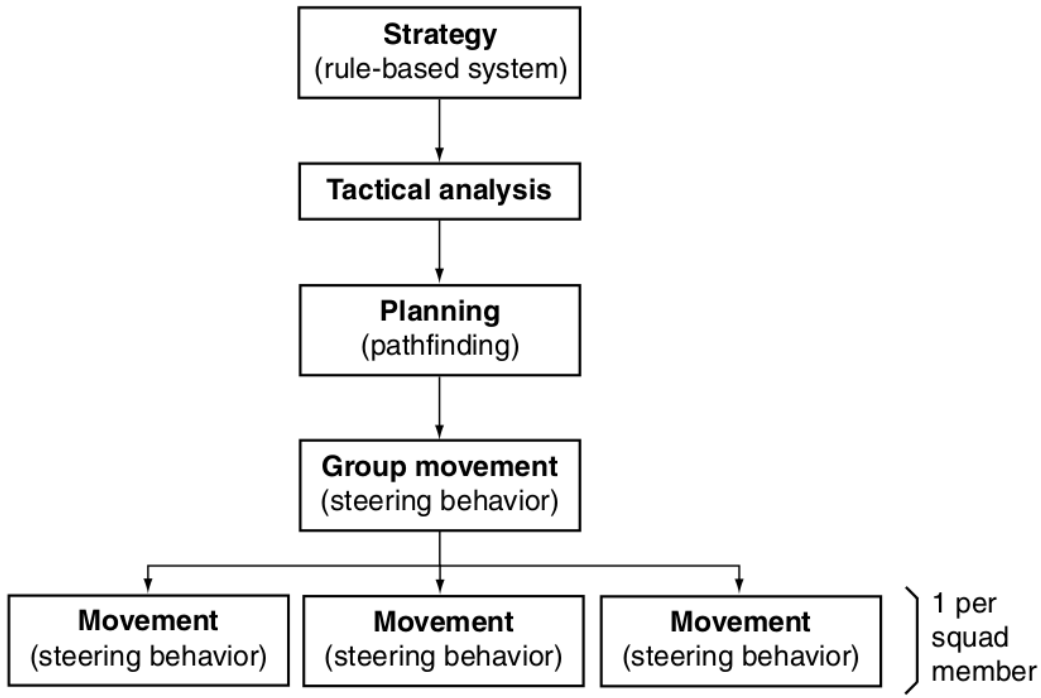
Coordinated Action

- a whole side in a real-time strategy game
- squads or pairs of individuals in a shooter game.
- AI characters acting in a squad led by the player: cooperation occurs without any explicit orders being given.
  Characters need to detect the player's intent and act to support it.

AI characters need to tell each other what they are planning through some kind of messaging system,

Multi-Tier AI

top-down vs bottom-up (emergency) approach
Hierarchy of behaviors: at some levels decisions shared among multiple characters

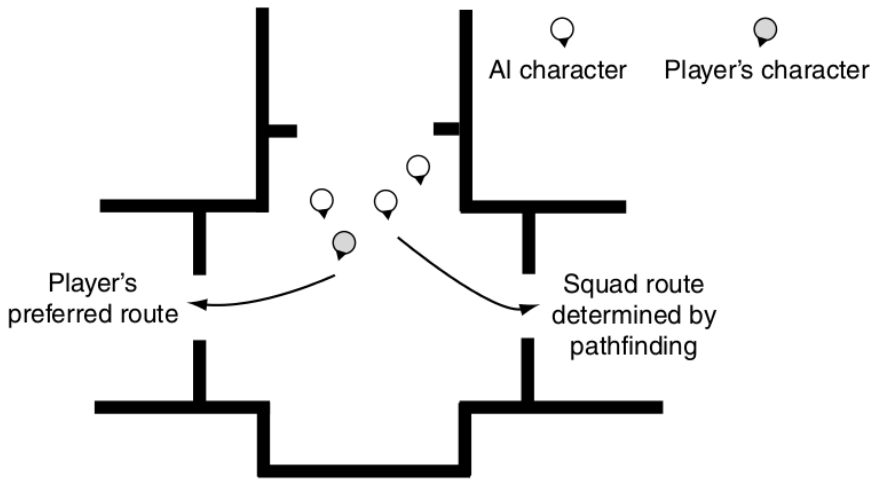


- Group Decisions
  standard expert systems or state machines, tactical analysis or waypoint tactic algorithms
  difference: actions are not scheduled for execution by the character, but are orders passed down

- Group Pathfinding
  different blend of tactical concerns for pathfinding that any individual would have alone
  heuristic of weakest character
  decision tree, expert system, or other decision making technology to detect constraints and separate the group

- Group Movement
  steering behaviors (feasibility issues)

- Including the Player
  high-level decision making may make a decision that the player subverts



  The game design may involve giving the player orders, but ultimately it is the player who is responsible for determining how to carry them out.

  Solution: put the Player at the top level of the hierarchy and conflicts avoided.

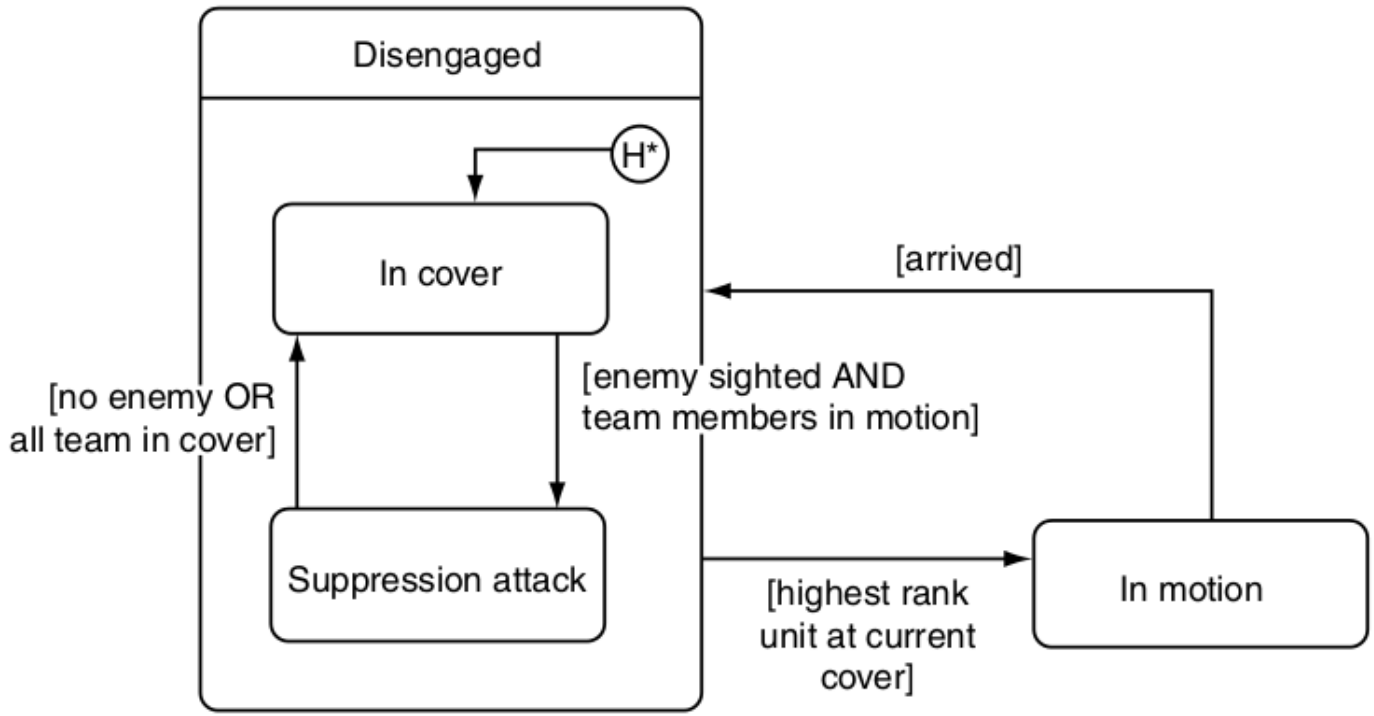
- Explicit Player Orders, eg. in RTS games

Emergent Cooperation

- Less centralized techniques to make a number of characters appear to be working together.
- Each character has its own decision making

$\rightsquigarrow$ the decision making takes into account what other characters are doing

- If any member of the team is removed, the rest of the team will still behave relatively efficiently
- most squad based games


Disengaged
H*
In cover
Suppression attack
[no enemy OR
all team in cover]
[enemy sighted AND
team members in motion]
[arrived]
[highest rank
unit at current
cover]
In motion

Behaviour of Emergent Systems

- Scalability: Some behavior scales better than other (flocking vs previous slide). Generally stable behavior scales better.
- Predictability: Often hard to predict how emergent systems will evolve. May end up in unintelligent behaviors.
- Consider combination of multi-tiered AI and emergent behavior, for example with a number of independent squads, each with their own multi-tiered ai.

Resume

1. Movement
2. Pathfinding
3. Decision making
4. Tactical and strategic AI
5. Board game AI

Outline

Board game AI

- different techniques from the ones seen so far
  tree-search algorithms defined on a special tree representation of the game.

- Limited applicability for real-time games, a strategic layer only occasionally used. Eg. making long-term decisions in war games.

Game Theory

- Game theory is a mathematical discipline concerned with the study of abstracted, idealized games.
  (Only a few definitions from this theory are needed in games.)

- classification of games according to:
  - number of players
  - kinds of goal
  - information each player has about the game.

Number of players

- Many traditional board games have two players.
- ply one player's turn (aka half-move with 2 players)
- move One round of all the players' turns (aka turn)

Goal

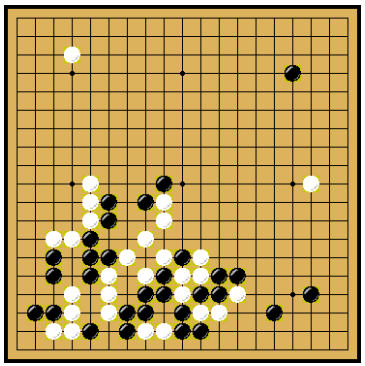
- zero-sum game: your win is the opponent's loss $(1; -1)$
  trying to win $\equiv$ trying to make your opponent loose.
- non-zero-sum game: you could all win or all lose
  focus on your own winning, rather than your opponent losing
- with more than two players and zero-sum games, best strategy may not be making every opponent loose.

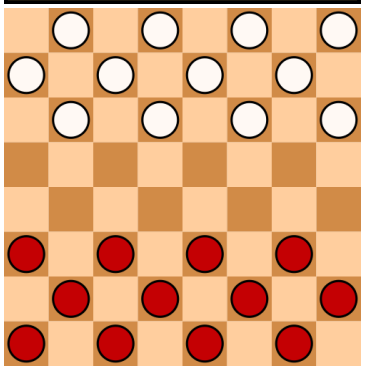
Information

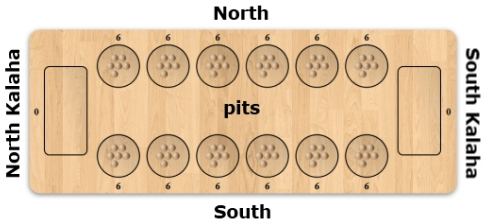
- perfect information fully observable environment
  complete knowledge of every move your opponent could possibly make
- imperfect information partially observable environment
  non-complete knowledge of the game state, different information available to the players.

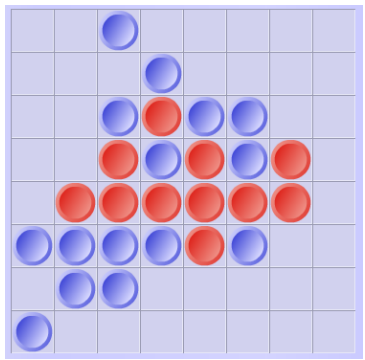
Types of Games

	deterministic	chance
perfect information	chess, hex, checkers, kalaha go, othello	backgammon, monopoly
imperfect information	battleships, stratego	bridge, poker, scrabble



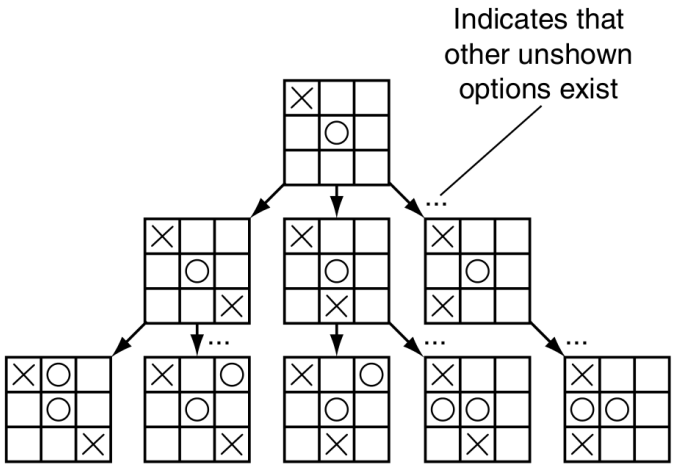


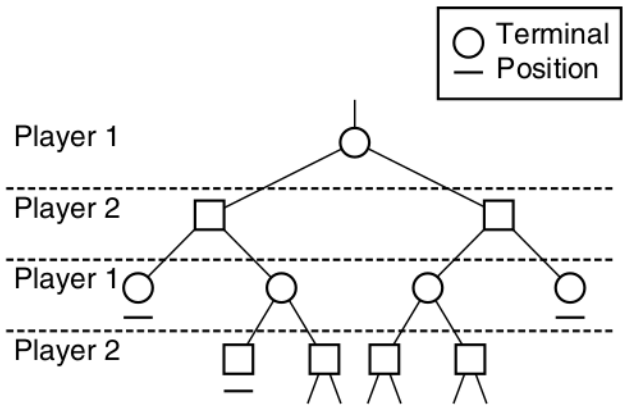




Game Tree

For turn-based games: each node in the tree represents a board position, and each branch represents one possible move.





terminal positions: no possible move, represent end of the game. Score given to players

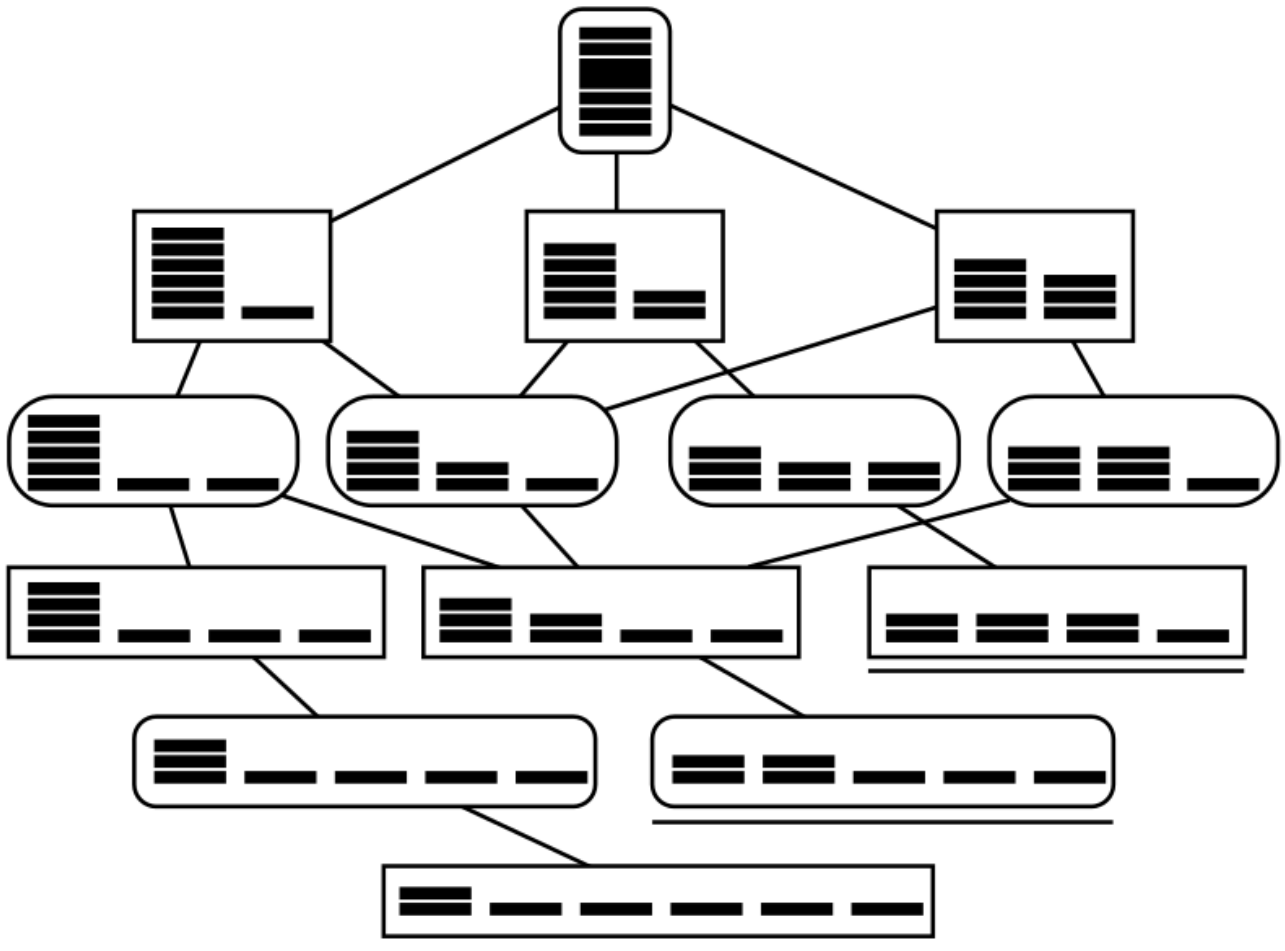
branching factor: number of branches at each branching point in the tree

tree depth: finite or infinite

transposition same board position from different sequences of moves $\rightsquigarrow$ cycles

Example

7-Split Nim: split one pile of coins into two non-equal piles. The last player to be able to make a move wins



Measures of Game Complexity

- state-space complexity: number of legal game positions reachable from the initial position of the game.

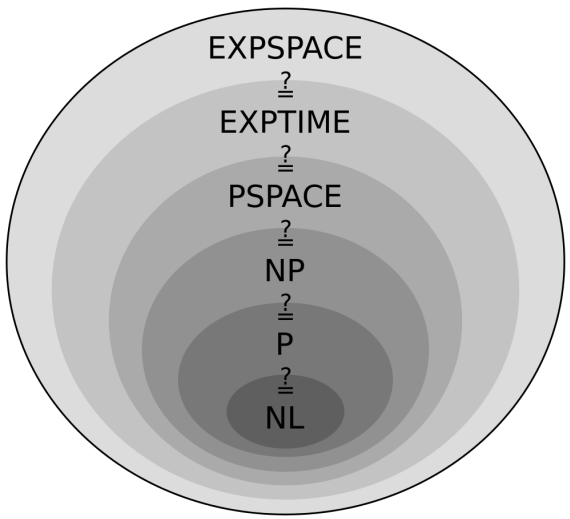
  an upper bound can often be computed by including illegal positions
  Eg, TicTacToe:
  $3^9 = 19.683$
  5.478 after removal of illegal
  765 essentially different positions after eliminating symmetries

- game tree size: total number of possible games that can be played: number of leaf nodes in the game tree rooted at the game's initial position.

  Eg: TicTacToe:
  $9! = 362.880$ possible games
  255.168 possible games halting when one side wins
  26.830 after removal of rotations and reflections

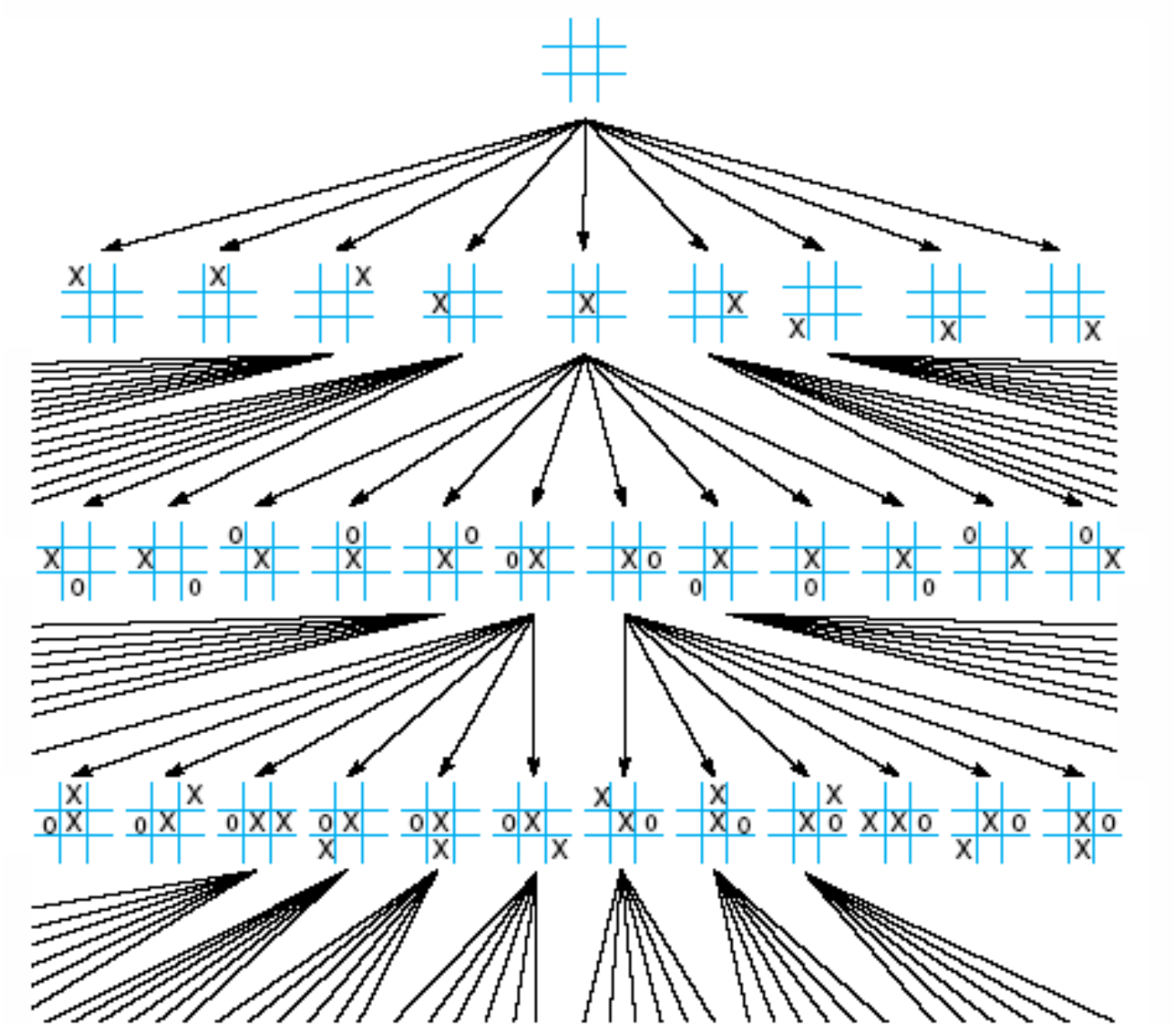
Measures of Game Complexity

- Computational complexity of the generalized game (eg, played on a $n \times n$ board): often PSPACE-complete (set of all decision problems that can be solved by a Turing machine using a polynomial amount of space and for which every other problem that can be solved in polynomial space can be transformed to one of these problems in polynomial time.)



Eg: Quantified Boolean formulas

$$\exists x_1, \forall x_2 \exists x_3 : (x_1 \vee x_2) \wedge (x_2 \vee x_3)$$



First three levels of the tic-tac-toe state space reduced by symmetry: $12 \times 7!$