Dr. Denton Bobeldyk

CIS 365 Artificial Intelligence

Uninformed Search

Week in Review

Blackboard Check-in

False Coin Problem

- You have a set of n coins, one of which is counterfeit. The counterfeit coin differs in weight from the genuine coins (it is either lighter or heavier). Using a balance scale, determine which coin is counterfeit and whether it is lighter or heavier with the minimum number of weighings

False Coin Problem AI Concepts

1. Search and Optimization
2. Decision Trees
3. Information Theory
4. Logical Reasoning and Constraint Satisfaction
5. Algorithm Design and Analysis

False Coin Problem AI Concepts

1. Search and Optimization
    1. AI algorithms often need to search through vast solution spaces
2. Decision Trees
    1. Each weighing represents a decision point
3. Information Theory
    1. Each weighing provides information that narrows down he possibilities
4. Logical Reasoning and Constraint Satisfaction
    1. Use logical deductions to eliminate possibilities based on weighing outcomes
5. Algorithm Design and Analysis
    1. Develop and analyze algorithms that can solve the False Coin Problem optimally

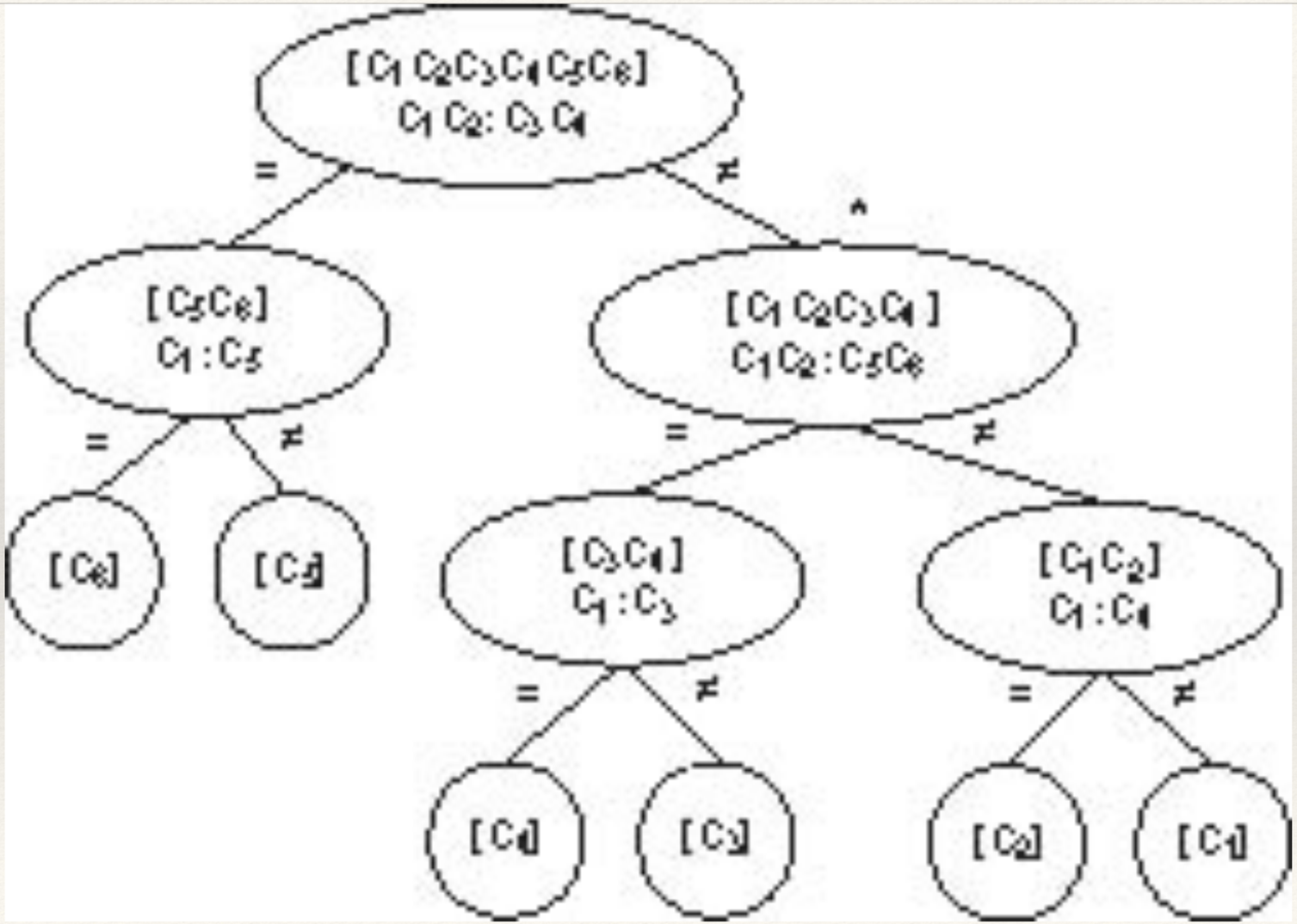
False Coin Problem Group Work

- You have a set of n coins, one of which is counterfeit. The counterfeit coin differs in weight from the genuine coins (it is either lighter or heavier). Using a balance scale, determine which coin is counterfeit and whether it is lighter or heavier with the minimum number of weighings

- Given 6 coins, determine which is the false coin?

Solution Next Slide

False Coin Problem Solution



State Space

- A state is a specific configuration or condition of the system/probelm at a given point in time.

State Space

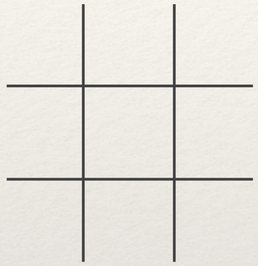
- Initial state - starting point or initial condition of the problem.
- Goal state - the desired final configuration or outcome
- State space - set of all possible states that can be reached by applying a series of actions/transitions

State Space (additional terms)

- **Operators (actions):** the moves/transitions that can be applied to a state to transform it into another state.
  - Example: A sliding puzzle an operator might “move the empty space up”
- **Path** - sequence of states connected by operators starting from the initial state and leading towards the goal state. Finding an optimal or valid path is often the primary objective in solving state space problems.

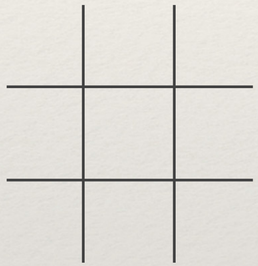
State Space Example

Initial State

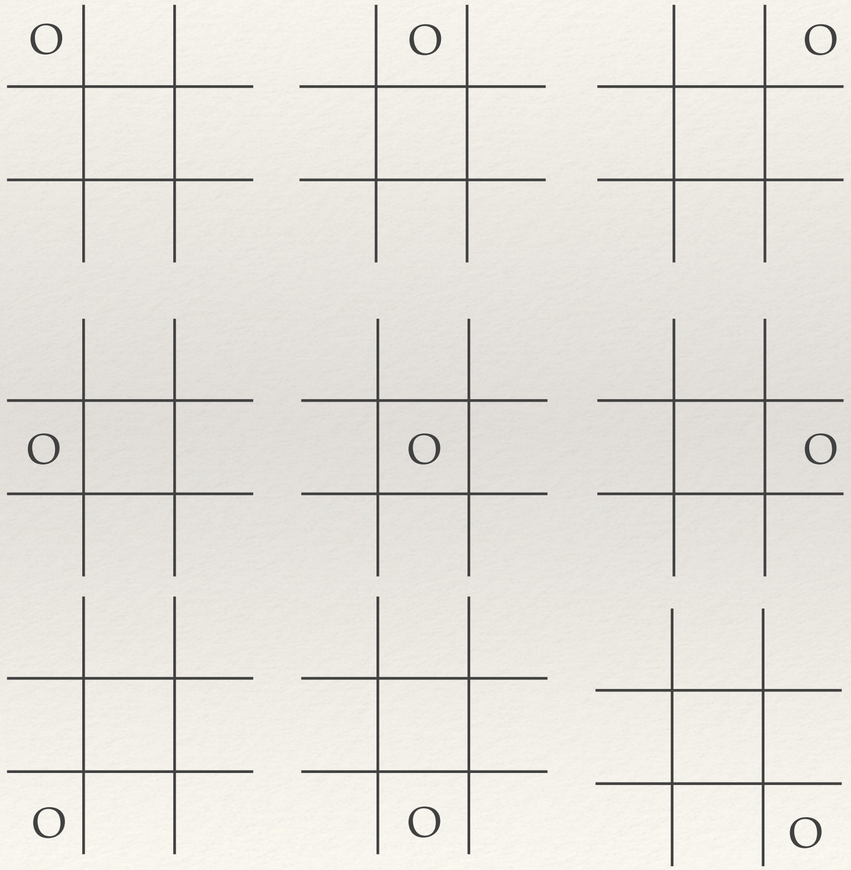


State Space Example

Initial State

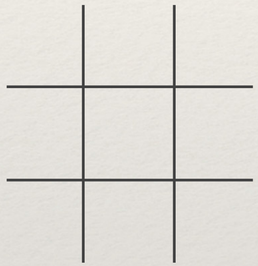


First Move Possible States

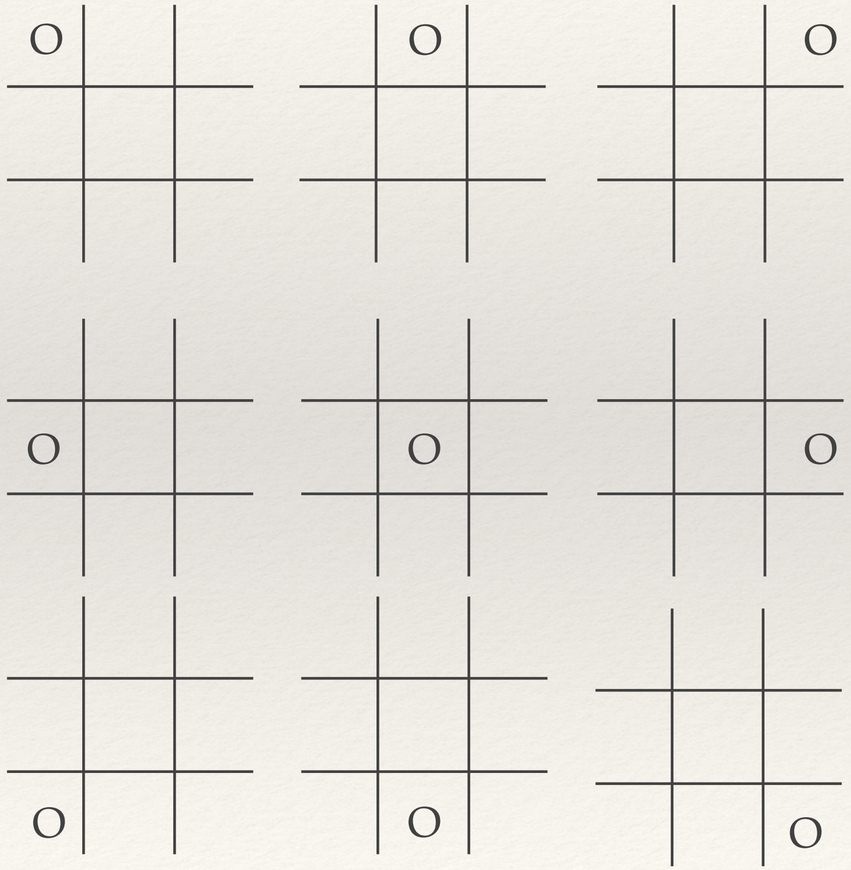


State Space Example

Initial State



First Move Possible States

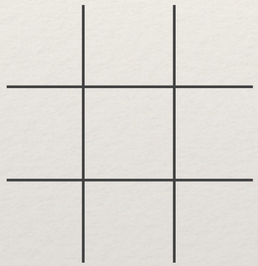


Anything we can do to reduce the possible state size here?

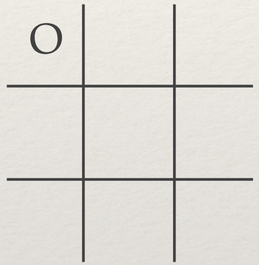
State Space Example

Possible State Space Path

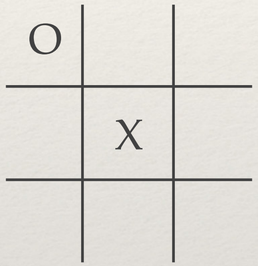
Initial State



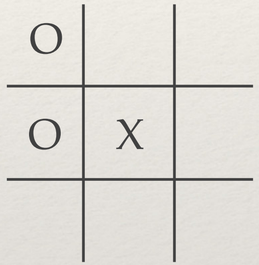
Place O in top left corner



Place X in middle square



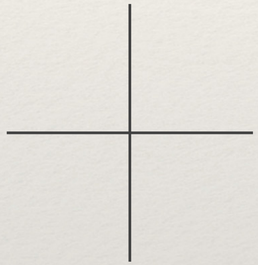
Place O in left middle square



State Space Example

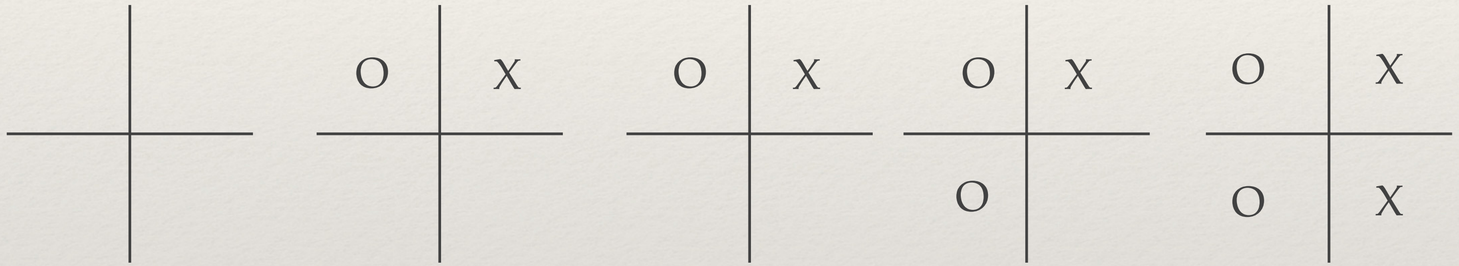
How many possible states for this scaled down (completely stupid and only for demonstration purposes) version of tic-tac-toe

Initial State



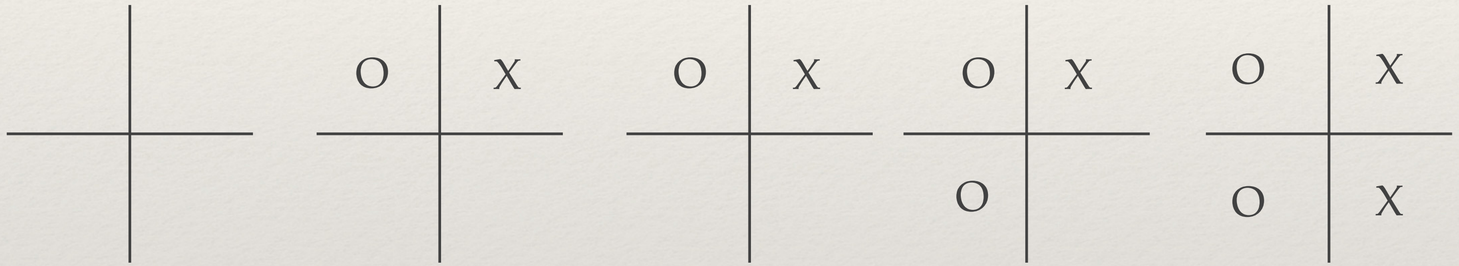
State Space Example

How many possible states for this scaled down (completely stupid and only for demonstration purposes) version of tic-tac-toe



State Space Example

How many possible states for this scaled down (completely stupid and only for demonstration purposes) version of tic-tac-toe



3^4

State Space Tic Tac Toe

How many different possible states for TicTacToe?

Group Class Exercise (3-5 mins)

What's the possible state space for:
Checkers
Chess

Find some other games you have interest in, what about them?

Artificial Intelligence Uninformed Search

N Queens Problem

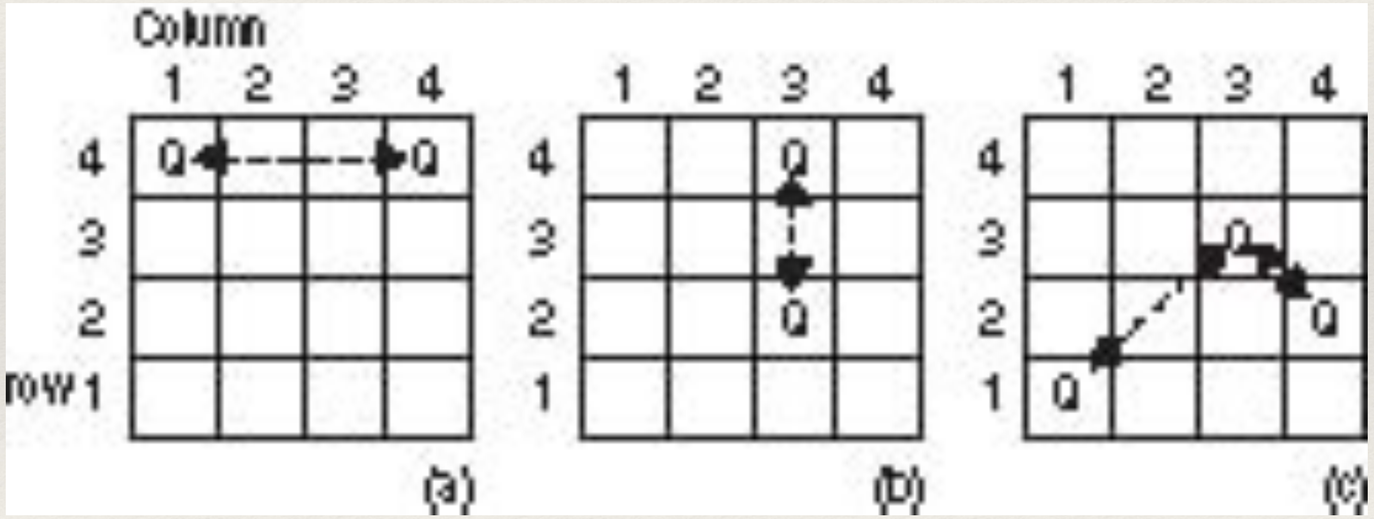
Artificial Intelligence Uninformed Search

N Queens Problem

Classic AI Problem. It involves placing N number of queens on an NxN chess board. Goal is to place all N queens on the board where no queen can attack each other.

Artificial Intelligence Uninformed Search

N Queens Problem



Artificial Intelligence Uninformed Search

Get in a small group and determine some possible algorithms for solving the N queens problem

Searching State Space

Depth First Search
Breadth First Search

Searching State Space

Depth First Search
Dive down into each branch before backtracking and exploring other branches

Searching State Space

Depth First Search

Dive down into each branch before backtracking and exploring other branches

Uses a stack to keep track of the path being explored

Breadth First Search

Examine all possible states at one depth before moving to the next depth.

Uses a queue to keep track of the states to be explored next

Depth First Search

```
Begin
Open? [Start State] // The Open List
//Maintained as a stack. i.e., a list in which the last item inserted is the first item deleted
// This is often referred to as a LIFO list
Closed ? [] // The closed list contains nodes that have already been inspected; it is initially empty

While Open not empty
  Begin
    Remove first item from open, call it X
    If X equals goal then return Success
    Else
     Generate immediate descendants of X
    Put X on Closed List
    If children of X already encountered then discard
  //loop check

    Else place children not yet encountered on Open
    // end else
  // end While
Return Goal Not Found
```

Artificial Intelligence Uninformed Search

Breadth First Search vs. Depth First Search

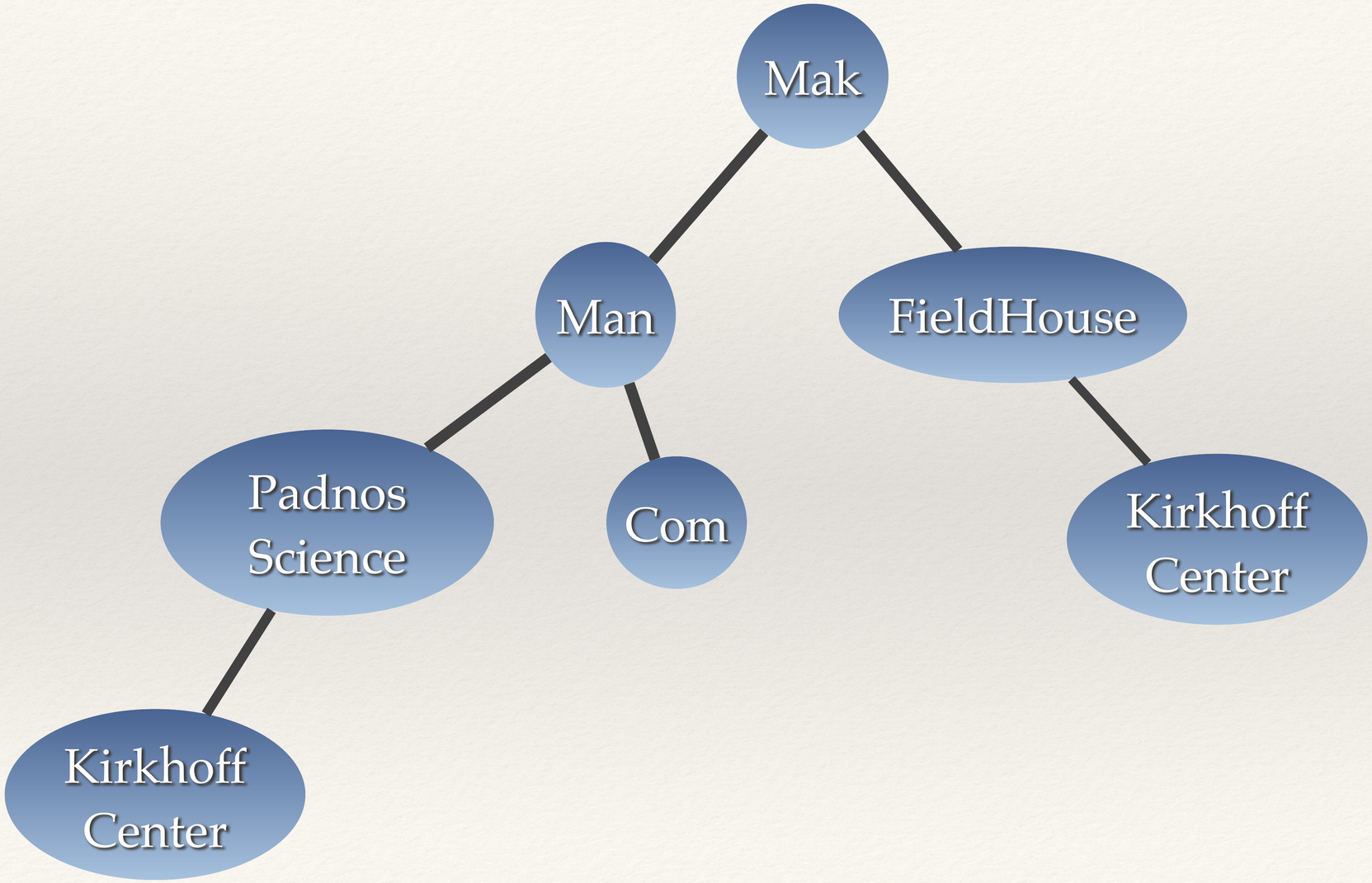
Artificial Intelligence Uninformed Search

Breadth First Search vs. Depth First Search

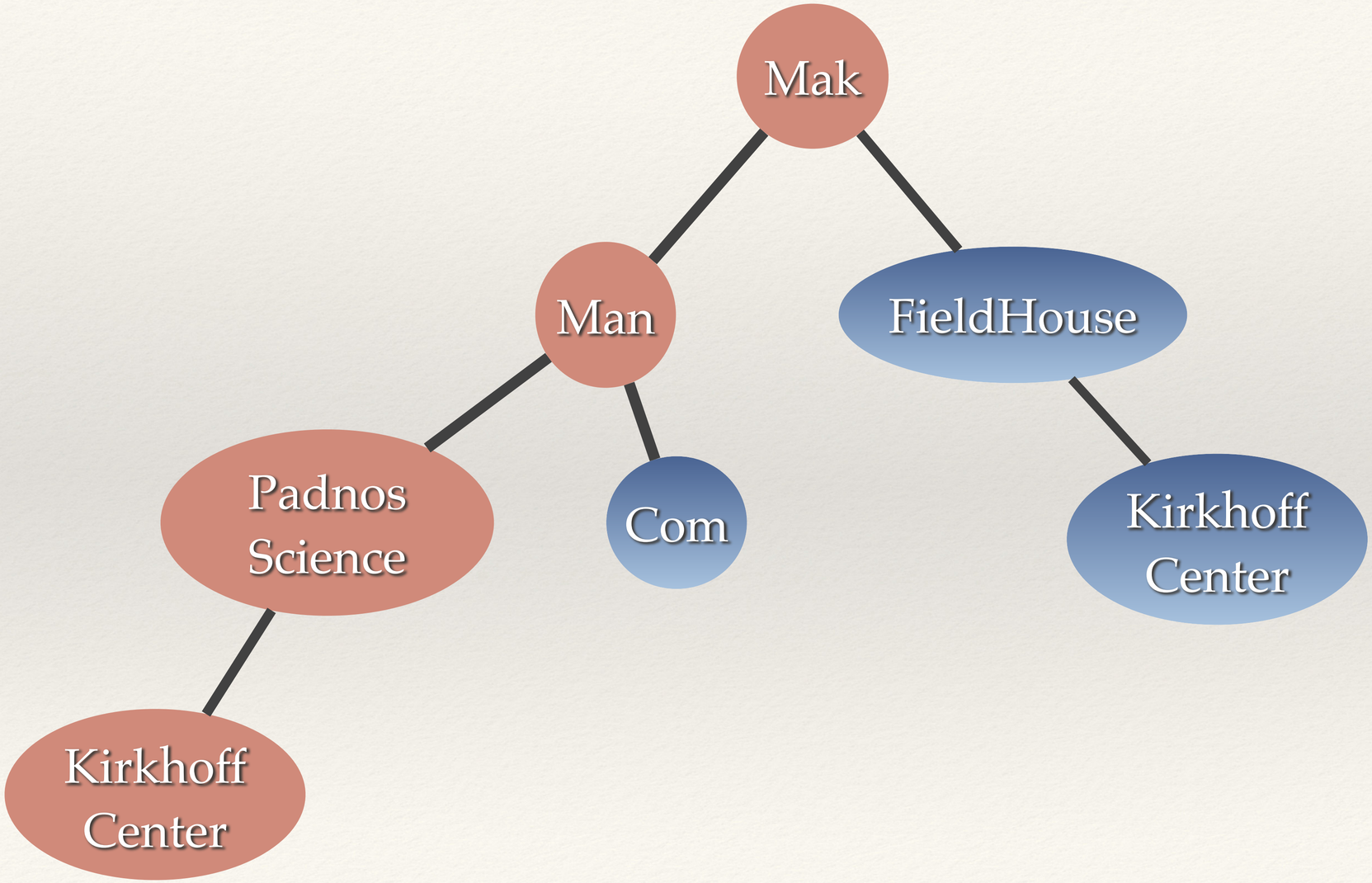
DFS preferred for:
The tree is deep
The branching factor is not excessive
Solutions occur relatively deep in the tree

BFS preferred for:
The branching factor of the search tree is not excessive (reasonable to b)
A solution occurs at a reasonable level in the tree (d is reasonable)
No path is excessively deep

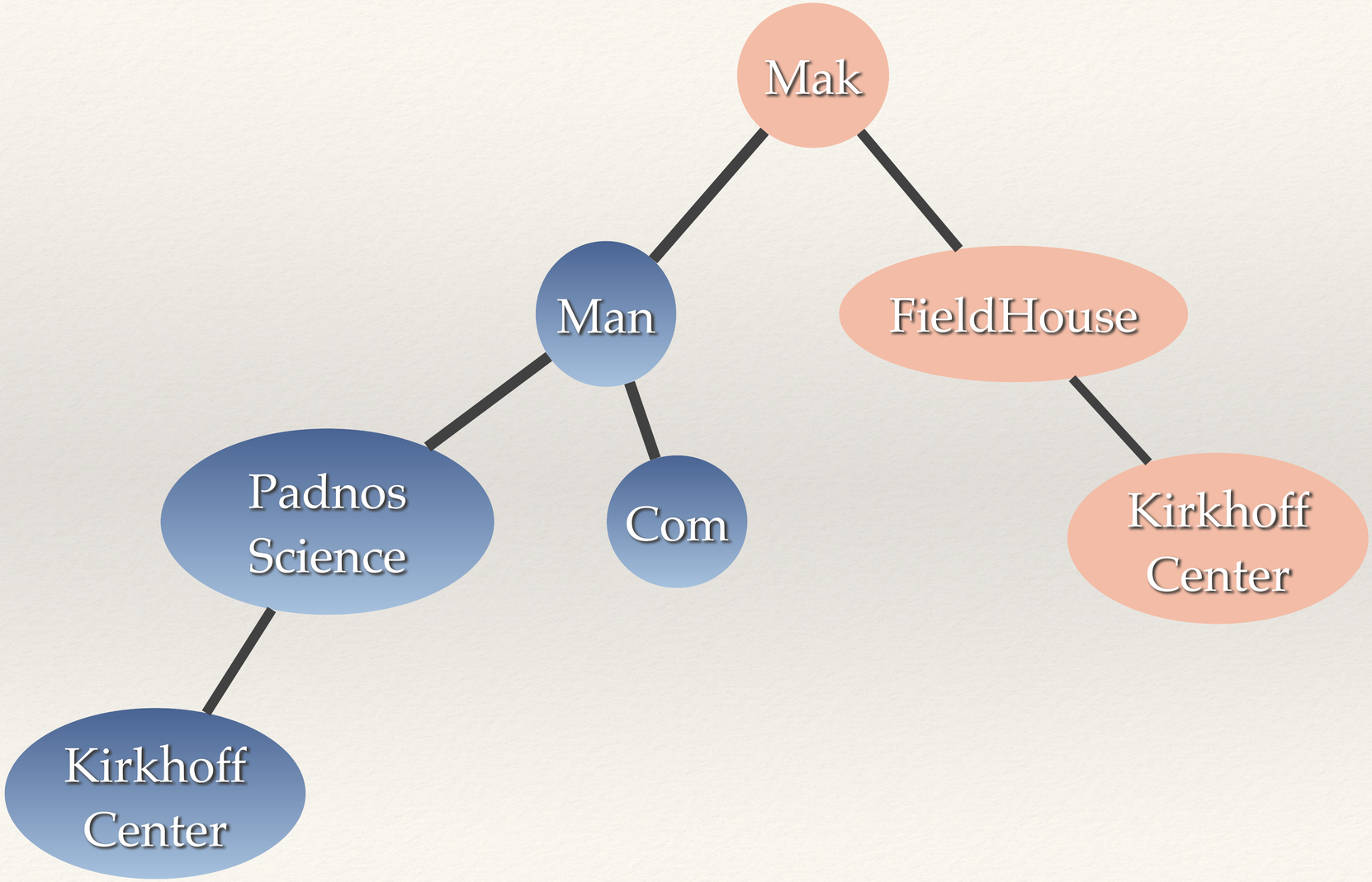
State Space


Mak
Man
FieldHouse
Padnos Science
Com
Kirkhoff Center
Kirkhoff Center

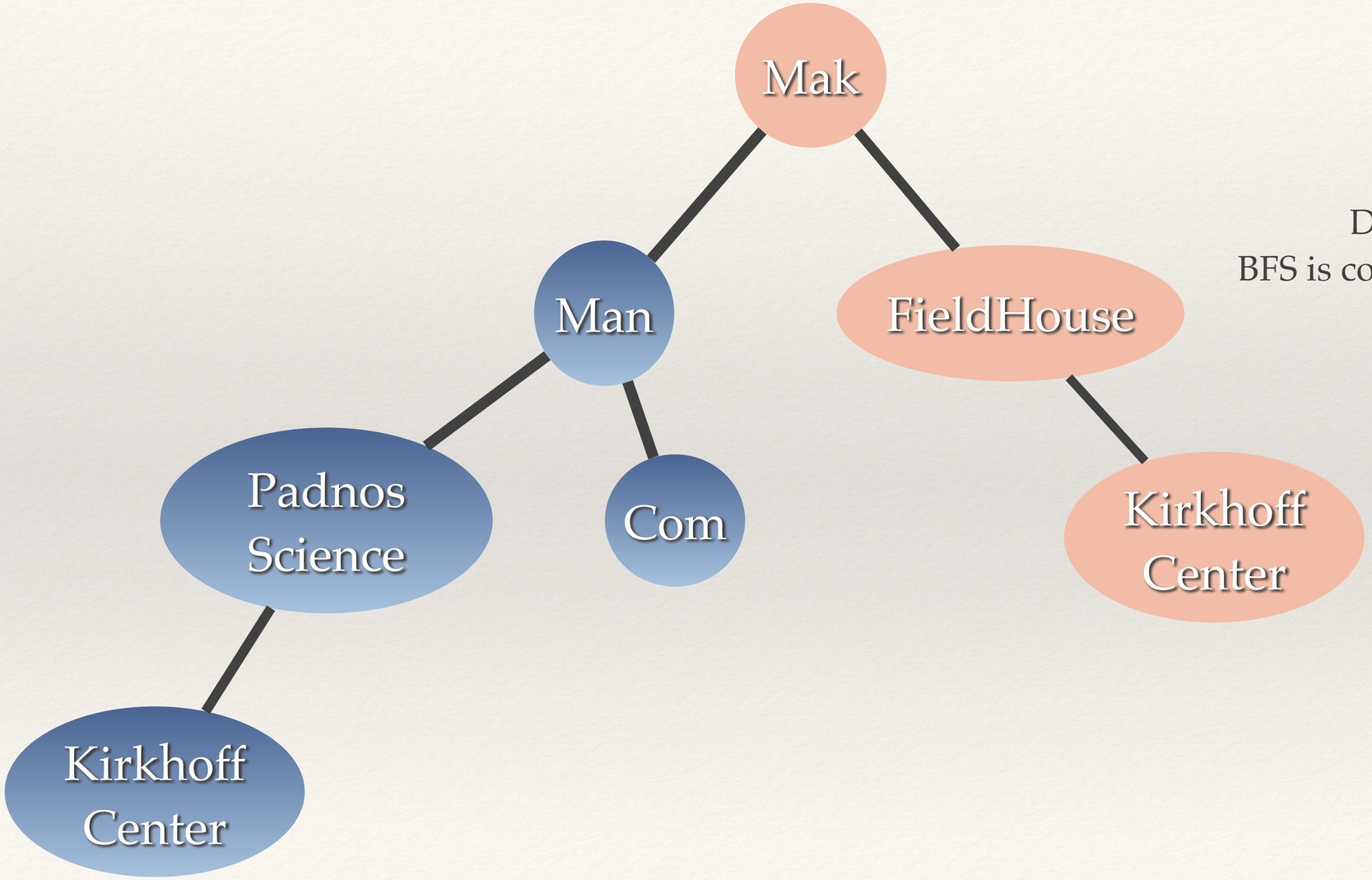
State Space - DFS Solution



State Space - BFS Solution



State Space - BFS Solution



DFS is not complete
BFS is complete (optimal solution)

Artificial Intelligence Uninformed Search

1. Bidirectional Search
    1. Runs two searches. One from the goal state and one from the initial state
2. Depth-Limited Search (DLS)
    1. Like DFS, but limits the depth it goes
    2. Can also iteratively increase the depth limit (Iterative Deepening Depth-First Search)
3. Random Walk
    1. Randomly select an action and step down that path

Donald Knuth - Advice to young people