# Midterm Review

Chen-Yu Wei

# Logistics

- Time:  9:30AM-10:45AM, Oct. 17
- Location:  Olsson Hall 009  (usual classroom)
- Open notes
  - Just paper-based notes/notebooks, no electronic devices
  - No textbook
- Let me know if you need accommodation
  - Please apply through Student Disability Access Center (SDAC)

# Topics

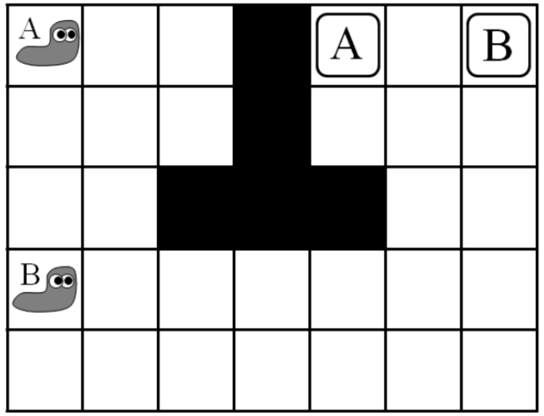
- Search
- Search in Games
- Constraint Satisfaction Problems
- Logic
- Probabilistic Models (up to the content we have covered)

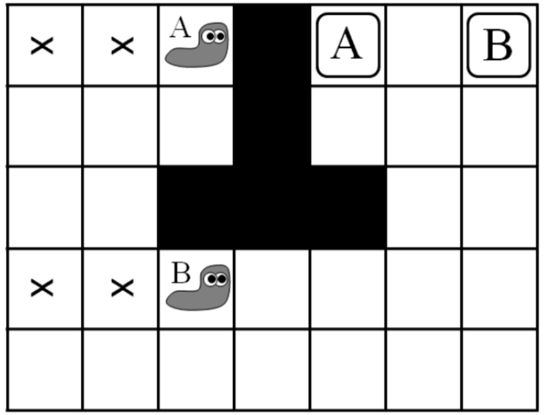
# Search

# Learning Objective 1

- For a given statement, formulate it as a search problem with well-defined
  - States
  - Actions
  - Successor function
  - Goal test

# Learning Objective 1 – Example



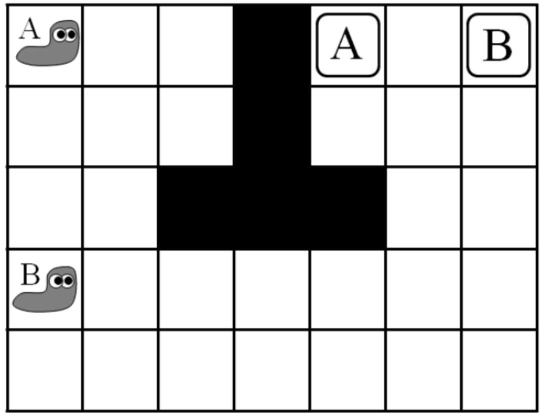



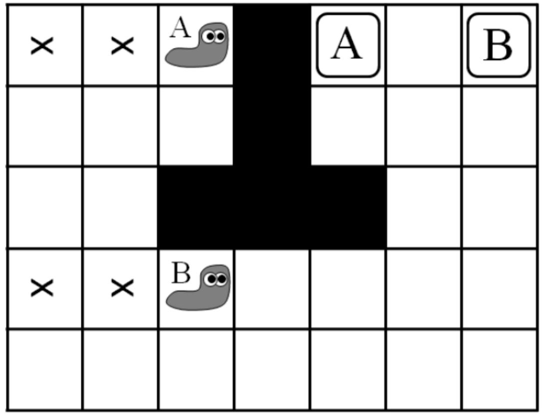

- You want to make two slugs go to their respective destination in an M by N grid.
- Every time, each slug can choose to move NSEW or stay
- For squares where slugs have passed by, a sticky substance is left. Those squares cannot be passed by again.
- You want to find the shortest-time solution.

**Questions:**

- How do you define a “state” in this problem?
- How many states are there?   (just estimate the order)
- What’s the maximum “branching factor” in your formulation?

# Learning Objective 1 – Answer




**Questions:**

- How do you define a “state” in this problem?

  A state can be defined as the combination of the following:
  a) the location of slug A
  b) the location of slug B
  c) a boolean for each square indicating whether it has been passed by

- How many states are there? (just estimate the order)

  $(MN) \times (MN) \times 2^{MN}$

- What’s the maximum “branching factor” in your formulation?

  $5 \times 5$

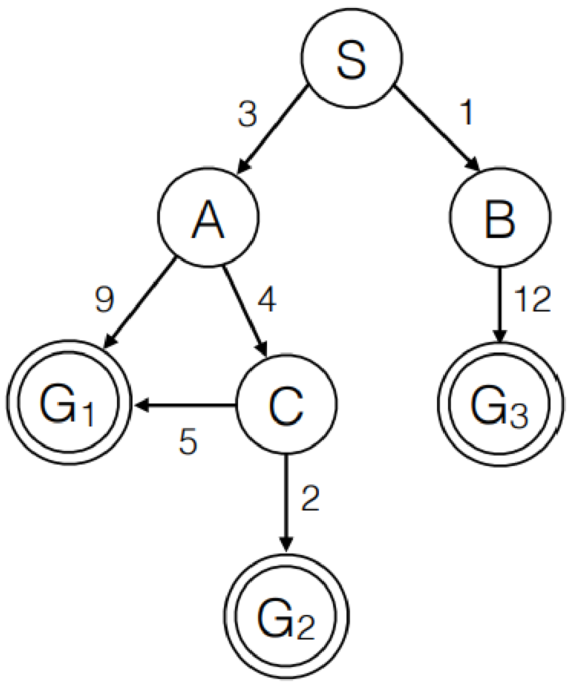
# Learning Objective 2

- Understand the execution of
  - BFS
  - DFS
  - Uniform-cost search
  - A* search
  - Best-first (Greedy) Search
- Variants of algorithms
  - Early/late goal test

# Learning Objective 2 – Example

**Question:**

Write out the sequence of nodes expanded by A* search using H-3 as the heuristic function?




| | A | B | C | S |
|---|---|---|---|---|
| H-1 | 0 | 0 | 0 | 0 |
| H-2 | 6 | 7 | 1 | 7 |
| H-3 | 7 | 7 | 1 | 7 |
| H-4 | 4 | 7 | 1 | 7 |

**Question:**
Should you use early or late goal test for A* search?

# Learning Objective 3

- Analysis of different variants of algorithms
  - Time complexity
  - Space complexity
  - Optimality guarantee
- Time and space tradeoff in DFS
  - Graph search/Tree search (tree search algorithm does not maintain the list of expanded states)
  - How to handle cycles in tree search?

# Learning Objective 3 (Review)

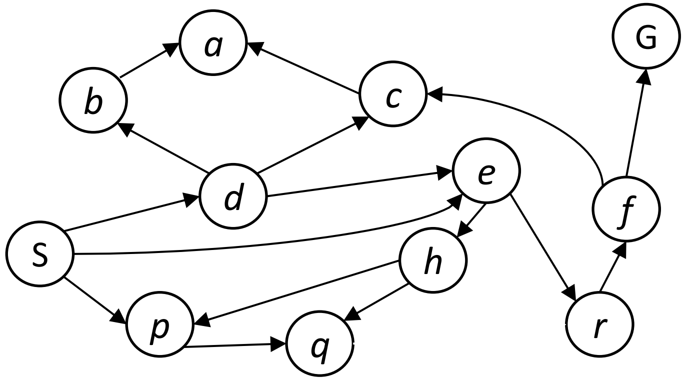
```
Frontier ← { initial_state }
While Frontier is not empty:
    Pop the newest node s from Frontier

    For all action a:
        s′ ← succ(s, a)
        If not Reached[s′]:
            If s′ is a goal state, terminate
            Push s′ to Frontier
            Reached[s′] ← True
```

Graph search

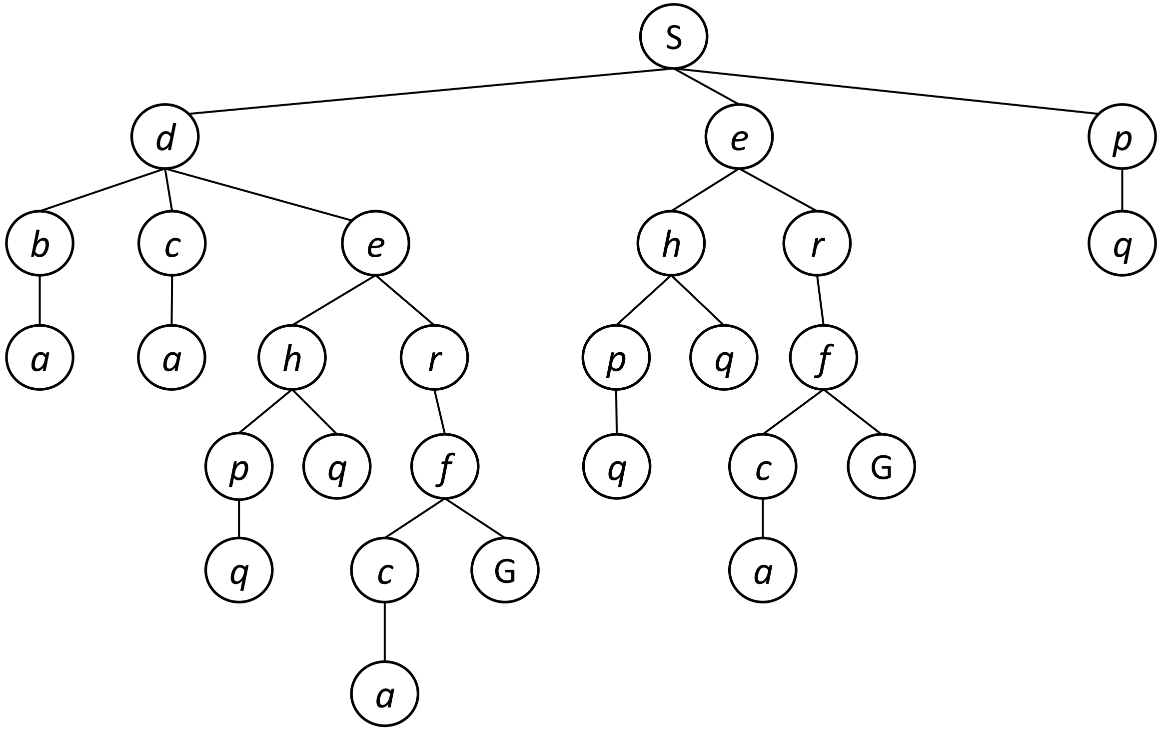
```
Frontier ← { initial_state }
While Frontier is not empty:
    Pop the newest node s from Frontier

    For all action a:
        s′ ← succ(s, a)
        If s′ is a goal state, terminate
        Push s′ to Frontier
```

Tree search

# Learning Objective 3 (Review)

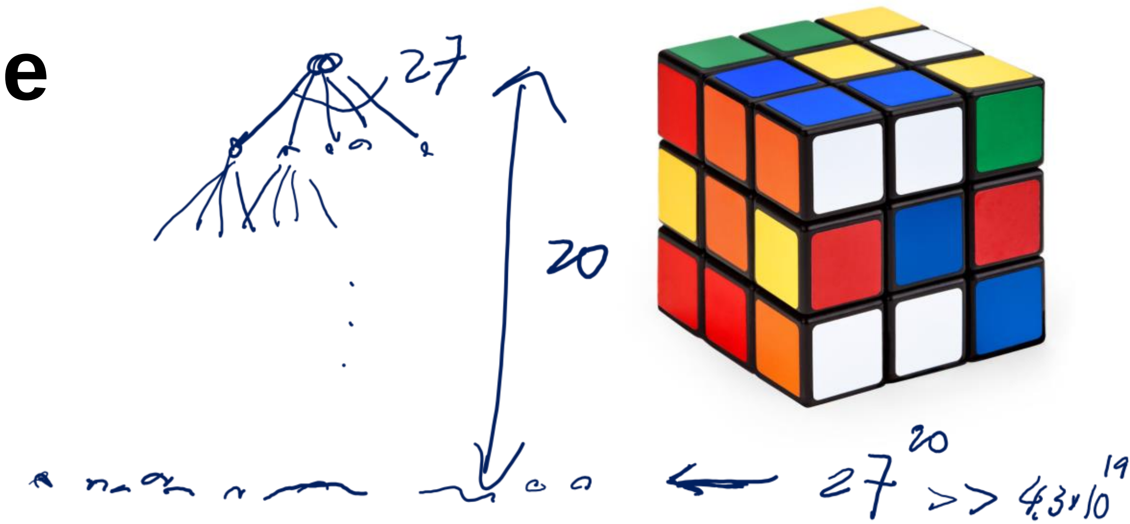



Graph search




Tree search

# Learning Objective 3 – Example




- Rubik's cube
  - $4.3 \times 10^{19}$ configurations
  - Branching factor = 27

**Question**

Assume that we start from a configuration where the shortest solution is 20 steps away.

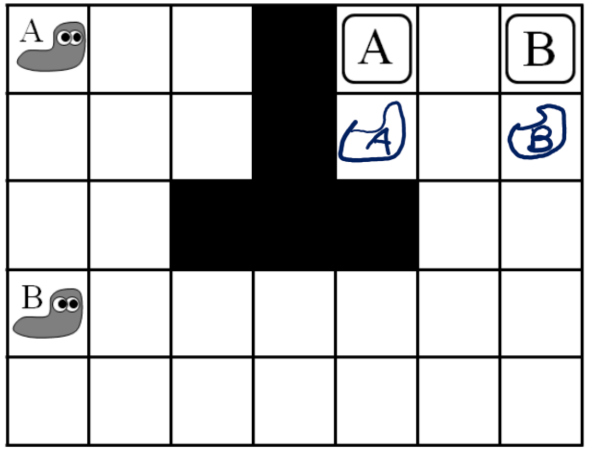
- Worst-case time complexity of BFS (graph search) to find the goal $4.3 \times 10^{19}$
- Worst-case time complexity of BFS (tree search) to find the goal $27^{20}$
- Worst-case time complexity of DFS (graph search) to find the goal $4.3 \times 10^{19}$
- Worst-case time complexity of DFS (tree search) to find the goal $27^{20}$

Best case DFS 20

# Learning Objective 4

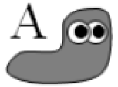
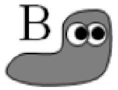
- Design of heuristic functions
- Analysis of heuristic functions
  - Admissibility
  - Consistency

# Learning Objective 4 – Example




Suppose every time step incurs a cost of 1.

Define

$h(s) =$ Manhattan(  , A ) + Manhattan (  , B )

$h^*(s) = 1$

$h(s) = 2$

$h(s) > h^*(s)$

$\leq$

**Question:** Is h(s) admissible?

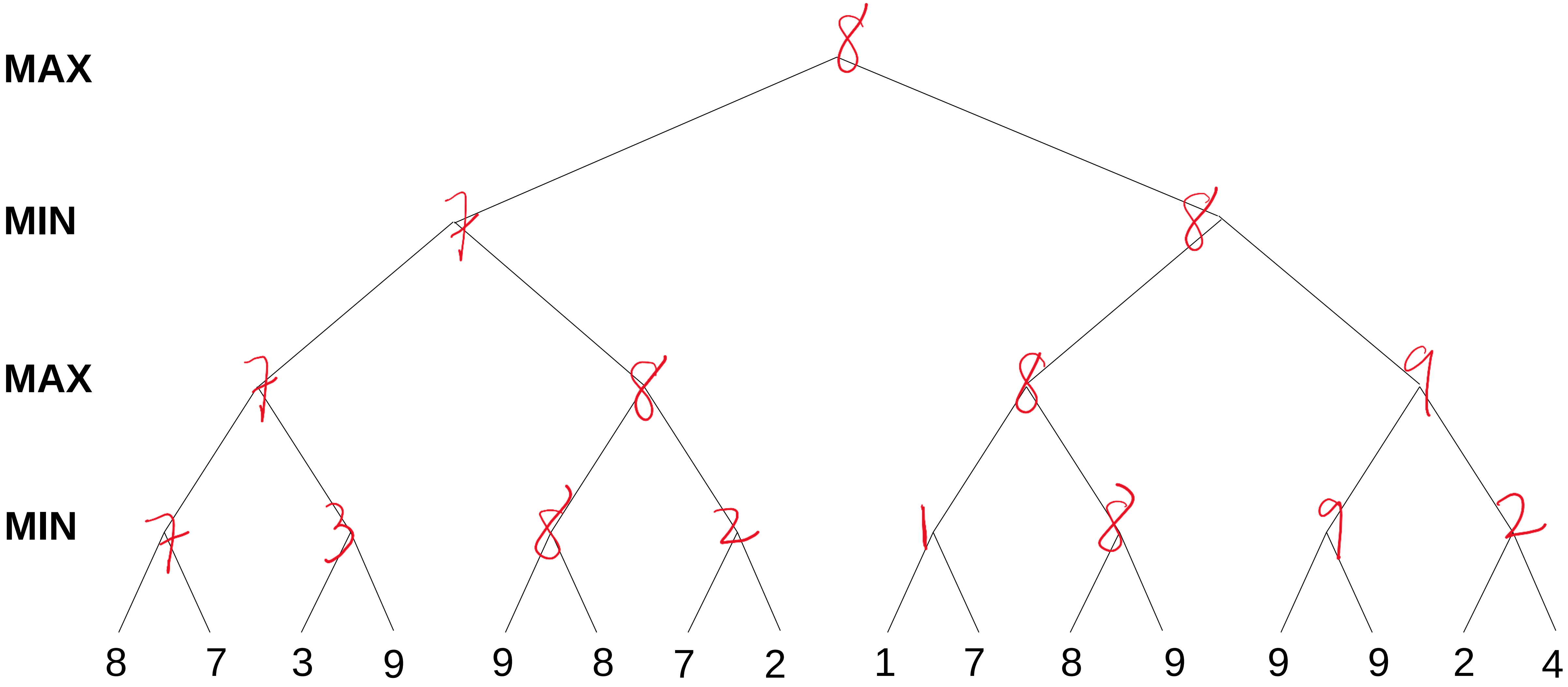
# Search in Games

# Learning Objective 1

- Minimax tree search algorithm
- Alpha-beta pruning
- Expectimax search algorithm

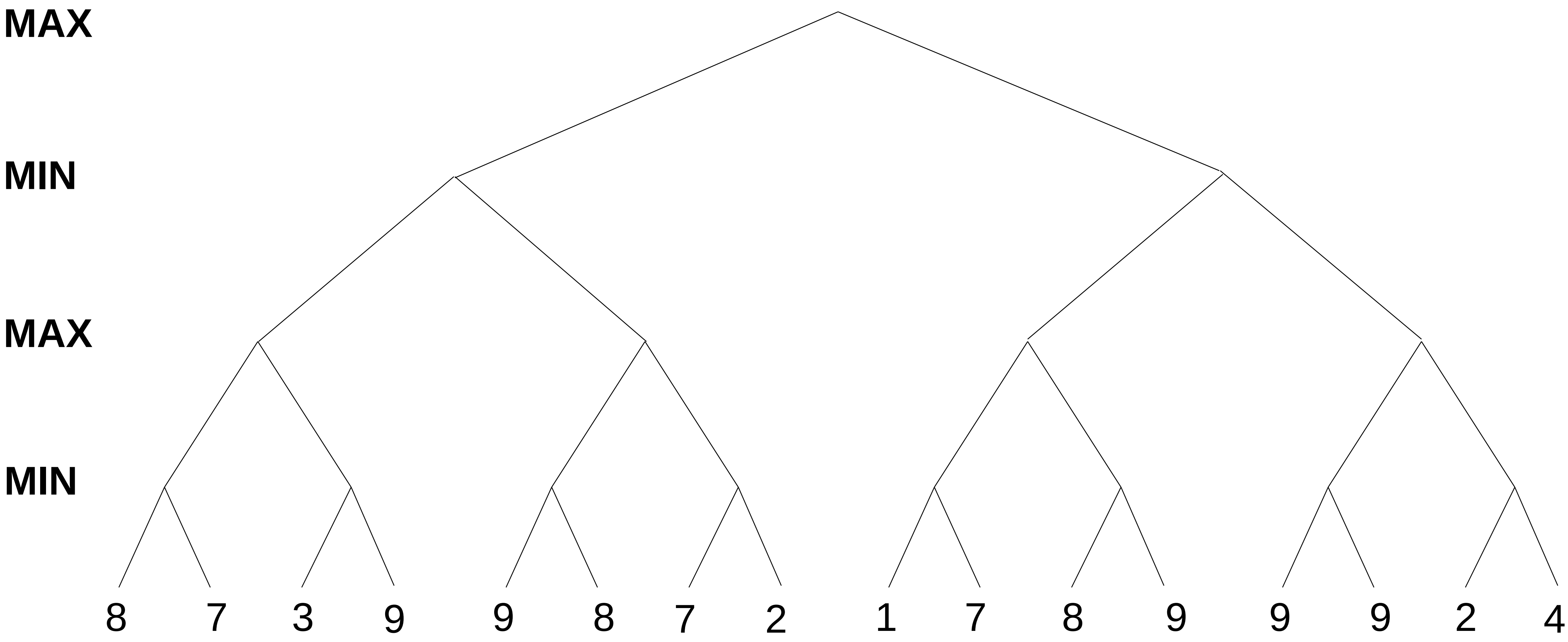
# Learning Objective 1 – Example

**Question:** What are the game value at each node?



# Learning Objective 1 – Example

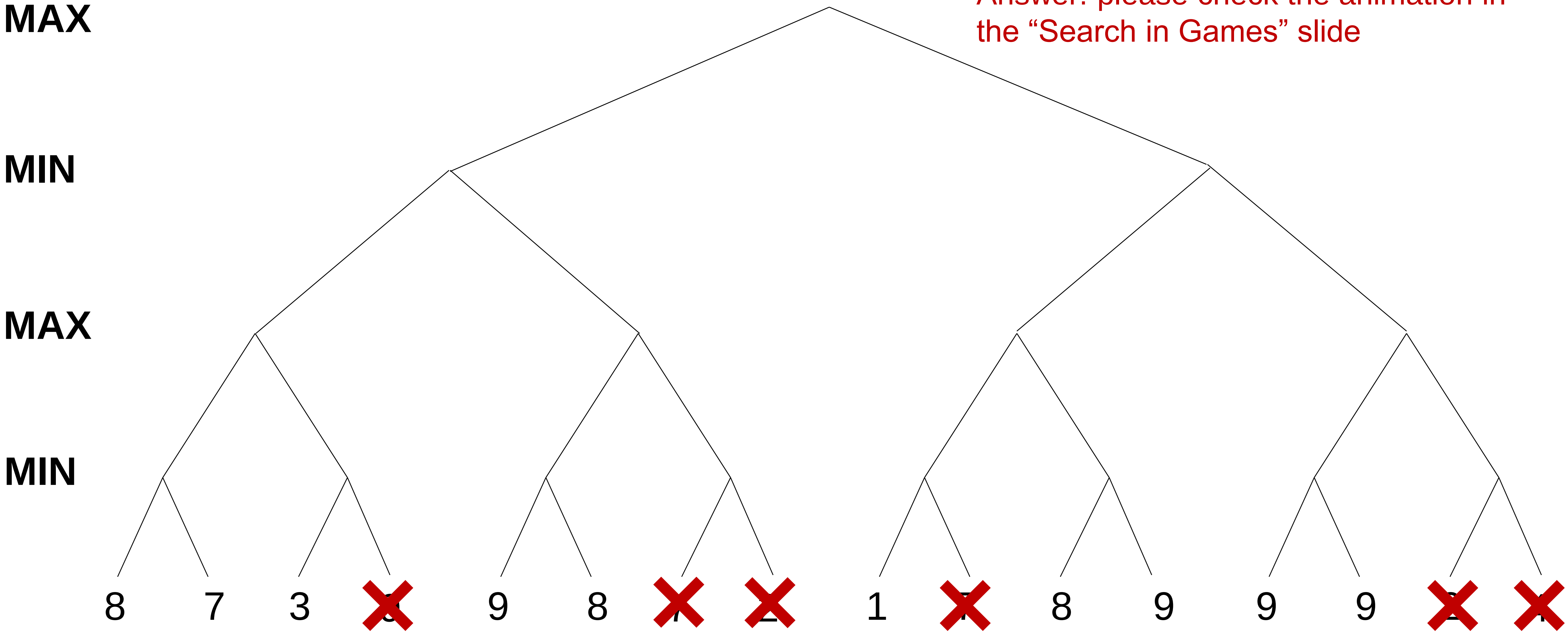
**Question:** which numbers will be pruned with alpha-beta pruning?



# Learning Objective 1 – Answer

**Question:** which numbers will be pruned with alpha-beta pruning?

Answer: please check the animation in the "Search in Games" slide



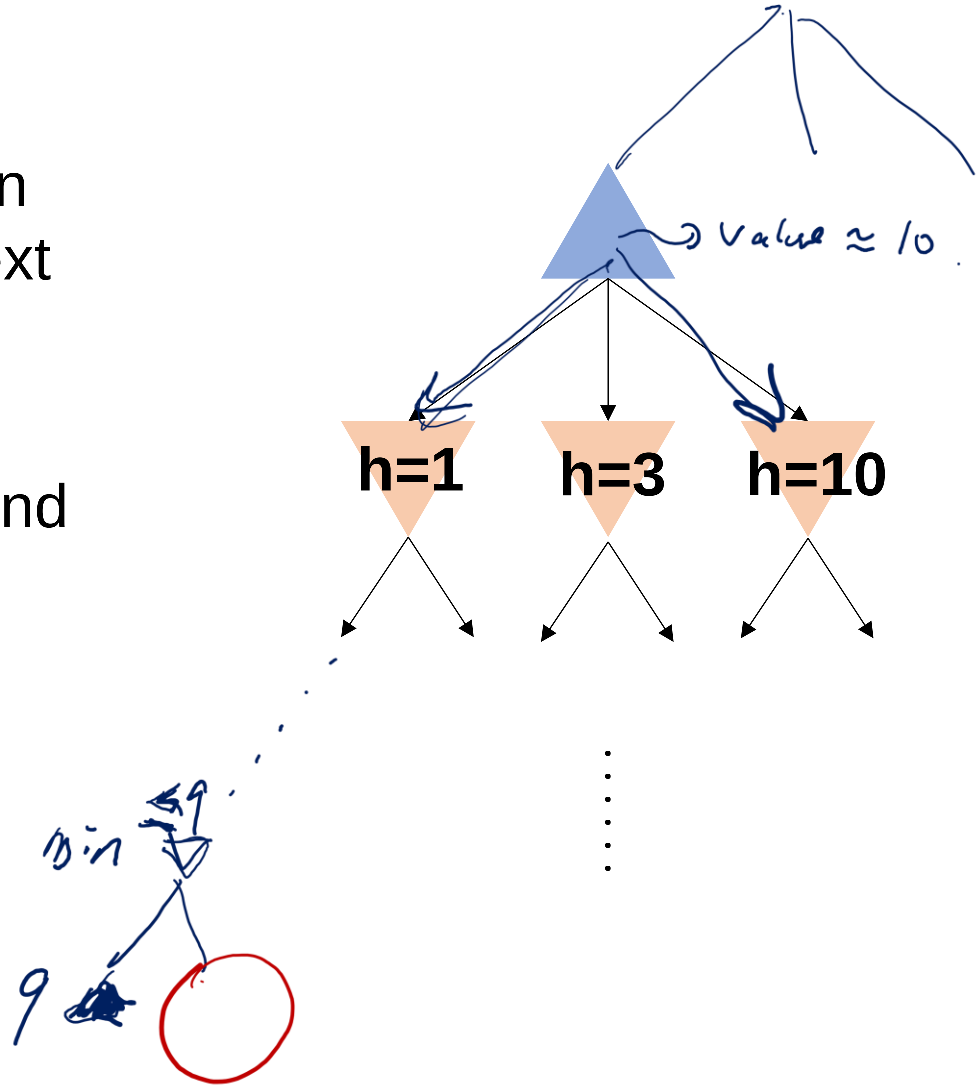
# Learning Objective 2

- How to leverage evaluation functions?
  - Allows depth-limited search
  - Helps deciding the next branch to expand in minimax search

# Learning Objective 2 – Example

For a MAX node, suppose you have an evaluation function that can evaluate the MIN node in the next layer accurately.

**Question:** Which subtree would you like to expand first in order to gain more pruning?



# Learning Objective 3

- Monte-Carlo tree search algorithm
  - Selection, Expand, Simulation, Backup
  - How to explore the tree? (the idea behind UCB1)

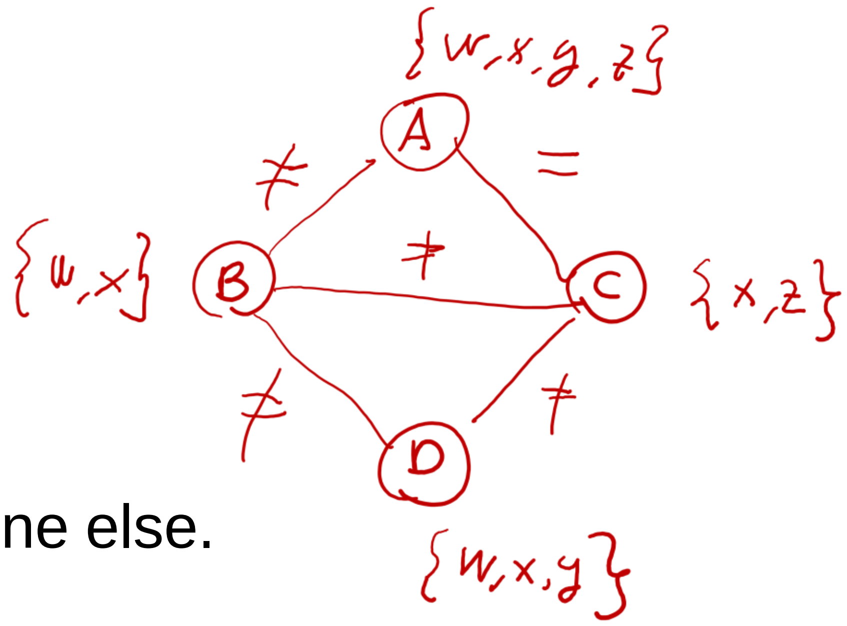
# Constraint Satisfaction Problem

# Learning Objective 1

- Problem formulation
  - Variables
  - Domain
  - Constraints

# Learning Objective 1 – Examples

A, B, C, D are picking their entries at a restaurant.  The entrees are w, x, y, z. They have the following restriction:




- C will not order w or y
- D will not order z
- B will only order w or x
- A and B will order different dishes
- C want to be the same as A, but different with anyone else.
- B wants to be different with all others

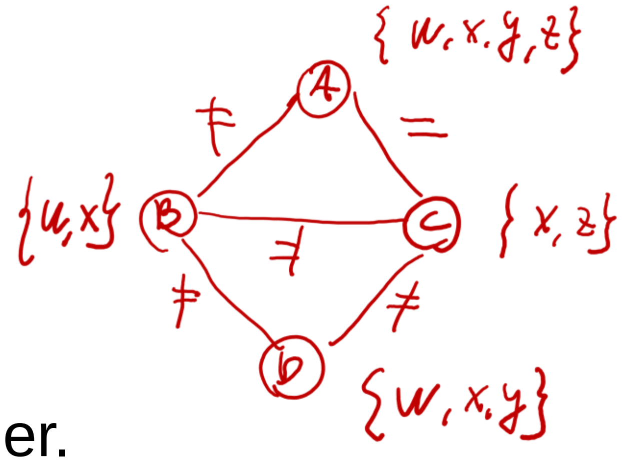
**Question**

- Draw a constraint graph for this problem.

# Learning Objective 2

- Backtracking search algorithm
- Forward checking
- Maintaining arc consistency
- The tradeoff between “more checking” and “late failure detection”

# Learning Objective 2 – Example




(Continued from the previous problem)

- Suppose we iterate variables and values in alphabetical order.
- Execute the algorithm using "backtracking search" + "forward checking"
- Execute the algorithm using "backtracking search" + "maintaining arc consistency"

| | A | B | C | D |
|---|---|---|---|---|
| | w, x, y, z | w, x | x, z | w, x, y |
| decide A → | w | x | ☐ | w, x, y |
| decide A → | x | w | x | w, x, y |
| decide B → | x | w | x | x, y |
| C → | x | w | x | y |

| A | B | C | D |
|---|---|---|---|
| w, x, y, z | w, x | x, z | w, x, y |
| w | x | ☐ | w, x, y |
| x | w | x | ~~w~~, ~~x~~, y |

# Learning Objective 3

- Variable ordering
  - Minimum Remaining Value principle
  - Degree heuristic
- Value ordering
  - Least constrained value principle

# Learning Objective 3 – Example

- Use “Minimum Remaining Value principle” to decide the variable ordering (keep value ordering as alphabetical, breaking ties alphabetically, do forward checking).

| | A | B | C | D |
|---|---|---|---|---|
| Original domain | w,x,y,z | w,x | x,z | w,x,y |
| Decide B | x,y,z | w | x,z | x,y |
| Decide C | x | w | x | y |
| Decide A | x | w | x | y |
| Decide D | x | w | x | y |

# Learning Objective 3 – Example

- Use “Least constrained value principle” to decide the value ordering (keep variable ordering as alphabetical, using forward checking to estimate how many values are eliminated).

| | A | B | C | D |
|---|---|---|---|---|
| Original domain | w,x,y,z | w,x | x,z | w,x,y |
| Decide A | w | x | | w,x,y |
| | x | w | x | w,x,y |
| | y | w,x | | w,x,y |
| | z | w,x | z | w,x,y |
| Decide B | z | w | z | x,y |
| | z | x | z | w,y |
| Decide C | z | w | z | x,y |
| Decide D | z | w | z | x |

← Eliminate 3 values in total

← Eliminate 2 values in total

← Eliminate 2 values in total

← Eliminate 1 values in total **(this eliminate least values, so we go with this)**

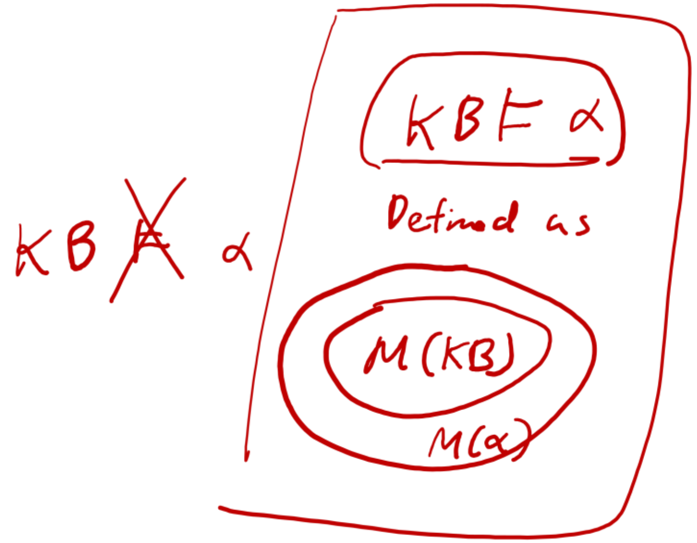
Both eliminate 1 values in total (break ties alphabetically)

# Logic

# Learning Objective 1

- Syntax of propositional logic
- Concepts in propositional logics
  - Model
  - Entailment
  - The relation between sentence, model, entailment
- Problem formulation (translation between PL and natural language)
- Inference
  - Soundness
  - Completeness

# Learning Objective 1 – Example



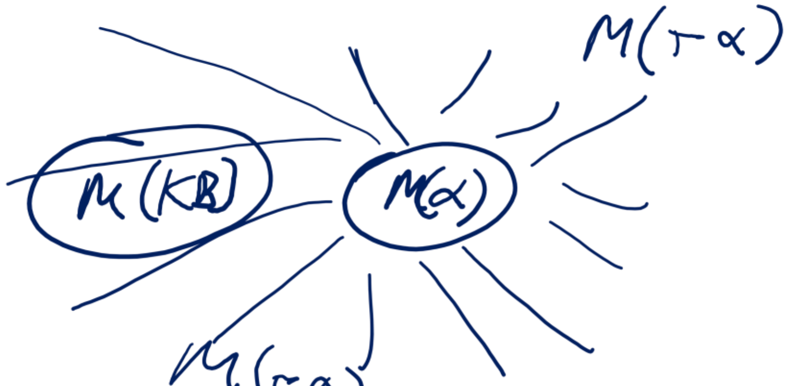

**Question:** Are the following statement true?

- If KB does not entail $\alpha$, then $\mathcal{M}(\text{KB}) \subset \mathcal{M}(\neg\, \alpha)$
- If KB contradicts $\alpha$, then $\mathcal{M}(\text{KB}) \subset \mathcal{M}(\neg\, \alpha)$

**Question**

There are three box. One contains gold, the other two are empty. Each box has a clue on it:

Box 1: “The gold is not here”
Box 2: “The gold is not here”
Box 3: “The gold is in Box 2”




We know exactly one clue is true; other are false.

Let $B_i$ represent “Gold is in box i”. Formalize the statements as propositional logic.

# Learning Objective 1 – Example

Exactly one box has gold: $(B_1 \wedge \neg B_2 \wedge \neg B_3) \vee (\neg B_1 \wedge B_2 \wedge \neg B_3) \vee (\neg B_1 \wedge \neg B_2 \wedge B_3)$

Box 1 clue: $C_1 \equiv \neg B_1$

Box 2 clue: $C_2 \equiv \neg B_2$

Box 3 clue: $C_3 \equiv B_2$

Exactly one clue is true: $(C_1 \wedge \neg C_2 \wedge \neg C_3) \vee (\neg C_1 \wedge C_2 \wedge \neg C_3) \vee (\neg C_1 \wedge \neg C_2 \wedge C_3)$
$\equiv (\neg B_1 \wedge B_2 \wedge \neg B_2) \vee (B_1 \wedge \neg B_2 \wedge \neg B_2) \vee (B_1 \wedge B_2 \wedge B_2)$
$\equiv (B_1 \wedge \neg B_2) \vee (B_1 \wedge B_2)$
$\equiv B_1 \wedge (\neg B_2 \vee B_2)$
$\equiv B_1$

Thus, we know $B_1 = 1,\ B_2 = 0,\ B_3 = 0$

# Learning Objective 2

- Special forms of sentences
  - Horn clauses
  - CNF
- Inference rules in propositional logic
  - Modus ponens
  - Resolution
- Forward inference algorithm

# Learning Objective 2 – Examples

**Question**

- Convert propositional logic sentence to CNF

**Question**

- Ask you to execute the forward inference algorithm (see an example later)

# Learning Objective 3

- Syntax of first-order logic
- Translation between natural language and first-order logic
- Similar concepts as in propositional logic
  - Horn clause
  - CNF
  - Modus ponens
  - Resolution

# Learning Objective 3 – Examples

There is currently a war, and United States is desperate to round up all criminals. We want to determine whether *West* is a criminal.

We know: It is a crime for an American to sell weapons to hostile nations. The country Nono is Hostile. Furthermore, we know that Nono has some weapons, all of which were sold to it by *West*, who is American.

**Question.**

- Create a knowledge base using the following predicates: America(x), Criminal(x), Hostile(x), Weapon(x), Owns(x,y), Sells(x,y,z)
- Try to use inference rules to derive Criminal(*West*)

# Learning Objective 3 – Examples

(1) $\forall x,y,z,$ American(x) ∧ Weapon(y) ∧ Hostile(z) ∧ Sells(x,y,z) → Criminal(x)

(2) Hostile(Nono)

(3) $\exists x,$ Weapon(x) ∧ Owns(Nono, x)

(4) $\forall x,$ Weapon(x) ∧ Owns(Nono, x) → Sell(West, x, Nono)

(5) American(West)

By (3) and Existential Instantiation / Skolem function → (6) Weapon($W_1$) ∧ Owns(Nono, $W_1$)

Modus ponens on (6) + (4) with substitution {x / $W_1$} → (7) Sell(West, $W_1$, Nono)

And elimination on (6) → (8) Weapon($W_1$)

Modus ponens on (5) + (8) + (2) + (7) with substitution {x / West, y/ $W_1$, z/Nono} → (9) Criminal(West)

# Probability and Bayesian Network

# Learning Objectives

- Probability basics
- Independence and conditional independence
- From causal relation to Bayesian network
- From Bayesian network to probabilistic model
- In Bayesian network, decide if $X \perp\!\!\!\perp Y \mid Z$