CS2109S Tutorial 3

Adversarial Search and Local Search

(AY 24/25 Semester 2)

February 14, 2025

(Prepared by Benson)

Contents

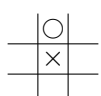
Q1. Tic-Tac-Toe

The following heuristic evaluation function is used at each leaf node n:

$$Eval(n) = P(n) - O(n)$$

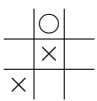
where $P(n)$ is the number of possible winning lines for the player while $O(n)$ is the number of possible winning lines for the opponent.

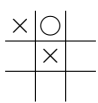
Consider this state where the $\times$ player moves next:

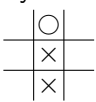


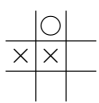
Q1. Tic-Tac-Toe

Poll: What is the optimal next move of the $\times$ player?

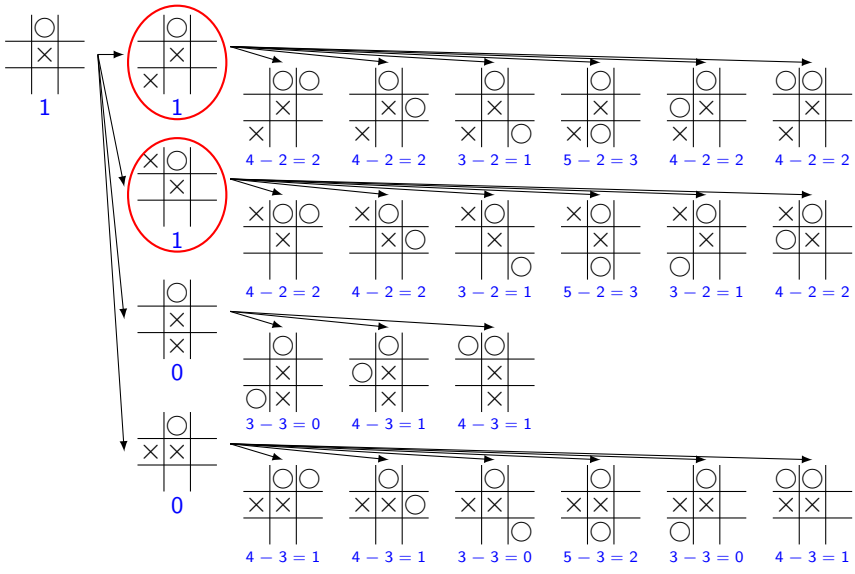
A. 

B. 

C. 

D. 

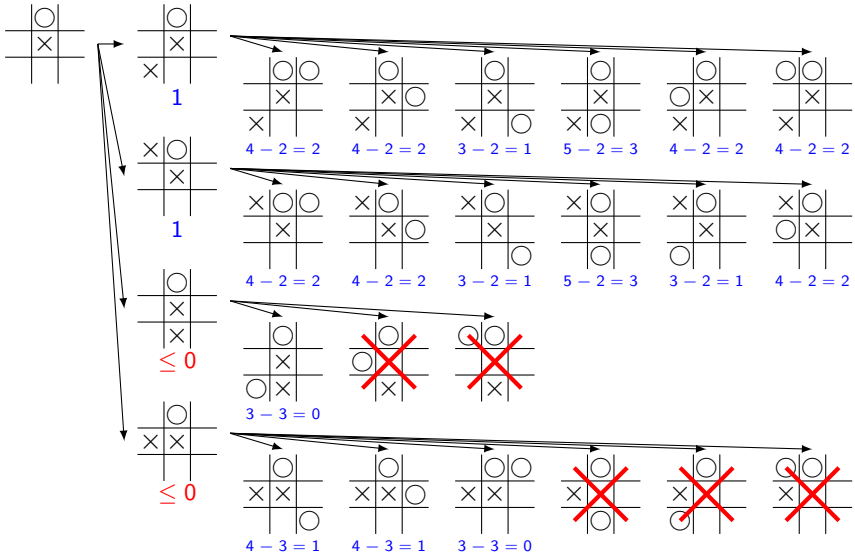
Q1. Tic-Tac-Toe



Q1. Tic-Tac-Toe

Explain how you could have saved time by eliminating some options early.

Q1. Tic-Tac-Toe



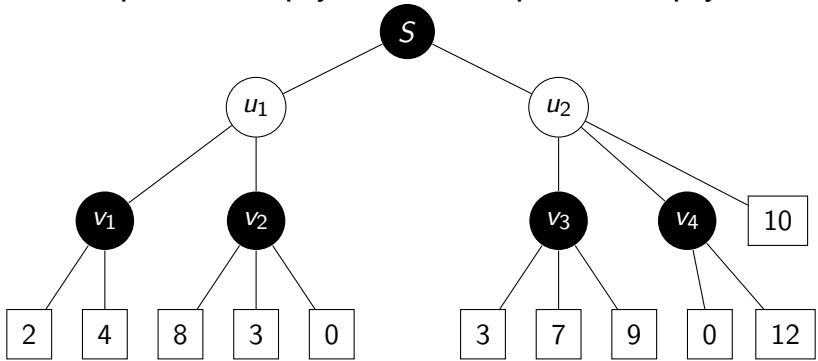
Alpha Beta Pruning

- α = Best already explored option along path to the root for MAX
  β = Best already explored option along path to the root for MIN
- We always wish to "reduce" the range $[\alpha, \beta]$. Hence, the MAX player increases α and the MIN player reduces β.
- Once $\alpha \geq \beta$, the current subtree isn't relevant anymore and we can prune the remaining branches.

Q2. Alpha Beta Pruning

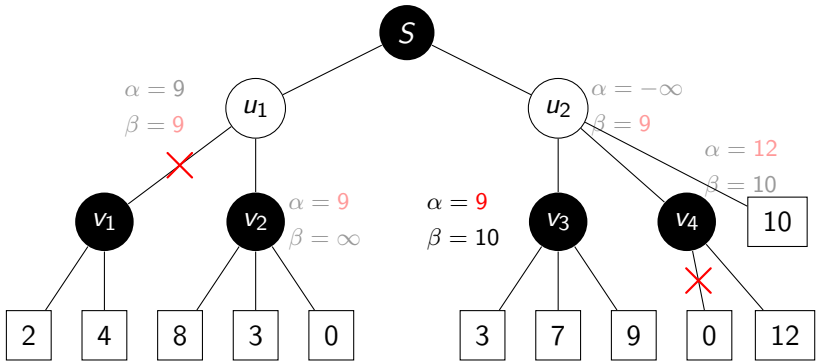
(a) Assume that we iterate over nodes from right to left; mark with an 'X' all arcs that are pruned by α-β pruning, if any.

NOTE: Black node represents MAX player, white node represents MIN player.



Q2. Alpha Beta Pruning

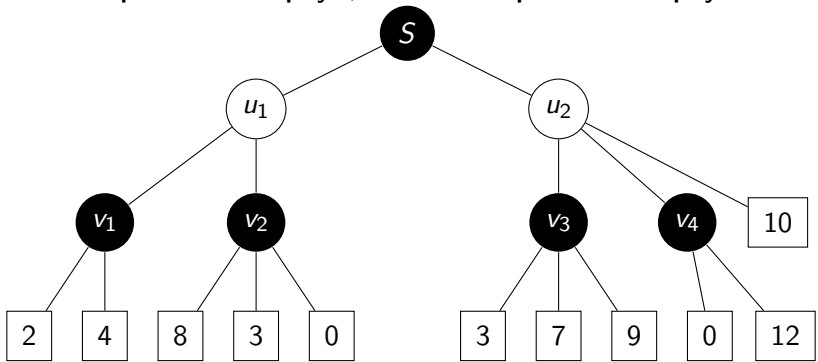
Solution:



Q2. Alpha Beta Pruning

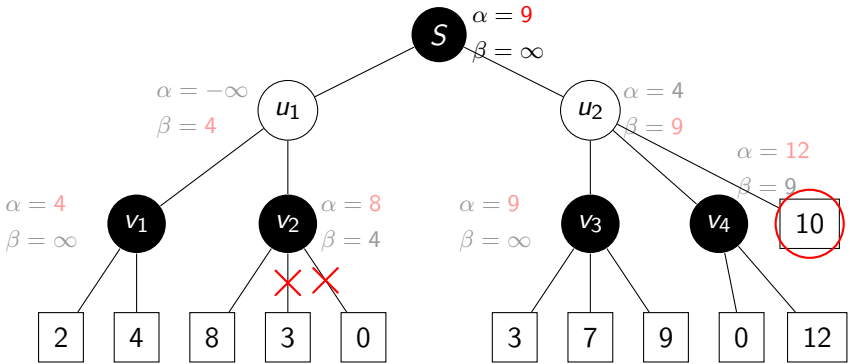
(b) Show that the pruning is different when we instead iterate over nodes from left to right. Your answer should clearly indicate all nodes that are pruned under the new traversal ordering.

NOTE: Black node represents MAX player, white node represents MIN player.



Q2. Alpha Beta Pruning

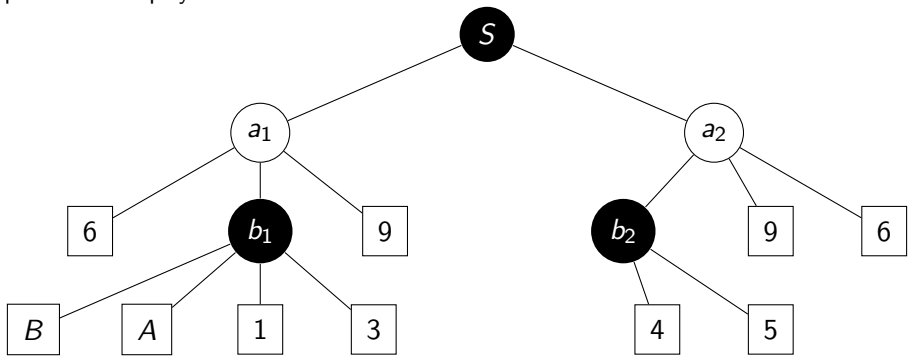
Solution:



Q4. Alpha Beta Pruning

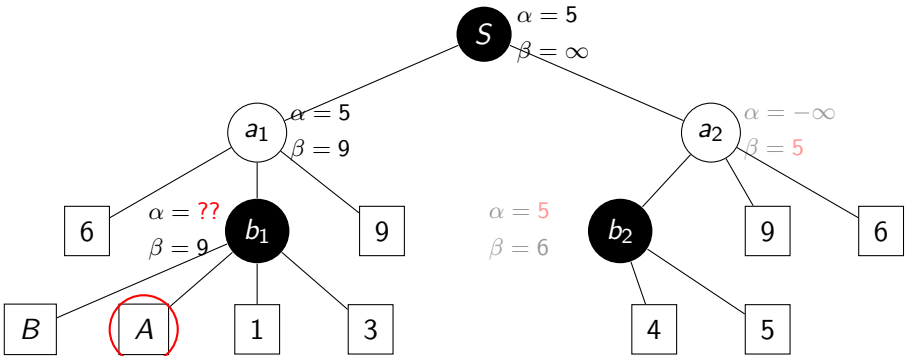
In order for node B to NOT be pruned, what values can node A take on?

NOTE: Assume that we iterate over nodes from right to left. Black node represents MAX player, white node represents MIN player.



Q4. Alpha Beta Pruning

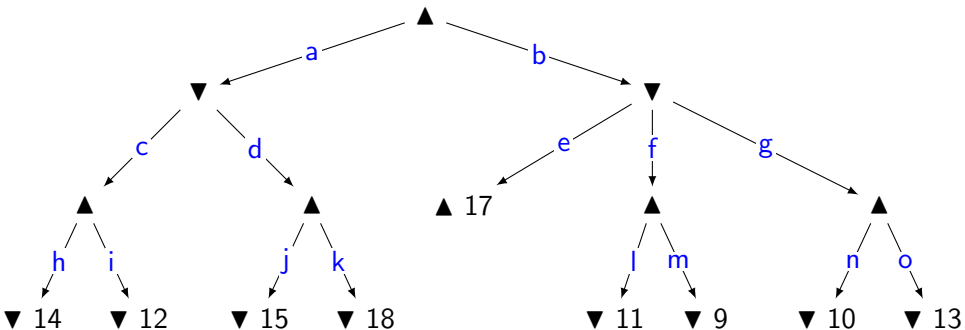
Solution:



Node B is NOT pruned $\Rightarrow \alpha < \beta \Rightarrow A < 9$

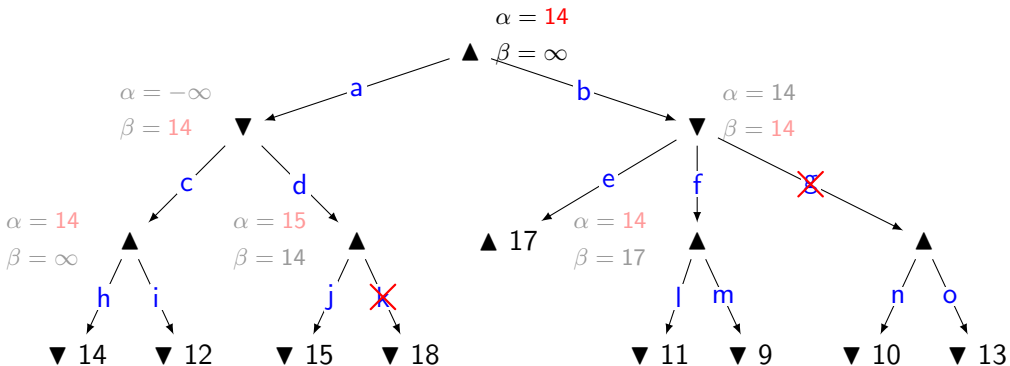
(Mini-)Practice

Poll: Suppose we traverse this tree with DFS from left to right. Select all the link(s) that would be pruned by alpha-beta.



(Mini-)Practice

Solution:



Q3. Nonogram

			3	1	1	4	4
1	1	1					
	1	2					
	2	2					
		2					
		1					

(a) Initial Nonogram

			3	1	1	4	4
1	1	1	■		■		■
	1	2	■			■	■
	2	2	■	■		■	■
		2				■	■
		1				■	

(b) Solved Nonogram

Q3. Nonogram

(a) Having learnt both informed search and local search, you think that local search is more suitable for this problem. Give 2 possible reasons why informed search might be a bad idea.

- The search space is very huge.
- We are not interested in obtaining the solution path, but rather, reaching the goal state.

Q3. Nonogram

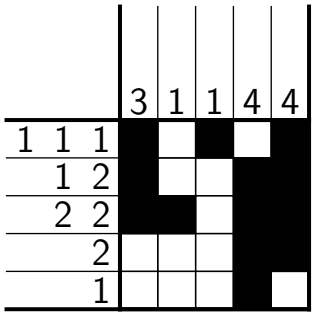
(b) Propose a state representation for the problem.

(c) What are the initial and goal states for the problem under your proposed representation?

(d) Define a reasonable transition function to generate new candidate solutions.

(e) Define a reasonable heuristic function to evaluate the "goodness" of a candidate solution. Explain how this heuristic can also be used as a goal test to determine that we have a solution to the problem.

Q3. Nonogram



State Representation:

1	0	1	0	1
1	0	0	1	1
1	1	0	1	1
0	0	0	1	1
0	0	0	1	0

Q3. Nonogram

	3	1	1	4	4
1 1 1					
1 2					
2 2					
2					
1					

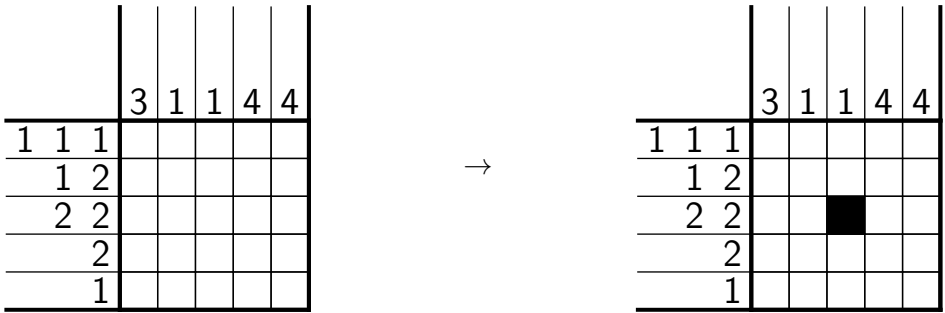
Initial State?

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

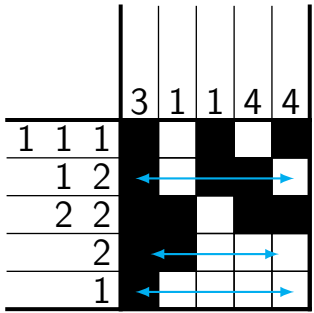
Q3. Nonogram

Transition function?: Pick a random cell and toggle its color.

- Is the goal state reachable?
- Is it reasonably efficient?



Q3. Nonogram



A "Better" Initial State:

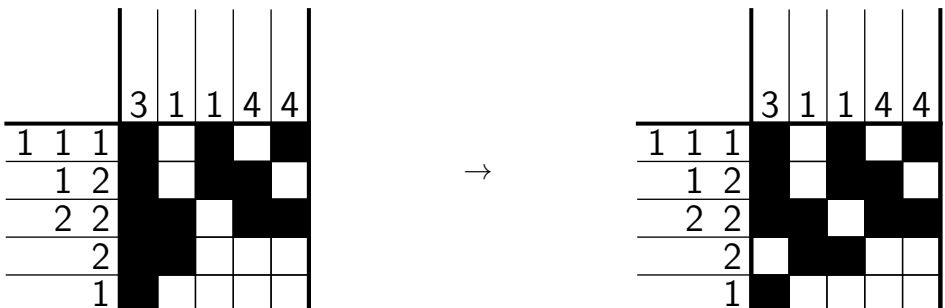
- Satisfies row constraints.
- Can be horizontally moved.

1	0	1	0	1
1	0	1	1	0
1	1	0	1	1
1	1	0	0	0
1	0	0	0	0

Q3. Nonogram

Transition function: Pick a random block on a random row. Move the block leftwards or rightwards. **(Row constraints are still satisfied)**

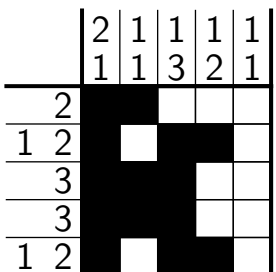
- Is the goal state reachable?
- Is it reasonably efficient?



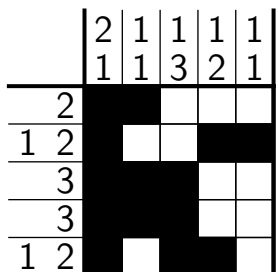
Q3. Nonogram

Heuristic function (how "good" is a state?):

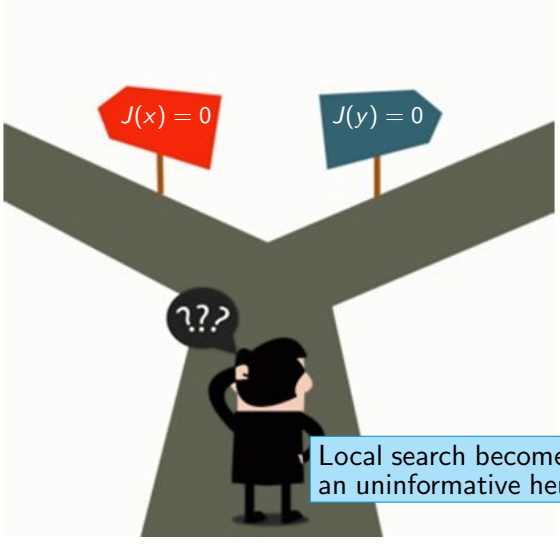
- 1 if all constraints are satisfied, 0 otherwise.
- Number of columns with any constraints satisfied/violated.
- Number of column constraints satisfied/violated.



→



Q3. Nonogram

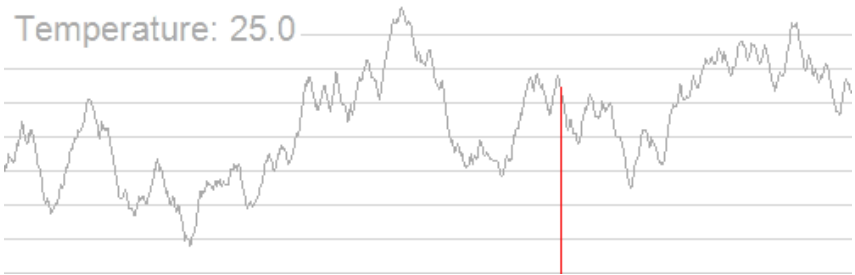


Local search becomes “wild guessing” with an uninformative heuristic function.

Q3. Nonogram

(f) Local search is susceptible to local minimas. Describe how you can modify your solution to combat this.

- Introduce random restarts by repeating local search from a random initial state.
- Use simulated annealing or stochastic beam search.



Q3. Nonogram

Poll: How does simulated annealing / stochastic beam search escape local minimas?

A. By introducing random restarts.

B. By accepting a worse solution probabilistically.

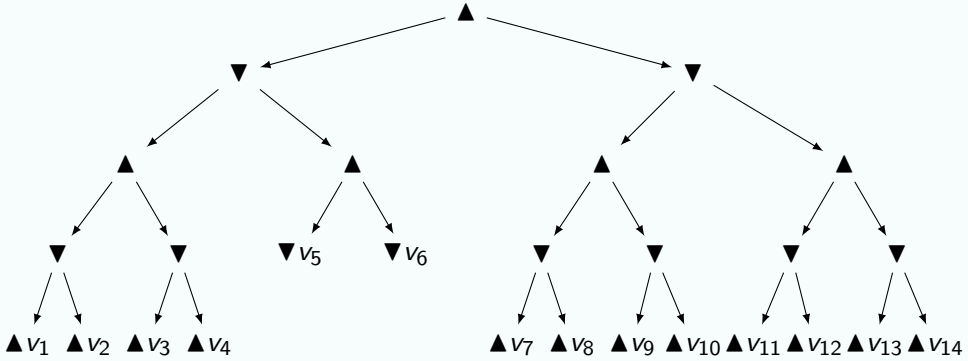
C. By considering multiple neighbouring states.

D. By running multiple instances of the algorithm in parallel.

Bonus. Maximum and Minimum Pruning

Extra Slide

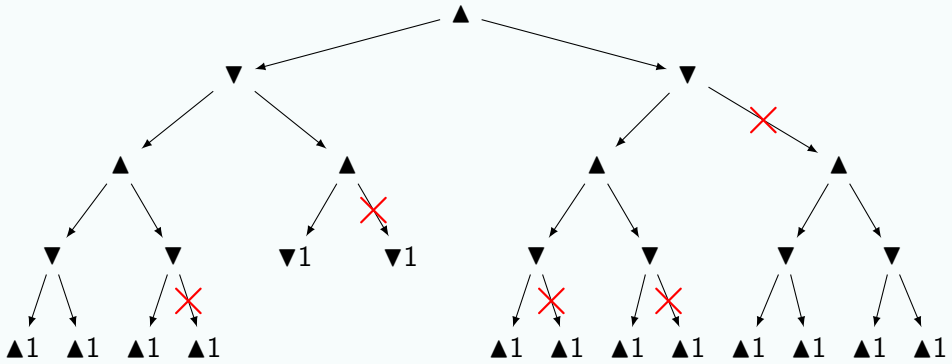
Leaf nodes take values which are either 0 or 1. Assume that we iterate over nodes from left to right. Write down the values of the leaf nodes where the **minimum** / **maximum** number of leaf nodes get skipped.



Bonus. Maximum and Minimum Pruning

Extra Slide

Maximum: 8 leaf nodes will be skipped.



Bonus. Maximum and Minimum Pruning

Extra Slide

Minimum: No leaf nodes will be skipped.