CS3243 INTRODUCTION TO ARTIFICIAL INTELLIGENCE

MID-TERM REVISION

nicholas.teh@u.nus.edu

CS3243 INTRODUCTION TO ARTIFICIAL INTELLIGENCE

MODELLING A SEARCH PROBLEM & HEURISTICS

nicholas.teh@u.nus.edu

MODELLING A SEARCH PROBLEM (AY19/20 SEM 2 FINAL EXAM Q1)

- Set of minions: $M = \{m_1, m_2, \ldots, m_n\}$
- Distinct set of bananas: $B = \{b_1, b_2, \ldots, b_k\}$
- Bananas are not divisible (minions can't take half a banana!).
- Each minion has possibly distinct utility for each piece of banana, and the utility of a minion for a particular banana may differ from the utility of another minion for the same banana. In more detail, the value of minion m_i for banana b_j is v_{ij}. If a minion m_i is given a bundle of bananas $S \subset B$, their utility for the bundle is given by $v_i(S) = \sum_{b_j \in S} v_{ij}$.

MODELLING A SEARCH PROBLEM (AY19/20 SEM 2 FINAL EXAM Q1)

- In an allocation $A = \{A_1, A_2, \ldots, A_n\}$ of bananas to minions, A_i is the set of bananas allocated to minion m_i.
- An allocation is *envy-free* if for every two minions, $m_i, m_j \in M, v_i(A_i) \geq v_i(A_j)$
- A *valid* allocation is one where each banana is allocated to exactly one minion.
- Given an allocation $A = \{A_1, A_2, \ldots, A_n\}$, we define the *envy graph* as follows: nodes are minions and there is a directed edge $m_i \rightarrow m_j$ if minion m_i envies minion m_j.

MODELLING A SEARCH PROBLEM (AY19/20 SEM 2 FINAL EXAM Q1)

- An envy cycle is a directed cycle (with minions as nodes) in the envy graph. For instance, in the 3-minion envy cycle $m_1 \to m_2 \to m_3 \to m_1$, m_1 envies m_2, m_2 envies m_3, and m_3 envies m_1.
- Our goal is to achieve a *valid*, *envy-free* allocation. We are allowed two possible actions:
  - AllocateBananaToUnenvied(b, m) assigns banana b to minion m, assuming that the minion m is not envied by anyone.
  - Decycle$(m_{i1}, m_{i2}, \ldots, m_{ik})$ assigns the bundle of minion m_{i2} to minion m_{i1}, the bundle of m_{i3} to m_{i2} and the bundle of m_{i1} to m_{ik}.
- **Both actions have a cost of 1**. We start with all items unallocated (i.e. the bundle of banana assigned to all minions is empty) and take one of the above actions at each round (if they can be taken). You may assume that a valid, envy-free allocation exists.

MODELLING A SEARCH PROBLEM (AY19/20 SEM 2 FINAL EXAM Q1)

- Define the **state**, **goal state**, and **actions** for this problem.
- State:
- Goal State:
- Actions:

MODELLING A SEARCH PROBLEM (AY19/20 SEM 2 FINAL EXAM Q1)

- Define the **state**, **goal state**, and **legal actions at a state** for this problem.
- **State**: Any partial allocation of bananas to minions
- **Goal State**: A valid and envy-free allocation of bananas to minions
- **Actions**: AllocateFruitToUnenvied, Decycle

Information is already given to you the question. Extract it! if you put "the actions above" or "the conditions above", no marks will be awarded due to vagueness.

ADMISSIBLE HEURISTICS (AY19/20 SEM 2 FINAL EXAM Q2)

- **Select all of the admissible heuristics** from the list below.

(a) $h_1(s)$ = number of unallocated bananas

(b) $h_2(s)$ = number of unenvied minions

(c) $h_3(s)$ = number of envied minions

(d) $h_4(s)$ = number of envy cycles

(e) None of the heuristics are admissible

ADMISSIBLE HEURISTICS (AY19/20 SEM 2 FINAL EXAM Q2)

- **Select all of the admissible heuristics** from the list below.

(a) $h_1(s)$ = number of unallocated bananas

(b) $h_2(s)$ = number of unenvied minions

(c) $h_3(s)$ = number of envied minions

(d) $h_4(s)$ = number of envy cycles

(e) None of the heuristics are admissible

ADMISSIBLE HEURISTICS (AY19/20 SEM 2 FINAL EXAM Q2)

- Let's analyse each of the heuristics for admissibility.

(a) $h_1(s)$ = number of unallocated bananas

- At every round, one banana is allocated. The minimum number of rounds (i.e. "best possible case") is the number of unallocated bananas. All bananas have to be allocated at the goal state. So it is definitely **admissible**.

ADMISSIBLE HEURISTICS (AY19/20 SEM 2 FINAL EXAM Q2)

- Let's analyse each of the heuristics for admissibility.

(b) $h_2(s)$ = number of unenvied minions

- Consider a scenario where there is currently no envy, and there are only a few more bananas to allocate (a few decycles can remove any envy). Then if there is a large number of minions, the heuristic overestimates. Or the goal state has *n* unenvied minions, $h_2(G) = n >> 0$. **Inadmissible**.

ADMISSIBLE HEURISTICS (AY19/20 SEM 2 FINAL EXAM Q2)

- Let's analyse each of the heuristics for admissibility.

  (c) $h_3(s)$ = number of envied minions

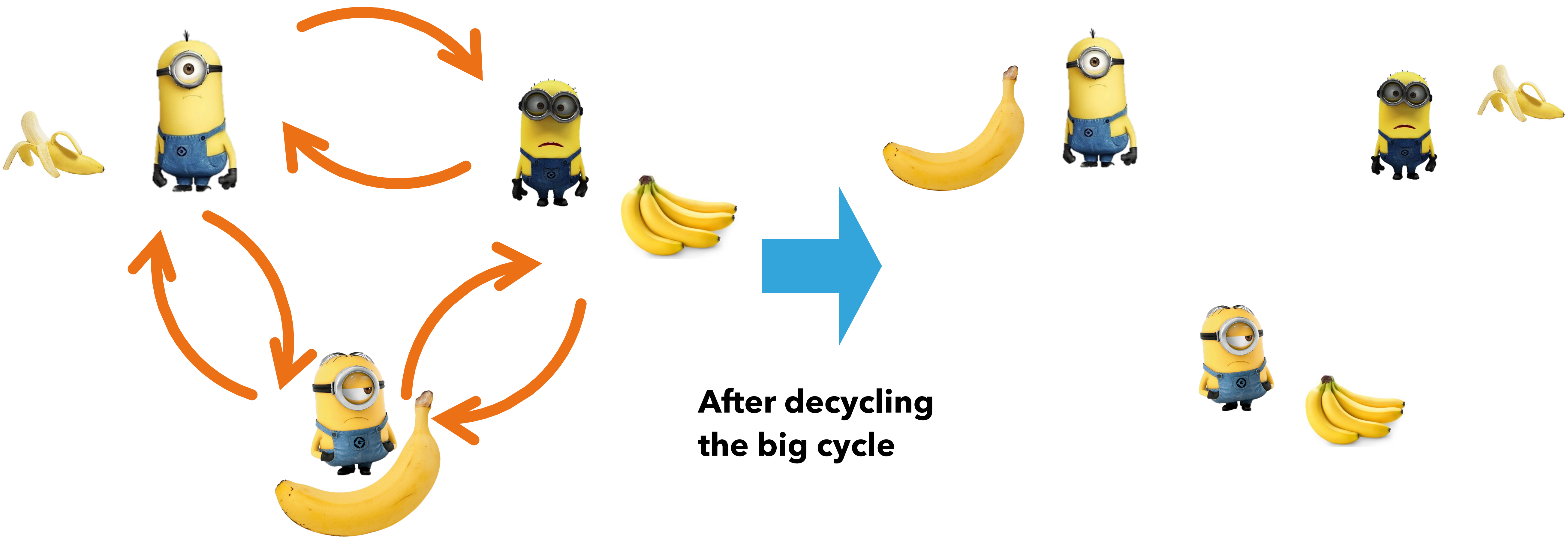
- Consider a scenario where all minions are in one envy cycle, and all bananas have been allocated. Then, the number of envied minions is *n*, but just one decycling action can bring us to the goal state. The heuristic overestimates. **Inadmissible**.

ADMISSIBLE HEURISTICS (AY19/20 SEM 2 FINAL EXAM Q2)

- Let's analyse each of the heuristics for admissibility.

(d) $h_4(s)$ = number of envy cycles

Consider a scenario where there are multiple overlapping envy-cycles. Then decycling one envy cycle can break all envy cycles. So the number of envy cycles overestimates. A counterexample is as follows:



DOMINATING HEURISTICS (AY19/20 SEM 2 FINAL EXAM Q3)

- With reference to the heuristics in Question 2, select all of the following that are true.

  (a) $\min(h_1, h_2)$ dominates h_1

  (b) $\max(h_2, h_3)$ dominates one heuristic in the set $\{h_1, h_4\}$

  (c) $\max(h_1, h_4)$ dominates both h_1 and h_4

  (d) $\min(h_1, h_3)$ does not dominate h_1

Properties max/min admissible/inadmissible can give you an idea of the answer. Though some times it's not straightforward to use abstract reasoning to answer. Need contextual explanation.

$h_1(s)$	= number of unallocated bananas
$h_2(s)$	= number of unenvied minions
$h_3(s)$	= number of envied minions
$h_4(s)$	= number of envy cycles

DOMINATING HEURISTICS (AY19/20 SEM 2 FINAL EXAM Q3)

- With reference to the heuristics in Question 2, select all of the following that are true.

(a) $\min(h_1, h_2)$ dominates h_1

(b) $\max(h_2, h_3)$ dominates one heuristic in the set $\{h_1, h_4\}$

(c) $\max(h_1, h_4)$ dominates both h_1 and h_4

(d) $\min(h_1, h_3)$ does not dominate h_1

Properties max/min admissible/inadmissible can give you an idea of the answer. Though some times it's not straightforward to use abstract reasoning to answer. Need contextual explanation.

$h_1(s)$	= number of unallocated bananas
$h_2(s)$	= number of unenvied minions
$h_3(s)$	= number of envied minions
$h_4(s)$	= number of envy cycles

DOMINATING HEURISTICS (AY19/20 SEM 2 FINAL EXAM Q3)

- Let's analyse each option for dominance.

(a) $\min(h_1, h_2)$ dominates h_1

- Consider a scenario where all minions are envied but there is still unallocated bananas, then $h_1(s) > 0$ and $h_2(s) = 0$. So $\min(h_1, h_2) = 0 \mathbf{<} h_1(s) > 0$. **False**.

Inadmissible heuristic may not necessarily dominate an admissible heuristic at **a** particular node.

$h_1(s)$	=	number of unallocated bananas
$h_2(s)$	=	number of unenvied minions
$h_3(s)$	=	number of envied minions
$h_4(s)$	=	number of envy cycles

DOMINATING HEURISTICS (AY19/20 SEM 2 FINAL EXAM Q3)

- Let's analyse each option for dominance.

  (b) $\max(h_2, h_3)$ dominates one heuristic in the set $\{h_1, h_4\}$

- Consider a scenario where there are way more bananas than minions (we can adversarially say so), and half of the minions (n/2) are envied, we allocate sufficient number of bananas such that there's still > n/2 bananas left, and create > n/2 envy cycles among the minions. Then $h_2 = h_3 = \max(h_2, h_3) = n/2$, but $h_1 > n/2$ and $h_4 > n/2$. **False**.

$h_1(s)$	= number of unallocated bananas
$h_2(s)$	= number of unenvied minions
$h_3(s)$	= number of envied minions
$h_4(s)$	= number of envy cycles

DOMINATING HEURISTICS (AY19/20 SEM 2 FINAL EXAM Q3)

- Let's analyse each option for dominance.

  (c) $\max(h_1, h_4)$ dominates both h_1 and h_4

- The maximum of two heuristics will obviously dominate both of them. **True**.

$h_1(s)$	= number of unallocated bananas
$h_2(s)$	= number of unenvied minions
$h_3(s)$	= number of envied minions
$h_4(s)$	= number of envy cycles

DOMINATING HEURISTICS (AY19/20 SEM 2 FINAL EXAM Q3)

- Let's analyse each option for dominance.

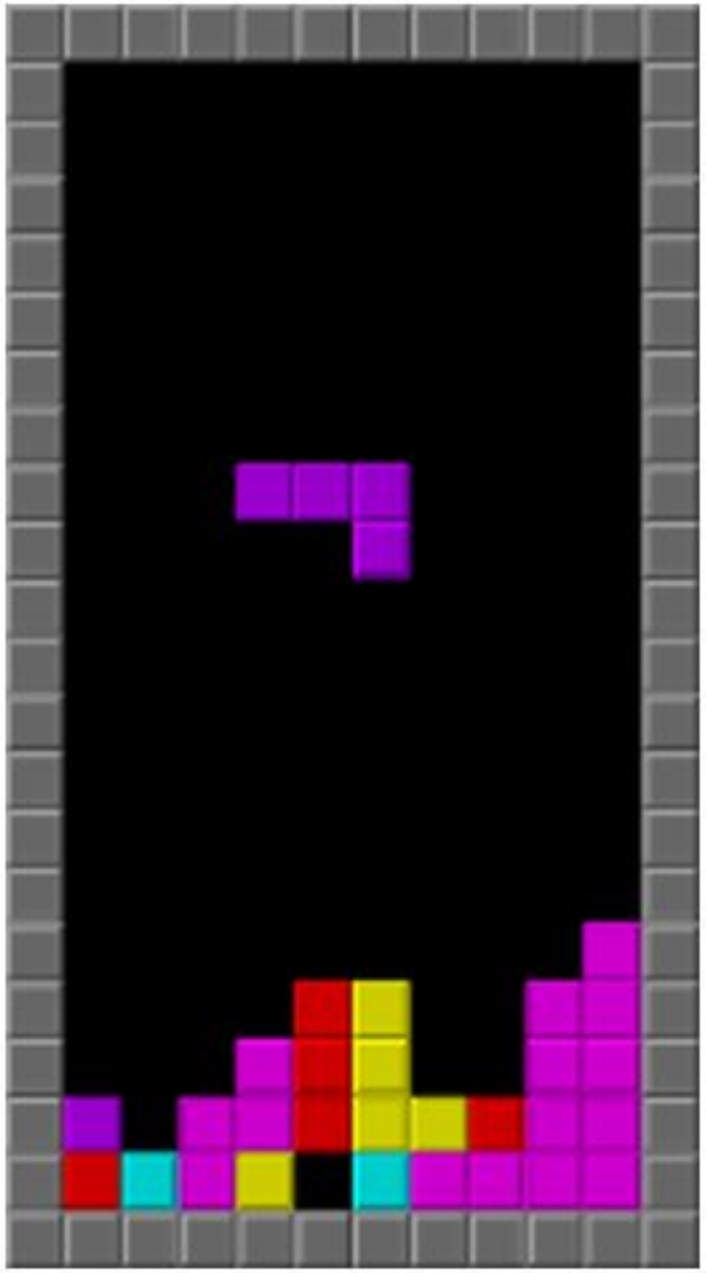
(d) $\min(h_1, h_3)$ does not dominate h_1

- Consider a scenario where no minions are envied, but not all bananas have been allocated, e.g. start state, then $h_1(s) > 0$ and $h_3(s) = 0$. $\min(h_1, h_3) = 0$ **<** h_1. **True**.

$h_1(s)$	= number of unallocated bananas
$h_2(s)$	= number of unenvied minions
$h_3(s)$	= number of envied minions
$h_4(s)$	= number of envy cycles

INFORMED SEARCH: TETRIS

- Tetris (fill-the-board variant) is a tile-matching game in which pieces of different geometric forms, called tetriminos, descend from the top of the field. During this descent, the player can move the pieces laterally (move left, move right) and rotate (rotate right, rotate left) them until they touch the bottom of the field or land on a piece that had been placed before it. The player can neither slow down the falling pieces nor stop them.

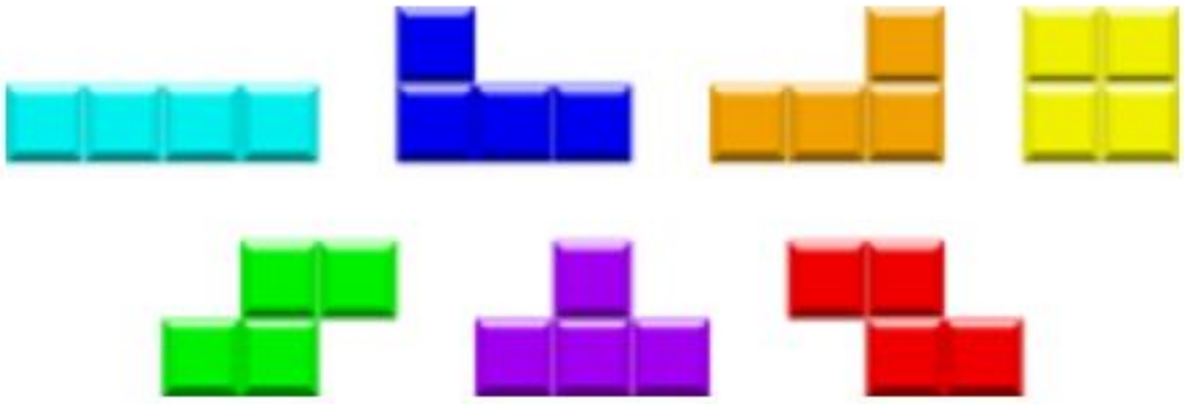


INFORMED SEARCH: TETRIS

- The objective of the game is to **configure the pieces to fill a board completely without any blocked gaps**.
- A **gap** is defined as an empty cell on the board.
- A row (or column) is **complete** if there are no gaps in that row (or column respectively).
- A gap is called a **blocked gap** if for the corresponding column where the gap belongs to, there exists an occupied cell somewhere above that gap.

INFORMED SEARCH: TETRIS

- There are 7 kinds of tetriminos. Assume that we start with a **fixed number of tetriminos**, N (comprising some of each kind), and **all are required to be used to fill the board** (i.e. there exists a way to place all these tetriminos such that the board is filled).



INFORMED SEARCH: TETRIS

- **States**: different partially filled tetris fields with a tetrimino that is about to be placed in the field next (but not placed yet);
- **Initial state**: an empty field with a starting tetrimino;
- **Action**: a sequence of lateral movement(s) and/or rotation(s) of a tetrimino (assume the player has enough time to shift the tetrimino to an intended configuration before descent);
- **Transition model**: takes in a state, applies the sequence of actions on the tetrimino that enters the field, and outputs a state where the tetrimino of that specified configuration descended onto the field;
- **Goal state**: a completely filled board where there are no gaps (and every tetrimino fits perfectly); you may assume there exists such a goal state.
- Transition **cost** is 1.

INFORMED SEARCH: TETRIS

- **Select all of the heuristics that are admissible. If you feel that none are admissible, select only the option "None of the options are admissible". For each option, briefly, but clearly explain, why it is admissible/inadmissible.**

- $h_1(n)$ = number of unfielded tetriminos
- $h_2(n)$ = number of gaps
- $h_3(n)$ = number of complete rows
- $h_4(n)$ = number of blocked gaps

INFORMED SEARCH: TETRIS

- **Select all of the heuristics that are admissible. If you feel that none are admissible, select only the option "None of the options are admissible". For each option, briefly, but clearly explain, why it is admissible/inadmissible.**

- $h_1(n)$ = number of unfielded tetriminos — **Admissible**. This is the minimum number of steps required to even get to the goal, obviously admissible.
- $h_2(n)$ = number of gaps — **Inadmissible**. Consider a field with horizontal gap line of size 4. Then cyan tetrimino can fill it in one move (cost to goal is 1), but heuristic returns 4, overestimate.
- $h_3(n)$ = number of complete rows — **Inadmissible**. Consider the goal state - it has non-zero complete rows.
- $h_4(n)$ = number of blocked gaps — **Both Admissible** and **Inadmissible** acceptable, with correct justification

INFORMED SEARCH: TETRIS

- **With reference to the heuristics in the first question of this section, select all of the following that are True. Upon making your choice, pick any one of the options, clearly indicate which option you picked and justify your answer for that option.**

- $\max(h_1, h_2)$ is admissible
- $\min(h_2, h_3)$ is admissible
- $\max(h_3, h_4)$ is inadmissible
- $\min(h_1, h_4)$ is admissible

$h_1(n)$ = number of unfielded tetriminos

$h_2(n)$ = number of gaps

$h_3(n)$ = number of complete rows

$h_4(n)$ = number of blocked gaps

INFORMED SEARCH: TETRIS

- **With reference to the heuristics in the first question of this section, select all of the following that are True. Upon making your choice, pick any one of the options, clearly indicate which option you picked and justify your answer for that option.**

- $\max(h_1, h_2)$ is admissible — **False**. Maximum of an admissible heuristic and inadmissible is inadmissible since we would pick the higher inadmissible value for some states
- $\min(h_2, h_3)$ is admissible — **False**. Minimum of two inadmissible heuristics could still inadmissible, since the lower of the two values for a given state may still be inadmissible
- $\max(h_3, h_4)$ is inadmissible — **True**. Maximum of inadmissible heuristic and another heuristic is inadmissible, we would pick the higher inadmissible value for some states
- $\min(h_1, h_4)$ is admissible — **True**. Minimum of two heuristics when one of them is admissible, is admissible.

INFORMED SEARCH: TETRIS

- **With reference to the heuristics in the first question of this section, select all of the following that are True. Upon making your choice, pick any <u>one</u> of the options, clearly indicate which option you picked and justify your answer for that option.**

- h_1 dominates h_2
- h_2 dominates h_4
- h_3 does not dominate h_2
- h_4 does not dominate $h_2/2$

$h_1(n)$ = number of unfielded tetriminos
$h_2(n)$ = number of gaps
$h_3(n)$ = number of complete rows
$h_4(n)$ = number of blocked gaps

INFORMED SEARCH: TETRIS

- **With reference to the heuristics in the first question of this section, select all of the following that are True. Upon making your choice, pick any one of the options, clearly indicate which option you picked and justify your answer for that option.**

- h_1 dominates h_2 — **False**. Admissible heuristics cannot dominate inadmissible heuristics
- h_2 dominates h_4 — **True**. Number of blocked gaps has to be ≤ number of gaps, since blocked gaps is a subset of gaps
- h_3 does not dominate h_2 — **True**. Consider the initial state, then the number of complete rows (h_3) is 0, whereas number of gaps (h_2) is a lot (non-zero)
- h_4 does not dominate $h_2/2$ — **True**. Consider the initial state where there are 0 blocked gaps but many gaps.

CS3243 INTRODUCTION TO ARTIFICIAL INTELLIGENCE

UNINFORMED SEARCH

nicholas.teh@u.nus.edu

UNINFORMED SEARCH: TRACING

- You must be familiar with tracing the three uninformed search algorithms taught in class:
  - BFS (unoptimized: goal test after popping from frontier; optimized: goal test before pushing into frontier)
  - DFS (unoptimized: stack (there's some reversal in order), optimized: recursion)
  - UCS (any alphabetical tie-breaking is done within and popping PQ, not when insert! // TREE SEARCH VS GRAPH SEARCH)
- **Explored** means you actually visited it (POP), not just see it (see means put in frontier)

Any other clarifications you'd like to make?

CS3243 INTRODUCTION TO ARTIFICIAL INTELLIGENCE

INFORMED SEARCH

nicholas.teh@u.nus.edu

INFORMED SEARCH: TRUE/FALSE

- True/False

Given that a King in the game of International Chess may move one square in any of the eight possible directions (refer to Figure below for a graphical representation - note that in the Figure, K represents the King piece), the Manhattan Distance heuristic would be admissible for the problem of moving the King from an initial square, s, to a goal square, t.

$\nwarrow$	$\uparrow$	$\nearrow$
$\leftarrow$	K	$\rightarrow$
$\swarrow$	$\downarrow$	$\searrow$

INFORMED SEARCH: TRUE/FALSE

- True/False

Given that a King in the game of International Chess may move one square in any of the eight possible directions (refer to Figure below for a graphical representation - note that in the Figure, K represents the King piece), the Manhattan Distance heuristic would be admissible for the problem of moving the King from an initial square, s, to a goal square, t.

$\nwarrow$	$\uparrow$	$\nearrow$
$\leftarrow$	K	$\rightarrow$
$\swarrow$	$\downarrow$	$\searrow$

False. The heuristic returns 2 even if the goal state is achievable in 1 step by diagonal move, so is an over-estimate.

INFORMED SEARCH: TRUE/FALSE

- True/False

If $h_1(s)$ and $h_2(s)$ are two admissible A* heuristics, let $h_3(s) = w.h_1(s)+(1-w).h_2(s)$ for any real number w in [0, 1]. Then $h_3(s)$ must also be admissible.

INFORMED SEARCH: TRUE/FALSE

- True/False

If $h_1(s)$ and $h_2(s)$ are two admissible A* heuristics, let $h_3(s) = w.h_1(s)+(1-w).h_2(s)$ for any real number w in [0, 1]. Then $h_3(s)$ must also be admissible.

True. (recall tutorial question)

For all n,
Argue in one line $h_3 = wh_1 + (1\text{-}w)h_2 \leq \max(h_1, h_2) \leq h^*$

Slightly insufficient to say the following:
If h_1 dominates h_2,
$h_3 = wh_1 + (1\text{-}w)h_2 \leq h_1 \leq h^*$

If h_2 dominates h_1,
$h_3 = wh_1 + (1\text{-}w)h_2 \leq h_2 \leq h^*$

INFORMED SEARCH: TRUE/FALSE

- True/False

Given an admissible heuristic, h_1, and an inadmissible heuristic, h_2, a third heuristic, $h_3 = h_1 \times h_2$, is always inadmissible.

INFORMED SEARCH: TRUE/FALSE

- True/False

Given an admissible heuristic, h_1, and an inadmissible heuristic, h_2, a third heuristic, $h_3 = h_1 \times h_2$, is always inadmissible.

False.

Consider the case where h_1 is always 0.

- Let G be a graph defined by:
  Vertices V ={s, a, b, t}
  Edges E = {(s, a, 2), (s, b, 4), (s, t, 10), (a, t, 6), (b, t, 3)}
- **Define ALL possible admissible heuristics for G, with source s and goal t.**

- Let G be a graph defined by:
  Vertices V ={s, a, b, t}
  Edges E = {(s, a, 2), (s, b, 4), (s, t, 10), (a, t, 6), (b, t, 3)}
- **Define ALL possible admissible heuristics for G, with source s and goal t.**
- Any heuristic function h that returns values satisfying:

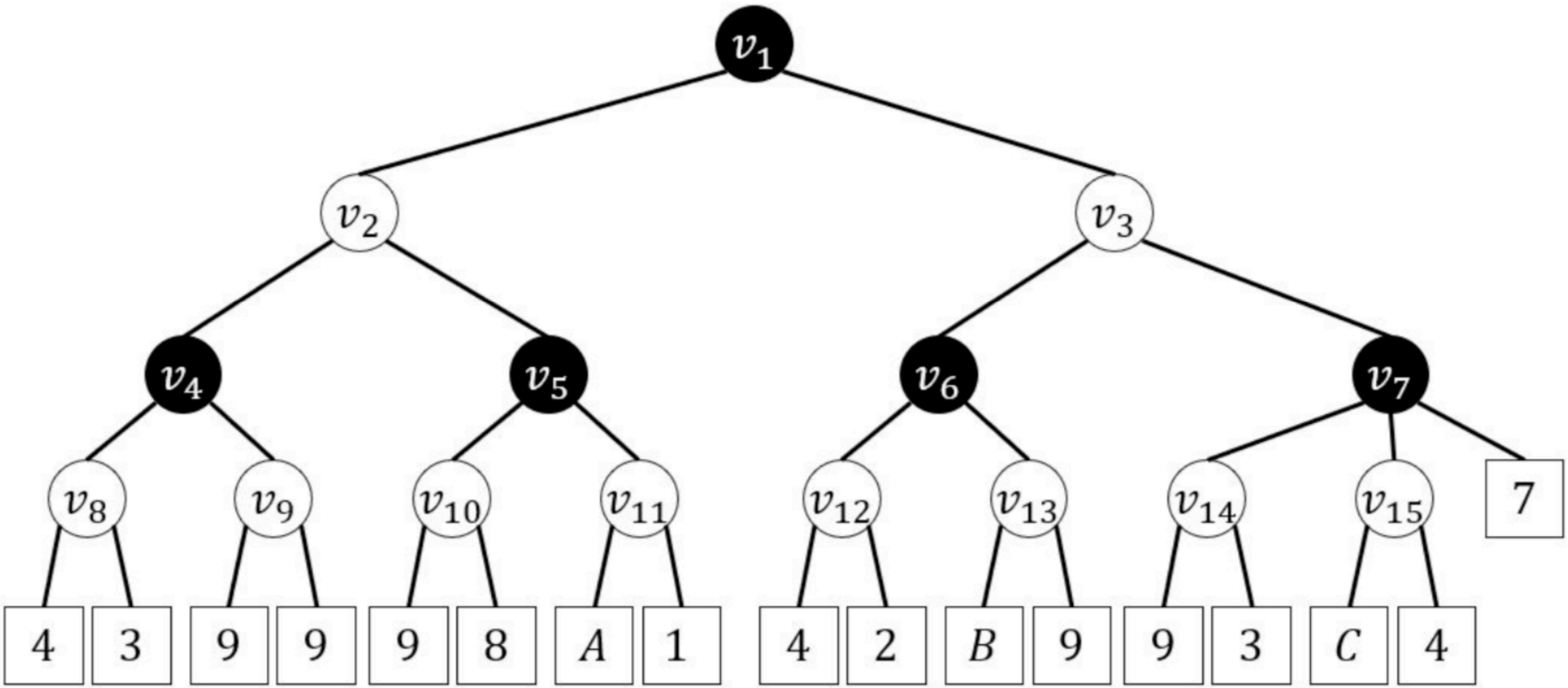
$$0 \leq h(s) \leq 7$$
$$0 \leq h(v_1) \leq 6$$
$$0 \leq h(v_2) \leq 3$$
$$h(t) = 0$$

CS3243 INTRODUCTION TO ARTIFICIAL INTELLIGENCE

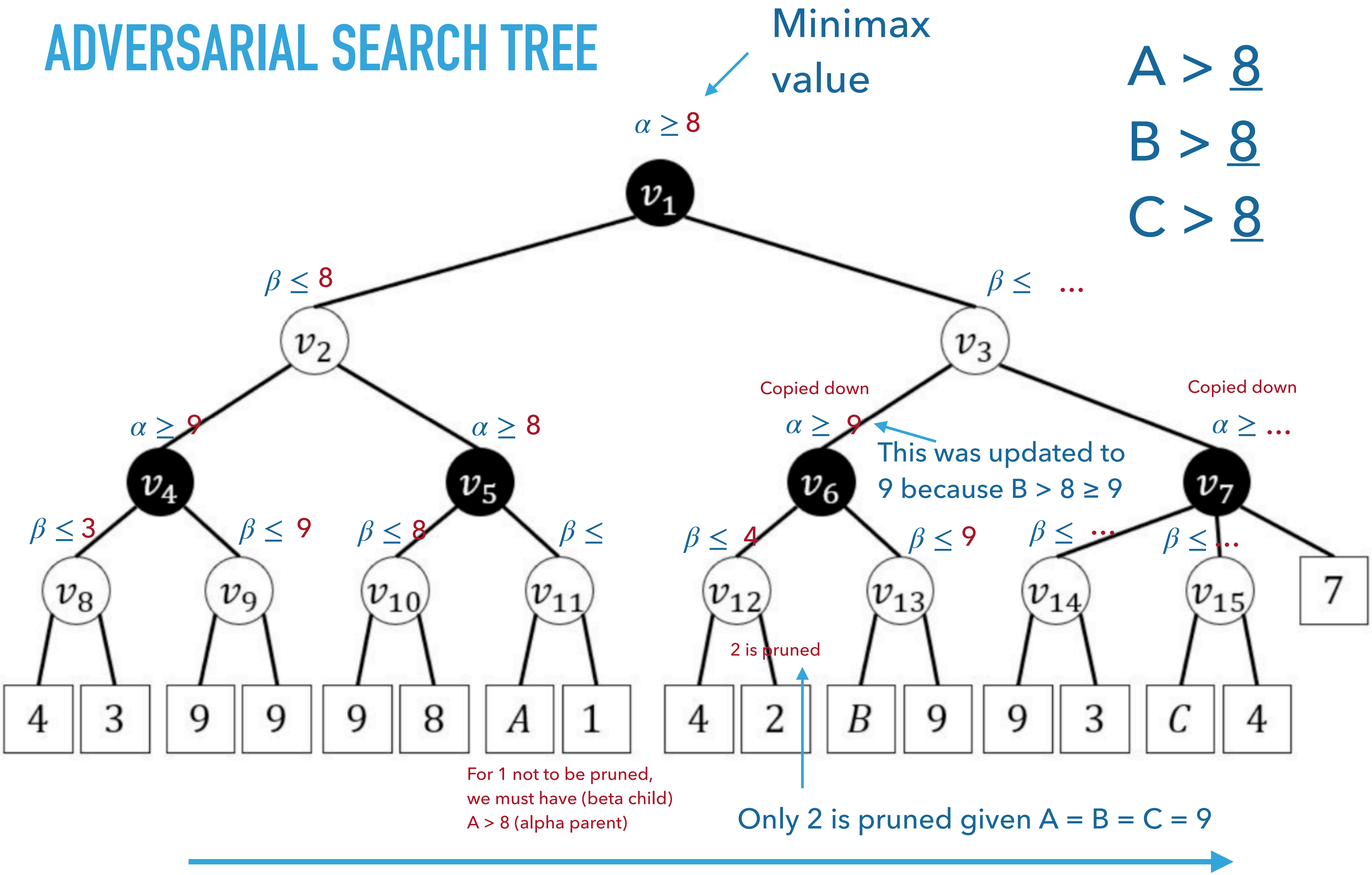
ADVERSARIAL SEARCH

nicholas.teh@u.nus.edu

ADVERSARIAL SEARCH TREE



ADVERSARIAL SEARCH TREE



IN SUMMARY…

- **Uninformed Search**
  - Modelling a search problem
  - Uninformed search algorithms: know exactly how to trace
  - Some uninformed search properties
- **Informed Search**
  - Designing and proving admissible heuristics, including its dominance relationship
  - Knowing what is admissibility and consistency
- **Adversarial Search**
  - Game tree
  - Minimax algorithm