CS3243 Tutorial 3

Eric Han

Sep 6, 2022

Annoucements

Important admin

1. Attendance Marking on telegram; Same as last week. Check-in if you are here!
2. Show me your bonus to collect your snacks
3. Assignment 1 results are out on turnitin, check scores and comments:

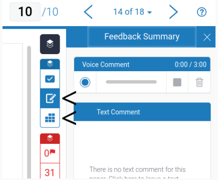

Figure 1: Turnitin, comments on Luminus; 2 places you can find the comments.

Tidbits from tutorials

1. Heuristics should be thought of as functions, so combining heuristics are like combining functions – $h(n) = \max(h_1(n), h_2(n))$, also $h = \max(h_1, h_2)$

Annoucements (New)

Tidbits from tutorials

1. **[From Prof]** In L3, slide 42 it says - 'dominance requires admissibility and that is applied in CS3243' - we should apply this in general. This would mean Tut3, Q2c is missing a statement - 'Here we do not require admissibility for dominance'.

Annoucement from Prof.

Error in the lecture slides.

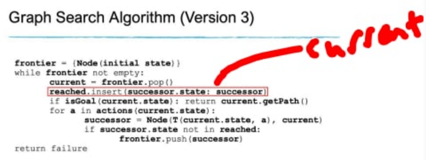

Figure 2: Graph Algo

I recommend you to see implementations at https://github.com/aimacode/aima-python.

Previously from T02, Q5

Recap

- What is an admissible/consistent heuristic?

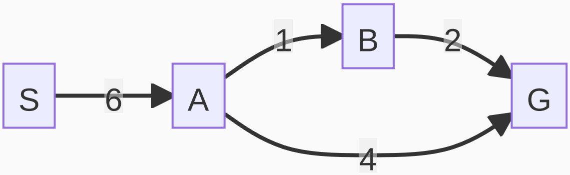

Figure 3: Illustration

a. In the search problem below, we have listed 5 heuristics. Indicate whether each **heuristic** is **admissible** and/or **consistent** in the table below.
b. Write out the order of the nodes that are explored by the **A* Search** algorithm. Assume a **graph search** (v3) implementation that utilises heuristic h_4.

c. Which heuristic would you use? Explain why.

d. Prove or disprove the following statement:

The heuristic $h(n) = max\{h_3(n), h_5(n)\}$ is admissible.

Answer 5a

s	S	A	B	G	Admissible	Consistent
$h_1(s)$	0	0	0	0	T	T
$h_2(s)$	8	1	1	0	T	F
$h_3(s)$	9	3	2	0	T	T
$h_4(s)$	6	3	1	0	T	F
$h_5(s)$	8	4	2	0	F	F
$\max\{h_3(s), h_5(s)\}$	9	4	2	0	F	F

Answer 5b

$S - A - B - G$

Answer 5c

h_3, as $h_3 = h^*$ is the optimal heuristic.

Answer 5d

$4 = h(A) > h^*(A) = 3$

Question 1

Given a two-dimensional, rectangular, $n \times m$ grid of coloured squares.

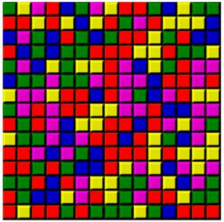

Figure 4: Q1 eg.

- **State Space**: $n \times m$ where each cell value is color, $[0, c]$.
- **Initial State**: Random matrix where each cell $[1, c]$.
- **Final State**: Zero matrix.
- **Action**: Delete a group of the same color.
- **Transition Model**: Replace group with 0, any cell which has 0 below will move down until zeros are ontop, columns move left if zero column.

Design an admissible heuristic for this puzzle game. Your heuristic may not be $h(s) = 0$ for all states s, the optimal heuristic, or a linear combination/simple function thereof. You may assume that the tile layout in the initial grid is solvable i.e. there is some path to a goal state. **You must prove that your heuristic is admissible.**

Recap

- How is a heuristic *useful*?
- What is the h^* heuristic?
- What is a potential downfall of choosing an optimal heuristic?

Question 1 - Answer crowd-sourced from TG4/TG5

$h(n) =$

1. No. of colors - See next slide for discussion.
2. No. of groups [Inadmissible]
    - BBGGBB $>$ BBBB $> \varnothing$ - $h(n) = 3 > h^*(n) = 2$
3. No. of singletons [Inadmissible]
    - GBRRBG $>$ GBBG $>$ GG $> \varnothing$ - $h(n) = 4 > h^*(n) = 3$
4. min(No. of groups, No. of singletons) [Inadmissible]

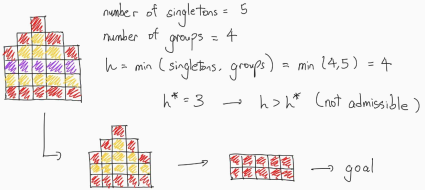

Figure 5: min(No. of groups, No. of singletons)

Question 1 - Answer

$h(s) =$ number of colors remaining.

Proof of Admissibility

$h^*(s)$ is the number of optimal moves from s to goal.

1. Each group contains exactly 1 colour.
2. So for each remaining colour, there can be 1 or more groups.
3. There are 2 possibilities from a particular move:
    - Reduce the number of colors - Number of groups (maximally) reduced by 1.
    - Do not reduce the number of colors.

Hence, $h(s)$ is less than or equals to at least the minimum moves on the board $\implies h(s) \leq h^*(s)$.

Question 2

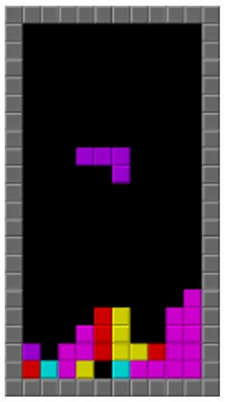

Figure 6: Tetriminos

Each turn, when a new piece appears to be placed, the player must select the location and orientation before it falls.

- **State Space**: Matrix where each cell is 0 empty or 1 filled.
- **Initial State**: Empty matrix.
- **Final State**: Filled matrix, with all 1s.
- **Action**: Orientation and column position.
- **Transition Model**: Transition cost - 1, add 1 to the position of the dropped tetriminos.

Recap

- What is an admissible heuristic? $h(N) \leq h^*(N)$
- What are some properties of admissiblity?

Question 2a

Admissible, or inadmissible?

- $h_1(n) =$ number of unfielded tetriminos
- $h_2(n) =$ number of gaps
- $h_3(n) =$ number of incomplete rows
- $h_4(n) =$ number of blocked gaps
- None of the options are admissible.

Question 2a

Admissible, or inadmissible?

- $h_1(n) =$ number of unfielded tetriminos
- $h_2(n) =$ number of gaps
- $h_3(n) =$ number of incomplete rows
- $h_4(n) =$ number of blocked gaps
- None of the options are admissible.

Answer

- $h_1(n)$ - Admissible, optimal steps must be $\geq$ than the number of teriminos.
- $h_2(n)$ - Inadmissible, yellow square block fills a gap of 4 but optimal cost is 1.
- $h_3(n)$ - Inadmissible, cyan vertical block fills a gap of 4 rows but optimal cost is 1.
- $h_4(n)$ - Admissible (Assuming that blocked gaps cannot be filled):
  - Whenever there is a blocked gap, h^* is infinite.
  - Whenever there is none, $h_4(.) = 0 \leq h^*(.)$

Question 2b

h	Admissiblity
h_1	Admissible
h_2	Inadmissible
h_3	Inadmissible
h_4	Admissible

Select all of the following that are True:

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

Question 2b - Answer

h	Admissiblity
h_1	Admissible
h_2	Inadmissible
h_3	Inadmissible
h_4	Admissible

Select all of the following that are True:

- **[False]** $\max(h_1, h_2)$ is admissible*
- **[False]** $\min(h_2, h_3)$ is admissible
  - L shaped piece gap, $h_2 = 4, h_3 = 2$ but $h^* = 1$
- **[True]** $\max(h_3, h_4)$ is inadmissible*
- **[True]** $\min(h_1, h_4)$ is admissible

Question 2b - Answer

h	Admissiblity
h_1	Admissible
h_2	Inadmissible
h_3	Inadmissible
h_4	Admissible

Select all of the following that are True:

- **[False]** $\max(h_1, h_2)$ is admissible*
- **[False]** $\min(h_2, h_3)$ is admissible
  - L shaped piece gap, $h_2 = 4, h_3 = 2$ but $h^* = 1$
- **[True]** $\max(h_3, h_4)$ is inadmissible*
- **[True]** $\min(h_1, h_4)$ is admissible

Analysis, cases, which are always true for general case?

- max(Admissible, Admissible) - Admissible
- max(Admissible, Inadmissible)
- max(Inadmissible, Inadmissible)
- min(Admissible, Admissible)
- min(Admissible, Inadmissible) - Admissible
- min(Inadmissible, Inadmissible)

Question 2c

h	Admissiblity
h_1	Admissible
h_2	Inadmissible
h_3	Inadmissible
h_4	Admissible

Recall the hueristics:

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

Question

Select all of the following that are True:

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

Recap

- What is dominates? $h_2(n) \geq h_1(n)$ for every state n, then h_2 dominates h_1.

Question 2c - Answer

h	Admissiblity
h_1	Admissible
h_2	Inadmissible
h_3	Inadmissible
h_4	Admissible

Recall the hueristics:

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

Select all of the following that are True:

- **[False]** h_1 dominates h_2 - Admissible cannot dominate inadmissible.
- **[True]** h_2 dominates h_4 - Gaps $\geq$ Blocked Gaps $\implies h_2 \geq h_4$
- **[True]** h_3 does not dominate h_2 - Consider inital state: $0 = h_3(s_0) < h_2(s_0) \neq 0$
- **[True]** h_4 does not dominate $h_2/2$ - Consider inital state: $0 = h_4(s_0) < h_2(s_0)/2 \neq 0$

Question 3

Assignment Question; we will go through this question next week.

For next question, we will assume **all hueristics are admissible**.

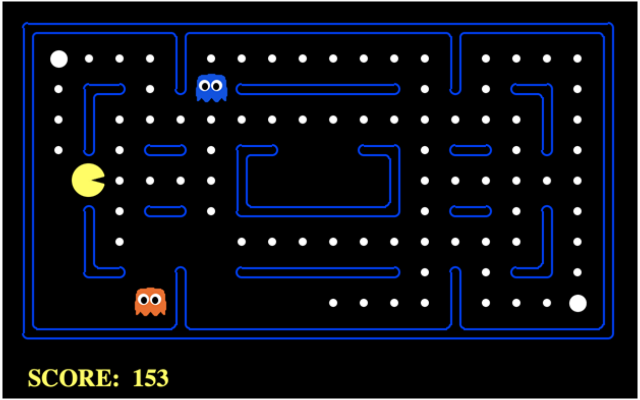

Figure 7: Pac-Man Example

Question 4

- h_1 : Number of pellets left at any point in time.
- h_2 : Number of pellets left + the minimum among all Manhattan distances from each remaining pellet to the current position of Pac-Man.
- h_3 : The Maximum among all Manhattan distances from each remaining pellet to the current position of Pac-Man.
- h_4 : The average over all Euclidean distances from each remaining pellet to the current position of Pac-Man.

Recap

- What is euclidean distance?
- What is manhattan distance?

Answer

- We pick h_1 to analyse dominate relationship:
    - h_2 **dominates** h_1; Trival to see $h_2(.) \geq h_1(.)$.
    - h_3 - **No RS**, h_4 - **No RS**; see example
        - Case where 1 pellet left, pacman is 10 units away: $h_1 = 1 < h_3 = h_4 = 10$
        - Case where 4 pellets left, pacman is 1 units away: $h_1 = 4 > h_3 = h_4 = 1$
- We pick h_2 to analyse dominate relationship.
    - h_3 - **No RS**, h_4 - **No RS**; see example
        - Case where 2 pellet left, pacman is $1, 9$ units away: $h_2 = 2 < h_3 = 9, h_4 = 5$
        - Case where 4 pellets left, pacman is 1 units away: $h_2 = 5 > h_3 = h_4 = 1$
- We pick h_3 with h_4 to analyse dominate relationship.
    - h_3 **dominates** h_4; We consider h_3' which is average of all manhattan distance. Then, h_3 (max man.) dominates h_3' (avg man.) dominates h_4 (avg. eucl.).

Bonus Question - Work for Snack

To help you further your understanding, not compulsory. Our task today is to just install Anaconda, OpenAI Gym and play around with a PacMan environment.

- Anaconda is a very popular tool for AI/ML.
- OpenAI Gym is a very good tool for RL/Env.

Tasks

1. Fork the repository https://github.com/eric-vader/CS3243-2223s1-bonus
2. Install Anaconda - https://www.anaconda.com/products/distribution
3. Install the conda environment: `cond env create -f tutorial3.yml`
4. Activate the environment: `conda activate tutorial3`
5. Run and see PacMan in action: `python3 tutorial3.py`
6. Fill in anywhere `TODO` in the code as appropriate.