

Artificial Intelligence Tutorial 1 - Answers

1) Suppose you are searching for a girl's name written using only the letters D, N and A. You have the letters ordered alphabetically (A, D, N) and you start writing down possibilities:

A, D, N, AA, AD, AN, DA, DD, DN, NA, ND, NN, ...

- 1a) How many strings of four or fewer letters are there where the letters are D, N or A?
- 1b) In the above possibilities, are you searching in a depth first or breadth first way?
- 1c) What are the next three possible names you would write down?
- 1d) How many possibilities will you write down before getting to the name ANNA? [difficult]
- 1e) Are you guaranteed to find all girls names with letters D, N and A in this manner?

1a) For strings of length 4, there are three choices for the first letter (D, N and A), three choices for the second, three choices for the third and three choices for the fourth letter. Hence, there are $3*3*3*3 = 81$ possible four-letter names with D, N and A in. Similarly, there are $3*3*3 = 27$ possibilities for three-letter names, $3*3 = 9$ possible two-letter names and 3 possible one letter names (there is a famous AI researcher called J Strother Moore – the J is his entire Christian name, which makes it difficult because he always gets forms sent back asking for his name, not his initial). Adding all the possibilities up, we get: $81 + 27 + 9 + 3 = 120$ possible names.

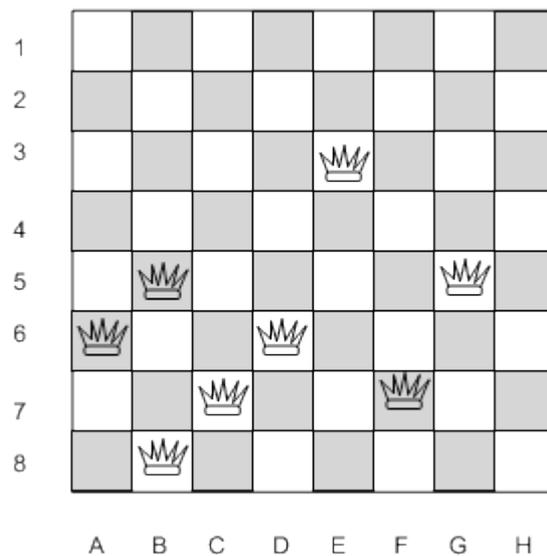
1b) This is a breadth first search (BFS), because all possibilities for two letter names are exhausted before moving on to three letter names, and so on. It's also a *uniform* search, because three nodes are expanded each time – one for A, one for D and one for N.

1c) The next names would be: AAA, AAD, AAN, because you expand the AA node (using the graph analogy for search). Similarly, the agenda items (AA, add A); (AA, add D); (AA, add N) would be at the top of the agenda by this stage.

1d) All three-or-fewer-letter names will have been exhausted by this stage, so there will be $3 + 9 + 27 = 39$ possibilities written down before ANNA. All four-letter possibilities starting with AAA, AAD, AAN, ADA, ADD, ADN, ANA and AND (no pun intended) will have also been written down. There are three four-letter possibilities starting AAA, namely AAAA, AAAD and AAAN, similarly three for AAD and so on. So, there will be $3+3+3+3+3+3+3+3 = 8*3 = 24$ possibilities with four letters in already written down. There are no more before ANNA, hence the total is $39 + 24 = 63$. Hence ANNA is found after roughly half the search (of 120 nodes). [Extra question for the really keen among you: how many would be written down if the search was a Russell & Norvig style depth first search? How many for an IDS?]

1e) Yes, all names will eventually be enumerated, because breadth first search is *complete*, i.e., guaranteed to cover all possibilities. It's worth bearing in mind that if you wanted to find a long name, say, up to 10 or 12 letters, then the Universe might end before you had exhausted your BFS.

2. Suppose you are using a gradient descent search with the evaluation function “number of queens which are threatened by another queen” in the 8-queens problem. Suppose that you’re at the following board state:



- 2a) What is the current score for the evaluation function?
- 2b) Write down some of the best moves from this position (queens can be moved anywhere).
- 2c) Give an example of an illegal move (in a gradient descent search).
- 2d) What do you do if there are no legal moves?

2a) In its current state, every queen is challenged by another, so the evaluation function scores 8.

2b) We want to move one queen to any other place on the board where it is not threatened by any other queen, as this will reduce the number of threatened queens by at least one. If we can do so in such a way that it frees up another queen, then this will be a bonus. These are some possibilities:

G5 – G4 (score after move is 6, because the queen at E3 is no longer threatened)

C7 – C2 (score after move is 7)

Etc.

2c) Gradient descent searches require the evaluation function to strictly decrease after every move. Hence, if we moved, say, B8 to C8, it would not improve the number of queens which are threatened, and this would be an illegal move.

2d) Give up on this situation. That is, start again by randomly putting your 8 queens on the board. This is called random-restart. No backtracking is done (so the method is very good on memory efficiency, as only the current board state has to be kept in memory).

3. A solved 8-puzzle game looks like this:

1	2	3
8		4
7	6	5

Given a puzzle state like the one below, where the numbers are in the wrong places, we want to search for a series of moves which ends in the solution above. A path cost could be calculated as the number of moves required.

1	2	3
8	6	
7	5	4

3a) Write down a heuristic function, h , for this game (remember that this *estimates* the path cost from a particular state to the solution).

3b) What is the value of h for the above board state?

3c) In a greedy search, what move would be chosen next? (i.e., move the 3, the 4 or the 6 into the gap?)

3d) Is this heuristic *admissible* [see notes]?

3e) [optional] Can you suggest a better heuristic measure? (Think about New York City....)

3a) The heuristic function must be designed to estimate the cost to the solution from the current state. One simple possibility is to calculate the number of pieces which are not in their final state (i.e., the position they will occupy in the solution state).

3b) In the case above, the numbers 4, 5 and 6 are out of place. Hence, this heuristic function would score 3 for the above board state.

3c) Moving the 6 into the gap would produce a value of h the same (namely three), because the 6 has not been moved into its final position. Moving the 3 would be disastrous, as the heuristic would now score four, as the 3 gets moved *out of*, rather than into its final position. Moving the four up would reduce the value of h to just two (the 5 and 6 are still out of place). Therefore, a greedy searching agent would move the four up.

3d) An admissible heuristic measure is one which underestimates the path cost to the solution. The advantage to these is that we know an A* search would be optimal (i.e., it would find the shortest path to the solution state), if the heuristic was admissible. In our search problem, the path cost is the number of moves required to get to the solution. Each out of place number must be moved at least once to get it into its final position, but in reality it is likely to be moved more than once. Hence the number of out-of-place pieces underestimates the path cost, and hence is admissible.

3e) Another possibility is to think about how the moves will be carried out in order to get the numbers into the correct positions. The 'Manhattan' distance for a number is calculated as the minimum number of moves (horizontal and vertical only) which will take it from its current position to its final position. We can add up all the Manhattan distances, and this will provide another heuristic measure. (This heuristic is so called because of the grid system of roads in Manhattan). See Russell and Norvig for more details of this heuristic measure.

4a) Translate this sentence into first order predicate logic:

“An elephant is happy if all its children can fly”

Use these predicates:

happy(x) is read as “x is happy”

fly(x) is read as “x can fly”

child(x,y) is read as “x is a child of y”

elephant(x) is read as “x is an elephant”

An elephant is happy if all its children can fly. This is hardly realistic (when are elephants ever happy?). However, the “all its children can fly” part needs to be read as: “if it’s a child of the elephant, then it can fly”. This will be inside another if-then statement: “if all the children of an elephant can fly, then the elephant is happy”. But how are we going to say this about all elephants? We will need a third implication to say that if something is an elephant, then, if all its children can fly, it is happy.

It is translated thus:

$$\forall X (\text{elephant}(X) \rightarrow ((\forall Y (\text{child}(Y, X) \rightarrow \text{fly}(Y)) \rightarrow \text{happy}(X))).$$

Here is an interesting wrong answer to this question:

$$\forall X (\text{elephant}(X) \& \text{happy}(X) \rightarrow \forall Y (\text{child}(Y, X) \rightarrow \text{fly}(Y))).$$

This is wrong, because it says that if an elephant is happy, then all its children can fly. This is not what we wanted to say. Another correct answer would be:

$$\forall X (\text{elephant}(X) \rightarrow ((\neg(\exists Y (\text{child}(Y, X) \& \neg\text{fly}(Y))) \rightarrow \text{happy}(X))).$$

Here, we’ve substituted the sentence: “there is no child which cannot fly” for the sentence “all its children can fly”.

4b) Translate this sentence into first order predicate logic:

“The rainfall in every Latin American country is at least 17cm a year”

Invent your own predicates for this sentence, and explain how they are meant to be read by an agent.

We first need to think of the predicates, functions and constants (ground variables) which will be in our logical representation. The only constant here is 17. There is an `is_latin_american_country` predicate there, and also an ‘at least’ relationship, which we can model with an `at_least` predicate. Rainfall per year is perhaps best thought of as a function, which takes in a country and outputs a number. There is obviously universal quantification here, also, because we are talking about *every* Latin American country. So, the quantified variable will appear in the `is_latin_american_country` predicate. Finally, there seems to be an implication going on here, if we think of the sentence as: “If X is a Latin American country, then its rainfall is at least 17cm a year”. Hence this is an answer:

$$\forall X (\text{is_latin_american_country}(X) \rightarrow \text{at_least}(\text{rainfall}(X), 17)).$$

We see that we’ve used the rainfall function inside the `at_least` predicate, which indicates that it is the output from the rainfall function which is more than 17cm.

4c) Identify the predicate names and the predicate arities, the constants, the quantifiers, the variables and the connectives in your translated sentence for part 1b.

The predicate names are: (i) `is_latin_american_country` (ii) `at_least` and (iii) `rainfall`. Note that this last one is still a predicate name, even though we are using it as a function. The arities of the predicates are both 1, and the function has one input – the name of the country – and one output, the amount of rainfall that the country gets.

There is only one constant here: 17.

There is only one quantifier here: \forall (forall), and only one quantified variable: X.

There is only one connective: \rightarrow