Τεχνητή Νοημοσύνη

1η άσκηση

Δεμερτζόγλου Ευστράτιος | TH20580

Table of Contents

Περιγραφή του προβλήματος

Το πρόβλημα των ιεραποστόλων και των κανιβάλων είναι ένα κλασικό πρόβλημα τεχνητής νοημοσύνης που αφορά τη μετάβαση ομάδας ανθρώπων από τη μία όχθη ενός ποταμού στην άλλη, ακολουθώντας συγκεκριμένους περιορισμούς:

- Υπάρχουν **3 ιεραπόστολοι** και **3 κανίβαλοι** στη μία όχθη.
- Διαθέτουν μια **βάρκα** που μπορεί να μεταφέρει το πολύ **2 άτομα**.
- Σε οποιαδήποτε όχθη ή στη βάρκα, οι ιεραπόστολοι **δεν μπορούν να είναι λιγότεροι από τους κανίβαλους**, αλλιώς οι κανίβαλοι θα τους σκοτώσουν.
- Στόχος είναι να μεταφερθούν **όλοι οι ιεραπόστολοι και οι κανίβαλοι με ασφάλεια στην απέναντι όχθη**.

Προσέγγιση Επίλυσης

Η λύση βασίζεται στην **αναζήτηση στο χώρο καταστάσεων**. Κάθε κατάσταση αναπαρίσταται ως **state(MissionariesLeft, CannibalsLeft, BoatSide, MissionariesRight, CannibalsRight)**, όπου:

- MissionariesLeft: Αριθμός ιεραποστόλων στην αριστερή όχθη.
- CannibalsLeft: Αριθμός κανιβάλων στην αριστερή όχθη.
- BoatSide: Θέση της βάρκας (left ή right).
- MissionariesRight: Αριθμός ιεραποστόλων στη δεξιά όχθη.
- CannibalsRight: Αριθμός κανιβάλων στη δεξιά όχθη.

Το πρόβλημα λύνεται με **αναζήτηση βάθους (DFS)**, η οποία αναζητά μονοπάτι από την αρχική κατάσταση προς την τελική, ακολουθώντας έγκυρες μεταβάσεις μεταξύ καταστάσεων.

Επεξήση Κώδικα:

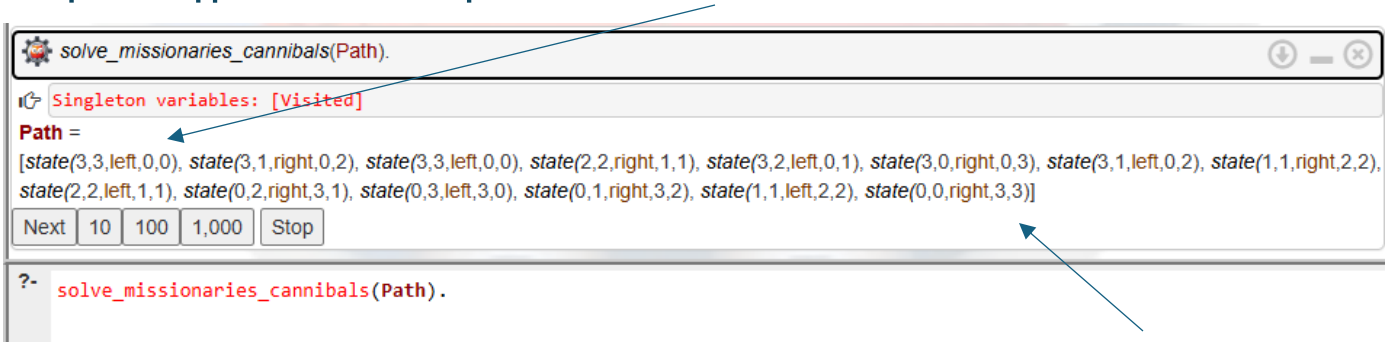
Ο κώδικας αποτελείται από τα παρακάτω τμήματα:

- **Ορισμός αρχικής και τελικής κατάστασης**.
- **Συνάρτηση ασφαλείας** που ελέγχει αν μια κατάσταση είναι έγκυρη.
- **Διαθέσιμες μετακινήσεις** (συνδυασμοί ανθρώπων που μπορεί να μεταφέρει η βάρκα).
- **Κανόνες μετάβασης** μεταξύ καταστάσεων.
- **Αλγόριθμος αναζήτησης DFS** για την εύρεση της λύσης.

Δεν παραθέτω τα ακριβή σημεία του κώδικα καθώς είναι επαρκώς τεκμηριωμένα με τον κατάλληλο σχολιασμό

Παράδειγμα εκτέλεσης

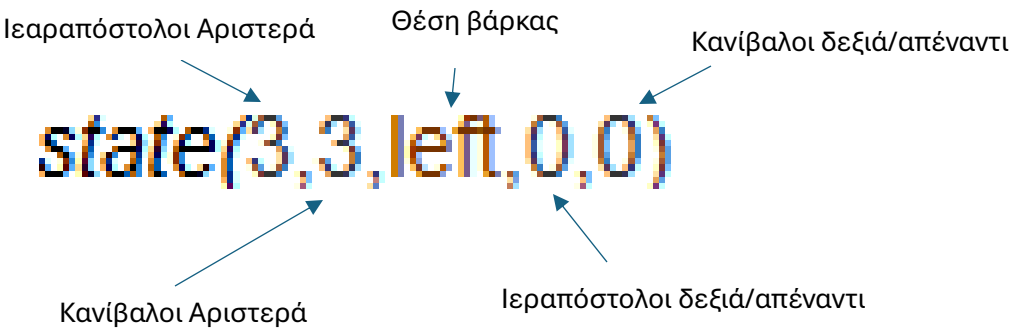
Αρχική κατάσταση



Τελική κατάσταση

Τι σημαίνει το παραπάνω παράδειγμα;

Στο 1° state έχουμε 3 ιεραποστόλους και 3 κανίβαλους στα αριστερά, η βάρκα βρίσκεται αριστερά και δεν είναι κανένας στην απέναντι όχθη.



Συνεπώς θέλουμε στην τελική κατάσταση να έχουμε 0 Ιεραποστόλους και 0 κανίβαλους στην αριστερή όχθη , 3 Ιεραποστόλους και 3 κανίβαλους στην δεξιά/απέναντι όχθη και φυσικά η βάρκα να βρίσκεται στην δεξιά/απέναντι όχθη, όπως δείχνει και το τελευταίο state του προγράμματος:

state(0,0,right,3,3)

Κώδικας που εκτελέστηκε

```
% Missionaries and Cannibals Problem in Prolog

% State Representation: state(MissionariesLeft, CannibalsLeft, BoatSide, MissionariesRight, CannibalsRight)
% BoatSide: left or right

% Initial state: 3 missionaries, 3 cannibals on the left, boat on the left
initial_state(state(3,3,left,0,0)).

% Goal state: all missionaries and cannibals on the right
goal_state(state(0,0,right,3,3)).

% Safe state check: missionaries should never be outnumbered by cannibals
safe(state(ML, CL, _, MR, CR)) :-
    (ML >= CL ; ML = 0),
    (MR >= CR ; MR = 0).

% Possible boat moves: (Missionaries, Cannibals) combinations
move(2,0). % Two missionaries
move(0,2). % Two cannibals
move(1,1). % One missionary, one cannibal
move(1,0). % One missionary
move(0,1). % One cannibal

% Transition between states
transition(state(ML, CL, left, MR, CR), state(ML2, CL2, right, MR2, CR2)) :-
    move(M, C),
```

```
    ML >= M, CL >= C,
    ML2 is ML - M, CL2 is CL - C,
    MR2 is MR + M, CR2 is CR + C,
    safe(state(ML2, CL2, right, MR2, CR2)).

transition(state(ML, CL, right, MR, CR), state(ML2, CL2, left, MR2, CR2)) :-
    move(M, C),
    MR >= M, CR >= C,
    MR2 is MR - M, CR2 is CR - C,
    ML2 is ML + M, CL2 is CL + C,
    safe(state(ML2, CL2, left, MR2, CR2)).

% Depth-first search for solution
solve(State, Path) :-
    dfs(State, [], Path).

dfs(State, Visited, [State]) :-
    goal_state(State).

dfs(State, Visited, [State | Path]) :-
    transition(State, NextState),
    \+ member(NextState, Visited),
    dfs(NextState, [NextState | Visited], Path).

% Find a solution
solve_missionaries_cannibals(Path) :-
    initial_state(StartState),
    solve(StartState, Path).
```