

Ερώτημα 1:

Δεν έχει κάτι ιδιαίτερο, ακολουθούμε τα βήματα της εκφώνησης

Ερώτημα 2:

Στην `AtLeastOne`, επιστρέφοντας μία λίστα με διαζεύξεις λεκτικών, ένα να είναι αληθές πιάνει το τουλάχιστον 1. Στην `AtMostOne` Επιστρέφουμε συζεύξεις αντίθετων λεκτικών με όλους τους συνδυασμούς, ώστε αν βγει ένα ζευγάρι αληθές να σημαίνει ότι 2 λεκτικά είναι αληθές, επομένως δεν ισχύει το `AtMostOne`. Στην `ExactlyOne` παίρνουμε την τομή του το πολύ 1 και τουλάχιστον 1

Ερώτημα 3:

Ακολουθούμε εκφώνηση πάλι

Ερώτημα 4:

Γεμίζουμε την KB με την αρχική θέση του πακμαν, με την έκφραση που επιτρέπει τον πακμαν για τις επόμενες 50 χρονικές στιγμές να βρίσκεται ακριβώς σε μία θέση από τις θέσεις τις οποίες δεν είναι τοίχος, επιπλέον προσθέτουμε στην KB την έκφραση ότι ο πακμαν για τις επόμενες 50 χρονικές στιγμές μπορεί να κάνει ακριβώς μία νόμιμη κίνηση. Επίσης προσθέτουμε την έκφραση για τις 49 χρονικές στιγμές (εκτός του 0) που επιστρέφει η συνάρτηση `pacmanSuccessorAxiomSingle` που φτιάξαμε στο προηγούμενο ερώτημα. Αν βρεθεί μοντέλο σε αυτές τις 50 χρονικές στιγμές που να πηγαίνει τον πακμαν σε `goal state` επιστρέφουμε τα `actions` του μοντέλου που βρήκαμε.

Ερώτημα 5:

Ίδια λογική με το παραπάνω ερώτημα, αλλάζοντας το `goal state` σε μία κατάσταση κατά την οποία ο πακμαν έχει φάει όλες τις κουκίδες και προσθέτοντας επιπλέον στην βάση γνώσης μας τις εκφράσεις του αν η κουκίδα υπάρχει ή δεν υπάρχει την επόμενη χρονική στιγμή αναλόγως με την θέση του πακμαν, αν την έχει φάει ή δεν την έχει φάει δηλαδή.