Project 4 - Τεχνητή Νοημοσύνη

Παναγιώτα Γύφτου , A.M.: 1115201900318

Φεβρουάριος 2023

Περίληψη

Θέμα εργασίας:

1. Ερμηνείες και Ικανοποίηση Προτάσεων
2. Λογική Πρώτης Τάξης
3. Συμπερασμός

Readme προγραμματιστικών προβλημάτων *Pacman*.

Προσοχή: Στις περισσότερες ερωτήσεις που εκτελείται ψευδοκώδικας της εκφώνησης έχω για ευκολία πάνω από κάθε μεταφρασμένο βήμα σε γλώσσα *python* το αντίστοιχο του ψευδοκώδικα.

---

*Question*1

sentence1()

Ορίζω τρία αντικείμενα $Expr$, ένα για κάθε σταθερό σύμβολο. Για το πρώτο σύνολο παραστάσεων χρησιμοποιούνται οι σταθερές A, B, C.

- $(\mathbf{A} \wedge \mathbf{B})$

  Για την $(\mathbf{A} \wedge \mathbf{B})$, χρησιμοποίησα την μέθοδο $disjoin$ με ορίσματα A, B, όπου επιστρέφει την ζητούμενη διαζευκτική πρόταση και κατόπιν το αποτέλεσμα αντιγράφεται στην μεταβλητή $A\_\_OR\_\_B$

- $(\neg\mathbf{A} \Leftrightarrow (\neg\mathbf{B} \vee \mathbf{C}))$

  Αρχικά βρίσκω τις αρνήσεις των A και B. Έπειτα λόγω της προτεραιότητας των παρενθέσεων, βρίσκω μεμονωμένα την $(\neg B \vee C)$, πάλι μέσω της $disjoin$. Αφού έχουμε αναπαραστήσει τις προτάσεις των δύων πλευρών της αμφίδρομης υποθετικής αναπαριστούμε την πρόταση.

- $(\neg\mathbf{A} \vee \neg\mathbf{B} \vee \mathbf{C})$

  Οι αρνήσεις έχουν αναπαρασταθεί προηγουμένως και απλώς καλούμε την μέθοδο $disjoin$, ώστε να μας επιστρέψει την διαζευκτική αλυσίδα $\neg A \vee \neg B \vee C$.

Τέλος επιστρέφουμε την αλυσίδα σύζευξης των τριών προτάσεων που αναπαριστήσαμε παραπάνω μέσω της μεθόδου $conjoin$.

sentence2()

Η διαδικασία αναπαράστασης των προτάσεων είναι παρόμοια με αυτή της συνάρτησης $sentence1()$.
Τέλος επιστρέφουμε την αλυσίδα σύζευξης των τεσσάρων προτάσεων που αναπαριστήσαμε μέσω της μεθόδου $conjoin$.

sentence3()

Σε αυτή την συνάρτηση χρησιμοποιούμε το $Pacphysics\ symbol\ \ PropSymbolExpr$, μέσω του οποίου δημιουργούμε τις σταθερές της εκφώνησης:

$$PacmanAlive\_0 \quad PacmanAlive\_1 \quad PacmanBorn\_0 \quad PacmanKilled\_0$$

Στην συνέχεια αναπαριστούμε τις προτάσεις που μας δίνονται σε φυσική γλώσσα σε προτάσεις της προτασιακής λογικής.

1. $Pacman\ is\ alive\ at\ time\ 1\ if\ and\ only\ if\ he\ was\ alive\ at\ time\ 0\ and\ he\ was\ not\ killed\ at\ time\ 0\ or\ he\ was\ not\ alive\ at\ time\ 0\ and\ he\ was\ born\ at\ time\ 0.$

2. $At\ time\ 0,\ Pacman\ cannot\ both\ be\ alive\ and\ be\ born.$

3. $Pacman\ is\ born\ at\ time\ 0$

Τέλος επιστρέφουμε την σύζευξη αυτών των προτάσεων που αναπαριστήσαμε.

findModelCheck():

Η συνάρτηση αυτή επιστρέφει το $cmbs$ του $findModel()$ με όρισμα μια πρόταση η οποία αποτελείται από μια σταθερά.Έστω ότι έχουμε την σταθερά α, τότε επιστρέφει ότι το μοντέλο της πρότασης $\varphi = \alpha$ είναι ικανοποιήσιμο. Η επιστρεφόμενη τιμή είναι ένα $dictionary$ με το μοντέλο και την αντίστοιχη $boolean$ τιμή του, η οποία είναι αληθής.

entails():

Η $entails()$ ελέγχει μέσω της $findModel()$ αν μια υπόθεση ικανοποιεί ένα συμπέρασμα.Αρκεί να περάσουμε την πρόταση (υπόθεση $\ \wedge \neg$ συμπέρασμα), στην $findModel()$, όπου εάν γυρίσει με αρνητική απάντηση η $findModel()$, τότε έχει αποδειχθεί ότι η υπόθεση ικανοποιεί την πρόταση του συμπεράσματος και επιστρέφει αληθές, διαφορετικά ψευδές.

plTrueInverse():

Επιστρέφει αλήθεια εάν και μόνο εάν η μη αντίστροφη δήλωση είναι αληθής. Για την επέκταση αυτής της συνάρτησης βοηθηθηκά από τα σχόλια της περιγραφής της συνάρτησης στον κώδικα, όπου αναφέρει σαν χρήσιμιμη συνάρτηση την pl_true.

Question2

atLeastOne(**literals**)

Επιστρέφει μια έκφραση CNF που είναι αληθής αν υπάρχει τουλάχιστον μία αληθής έκφραση στη λίστα που δόθηκε ως όρισμα στην συνάρτηση.Αυτό επιτυγχάνεται μέσω της κλήσης $disjoin$ όπου μας επιστρέφει την διάζευξη των στοιχείων της $literals$.

atMostOne(**literals**)

Επιστρέφει μια έκφραση CNF που είναι αληθής αν υπάρχει το πολύ μία αληθής έκφραση στη λίστα που δόθηκε ως όρισμα στην συνάρτηση.Αυτό επιτυγχάνεται μέσω της γεννήτριας συνάρτησης συνδυασμών ρυθμισμένη να παράγει συνδιασμούς των δύων $itertools.combinations$ λαμβάνοντας την λίστα $literals$, ζευγάρια των δύων στοιχείων τα οποία είναι διαφορετικά μεταξύ τους.Στην συνέχεια κάθε ζεύγος αναπαρίσταται ως μια διαζευξη, έτσι ώστε να βρεθεί το πολύ ένα.Τέλος ενώνουμε τα διαζευγμένα ζευγάρια με μία σύζευξη, όπου στην ουσία οποίοδήποτε διαζευγμένο ζεύγος είναι ψευδές τότε δεν υπάρχει κανένα στοιχείο αληθές, αυτό ήταν και το ζητούμενο.Το πολύ 1 σημαίνει στην λογική γλώσσα κανένα αληθές.

exactlyOne(literals)

Επιστρέφει μια έκφραση CNF που είναι αληθής αν υπάρχει ακριβώς μία αληθής έκφραση στη λίστα που δόθηκε ως όρισμα στην συνάρτηση.Αυτό επιτυγχάνεται με την ύπαρξη τουλάχιστον ενός αληθούς λεκτικού και την ύπαρξη το πολύ ένα λεκτικού.Στην ουσία είναι η υλοποίση των δύων παραπάνβ συναρτήσεων όπου τα αποτελέσματα τους ενώνονται με σύζευξη.Δηλαδή:

$$atLeastOne \ \& \ atMostOne$$

Question3

pacmanSuccessorAxiomSingle()

Επιστρέφουμε μία έκφραση όπου περιέχει όλες τις απαραίτητες προϋποθέσεις ώστε να βρίσκεται ο $Pacman$ την δεδομένη στιγμή t στην θέση x, y.Αυτό επιτυγχάνεται με μία αμφίδρομη υποθετική πρόταση ανάμεσα στην εξετάζουσα θέση και τις προϋποθέσεις για να βρίσκεται εκεί.

pacphysicsAxioms()

Δημιουργούμε τα αξιώματα του $Pacman$, όπου μας δίνονται στην εκφώνηση και τα μεταφράζουμε κατάλληλα σε λογικές προτάσεις κατανοητές στον κόσμο που ζει ο $Pacman$.

checkLocationSatisfiability()

Σε αυτή την συνάρτηση ελέγχουμε τα μοντέλα κινήσεων του $Pacman$ ποια είναι ικανοποιήσιμα και ποια όχι.Κατά την διάρκεια του έλεγχου των μοντέλων προσθέτουμε πληροφορίες στην KB. Δημιουργούμε την βάση γνώσης του $Pacman$, όπου εισάγουμε στην KB τις απαραίτητες πληροφορίες που μας ζητούνται στην εκφώνηση. Στην ουσία μεταφράζουμε τον ψευδοκώδικα της εκφώνησης με την βοήθεια του σχολιασμού της συνάρτησης σε $python$.

Question4

positionLogicPlan()

Ο $Pacman$ ψάχνει να βρει το μονοπάτι που θα τον οδηγήσει στον στόχο.Με βάση τα τρέχοντα δεδομένα του στην βάση γνώσης του ο $Pacman$ βρίσκει την επόμενη κίνηση που θα το φέρει πιο κοντά στον στόχο.Με κάθε βήμα η KB του $Pacman$ ανανεώνεται με βάση τα νέα δεδομένα που δημιουργούνται με την εκτέλεση μιας κίνησης του.Και εδώ έχουν μεταφραστεί τα βήματα του ψευδοκώδικα της εκφώνησης.

*Question*5

Η ερώτηση αυτή έχει παρόμοια μορφή με αυτή της ερώτησης 4.

foodLogicPlan()

Αλλαγές/Προσθήκες:

- Αρχικοποίηση $Food[x, y]\_t$
- Αλλαγή του ισχυρισμού του στόχου, όπου πρέπει να είναι αληθής, αν έχουν καταναλωθεί όλα τα τρόφιμα.Αυτό συμβαίνει όταν όλα τα τρόφιμα είναι $False$
- Προσθήκη ενός αξιώματος διαδοχής τροφίμων:

Πρέπει η βάση να ανανεώνεται με τις εξής πληροφορίες:

Όταν ο $Pacman$ τρώει φαγητό στην θέση (x, y) σε μία δεδομένη στιγμή, έστω t, τότε στον χρόνο $t + 1$, στην θέση x, y δεν θα υπάρχει φαγητό.

Όταν ο $Pacman$ βρίσκεται στην θέση (x, y) σε μία δεδομένη στιγμή, έστω t, και την στιγμή εκείνη στην θέση $(x1, y1)$ υπάρχει φαγητό τότε στον χρόνο $t + 1$, στην θέση $(x1, y1)$ θα υπάρχει ακόμα το φαγητό, αφού δεν το έχει φάει.

*Question*6

localization()

Ο $Pacman$ ξεκινάει με έναν γνωστό χάρτη, αλλά άγνωστη θέση εκκίνησης.Και εδω θα κωδικοποιήσουμε τις προτάσεις που βοηθούν τον $Pacman$ να καθορίσει τις πιθανές τοποθεσίες που μπορεί να είναι σε κάθε χρονική στιγμή.Και εδώ έχουν μεταφραστεί τα βήματα του ψευδοκώδικα της εκφώνησης.

*Question*7

mapping()

Ο $Pacman$ έχει γνωστή θέση εκκίνησης, αλλά δεν ξέρει πού είναι οι τοίχοι.Και εδω θα κωδικοποιήσουμε τις προτάσεις που βοηθούν τον $Pacman$ να καθορίσει τις θέσεις των τοίχων.Και εδώ έχουν μεταφραστεί τα βήματα του ψευδοκώδικα της εκφώνησης.Παρόμοια ιδέα με μικρές αλλαγές.

*Question*8

slam()

Στο $SLAM$ ($Simultaneous\ Localization\ and\ Mapping$), ο $Pacman$ έχει γνωστή θέση εκκίνησης, αλλά δεν ξέρει πού βρίσκονται οι τοίχοι. Στο $SLAM$, ο $Pacman$ μπορεί να επιλέξει παράνομες ενέργειες, όπου θα αυξήσει την αβεβαιότητα της θέσης του με την πάροδο του χρόνου. Και εδώ έχουν μεταφραστεί τα βήματα του ψευδοκώδικα της εκφώνησης.Στην ουσία είναι συγχώνευση του ερωτήματος 6 και 7 με μικρές αλλάγές ως προς τις συναρτήσεις των αξιομάτων.