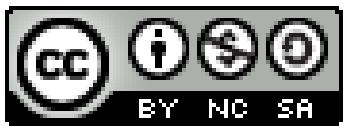
ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΙΚΑ
ΜΑΘΗΜΑΤΑ

Συστήματα Γνώσης

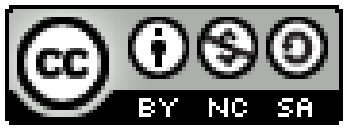
Πρακτικό Κομμάτι Μαθήματος
Πρόγραμμα Κίνησης Robot

Νίκος Βασιλειάδης, Αναπλ. Καθηγητής
Τμήμα Πληροφορικής



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

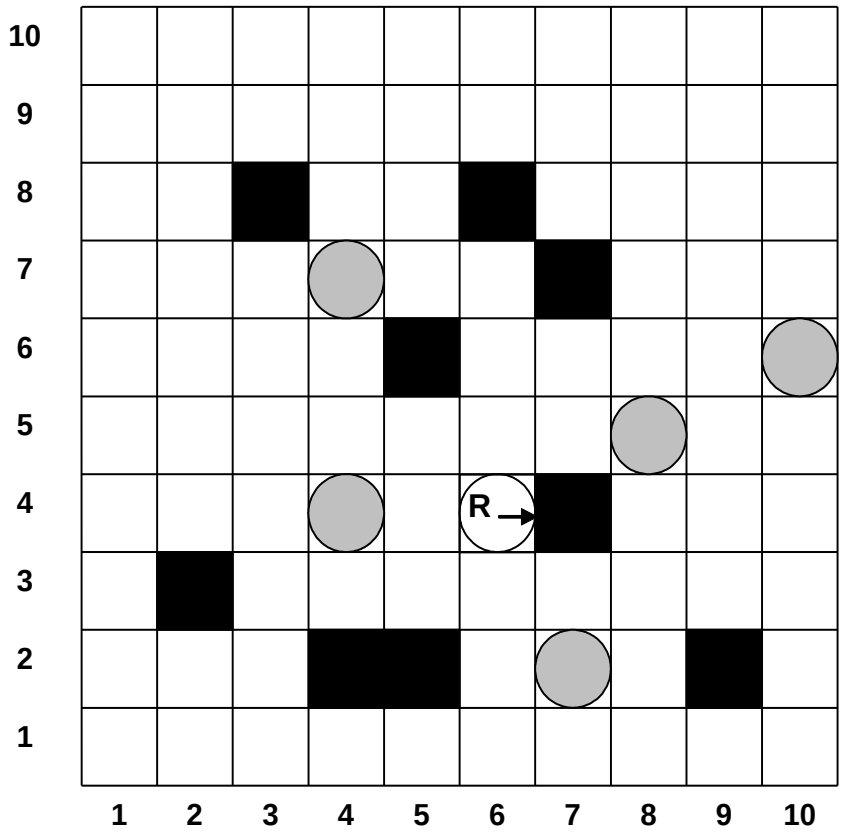


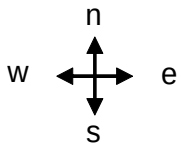
Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Πρόγραμμα Κίνησης Robot

Κίνηση Ρομπότ σε χώρο με εμπόδια






```
robot_at(6,4)
direction(e)
choice(w)
choice(s)
choice(n)
choice(e)
obstacle_at(7,4)
obstacle_at(6,8)
obstacle_at(7,7)
. . .
object_at(4,7)
. . .
```

Κίνηση Ρομπότ

- Υλοποίηση σε CLIPS του θεωρητικού παραδείγματος
- Διαφέρει σε μερικά σημεία
  - Οι στρατηγικές επίλυσης συγκρούσεων του CLIPS δε συμπίπτουν.
- Στο CLIPS υπάρχουν 7 προκαθορισμένες στρατηγικές οι οποίες **δεν μπορούν να συνυπάρχουν** την ίδια χρονική στιγμή,
  - Δεν μπορεί να υπάρξει ο συνδυασμός των στρατηγικών που απαιτεί το θεωρητικό παράδειγμα,
  - *Αποφυγή Επανάληψης*, *Επιλογή του πιο Ειδικού* και *Τυχαία Επιλογή*.

Στρατηγικές Επίλυσης

- Δεν μπορούν να συνυπάρξουν οι στρατηγικές της επιλογής του πιο ειδικού με την τυχαία επιλογή.
- Η τυχαία επιλογή είναι απαραίτητη για την επιλογή τυχαίας κατεύθυνσης όταν το ρομπότ πέφτει πάνω σε εμπόδια
- Η επιλογή του πιο ειδικού χρειάζεται για να δώσει προτεραιότητα στους κανόνες αποφυγής εμποδίων έναντι αυτών της κίνησης.
  - Υπάρχει εναλλακτική λύση μέσω της χρήσης salience
- Η στρατηγική που χρησιμοποιείται είναι η **random**.

Άλλες Διαφοροποιήσεις

- Ο κανόνας αρχικοποίησης που εισάγει τα δυναμικά γεγονότα
  - Η χρήση του δεν είναι απαραίτητη
- Η αποφυγή της εξόδου του ρομπότ από το πλέγμα
  - Ανάγεται σε αποφυγή εμποδίων
- Ο τερματισμός της εκτέλεσης όταν βρεθεί κάποιο αντικείμενο.

Στατικά Γεγονότα

```
(deffacts static-facts
  ;;; Εμπόδια
  (obstacle_at 4 2) (obstacle_at 5 2)
  (obstacle_at 9 2) (obstacle_at 2 3)
  (obstacle_at 7 4) (obstacle_at 5 6)
  (obstacle_at 7 7) (obstacle_at 3 8)
  (obstacle_at 6 8)
  ;;; Δυνατές κατευθύνσεις κίνησης
  (choice w)  (choice e)  (choice n)
  (choice s)
  ;;; Αντικείμενα
  (object_at 7 2)  (object_at 4 4)
  (object_at 8 5)  (object_at 10 6)
  (object_at 4 7))
```

Κανόνας αρχικοποίησης

```
;;; Αρχικά δυναμικά γεγονότα.
  Στρατηγική random.
(defrule begin
  (initial-fact)
=>
  (set-strategy random)
  (assert (robot_at 6 4))
  (assert (direction e))
)
```

Εντοπισμός Αντικειμένου

```
;;; Κανόνας εντοπισμού αντικειμένου
  και τερματισμού της εκτέλεσης
(defrule detect_object
  (robot_at ?x ?y)
  (object_at ?x ?y)
=>
  (printout t "object is found at
  position " ?x "-" ?y " !" crlf)
  (halt)
)
```

Μετακίνηση αριστερά-δεξιά

```
;;; Κανόνας μετακίνησης προς τα
  αριστερά
(defrule move_west
  ?f <- (robot_at ?x ?y)
  (direction w)
=>
  (retract ?f)
  (assert (robot_at (- ?x 1) ?y)))

;;; Κανόνας κίνησης προς τα δεξιά
(defrule move_east
  ?f <- (robot_at ?x ?y)
  (direction e)
=>
  (retract ?f)
  (assert (robot_at (+ ?x 1) ?y)))
```

Μετακίνηση πάνω-κάτω

```
;;; Κανόνας κίνησης ρομπότ προς τα πάνω
(defrule move_north
  ?f <- (robot_at ?x ?y)
  (direction n)
=>
  (retract ?f)
  (assert (robot_at ?x (+ ?y 1))))

;;; Κανόνας κίνησης ρομπότ προς τα κάτω
(defrule move_south
  ?f <- (robot_at ?x ?y)
  (direction s)
=>
  (retract ?f)
  (assert (robot_at ?x (- ?y 1))))
```

Αποφυγή Εμποδίου προς τα Κάτω

```
;;; Κανόνας αποφυγής εμποδίου κατά την
  κίνηση προς τα κάτω. Έχει μεγαλύτερη
  προτεραιότητα από τον αντίστοιχο κανόνα
  κίνησης. Εμπόδιο θεωρείται και η έξοδος
  από το πλέγμα

(defrule avoid_obstacle_south
  (declare (salience 1))
  (robot at ?x ?y)
  ?f <- (direction s)
  (or
     (obstacle at ?x =(- ?y 1))
     (test (= ?y 1))  )
  (choice ?nd)
=>
  (retract ?f)
  (assert (direction ?nd)))
```

Αποφυγή Εμποδίου προς τα Πάνω

```
(defrule avoid_obstacle_north
  (declare (salience 1))
  (robot at ?x ?y)
  ?f <- (direction n)
  (or
     (obstacle at ?x =(+ ?y 1))
     (test (= ?y 10))
  )
  (choice ?nd)
=>
  (retract ?f)
  (assert (direction ?nd))
)
```

Αποφυγή Εμποδίου προς τα Δεξιά

```
(defrule avoid_obstacle_east
  (declare (salience 1))
  (robot at ?x ?y)
  ?f <- (direction e)
  (or
     (obstacle at =(+ ?x 1) ?y)
     (test (= ?x 10))
  )
  (choice ?nd)
=>
  (retract ?f)
  (assert (direction ?nd))
)
```

Αποφυγή Εμποδίου προς τα Αριστερά

```
(defrule avoid_obstacle_west
  (declare (salience 1))
  (robot at ?x ?y)
  ?f <- (direction w)
  (or
     (obstacle at =(- ?x 1) ?y)
     (test (= ?x 1))
  )
  (choice ?nd)
=>
  (retract ?f)
  (assert (direction ?nd))
)
```

Εκτέλεση

```
CLIPS> (load "robot.clp")
TRUE
CLIPS> (reset)
CLIPS> (run)
==> f-19     (robot at 6 4)
==> f-20     (direction e)
<== f-20     (direction e)
==> f-21     (direction e)
<== f-21     (direction e)
==> f-22     (direction w)
<== f-19     (robot at 6 4)
==> f-23     (robot at 5 4)
<== f-23     (robot at 5 4)
==> f-24     (robot at 4 4)
object is found at position 4-4 !
CLIPS> (reset)
CLIPS> (run)
object is found at position 10-6 !
```

Παραλλαγές (1)

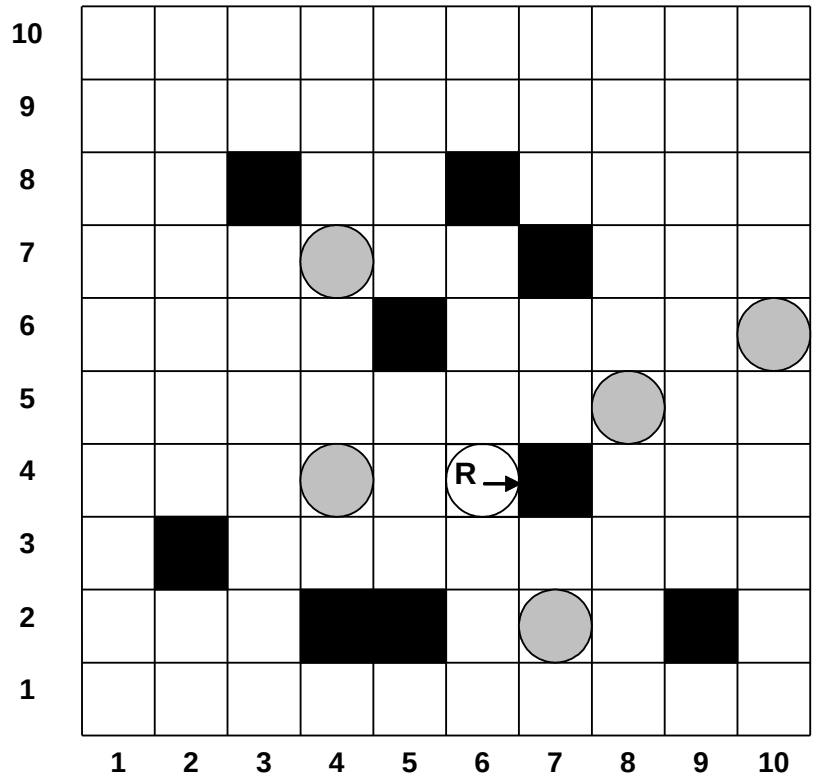
- Κάθε αντικείμενο που εντοπίζεται, να αφαιρείται από την λίστα γεγονότων
- Έτσι ο τερματισμός θα γίνεται όταν συλλεχθούν όλα τα αντικείμενα
- Το πρόγραμμα πέφτει σε ατέρμονα βρόχο γιατί όταν το robot μεταβεί στην περιφέρεια του λαβυρίνθου περιφέρεται για πάντα εκεί

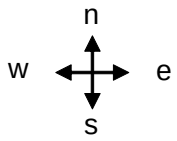
Παραλλαγή 1 - Εντοπισμός Αντικειμένου και Τερματισμός

```
; Κανόνας τερματισμού
(defrule stop
   (not (object_at ?x ?y))
=>
   (printout t crlf "I have found all objects!" crlf)
   (halt))

; Εντοπισμός αντικειμένου και αφαίρεσής του
(defrule detect_object
   (robot_at ?x ?y)
   ?f <- (object_at ?x ?y)
=>
   (printout t "object is found at position " ?x "-" ?y " !" crlf)
   (retract ?f))
```

Κίνηση Ρομπότ σε χώρο με εμπόδια






```
robot_at(6,4)
direction(e)
choice(w)
choice(s)
choice(n)
choice(e)
obstacle_at(7,4)
obstacle_at(6,8)
obstacle_at(7,7)
. . .
object_at(4,7)
. . .
```

Παραλλαγές (2)

- Να επιτρέπονται οι διαγώνιες κινήσεις

```
(deffacts static-facts
…
      ;;; Δυνατές κατευθύνσεις κίνησης
      (choice w)   (choice e)
      (choice n)   (choice s)
      (choice nw)   (choice ne)
      (choice sw)   (choice se)
…
)
```

Παραλλαγές (2)
Διαγώνιες κινήσεις

```
(defrule move_north_west
    ?f <- (robot_at ?x ?y)
    (direction nw)
=>
    (retract ?f)
    (assert (robot_at (- ?x 1) (+ ?y 1)))
)
```

Παραλλαγές (2)
Αποφυγή εμποδίων διαγώνια

```
(defrule avoid_obstacle_north_west
        (declare (salience 1))
        (robot_at ?x ?y)
        ?f <- (direction nw)
        (or
                (obstacle_at =(- ?x 1) =(+ ?y 1))
                (test (or (= ?x 1) (= ?y 10)))
        )
        (choice ?nd)
=>
        (retract ?f)
        (assert (direction ?nd))
)
```

Παραλλαγές (2)
Εκτέλεση

```
CLIPS> (run)
object is found at position 8-5 !
object is found at position 10-6 !
object is found at position 4-4 !
object is found at position 4-7 !
object is found at position 7-2 !

I have found all objects!
```

Παραλλαγές (3)

- Όταν η κίνηση γίνεται ευθεία (χωρίς εμπόδια) να υπάρχει κάποια πιθανότητα να στρίψει το ρομπότ

```
(defrule random_direction_change
        (declare (salience (random -10 0)))
        (robot_at ?x ?y)
        ?f <- (direction ?d)
        (choice ?nd)
=>
        (retract ?f)
        (assert (direction ?nd))
)
```

Παραλλαγές (3)
Τυχαία Στροφή

- Οι κανόνες κίνησης έχουν salience 0
- Ο κανόνας τυχαίας στροφής παίρνει at run-time τυχαίο salience στο διάστημα -10..0
  - Άρα έχει 1 στις 11 πιθανότητες να πάρει την τιμή 0
- Τότε έχει ίδια πιθανότητα με τους κανόνες κίνησης να επιλεχθεί
- Η στρατηγική random θα επιλέξει αν θα στρίψει ή αν θα συνεχίσει ευθεία
  - Πιθανότητα 3 στις 5 να στρίψει
    - 1 κανόνας **move_XXX**, 4 instances του κανόνα **random_direction_change** (1 προς την ίδια κατεύθυνση)
  - Συνολική πιθανότητα στροφής 3/55=5,45%

Παραλλαγές (3)
Εκτέλεση

```
CLIPS> (run)
object is found at position 7-2 !
object is found at position 4-7 !
object is found at position 4-4 !
object is found at position 10-6 !
object is found at position 8-5 !

I have found all objects!
```

ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΙΚΑ
ΜΑΘΗΜΑΤΑ

Τέλος Ενότητας

Επεξεργασία: Εμμανουήλ Ρήγας

Θεσσαλονίκη, 17/3/2014

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης