

# ΒΑΣΙΛΕΙΟΣ ΚΑΛΑΪΤΖΟΠΟΥΛΟΣ

AM 1066670

## 1ο ΕΤΟΣ ΦΟΙΤΗΣΗΣ

Langton's Ant: Υλοποίηση και πειραματισμοί.

### Περίληψη

Η παρακάτω ατομική εργασία αφορά το Langton's Ant μια universal διδιάστατη μηχανή Turing (UTM). Αυτές οι μηχανές, στον κλάδο της επιστήμης των υπολογιστών, μπορούν να προσομοιώσουν μια αυθαίρετη μηχανή Turing η οποία θα δέχεται μια αυθαίρετη είσοδο. Η αρχική ιδέα ήταν ένα «μυρμήγκι» το οποίο προχωράει πάνω σε λευκά και μαύρα «κουτάκια» και ανάλογα με το χρώμα πάνω στο οποίο βρίσκεται αποφασίζει την επόμενη του κίνηση. Βέβαια μετά η ιδέα έγινε πιο γενική αφού προστέθηκαν και άλλα χρώματα δημιουργώντας έτσι άπειρα patterns. Η εργασία περιέχει και εκτελέσιμους κώδικες που βασίζονται στις αρχές του Langton's Ant.

## I. ΕΙΣΑΓΩΓΗ

Το Langton's Ant εφευρέθηκε από τον Chris Langton το 1986 και αποδείχθηκε ότι ήταν μια universal Turing machine(UTM) το 2000. Είναι μια πάρα πολύ απλή ιδέα η οποία υπακούει σε δύο πολύ απλούς κανόνες:

- Όταν το κουτί στο οποίο βρίσκεται το μυρμήγκι είναι άσπρο στρίβει 90° δεξιά, κάνει το κουτί μαύρο και προχωράει ένα βήμα.
- Όταν το κουτί στο οποίο βρίσκεται το μυρμήγκι είναι μαύρο στρίβει 90° αριστερά, κάνει το κουτί άσπρο και προχωράει ένα βήμα.

Το Langton's Ant μπορεί να χαρακτηριστεί και ως cellular automaton (αντί για UTM) αν θεωρήσουμε ότι το πλέγμα πίσω από το μυρμήγκι είναι άσπρο ή μαύρο και το μυρμήγκι πρέπει να λάβει υπόψιν του τα οχτώ γειτονικά «κελιά» για το πώς θα μετακινηθεί. Αυτή η συμπεριφορά είναι πανομοιότυπη με αυτή που βλέπουμε στο Conway's Game of Life, 1970 του John Horton Conway το οποίο είναι ένας διδιάστατος κυτταρικός αυτοματισμός.

## II. LANGTON'S ANT

Εξαιτίας της απλότητας των κανόνων που υπακούει το μυρμήγκι, όταν βρίσκεται πάνω σε ένα τελείως λευκό πλέγμα, δηλαδή χωρίς κάποια κουτάκια να είναι ήδη χρωματισμένα, παρουσιάζει μια συγκεκριμένη συμπεριφορά ανάλογα με το πόσα βήματα έχει κάνει.

1. Στα πρώτα περίπου εκατό βήματα δημιουργεί πολύ απλά pattern τα οποία μπορεί να είναι και συμμετρικά.
2. Έπειτα επικρατεί απόλυτο χάος ένα τελείως ακανόνιστο μοτίβο εμφανίζεται πράγμα το οποίο συμβαίνει μέχρι το μυρμήγκι να φτάσει περίπου δέκα χιλιάδες βήματα.
3. Τέλος μετά από δέκα χιλιάδες βήματα αρχίζει και δημιουργείται το λεγόμενο «highway pattern» όπου επαναλαμβάνονται 104 βήματα επ' αόριστο.

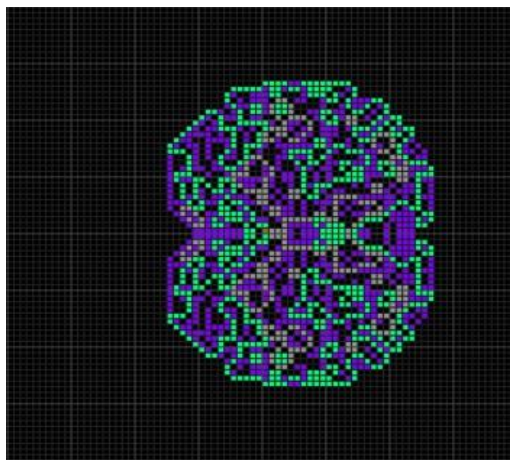
Έτσι δημιουργείται η εξής εικόνα:



**Εικόνα 1:** Langton's Ant μετά από 11,000 βήματα. Το κόκκινο pixel είναι η θέση του.

### III. ΕΠΕΚΤΑΣΕΙΣ ΤΗΣ ΙΔΕΑΣ

Μετά από την εφεύρεση αυτής της απλής ιδέας προέκυψαν διάφορες επεκτάσεις πάνω της. Η πρώτη και πιο απλή επέκταση της ιδέας προήλθε από τους Greg Turk και Jim Propp οι οποίοι σκέφτηκαν, πολύ απλά, να προσθέσουν κι άλλα χρώματα εκτός του μαύρου και του άσπρου τα οποία θα επηρέαζαν τις κινήσεις του μυρμηγκιού. Έτσι δόθηκαν οι ονομασίες «L» και «R» που υποδείκνυαν προς τα που θα στρίψει το μυρμήγκι. Η ιδέα του Langton πήρε την ονομασία «RL».

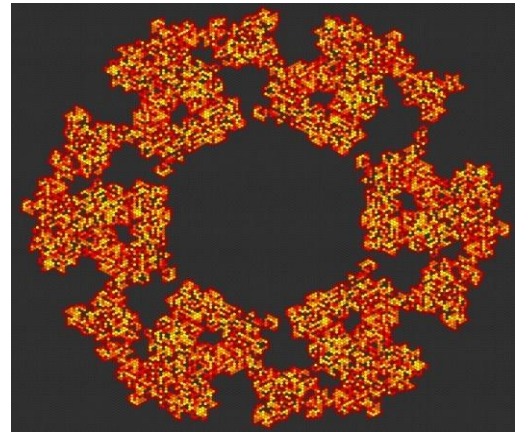


**Εικόνα 2:** Pattern LLRR, το συγκεκριμένο μοτίβο μεγαλώνει συμμετρικά.

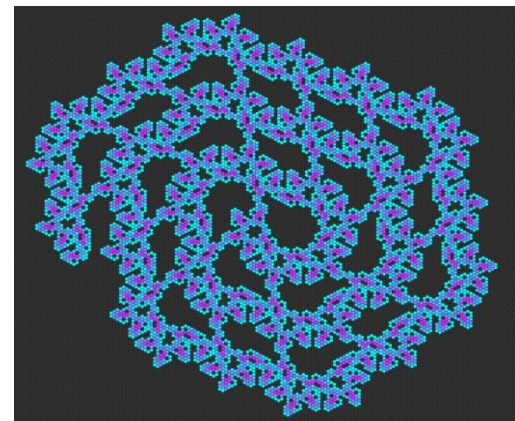
Έπειτα η ιδέα επεκτάθηκε περαιτέρω αφού προστέθηκαν κανόνες που άλλαζαν τις μοίρες της στροφής του μυρμηγκιού. Αυτές οι εντολές ήταν οι εξής:

- N (No change)
- R1 (60° clockwise)
- R2 (120° clockwise)
- U (180°)
- L2 (120° counter-clockwise)
- L1 (60° counter-clockwise)

Μετά από αυτήν αυτή την επέκταση τα αποτελέσματα γίνονταν όλο και πιο θεαματικά με το μυρμήγκι να δημιουργεί πολύ περίπλοκα σχήματα όπως τα παρακάτω.

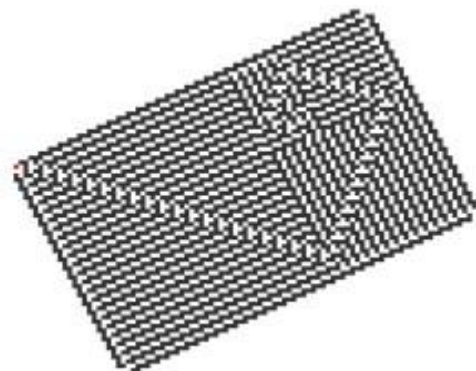


**Εικόνα 3:** Pattern L2NNL1L2L1: Εξάγωνο πλέγμα που μεγαλώνει κυκλικά.



**Εικόνα 4:** Pattern L1L2NUL2L1R2: Εξάγωνο πλέγμα που έχει σπειροειδή ανάπτυξη.

Τελευταία επέκταση του Langton's Ant ήταν η «συμβίωση» δυο και παραπάνω μυρμηγκιών σε μια δισδιάστατη επιφάνεια. Έτσι είχε δοθεί η ικανότητα να αναπτυχθούν οποιοδήποτε είδους σχήματα, που μπορεί και να αντιπροσώπευαν κάτι ουσιαστικό.



**Εικόνα 5:** Δημιουργία μιας σπείρας Fibonacci.

## IV. ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΠΕΙΡΑΜΑΤΙΣΜΟΙ

Η υλοποίηση της ιδέας αυτής είναι πολύ απλή ιδίως στο περιβάλλον της Python η οποία προσφέρει γραφικό περιβάλλον με πολύ απλές και κατανοητές εντολές μέσω της βιβλιοθήκης Turtle.

```
if pos not in maps or maps[pos] == "white":  
  
    ant.fillcolor("red")  
    invert(maps, ant, "red")  
    ant.stamp()  
    ant.right(90)  
    ant.forward(step)  
    pos = coordinate(ant)  
    step_counter=step_counter+1
```

**Εικόνα 6:** Κώδικας για την δημιουργία κανόνων στους οποίους υπακούει το μυρμήγκι.

Η υλοποίηση των κανόνων γίνεται με τις εξής εντολές ( Έχει προηγηθεί η εντολή `ant = turtle.Turtle()` ):

- **ant.fillcolor():** Με την οποία διαλέγουμε το χρώμα που θα μετατρέψουμε το κελί κάτω από το μυρμήγκι (π.χ. "red").
- **invert()** και **ant.stamp()**: Με τις οποίες πραγματοποιείται η αλλαγή χρώματος.
- **ant.right()** και **ant.left()**: Με τις οποίες διαλέγουμε πόσες μοίρες και προς τα πού θα στρίψει το μυρμήγκι (π.χ. 90).
- **ant.forward()**: Με την οποία το μυρμήγκι πάει μπροστά κατά όσο **step** έχουμε ορίσει(π.χ. `step = 10`).
- **coordinate()**: Με την οποία παίρνουμε τις συντεταγμένες που βρίσκεται το μυρμήγκι για να επαναλάβουμε την διαδικασία.

Τέλος αντικρίζουμε την «εντολή» **step\_counter=step\_counter+1** η οποία δεν παίζει κανένα ρόλο στην υλοποίηση του κανόνα αλλά καταμετράει τα βήματα τα οποία έχει κάνει το μυρμήγκι. Επίσης με ένα πολύ απλό σετ εντολών που προσφέρει η Turtle μπορούμε να δείχνουμε στα πόσα βήματα είναι το μυρμήγκι όσο αυτό προχωράει.

```
counter.undo()  
counter.penup()  
counter.setposition(-600,350)  
counter.write("Steps="+str(step_counter),font=("Arial",20,"normal"))
```

**Εικόνα 7:** Κώδικας για την εκτύπωση των βημάτων στο γραφικό περιβάλλον.

Για την εκτύπωση χρησιμοποιούνται οι ακόλουθες εντολές ( Έχει προηγηθεί η εντολή

`counter=turtle.Turtle(visible=False)` ):

- **counter.undo():** Για να σβήσουμε ότι είχε πριν στην θέση του counter.
- **counter.penup():** Για να σηκώσουμε το counter και να μην σχεδιάσει κάποια γραμμή μέχρι να πάει στην θέση του.
- **counter.setposition():** Για να ορίσουμε την θέση του counter (π.χ. -600,350).
- **counter.write():** Με την οποία τυπώνουμε τον μήνυμα που θέλουμε.

Κυρίως με την χρήση των προαναφερθέντων εντολών μπορούμε να υλοποιήσουμε όποιο μοτίβο θέλουμε και να τυπώνουμε τα βήματα του μυρμηγκιού.

Στους πέντε κώδικες που παραδίδονται εμπεριέχονται τα εξής pattern (παρέχονται εικόνες όλων των παρακάτω):

1. **«RL» (Langton's Ant)** του οποίου η συμπεριφορά αναφέρεται στην αρχική σελίδα)
2. **«RLLLLLLRLRR»:**  
Το συγκεκριμένο pattern είναι μια τυχαία προσωπική επιλογή εντολών η οποία είχε σαν αποτέλεσμα το μυρμήγκι να δημιουργεί πρακτικά ένα τετράγωνο με κόκκινο περίγραμμα μέσα στον οποίο κυριαρχεί το χάος.
3. **«RRLLLRLLLR»:**  
Το συγκεκριμένο είναι ένα αρκετά γνωστό pattern το οποίο στην αρχή

απλά επικρατεί χάος αλλά καταλήγει να δημιουργεί μια «προέκταση» η οποία είναι συγκεκριμένα ένα τρίγωνο.

4. **«RLLR»:**

Το συγκεκριμένο pattern είναι επίσης μια τυχαία προσωπική επιλογή η οποία είχε ως μοναδικό σκοπό την συμμετρική ανάπτυξη η οποία επιτυγχάνεται και δημιουργείται ένα συμμετρικό «τετράπλευρο σχήμα» μέσα στο οποίο σχηματίζεται ένα σχήμα που θυμίζει εικόνα εγκεφάλου και τεστ Ρόσαρχ.

5. **«L1L2NUL2L1R2»:**

Τέλος το τελευταίο μοτίβο είναι επίσης ένα γνωστό μοτίβο το οποίο αναπτύσσεται σπειροειδώς, δεν επικρατεί το χάος όπως στα περισσότερα pattern και δημιουργεί ένα σχήμα που θυμίζει χιονονιφάδα.

[2].<https://docs.python.org/3/library/turtle.html#module-turtle>

[3].[https://en.wikipedia.org/wiki/Langton%27s\\_ant](https://en.wikipedia.org/wiki/Langton%27s_ant)

[4].[https://en.wikipedia.org/wiki/Universal\\_Turing\\_machine](https://en.wikipedia.org/wiki/Universal_Turing_machine)

[5].[https://en.wikipedia.org/wiki/Turing\\_machine](https://en.wikipedia.org/wiki/Turing_machine)

## V. ΣΥΝΟΨΗ-ΕΠΙΛΟΓΟΣ

Τέλος καταλήγουμε στο συμπέρασμα ότι το Langton's Ant είναι μια Universal Turing Machine με άφθονες χρησιμότητες και μπορεί να επεκταθεί με πάρα πολλούς τρόπους, ακόμη και να συνδυαστεί με άλλους κυτταρικούς αυτοματισμούς. Ένας τέτοιος διδιάστατος αυτοματισμός είναι το Conway's Game of Life μέσα από το οποίο το μυρμήγκι θα αναγκαζόταν να «συμβιώσει» με άλλα κινούμενα κύτταρα. Επιπλέον, στις μέρες μας η ιδέα αυτή έχει αναπτυχθεί τόσο πολύ που αναπαρίσταται μέχρι και σε τρισδιάστατους χώρους, έχει υλοποιηθεί σε πάρα πολλές γλώσσες προγραμματισμού και υπάρχει πληθώρα δωρεάν εφαρμογών δίνοντας την δυνατότητα στον καθένα να πειραματιστεί και να δοκιμάσει τυχαία μοτίβα.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

[1]. Διάλεξη, Κ. Σγάρμπας, Μηχανισμοί Επεξεργασίας Πληροφορίας.