ΤΣΙΑΤΣΙΑΝΑΣ ΕΥΑΓΓΕΛΟΣ

Α.Μ.: 1072369

ΕΤΟΣ ΦΟΙΤΗΣΗΣ: Ά Έτος

Υλοποίηση Κυκλικών Κυτταρικών Αυτομάτων σε γλώσσα Python

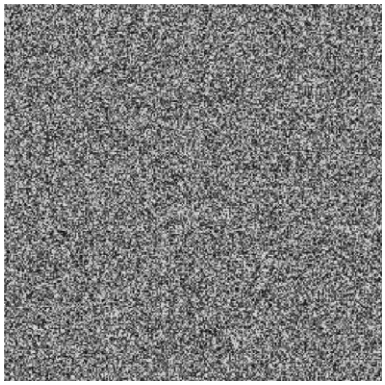
Περίληψη

Στα πλαίσια του μαθήματος "Εισαγωγή στην επιστήμη του Ηλεκτρολόγου μηχανικού", επιλέχθηκαν σαν θέμα της ατομικής εργασίας τα «Κυκλικά Κυτταρικά Αυτόματα». Έτσι, αφού μελετήθηκε η διαθέσιμη βιβλιογραφία, πάρθηκε η απόφαση να δημιουργηθεί μια μηχανή κυκλικών κυτταρικών αυτομάτων σε γλώσσα προγραμματισμού Python, με γραφικό περιβάλλον το οποίο επιτρέπει στον χρήστη να μεταβάλλει τους κανόνες και να οπτικοποιεί τα κυκλικά κυτταρικά αυτόματα που προκύπτουν σε πραγματικό χρόνο.

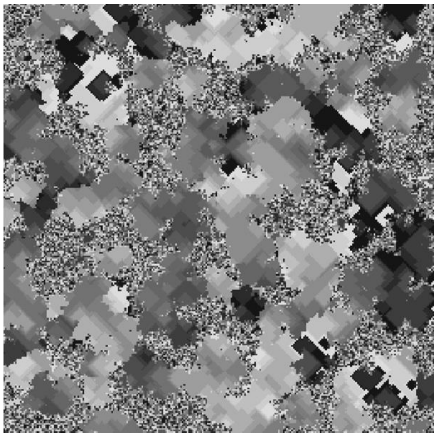
I. ΕΙΣΑΓΩΓΗ

Τα κυκλικά κυτταρικά αυτόματα είναι ένα είδος κανόνα κυτταρικών αυτομάτων που αναπτύχθηκαν από τον David Griffeath στο Πανεπιστήμιο του Ουισκόνσιν και μελετήθηκαν ακόμη από πολλούς άλλους ερευνητές κυτταρικών αυτομάτων. Σε αυτά τα δισδιάστατα κυτταρικά αυτόματα κάθε κελί μπορεί να αλλάξει κατάσταση, αυξάνοντάς την κατά 1, μόνο όταν ένα πλήθος γειτονικών κελίων έχει την αμέσως επόμενη κατάσταση από αυτό.[1]

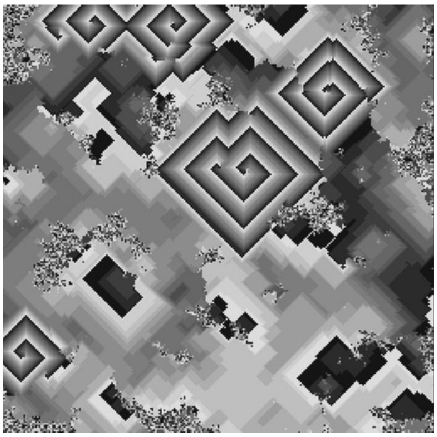
Ορισμένοι συνδυασμοί παραμέτρων που καθορίζουν τους κανόνες των κυκλικών κυτταρικών αυτομάτων έχουν μια τάση αυτό-οργάνωσης που παρουσιάζει τοπικά περιοδικές δομές μεγάλης κλίμακας [2]. Η εξέλιξη αυτού του είδους κυκλικών κυτταρικών αυτομάτων χωρίζεται σε 4 στάδια: πρώτο στάδιο είναι αυτό στο οποία κάθε κελί αποκτά μια τυχαία κατάσταση και ξεκινάει ο υπολογισμός του κυτταρικού αυτομάτου, ωστόσο τα περισσότερα κελία δεν πληρούν τους κανόνες που τους επιτρέπουν να αλλάξουν κατάσταση. Στο επόμενο στάδιο έχουν δημιουργηθεί κάποιες «κρίσιμες σταγόνες» οι οποίες επεκτείνονται γραμμικά με μια κυματική δραστηριότητα, έως ότου να εισαχθούν σε αυτή σχεδόν όλα τα κελία του αυτομάτου. Κατά το τρίτο στάδιο ορισμένες ατέλειες οδηγούν στη δημιουργία αντικειμένων που ονομάζονται «δαίμονες», αυτοί εναλλάσσουν κυκλικά κάθε κατάσταση και καθώς περιστρέφονται δημιουργούν κύματα τα οποία επεκτείνονται με μορφή ελίκων. Τέταρτο και τελευταίο στάδιο είναι αυτό στο οποίο κάθε κελί έχει επηρεαστεί από κάποια έλικα και ακολουθεί την περίοδο του δαίμονα που την δημιουργεί [2]. Τα παραπάνω στάδια απεικονίζονται στις εικόνες 1 έως 4. Ωστόσο, αποτελούν παρατηρήσεις που έγιναν πάνω σε κυκλικά κυτταρικά αυτόματα που προκύπτουν από συγκεκριμένες παραμέτρους και δεν περιγράφουν πλήρως κάθε συνδυασμό παραμέτρων για τα κυτταρικά αυτόματα.



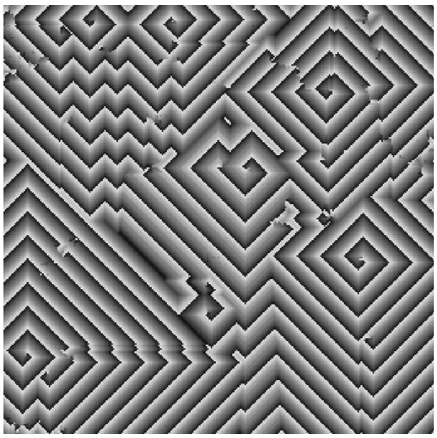
Εικόνα 1. Πρώτο στάδιο



Εικόνα 2. Δεύτερο στάδιο



Εικόνα 3. Τρίτο στάδιο



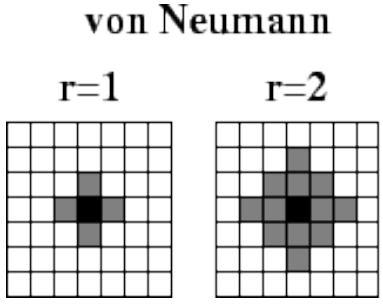
Εικόνα 4. Τέταρτο στάδιο

II. ΑΝΑΠΤΥΞΗ ΘΕΜΑΤΟΣ

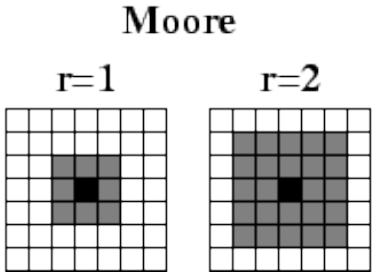
A. Κανόνες

Οι κανόνες που καθορίζουν τα κυκλικά κυτταρικά αυτόματα είναι: ο αριθμός των καταστάσεων (**states**), ο αριθμός κατωφλιού (**threshold**), το είδος της γειτονικής περιοχής (**neighbourhood**) και η εμβέλεια της γειτονικής περιοχής (**range**). Με αυτές τις παραμέτρους μπορούμε να ορίσουμε τους κανόνες υπολογισμού των κυτταρικών αυτομάτων. Έτσι, σε έναν πίνακα δύο διαστάσεων κάθε κελί μπορεί αρχικά να έχει μια τιμή ανάμεσα στο 0 και το n-1 όπου n ο αριθμός των καταστάσεων, στη συνέχεια και για κάθε επόμενο βήμα, κάθε κελί μπορεί να αναβαθμιστεί στην επόμενη κατάσταση αν το πλήθος κελίων που βρίσκονται στην γειτονία του με την αμέσως επόμενη κατάσταση από αυτό, ξεπερνάνε σε αριθμό τον αριθμό κατωφλιού. Να σημειωθεί πως η διαδοχική κατάσταση της κατάστασης n-1 είναι το 0.[3]

Όσον αφορά το είδος της γειτονίας υπάρχουν δύο τύποι, αυτός του Moore και αυτός του Von-Neumann. Έστω $N_{i,j}$ το κελί του οποίου θέλουμε να βρούμε τους γείτονες και r η εμβέλεια της γειτονίας, τότε σύμφωνα με τον Moore γείτονες του είναι τα κελία με συντεταγμένες k,l για τις οποίες ισχύει $|k - i| \leq r$ και $|l - j| \leq r$, ενώ σύμφωνα με τον Von-Neumann, γειτονικά κελιά του N είναι αυτά για τα οποία ισχύει $|k - i| + |l - j| \leq r$.[4] Στις εικόνες 5 και 6 παρουσιάζονται παραδείγματα των δύο διαφορετικών τύπων γειτονιών για εμβέλειες r = 1 και r = 2.



Εικόνα 5. Γειτονικά κελία κατά Von Neumann[4]



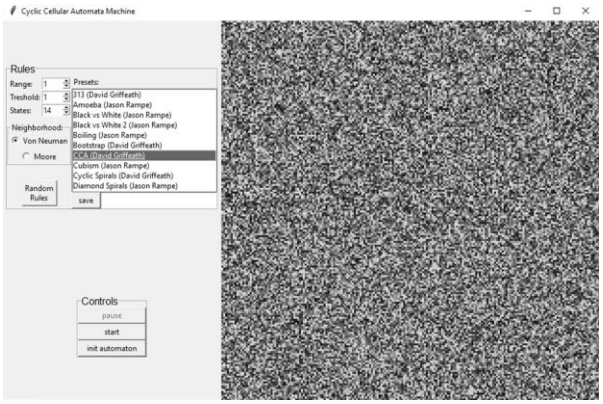
Εικόνα 6. Γειτονικά κελία κατά Moore[4]

Οποιοσδήποτε από τους δύο τύπους μπορεί να χρησιμοποιηθεί ως κανόνας για ένα κυκλικό κυτταρικό αυτόματο, ωστόσο κάθε ένας παρουσιάζει διαφορετική συμπεριφορά, επομένως και διαφορετικά μοτίβα στο τελικό στάδιο.

Β. Υλοποίηση

Για την υλοποίηση της μηχανής των κυκλικών κυτταρικών αυτομάτων επιλέχθηκε η γλώσσα Python 3.9 καθώς η δημιουργία γραφικού περιβάλλοντος για τον χρήστη είναι αρκετά πιο απλή σε σύγκριση με την γλώσσα C, χρησιμοποιώντας την βασική βιβλιοθήκη tkinter.

Στην βιβλιοθήκη αυτή, βασίζεται η δημιουργία του γραφικού παραθύρου, των αντικειμένων ελέγχου (κουμπιά, επιγραφές κ.λπ.), η διάταξή τους στο γραφικό περιβάλλον αλλά και η σύνδεση τους με συναρτήσεις του προγράμματος για την πραγματοποίηση των καθορισμένων ενεργειών. Στην εικόνα 7 φαίνεται ένα στιγμιότυπο οθόνης από το γραφικό περιβάλλον.



Εικόνα 7. Το γραφικό περιβάλλον «Μηχανή Κυκλικών Κυτταρικών Αυτομάτων»

Ο πίνακας που χρησιμοποιείται για την αναπαράσταση των κελίων του κυκλικού κυτταρικού αυτομάτου στην «μηχανή» έχει δημιουργηθεί με χρήση της βιβλιοθήκης NumPy και έχει μέγεθος ίσο με 200 κελιά σε κάθε διάσταση [5]. Τα διανύσματα της NumPy επιλέχθηκαν έναντι των προκαθορισμένων λιστών της γλώσσας Python διότι επιτυγχάνονται καλύτερες επιδόσεις όσον αφορά τους χρόνους υπολογισμού αλγεβρικών πράξεων και την διαχείριση της μνήμης.

Για την απεικόνιση του πίνακα με τα κελία του κυτταρικού αυτομάτου χρησιμοποιήθηκε η βιβλιοθήκη matplotlib [6] η οποία είναι περιέχει κατάλληλες συναρτήσεις για την γραφική απεικόνιση δεδομένων. Πιο συγκεκριμένα χρησιμοποιήθηκε η συνάρτηση pcolormesh για την αποτύπωση του δισδιάστατου πίνακα τιμών του κυτταρικού αυτομάτου, αναπαριστώντας κάθε κατάσταση με διαφορετικό χρώμα. Τα χρώματα αυτά επιλέγονται αυτόματα από ένα συγκεκριμένο χρωματικό φάσμα αναθέτοντας την μικρότερη και την μεγαλύτερη κατάσταση στα άκρα του φάσματος και τις ενδιάμεσες καταστάσεις ισοκατανεμημένες στο υπόλοιπο φάσμα. Στη συνέχεια χρησιμοποιήθηκε η συνάρτηση FuncAnimation η οποία πρώτα καλεί σε επανάληψη μια δοσμένη συνάρτηση του χρήστη για την τροποποίηση των δεδομένων (στην προκειμένη περίπτωση η συνάρτηση αυτή κάνει έλεγχο τον κανόνων για κάθε κελί και αλλάζει την κατάσταση του αν επιτρέπεται) και έπειτα ανανεώνει το αντικείμενο πάνω στο οποίο έχουν αποτυπωθεί τα δεδομένα.

Η πρώτη εκδοχή της συνάρτησης που αναφέρθηκε νωρίτερα για τον έλεγχο των κανόνων και μεταβολή της κατάστασης του κάθε κελιού είχε υλοποιηθεί με χρήση δυο διαδοχικών βρόχων for της Python. Ωστόσο η υλοποίηση αυτή ήταν αρκετά χρονοβόρα στον υπολογισμό και παρουσίασε μια δυσκολία στην παραμετροποίηση του είδους και της εμβέλειας της γειτονικής περιοχής. Σε αυτά λύση έδωσε η βιβλιοθήκη SciPy και η υποβιβλιοθήκη αυτής: ndimage [7], η οποία εξειδικεύεται στην ανάλυση και την επεξεργασία εικόνων αντιμετωπίζοντάς τες ως δισδιάστατους πίνακες. Συγκεκριμένα χρησιμοποιήθηκε η συνάρτηση generic_filter η οποία δέχεται ως βασικά ορίσματα τον πίνακα που θέλουμε να

επεξεργαστούμε, μια συνάρτηση επεξεργασίας και έναν πίνακα δυαδικών τιμών. Στη συνέχεια διατρέχει κάθε στοιχείο του δοσμένου πίνακα και καλεί την συνάρτηση δίνοντας της ως όρισμα ένα διάνυσμα με τις τιμές των γειτονικών στοιχείων που αντιστοιχούν σε άσους του δυαδικού πίνακα. Για τον δημιουργία του δυαδικού πίνακα χρησιμοποιήθηκαν οι συναρτήσεις iterate_structure και generate_ binary_structure. Με αυτόν τον τρόπο επιτεύχθηκε η παραμετροποίηση της γειτονικής περιοχής αλλά και μειώθηκε ο χρόνος υπολογισμού ενός βήματος κατά 70%. Στον παρακάτω πίνακα φαίνονται οι σχετικοί μέσοι χρόνοι υπολογισμού σε περίοδο 100 βημάτων.

Πίνακας Ι.

	Έκδοση no.1	**Έκδοση no.2**
Μέσος χρόνος υπολογισμού βήματος	0.73sec	0.20sec

Ακόμη να σημειωθεί πως υπάρχουν αρκετά σημεία τα οποία θα μπορούσαν να βελτιώσουν τις επιδόσεις της μηχανής, όπως ο τρόπος που απεικονίζεται ο πίνακας στο γραφικό περιβάλλον αλλά και περεταίρω ο τρόπος υπολογισμού κάθε βήματος.

C. Γραφικό περιβάλλον

Χρησιμοποιώντας το γραφικό περιβάλλον ο χρήστης μπορεί μεταβάλλει τους κανόνες που ορίζουν ένα κυκλικό κυτταρικό αυτόματο. Οι παράμετροι αυτοί αναλύθηκαν στην ενότητα ΙΙ.Α. Για να επιτευχθεί αυτό ο χρήστης μπορεί **α.** να εισάγει απευθείας τους κανόνες, **β.** να πατήσει το κουμπί «random rules» έτσι ώστε η μηχανή να εισάγει ψευδοτυχαίους συνδυασμούς και **γ.** να επιλέξει από την στήλη μια από τις προεπιλογές[3]. Στη συνέχεια ο χρήστης πρέπει να κάνει κλικ στο κουμπί «init automaton» για να φορτώσει τους κανόνες που επιλέχθηκαν νωρίτερα στην μηχανή. Τέλος ο χρήστης για να ξεκινήσει την αναπαραγωγή του κυτταρικού αυτομάτου πρέπει να πατήσει το κουμπί «start». Επιπλέον αν είναι επιθυμητό ο χρήστης μπορεί να σταματήσει και να συνεχίσει την αναπαραγωγή, ενώ υπάρχει και η δυνατότητα αποθήκευσης κανόνων του χρήστη στην λίστα προεπιλογών για μεταγενέστερη φόρτωση τους στην μηχανή κυτταρικών αυτομάτων, αφού πρώτα τους έχει δώσει μια ονομασία.

ΙΙΙ. ΣΥΝΟΨΗ

Μπορούμε να συμπεράνουμε πως παρ' όλο που εκ πρώτης όψεως τα κυκλικά κυτταρικά αυτόματα φαίνονται εντελώς χαοτικά, *διέπονται* από πολύ συγκεκριμένους κανόνες και αρκετές φορές έχουν προβλέψιμη συμπεριφορά.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1]. https://en.wikipedia.org/wiki/Cyclic_cellular_automaton

[2] R. Fisch, D. Griffeath and J. Gravner, Cyclic Cellular Automata in Two Dimensions, 1991

[3]. https://softologyblog.wordpress.com/2013/08/29/cyclic-cellular-automata/

[4]. http://www.jcasim.de/main/node5.html

[5]. https://numpy.org/

[6]. https://matplotlib.org/

[7]. https://docs.scipy.org/doc/scipy/reference/ndimage.html#module-scipy.ndimage