

Turing gépek

Rekurzív függvények

Alan Turing (1912-1954)

1. A modern számítógép-tudomány atyja
2. Nagy hatással volt az algoritmus és a számítógépes adatfeldolgozás kidolgozásában
3. Kidolgozta a Church-Turing tézist és a Turing gépet

Elméletét először az 1936-ban megjelent On Computable Numbers, with an Application to the Entscheidungsproblem című cikkében publikálta.

A második világháború alatt sikeres erőfeszítéseket tett a rejtjelkódok feltörésére. (Enigma)

Továbbá a Computing Machinery and Intelligence írásában vetette fel a mai mesterséges intelligencia alaptézisét, miszerint :

“Azt javaslom megfontolásra, hogy tudnak-e a gépek gondolkodni?”



Turing gépek alkotóelemei:

- I. két irányban végtelen, cellákra osztott szalag
- II. Író / olvasófej
- III. Központi egység

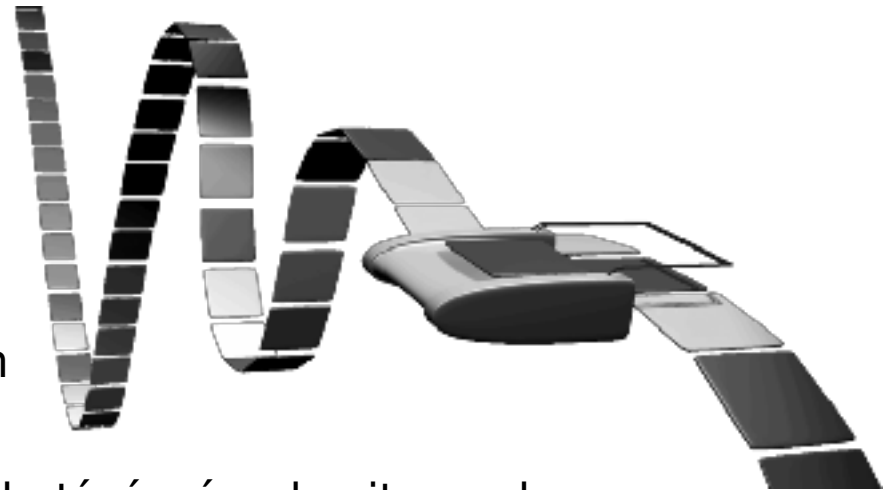
Működésének elve:

Elindítás előtt minden inputot a szalagra írunk
A fejet a szalag elejére állítjuk, majd a START
állapotból elindítjuk.

Működése folyamán különböző belső állapotokban
lehet a TM.

A következő lépésben : belső állapot + olvasott jel hatására ír valamit a szalagra.
Ezután mozdul , majd belső állapotot vált -> ezt irányítja az átmeneti függvény.

Megáll, ha eléri a HALT állapotot.



Sokféle kombinációja létezik(pl több start-halt állapot) az univerzális TM-nek, de ezek ekvivalensek egymással.

Azonban a véges hosszúságú szalaggal ellátott TM sok olyan problémát nem tud megoldani, mint nem korlátozott társa.

Halting problem: Nem létezik olyan program, amely tetszőleges programról és inputról eldöntené, hogy az adott program az adott inputtal megáll-e.

Totik Vilmos (1954 -)

Széchenyi-díjas magyar matematikus, egyetemi tanár, a Magyar Tudományos Akadémia rendes tagja.

Kutatási területei az ortogonális polinomok, az approximációelmélet és a potenciálmélet.



Az élet játéka:

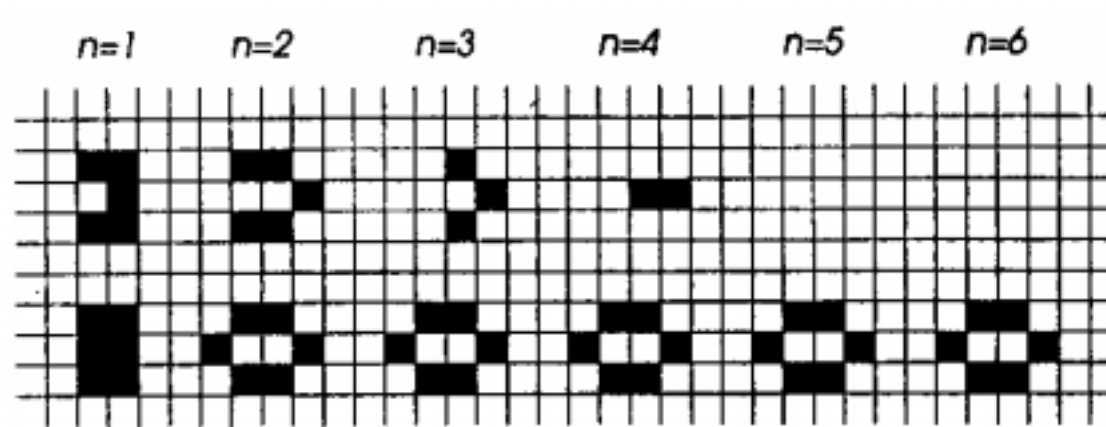
Osszunk egy síkot cellákra. Minden cellának 8 szomszéda van. Minden cellának 2 állapota (aktív, passzív).

Szinkronizáló óra hatására minden cella állapotot vált (egyszerre).

- (I) Egy aktív cella az $(n+1)$ időpillanatban aktív marad, ha az n -edikben pontosan kettő vagy három aktív szomszédja volt, különben passzívvá válik. -> **túlélés**
- (II) Egy passzív cella az $(n+1)$ időpillanatban aktívvá válik, ha az n -edikben pontosan három aktív szomszédja volt, egyébként passzív marad. -> **születés**

Ezután kezdődik a játék a szabályoknak megfelelően. Már csak meg kell határozni egy kezdőállapotot.

Például :

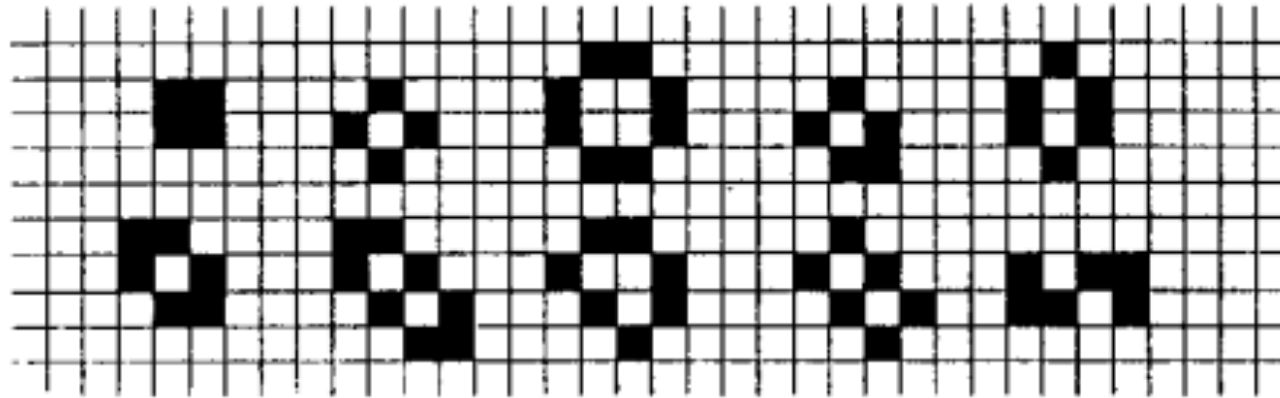


Az aktív cellákat szokás sejteknek nevezni.

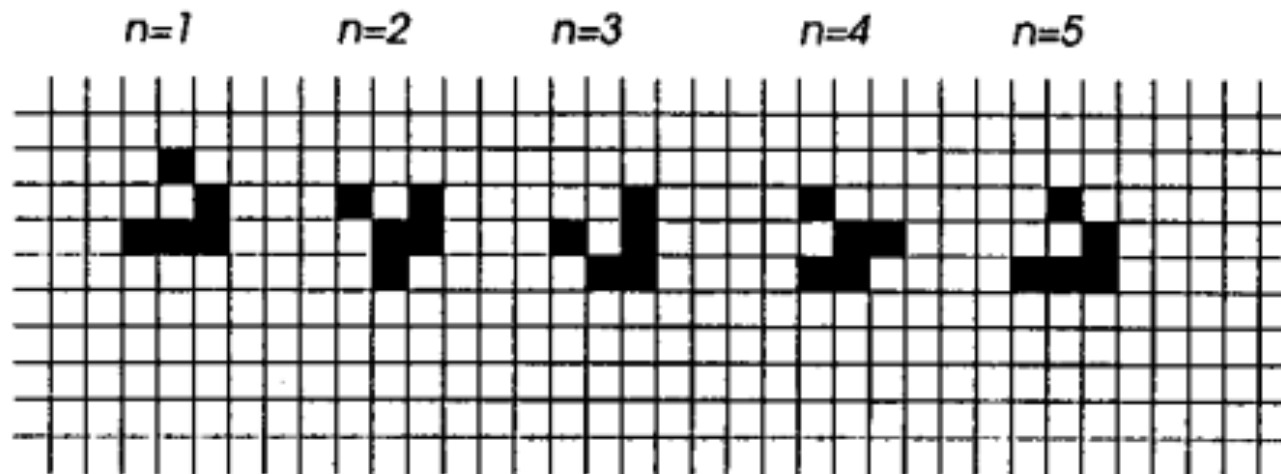
Sejt pusztulása kétféle okból következhet be:

- I. magányosság (≤ 1 szomszéd)
- II. túlzsúfoltság (≥ 4 szomszéd)

Néhány további egyszerű példa:



Tetszőleges k -ra k generáció után megismétlődnek, de kezdeti helyzetükhöz képest kissé eltolva. Legfontosabb közülük a glider ~ sikló, ami egy golyót lő ki 45 fokos szögben.



Felvetődött a kérdés, hogy lehet-e olyan konfigurációt megadni, amelyből kiindulva az aktív cellák száma minden határon túl nő?

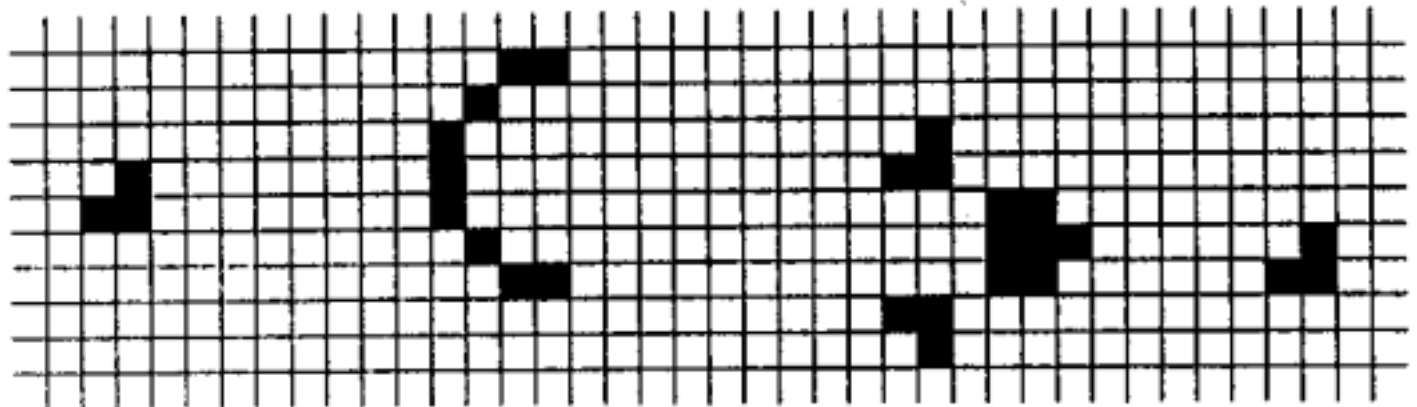
Ennek megoldását R. William Gosper alkotta meg.

R. William Gosper (1943-)

- MIT-n szerzi meg BSc-ját.
- Jelentős előrelépést ért el az Élet játékában, hátizsák problémákban.
- Jelenleg a számítógépes algebra területén tevékenykedik.



Ő alkotta meg azt az “ágyút” amely minden 30. generáció után visszatér kiindulási állapotába miközben egy golyót repít ki.



Sejtautomaták

A life egy tipikus sejtautomata.

- rendszer : négyzethálók négyzetei
- véges sok állapot : aktív/passzív
- szomszédság
- minden sejt egyszerre vált állapotot

Elmélete Neumann Jánostól eredeztethető. Megalkotta az önmagát reprodukáló sejtautomatát. A négyzethálós síkon Neumann megalkotott egy olyan sejtautomatát, amely minden függvényt képes volt kiszámolni (univerzális számológép).

Univerzális alkotó-gép: képes minden alakzatot megépíteni, ezáltal önmagát is képes reprodukálni.

A gép adatai: 29 belső állapot, 200 000 aktív sejt. Magában hordozott egy univerzális számológépet - ez tekinthető az ideális PC-nek -.

A life univerzális gép, azaz vele bármi kiszámítható, amely más gépekkel is kiszámítható, tehát van olyan life kezdőállapot, amely ugyan azt a végállapotot adja, mint egy tetszőlegesen írt program PC-n.