

Turing-gép

Témavázlat

- Turing-gép
 - Determinisztikus, 1-szalagos Turing-gép
 - A gép leírása, példák
 - k-szalagos Turing-gép
 - Univerzális Turing-gép
 - Egyéb Turing-gépek
 - Nemdeterminisztikus Turing-gép
 - Randomizált Turing-gép
- Turing-géppel ekvivalens gépek
 - Ram gép
- Church-tézis
- Nyelvek
 - Rekurzíve felsorolható nyelv, rekurzív nyelv
- Algoritmikusan megoldhatatlan problémák
- Algoritmusok hatékonysága
- Gyakorlati kiszámíthatóság
- Irodalomjegyzék

A Turing-gép nem valóságos gép, hanem absztrakt automata. Alan Turing alkotta meg 1936-ban azzal a céllal, hogy algoritmusok bonyolultságát vizsgálja, valamint azt, hogy melyek azok a feladatok, amelyek megoldására létezik algoritmus, s melyekre nem létezik. A Turing-gép segítségével megmutatható többek között az is, hogy van algoritmikusan megoldhatatlan probléma.

A Turing-gépnek számos változata lelhető fel az irodalomban, ezekről azonban be lehet látni, hogy egymással ekvivalensek.

Determinisztikus, 1-szalagos Turing-gép

A determinisztikus, 1- szalagos Turing-gép leírása

Az 1-szalagos Turing gépet a következő részek alkotják:

- Egy véges szalag, ami mindkét irányban tetszőlegesen meghosszabbítható és mezőkre van felosztva.
- Szalag-ábécé. Ennek a jelei fordulhatnak elő a szalagon, egy mezőben az ábécé egyik jele áll, vagy üres a mező.
- Író-olvasó fej, amelyik adott pillanatban pontosan egy mezőn áll.
- Vezérlőmű, ami véges sok állapot valamelyikében lehet.

Egy adott pillanatban a fej elolvassa azon a mezőn lévő jelet, ahol áll, és a jeltől, valamint a vezérlőmű pillanatnyi állapotától függően ír a mezőre (esetleg ugyanazt, ami ott volt), ezután vagy jobbra lép, vagy balra lép, vagy helyben marad, a vezérlőmű pedig új állapotba kerül.

A gép állapotai között van egy kitüntetett állapot, a kezdőállapot.

Megállapodunk abban, hogy induláskor a szalagon véges sok egymásmelletti mezőn vannak jelek, és az író-olvasó fej a bal szélső jel fölött áll.

Definíció.

Determinisztikus, 1-szalagos Turing-gép a következő rendezett hetes: $M = \langle Q, V, W, \delta, q_0, S, F \rangle$

- Q a gép *belső állapotainak* halmaza
- V a szalag *bemenő ábécéje*
- W a *szalag-ábécé*. $V \subseteq W$.
- δ az *átmenetfüggvény*. $\delta : Q \times W \rightarrow Q \times W \times \{B, H, J\}$
(B esetén az író-olvasó fej balra lép, J esetén jobbra, H esetén helyben marad.)
- $q_0 \in Q$ *kezdőállapot*
- S az *üres mező* jele
- $F \subseteq Q$ a *felismerő (elfogadó)* állapotok halmaza.

A gép akkor áll le, ha az aktuális állapot és az éppen beolvasott jel esetén a δ átmenetfüggvény nincs értelmezve. Ha leállítás esetén az állapot eleme F-nek, akkor azt mondjuk, hogy a gép *elfogadta* (*felismerte*) a bemeneti szót, egyébként *nem fogadta el* (*nem ismerte fel*).

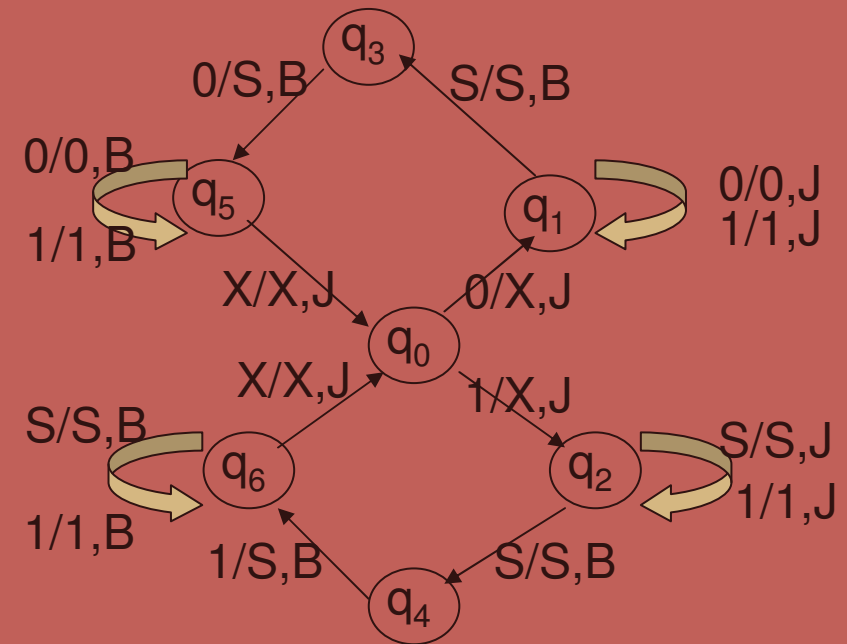
Lehetséges, hogy a gép nem áll le véges sok lépésben valamilyen bemenet esetén.

Teljes számítási folyamatról beszélünk, ha adott bemenet esetén sorban leírjuk a gép lépéseit, vagyis azt, hogy milyen állapotba kerül, milyen jelek vannak a szalagon, és melyik jel fölött áll az író-olvasó fej.

1. példa

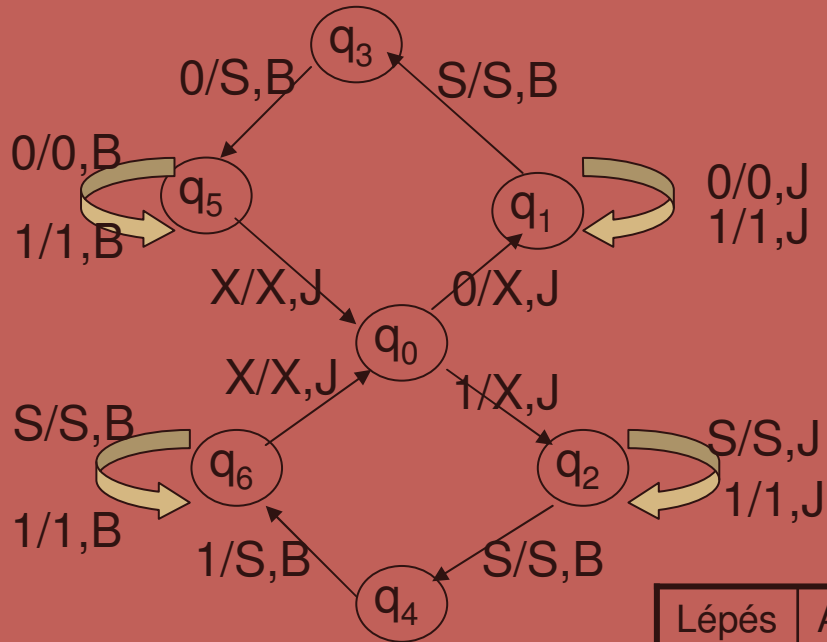
$M = \langle \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{0, 1\}, \{0, 1, X, S\}, \delta, q_0, \{q_0\} \rangle$

Régi állapot	Olvasott jel	Új állapot	Írt jel	Fej- mozgás
q_0	0	q_1	X	J
q_0	1	q_2	X	J
q_1	0	q_1	0	J
q_1	1	q_1	1	J
q_0	S	q_3	S	B
q_2	0	q_2	0	J
q_2	1	q_2	1	J
q_2	S	q_4	S	B
q_3	0	q_5	S	B
q_4	1	q_6	S	B
q_5	0	q_5	0	B
q_5	1	q_5	1	B
q_5	X	q_0	X	J
q_6	0	q_6	0	B
q_6	1	q_6	1	B
q_6	X	q_0	X	J



1. példa folytatás

A teljes számítási folyamat:



Lépés	Állapot	Szalag állapota
1	q ₀	<u>0</u> 01
2	q ₁	X <u>0</u> 1
3	q ₁	X0 <u>1</u>
4	q ₁	X01 <u>S</u>
5	q ₃	X0 <u>1</u>

Lépés	Állapot	Szalag állapota
1	q ₀	<u>1</u> 001
2	q ₂	X <u>0</u> 01
3	q ₂	X0 <u>0</u> 1
4	q ₂	X00 <u>1</u>
5	q ₂	X001 <u>S</u>
6	q ₄	X00 <u>1</u>
7	q ₆	X0 <u>0</u>
8	q ₆	X <u>0</u> 0
9	q ₆	<u>X</u> 00
10	q ₀	X <u>0</u> 0
11	q ₁	XX <u>0</u>
12	q ₁	XX0 <u>S</u>
13	q ₃	XX <u>0</u>
14	q ₅	XX <u>S</u>
15	q ₀	<u>X</u> X

2. példa

$$M = \langle \{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, 1, X, Y, S\}, \delta, q_0, \{q_4\} \rangle$$

Régi állapot	Olvasott jel	Új állapot	Írt jel	Fej-mozgás
q_0	0	q_1	X	J
q_0	Y	q_3	Y	J
q_1	0	q_1	0	J
q_1	1	q_2	Y	B
q_1	Y	q_1	Y	J
q_2	0	q_2	0	B
q_2	X	q_0	X	J
q_2	Y	q_2	Y	B
q_3	Y	q_3	Y	J
q_3	S	q_4	S	J

Adjuk meg ennek a Turing-gépnek az átmenetdiagramját. Írjuk le, hogy az alábbi bemenetek esetén mi a teljes számítási folyamat:

- 000111
- 00111
- 0101

Felismeri-e ezeket a szavakat a gép?

Definíció:

A T Turing-gép *előállítja az f függvényt*, ha minden x bemenet esetén az $f(x)$ kimenetet állítja elő, és leáll, ha x eleme az értelmezési tartománynak.

3. példa

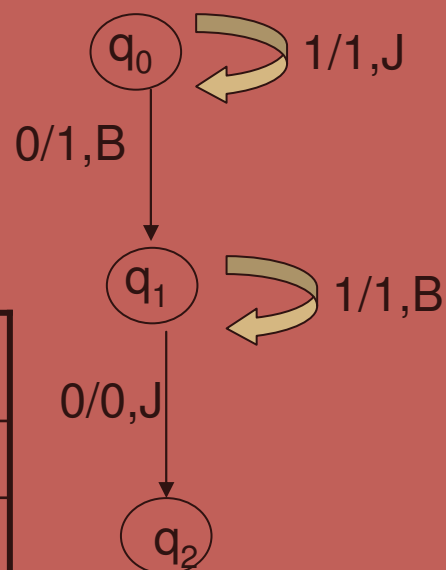
Állítsunk elő olyan Turing gépet, amelyik az $f(n)=n+1$ függvényt állítja elő. Az n szám $n+1$ darab 1-es számmal van reprezentálva.

Pl. 0: 1, 1: 11, 2: 111,...

A 0 jelöli az üres mezőt.

Régi állapot	Olvasott jel	Új állapot	Írt jel	Fej-mozgás
q_0	0	q_1	1	B
q_0	1	q_0	1	J
q_1	0	q_1	0	J
q_1	1	q_1	1	B
q_1	0	q_2	0	B

$$M = \langle \{q_0, q_1, q_2\}, \{1\}, \{0, 1\}, \delta, q_0, \{q_2\} \rangle$$



A teljes számítási folyamat $n=3$ esetén:

Lépés	Állapot	Szalag állapota
1	q_0	<u>1</u> 111
2	q_0	1 <u>1</u> 11
3	q_0	11 <u>1</u> 1
4	q_0	111 <u>1</u> 0
5	q_0	1111 <u>0</u>
6	q_1	1111 <u>1</u>
7	q_1	11 <u>1</u> 11
8	q_1	1 <u>1</u> 111
9	q_1	<u>1</u> 1111
10	q_1	0 <u>1</u> 1111
11	q_2	<u>1</u> 1111

4. példa

Állítsunk elő olyan Turing gépet, amelyik az $f(n, m)=n+m$ függvényt állítja elő. Az n szám $n+1$ darab 1-es számmal van reprezentálva, az m hasonlóképpen.

Pl. 0: 1, 1: 11, 2: 111, ...

A n számnak megfelelő 1-esek után 0 következik, majd az m -nek megfelelő egyesek.

Pl. $2+3$ esetén a bemenet: 11101111, a kimenet: 111111.

Adjuk meg az átmenetfüggvényt táblázatos formában, adjuk meg az átmenet diagramot, és $n=2$, $m=3$ esetén írjuk le a teljes számítási folyamatot.

k-szalagos Turing-gép

k-szalagos Turing-gép

A k-szalagos Turing-géphez $k \geq 2$ szalag, és k író-olvasó fej tartozik.

Definíció.

Legyen T k+1 szalagos, M pedig k szalagos Turing-gép Σ szalag-ábécével. Legyen $p \in \Sigma^* - \{S\}$. Azt mondjuk, hogy T a p programmal *szimulálja M-et*, ha tetszőleges $x_1, \dots, x_k \in \Sigma^* - \{S\}$ szavakra a T az (x_1, \dots, x_k, p) bemeneten akkor és csak akkor áll meg véges számú lépésben, ha M az (x_1, \dots, x_k) bemeneten megáll, és megálláskor T első k szalagján rendre ugyanaz áll, mint M szalagjain.

Tétel.

Minden k szalagos M Turing géphez van olyan 1-szalagos T Turing-gép, amely *M-et helyettesíti* a következő értelemben: minden $x \in \Sigma^* - \{S\}$ szóra M akkor és csak akkor áll meg véges sok lépésben az x bemeneten, ha T megáll az x bemeneten, és M első szalagjára ugyanaz lesz írva, mint T szalagjára.

Bizonyítás.

Lásd [3]-ban.

Univerzális Turing-gép

Univerzális Turing-gép

Definíció.

Akkor mondjuk, hogy a $k+1$ szalagos T Turing-gép *univerzális* (a k szalagos Turing-gépekre nézve), ha bármely k szalagos Σ szalag-ábécével rendelkező M Turing-géphez létezik olyan p szó (program), mellyel a T szimulálja M -et.

Tétel.

Minden $k \geq 1$ számhoz és minden Σ ábécéhez létezik $k+1$ szalagú univerzális Turing-gép.

Bizonyítás.

Lásd [3]-ban.

Egyéb Turing-gépek

Nemdeterminisztikus Turing-gép

Nemdeterminisztikus Turing-gépről beszélünk, ha az átmenetfüggvény nem $Q \times W \times \{B, H, J\}$ elemeire képez le, hanem részalmazaira.

Belátható, hogy nyelvfelismerés szempontjából ez a gép ekvivalens a determinisztikus Turing-géppel.

Randomizált Turing-gép

A randomizált Turing-gép egy nemdeterminisztikus Turing-gép kibővítve egy valószínűségi eloszlással, mely alapján a megengedett lépések közül választunk, s így a megengedett számításokhoz is konkrét valószínűségérték rendelhető.

Turing-géppel ekvivalens gépek

Ram gép

A Turing-gép egyik hátránya az, hogy egy távoli mező eléréséhez az összes addigi mezőn át kell haladnunk. Ezt a hátrányt kiküszöböli a RAM-gép (Random Access Machine=tetszőleges, vagy közvetlen elérésű gép). A RAM-gép memóriarekeszeit közvetlenül lehet elérni írás vagy olvasás céljából.

Tétel.

Tetszőleges Turing-gép szimulálható RAM programmal. Minden RAM program szimulálható Turing-géppel.