A számításelmélet alapjai 2.

8. gyakorlat

Számító Turing gép

Példa

M=<{0,1,igen,nem}, {a,b}, {a,b,x,ü}, δ, 0, igen, nem >

δ	a	b	x	ü
0	1 x →	1 x →	nem	igen
1	1 a →	1 b →	nem	2 ü ←
2	3 ü ←	3 ü ←	nem	nem
3	3 a ←	3 b ←	0 x →	nem

a) Elfogadja-e az **a** szót? (Indoklás.)

0a ├ x1ü ├ 2x ├ nem

b) Elfogadja-e a **b** szót? (Indoklás.)

0b ├ x1ü ├ 2x ├nem

c) Elfogadja-e az **aaa** szót?

0aaa ├x1aa ├xa1a ├xaa1ü ├xa2a ├x3a ├3xa ├x0a ├xx1ü ├x2x ├nem

d) Milyen nyelvet ismer fel az adott Turing-gép?
Páros hosszú szavakat ismeri fel.

e) Mit számol ki egy 3 illetve egy 4 hosszú szóra?
*3 és 4 hosszú szó esetén is **xx** marad a szalagon.*

f) Általánosan egy tetszőleges szóra mit számol ki?
$f(u) = x^k$, ahol $k=\lceil l(u)/2 \rceil$ és u $u \in \{a,b\}^*$

Gyakorlat

Adott a következő Turing-gép.
M=<{0,1,2,igen,nem}, {a,b}, {a,b,c,ü}, δ, 0, igen, nem>

δ	a	b	c	ü
0	0 a →	1 ü →	igen	nem
1	1 a →	1 b →	1 c →	2 c ←
2	2 a ←	2 b ←	2 c ←	0 ü →

Kérdések:

a) Adja meg az **0abab** kezdő konfigurációból kiindulva, hogy a gép az „**abab**” inputra milyen outputot állít elő!
b) Adja meg általánosan, hogy csak a-t és b-t tartalmazó szavakra mit számol ki a gép!

Nemdeterminisztikus Turing gépek

Jelölés: $\mathcal{P}(X) = \{Y | Y \subseteq X\}$ az X halmaz hatványhalmaza.

Definíció Az egyszalagos nemdeterminisztikus Turing gép (továbbiakban röviden NTG) egy $M = <Q,\Sigma,\Gamma,\delta,q_0,q_i,q_n>$ rendezett hetes, ahol

- Q az állapotok véges, nemüres halmaza,
- $q_0,q_i,q_n \in Q$, q_0 a kezdő- q_i az elfogadó- és q_n az elutasító állapot,
- Σ és Γ ábécék, a bemenő jelek és a szalagszimbólumok ábécéje úgy, hogy $\Sigma \subseteq \Gamma$ és ü$\in \Gamma \setminus \Sigma$,
- $\delta : (Q\setminus\{q_i,q_n\})\times\Gamma \rightarrow \mathcal{P}(Q\times\Gamma\times\{L,S,R\})$.

Míg a determinisztikus esetben a δ átmenetfüggvény minden egyes $(Q\setminus \{q_i,q_n\})\times\Gamma$-beli párhoz pontosan egy, addig egy nemdeterminisztikus TG akárhány (pl. 0,1,5,100) darab $Q\times\Gamma\times\{L,S,R\}$-beli rendezett hármast rendelhet hozzá. $|Q\times\Gamma\times\{L,S,R\}|$véges.

Példa

Adott a következő <u>nemdeterminisztikus</u> Turing-gép.
M=<{0,1,igen,nem}, {a,b}, {a,b,ü}, δ, 0, igen, nem >

δ	a	b	ü
0	1 a →	nem 0 b —	nem
1	1 a — 0 a ←	nem 0 b →	igen 1 ü ←

a) Mutassa meg az **0aba** kezdő konfigurációból kiindulva, hogy a gép elfogadja az „**aba**” szót! Adja meg a konfigurációs gráfot!

Megoldás:

0aba
|
a1ba
/ \
nem ab0a
|
aba1ü
/ \
igen ab1a
/ \
ab1a a0ba
... / \
nem a0ba
…

b) Elfogadja-e az „aa” szót? (Indoklás.)
c) Milyen nyelvet ismer fel az adott Turing-gép? *L(M) = a(ba)**
d) Ezt a nyelvet el is fogadja? (Indoklás.)

Gyakorlat

Adott a következő nemdeterminisztikus Turing-gép.

M=<{0,1,2,igen,nem}, {a,b}, {a,b,ü}, δ, 0, igen, nem >

δ	a	b	ü
0	0 a → 1 ü →	0 b →	nem
1	1 a → 2 ü →	1 b →	nem
2	2 a →	2 b →	igen

a) Mutassa meg a **0aab** kezdő konfigurációból kiindulva, hogy a gép elfogadja az „**aab**” szót! Adja meg a konfigurációs gráfot!
b) Elfogadja-e a gép a „**ba**” szót?
c) Milyen nyelvet ismer fel az adott Turing-gép?

Feladatok kódolása

1.) Adott a következő *valós* probléma.

Egy édesanya a két fia közt igazságosan szét akar osztani néhány cukorkát.

Az ennek megfelelő *absztrakt* feladat az, hogy inputként adott egy természetes szám, és outputként pedig egy olyan természetes számot kell kapnunk, amelynek kétszerese egyenlő az inputtal vagy eggyel kevesebb. (Az output adja meg, hogy mennyi cukrot kell adni az egyes gyerekeknek.)

Formálisan: input: n és $n \in N$; output: int(n/2)

A *konkrét* kiszámítási feladatban az inputot és az outputot kettes-számrendszerben kódoljuk.

Eldöntési problémának felfogva a feladatot, az inputot az outputtól a # jellel választjuk el.

Tegyük fel, hogy maximum 6 cukorka van, így a kódolásnál a számokat 3 bit hosszan ábrázoljuk.

Kérdések:

a) Milyen ábécé felett értelmezzük a szavakat? *Megoldás:* {0,1,#}

b) Jó szó-e a következő? **101#010** *Megoldás:* igen, mert int(5/2)=2

c) Adjuk meg az összes lehetséges jó szót!

Megoldás: L(M)={ 000#000, 001#000, 010#001, 011#001, 100#010, 101#010, 110#011}

2.) Adott a következő *valós* probléma.

Egy ábrát egy vonallal akarunk megrajzolni.

Az ennek megfelelő *absztrakt* feladat az, hogy inputként adott egy gráf és keressük egy olyan bejárását, amikor minden élt pontosan egyszer érintünk. (Euler-kör)

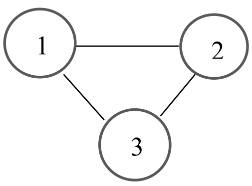
A *konkrét* kiszámítási feladatban az inputot a következőképpen kódoljuk. Megadjuk a csúcspontok számát majd egy # után a csúcsmátrixát sorfolytonosan kódolva. Az outputban vesszővel elválasztva felsoroljuk a csúcspontokat. A felsorolás, akkor helyes, ha egy Euler-körnek felel meg. A leírásban szereplő számokat 10-es számrendszerben kódoljuk.

Eldöntési problémának felfogva a feladatot az inputot az outputtól is a # jellel választjuk el.

Kérdések:

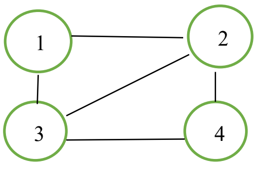
a) Milyen ábécé felett értelmezzük a szavakat? [0123456789#,]

b) Jó szó-e a következő? (Rajzolja fel a megfelelő gráfot!)
3#011101110#3,2,1,3



c) Jó szó-e a következő, ha csak a felsőháromszög részét adjuk meg a kódban a csúcsmátrixnak?

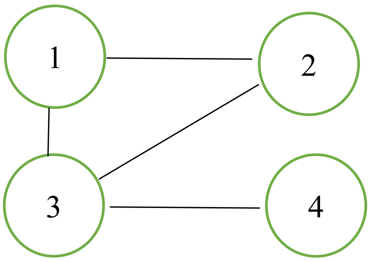
4#0010011010#1,3,4,2,3



3.) Adott a következő *valós* probléma. Egy irányítatlan gráfról el kell dönteni, hogy **összefüggő-e**. Szóproblémává fogalmazva a feladatot, azok a jó szavak, amelyek olyan gráfot kódolnak, amelyben mindenhonnan mindenhova el lehet jutni.
A kódolás a következőképpen történik. Megadjuk, hogy hány csúcs van a gráfban. Ezt a számot unárisan kódoljuk a 0, 1 jelekkel, azaz annyi darab nullát írunk egymás után, amekkora a szám és utána egy egyest. Ezután sorfolytonosan megadjuk a gráf szomszédsági mátrixát, de csak a felsőháromszög részét. Ha i és j között van él, akkor a mátrixba 1-et írunk az i-edik sor j-edik helyére, különben 0-t.

a) Jó szó-e a következő? Indoklásként rajzolja fel a gráfot!

00001011001001 0



b) Adja meg, hogy melyik szó felelne meg annak a konkrét esetnek, amikor 3 csúcspontunk van és az 1-es és 2-es pont között és az 1-es és 3-as pont között van él!

Megoldás: 0001011000