

1. Legyen $\Sigma = \{1\}$, az 1-szalagos Turing-gép átmeneti függvénye $\delta(q_0, 1) = (q_1, 1, J)$, $\delta(q_0, *) = (q_2, *, J)$, $\delta(q_1, 1) = (q_3, 1, J)$, $\delta(q_3, 1) = (q_0, 1, J)$, kezdő állapot a q_0 , elfogadó a q_3 . Mi a gép által elfogadott nyelv?

Megoldás: Vegyük észre, hogy ez a Turing-gép nem változtatja meg a szalag tartalmát, mindig az ott talált karaktert írja vissza és mindig tovább lép jobbra.

Az üres szóra a $\delta(q_0, *) = (q_2, *, J)$ szabállyal q_2 -be lép, és mivel ott nincs átmenet definiálva, megáll. A q_2 nem elfogadó állapot, ezért nem fogadja el az üres szót.

Az 1 karakterek hatására a q_0, q_1, q_3 állapotokon mozog körbe-körbe. Akkor fogad el, ha ez a q_3 -ban áll le, azaz az elfogadott nyelv: $\{1^k \mid k \equiv 2 \pmod{3}\}$.

(És mi a nyelv, ha $\{0, 1\}$ az ábécé?)

2. A 2-szalagos M Turing-gép átmeneti függvényét a következő táblázat írja le, ahol $*$ jelöli a szalagon az üres jelet és q_0 a kezdő állapotot:

állapot	1. szalag	2. szalag	1. szalag	2. szalag	új állapot
q_0	0	*	0	X	q_1
	1	*	1	X	q_1
	*	*	*	H	q_5
q_1	0	*	0	J	q_1
	1	*	1	J	q_1
	*	*	*	B	q_2
q_2	*	0	*	0	q_2
	*	1	*	1	q_2
	*	X	*	X	q_3
q_3	0	0	0	J	q_4
	1	1	1	J	q_4
q_4	0	0	0	H	q_3
	0	1	0	1	q_3
	1	0	1	0	q_3
	1	1	1	1	q_3
	0	*	0	*	q_5
	1	*	1	*	q_5

- (a) Mi a 2. szalag tartalma, amikor a gép q_2 állapotba kerül?
- (b) Mi az $L(M)$ nyelv, ha q_5 az egyetlen elfogadó állapot?
- (c) Legfeljebb hány lépést tehet a gép egy n hosszú bemeneten, mielőtt megáll?

Megoldás: Nézzük, mi történik:

q_0 : Ha nem üres a bemenet, akkor egy X kerül a 2. szalagra (ez jelzi majd a szalag elejét).

q_1 : Lemásolja az 1. szalagon talált karaktert a 2. szalagra. Akkor lép csak ki innen, ha a bemenet végére ért (ekkor, ha az első szalagon a w szó van, akkor a 2. szalagon Xw). Ilyenkor következik a q_2 állapot, ahova úgy lépünk át, hogy az 1. szalagon maradunk az első üres mezőn, a 2. szalagon visszalépünk az utolsónak írt karakterre.

q_2 : Az első szalagon nem mozdulunk, mialatt visszamegyünk a 2. szalag elejére. Végül úgy lépünk át q_3 -ba, hogy az 1. szalagon egyet visszalépünk (az utolsó nem üres mezőre), a 2. szalagon egyet előre lépünk (az első nem X karakterre).

q_3 : Ha ugyanazt látjuk mindkét szalagon, akkor az elsőn nem mozdulunk, a másodikon egyet előre lépünk és átkerülünk q_4 -be. (Első alkalommal tehát az 1. szalag utolsó és a 2. szalag X utáni első karakterét, azaz a w első és utolsó karakterét hasonlítjuk össze. Ha nem egyeznek, akkor ebben a nem elfogadó állapotban leállunk.)

q_4 : Ha nem értük el a 2. szalag végét, akkor semmit nem változtatva az 1. szalagon visszafelé lépünk egyet, a 2. szalagon helyben maradunk és a q_3 -ban folytatjuk. Azaz a q_3 -beli hasonlítás és a q_4 -beli lépés felváltva addig történik, amíg el nem érünk a 2. szalag végére. Ekkor átlépünk a q_5 elfogadó állapotba és a számítás sikeresen véget ért.

Az üres szón érdemi munka nélkül egyből átjutunk az elfogadó állapotba.

(a) Ha w a bemeneti szó, akkor Xw .

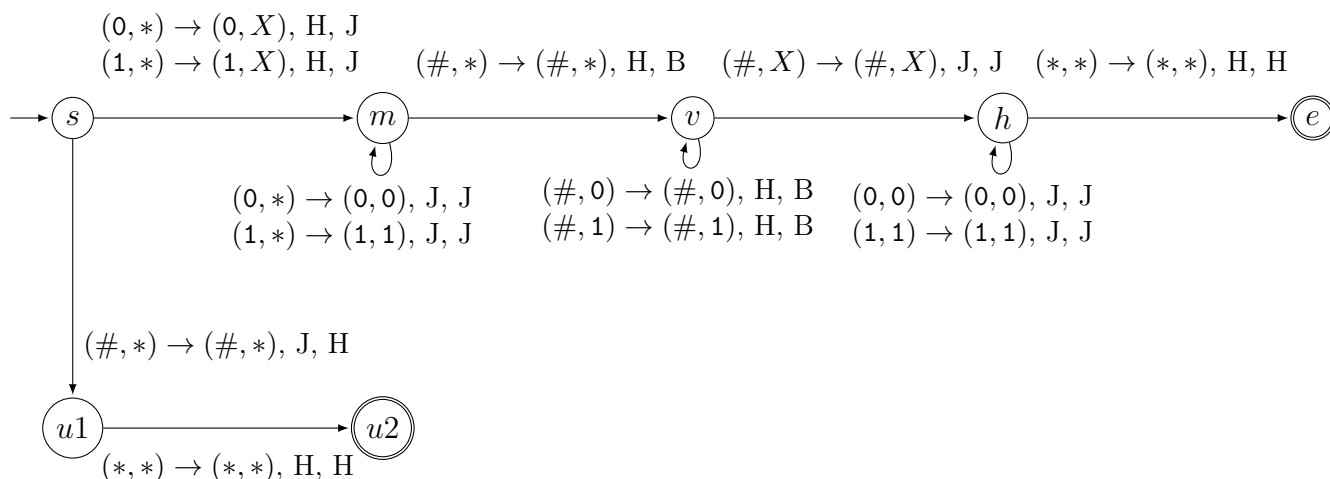
(b) A palindromok nyelve.

(c) Egy n hosszú palindromon a lépések száma: $1 + (n + 1) + (n + 1) + (2n - 1) + 1 = 4n + 3$. Ha a szó nem palindrom, akkor előbb elakadhat (de a másolás és a 2. szalag elejére visszalépegetés akkor is megtörténik, $2n + 3$ lépés akkor is lesz, amennyiben $n > 0$).

3. Adjon Turing-gépet a $\{w\#w \mid w \in \{0,1\}^*\}$ nyelvhez! Adjon felső becslést a Turing-gép lépésszámának nagyságrendjére!

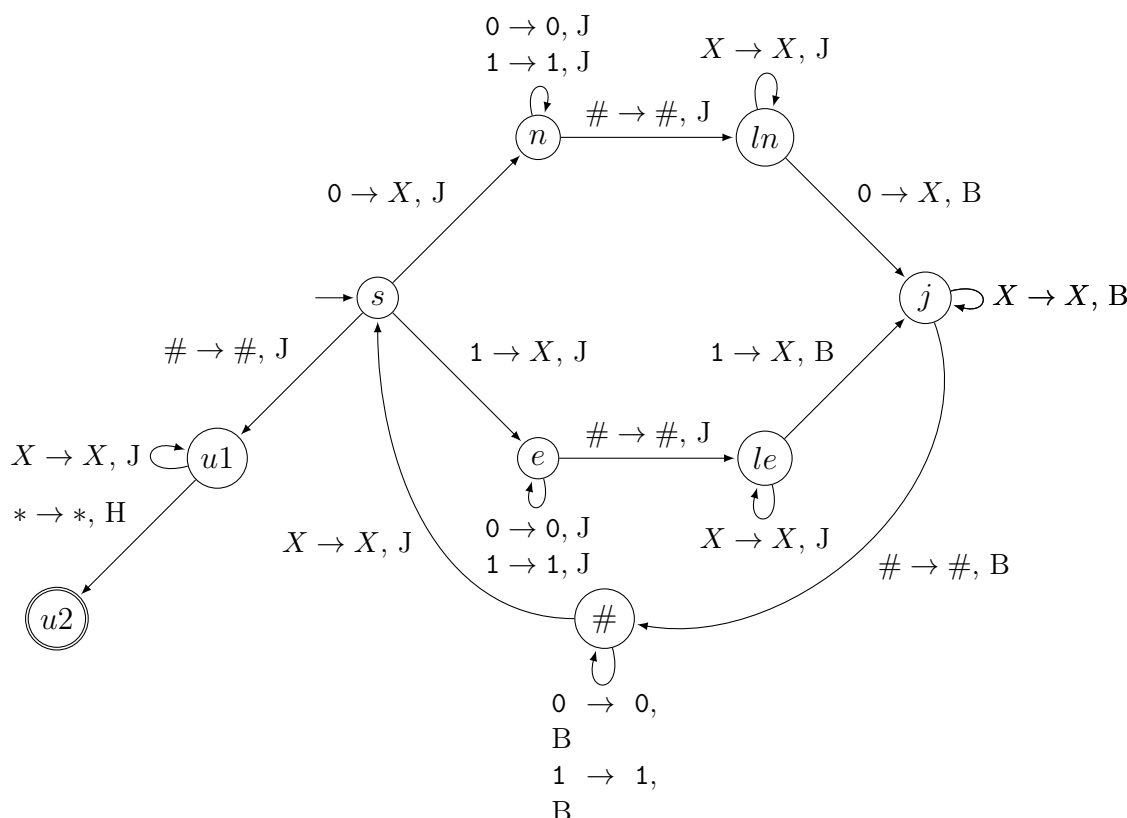
Megoldás: Lássunk előbb egy 2 szalagos gépet! Az ötlet, hogy a 2. szalagra lemásoljuk a szó elejét (az m =másolás állapotban). A $\#$ jelhez érve átlépünk egy újabb v állapotba, amivel visszamegyünk a 2. szalag elejére (v =vissza állapot), és onnan kezdve majd ezt hasonlítjuk az 1. szalagon levő szó második feléhez (h állapot). Arra most is figyelni kell, hogy a 2. szalag elejére tegyünk egy jelet ($s \rightarrow m$), hogy visszafelé jövet tudjuk, hol az eleje.

Azt az esetet is kezelni kell, ha a w az üres szó, az egyetlen $\#$ karakterből álló bemenetet is el kell fogadni, ez történik az $u1$ és $u2$ állapot segítségével.



Lépésszám: Az n hosszú bemeneteken az m , v és h állapotok bármelyikében legfeljebb n lépést töltünk, ezeken kívül csak konstans sok lépés van, tehát a lépésszám $O(n)$. (Könnyű látni, hogy az elfogadott szavakra $\Theta(n)$ is igaz. Miért nem igaz ez minden bemenetre?)

Egy másik megoldás 1 szalaggal: most azt csináljuk, hogy oda-vissza mozogva a szalagon hasonlítjuk a párokat, amiken túl vagyunk, azokat átírjuk X -re. Részletesebben: az első karaktert felülírjuk, és attól függően, hogy ez 0 vagy 1 volt, megyünk az n vagy e állapotba. A két ágon hasonlóan járunk el, az n és e állapotban elmegyünk a $\#$ jelig, majd a bemenet második felében átlépjük az esetleges X jeleket (ln, le). Ha az ez után jövő első karakter megfelel annak, amit várunk (az n ágon 0, az e ágon 1), akkor a szó eddig jó, ezt a karaktert felülírjuk, és visszamegyünk az X -eken a $\#$ jelig, ami után a $\#$ állapotban átlépünk a szó első felében még megmaradt 0 és 1 karakterekre. Amikor elérjük az első X jelet, akkor az s állapotból folytathatjuk a következő karakterpár ellenőrzését. Ha ide érve a $\#$ karaktert látjuk, akkor a szó első felét feldolgoztuk. Ha ilyenkor a második felében sem maradt X -en kívül semmi, akkor elfogadunk. (Ugyanez történik, ha $w = \varepsilon$).



A lépésszám becsléséhez elég azt észrevenni, hogy egyrészt minden állapotban egyfolytában legfeljebb n lépésig maradunk (hiszen a nem X karakterek száma minden körben eggyel csökken), másrészt az s állapotba legfeljebb n -szer térünk vissza. Ezért a lépésszám összesen $O(n^2)$.

4. Vázoljon Turing-gépet az alábbi nyelvekhez! Nem szükséges precízen megadni az átmeneteket, elegendő a működés elvét (részletesen) leírni.

- (a) $\{a^n b^{2n} \mid n \geq 1\}$
- (b) $\{a^i b^j c^k \mid i + j = k \text{ és } i, j, k \geq 1\}$
- (c) $\{a^i b^j c^k \mid i \cdot j = k \text{ és } i, j, k \geq 1\}$

Megoldás: (a) Csináljuk 2 szalaggal: Először, a korábbi példák szerint az első szalagon helyben maradva a második elejére írunk egy X jelet és ezzel átmegyünk egy $A1$ állapotba.

A következőkben az a betűket dolgozzuk fel, mindegyik olvasásakor két újabb a betűt írunk a 2. szalagra. Ezt két lépésben tudjuk megtenni, $\delta(A1, a, *) = (A2, a, a, H, J)$ és $\delta(A2, a, *) = (A1, a, a, J, J)$.

Az első b érkezésekor a $\delta(A1, b, *) = (B, b, *, H, B)$ szabállyal átlépünk a B állapotba, ahol minden, az 1. szalagon olvasott b karakterre balra lépünk a 2. szalagon. Ha egyszerre érünk el az 1. szalagon a szó végére és a 2. szalag elejére írt X jelhez, akkor elfogadunk. (Megoldható 1 szalaggal is az előző feladat mintájára, csak most egy a karakter feldolgozásakor két b karaktert írunk felül.)

(b) Szintén egy 2 szalagos megoldás: Az előzőhöz hasonlóan a 2. szalagra rakunk egy X karaktert, majd rámásoljuk az első szalagon levő a betűket. Az első b karakternél új állapotba lépünk, minden b karakternél továbbra is egy-egy a -t írunk a 2. szalagra. Az első c -től kezdve (egy újabb állapotban) az első szalagon levő minden c -nél balra lépünk a 2. szalagon. Ha egyszerre érünk el az 1. szalagon a szó végére és a 2. szalagon az elejére írt X jelhez, akkor elfogadunk.

(c) Legyen most 3 szalagunk. Először a 2. és 3. szalagra is írunk egy-egy X -et, hogy megjelöljük az elejüket. Ez után a 2. szalagra lemásoljuk az a betűket. Az első b -nél egy újabb állapotba lépünk, az első szalagon helyben maradunk, a másodikonál meg visszamegyünk az X -ig. A cél az, hogy innentől minden, a 2. szalagon levő a -ra lemásoljuk az első szalag b -jeit a 3. szalagra.

A másolás fő lépései: a 2. szalagon levő a hatására átkerülünk az m másoló állapotba, amikor az 1. és 3. szalagon haladunk, a 3. üres mezőibe b betűt írva. Ha az 1. szalagon egy c jön, akkor átlépünk egy v állapotba, ahol az 1. szalagon visszamegyünk az első b elé. Ezen, és a 2. szalagon is jobbra lépve bekerülünk újra az m állapotba.

Amire a 2. szalag a betűjei elfogynak, összesen $i \cdot j$ darab b kerül a 3. szalagra, amiknek a számát a szokásos módon össze tudjuk vetni az 1. szalagon levő c -k számával.

5. Legyen $\Sigma = \{0, 1, +, =\}$. Vázzon egy Turing-gépet ahhoz a nyelvhez, amelyik az olyan $x+y=z$ alakú szavakból áll, ahol $x, y, z \in \{0, 1\}^*$ nem üres bitsorozatok, és ha ezeket binárisan felírt számoknak tekintjük, akkor valóban z az x és y összege. Adjon becslést a Turing-gép lépésszámára!

Megoldás: Az egyszerűség kedvéért használjunk 4 szalagot. Először a bemenetről a $+$ -ig terjedő részt (x) a szokott módon lemásoljuk a 2. szalagra (előtte megjelölve a szalag elejét). Majd az $=$ -ig a következő részt (y) hasonlóan átmásoljuk a 3. szalagra. Álljunk vissza a fejjel a 2. és 3. szalag utolsó nem üres mezőjére (azaz x és y utolsó bitjére), hiszen az összeadást az utolsó bitnél jó kezdeni.

A 4. szalagra írjuk az összeget a szám végéről kezdve. Az összeadásnál két állapotot használunk, az egyik jelképezi, hogy volt átvitel az előző számjegy összeadásából, a másik meg, hogy nem volt. Ennek megfelelően a két állapotban mást-mást írunk a 4. szalagra ugyanannál az olvasott karakterpárnál.

Ha az összeadás véget ért, össze lehet hasonlítani az eredményt, azaz a 4. szalag tartalmát visszafelé, az első szalagon az $=$ utáni (z) résszel.

(Az összeg felírását el is kerülhetjük, ha a keletkező karaktereket egyből a z rész megfelelő karakterével összehasonlítjuk.)

6. Legyen $\Sigma = \{0, 1, +\}$. Vázzon egy Turing-gépet, amelyik az $x+y$ alakú inputon (ahol $x, y \in \{0, 1\}^*$ nem üres bitsorozatok) egy idő után megáll, és az 5. szalagon az x és y binárisan felírt számok összege áll. Adjon becslést a Turing-gép lépésszámára!

Megoldás: Használjunk 5 szalagot. Először a bemenetről a $+$ -ig terjedő részt (x) a szokott módon lemásoljuk a 2. szalagra (előtte megjelölve a szalag elejét). Majd a következő részt (y) hasonlóan átmásoljuk a 3. szalagra. Álljunk vissza a fejjel a 2. és 3. szalag utolsó nem üres mezőjére (azaz x és y utolsó bitjére), hiszen az összeadást az utolsó bitnél jó kezdeni.

A 4. szalagra írjuk az összeget a szám végéről kezdve. Az összeadásnál két állapotot használunk, az egyik jelképezi, hogy volt átvitel az előző számjegy összeadásából, a másik meg, hogy nem volt. Ennek megfelelően a két állapotban mást-mást írunk a 4. szalagra ugyanannál az olvasott karakterpárnál. Végül a 4. szalagon fordított sorrendben levő eredményt másoljuk a jó sorrendben az 5. szalagra.

A lépésszám $O(n)$.

7. Adjon 1-szalagos determinisztikus Turing-gépet a $\{0^{2^n} \mid n \geq 0\}$ nyelvhez! Nem szükséges precízen megadni az átmeneteket, elegendő a működés elvét leírni.

Megoldás: Az első 0-t átírjuk 1-re, ez jelöli majd az input elején az első 0-t. Jobbra haladva minden második 0-t átírjuk x -re. Ha az utolsót nem írtuk át, akkor páratlan hosszú volt az input, elutasítjuk. Ha átírtuk, akkor páros hosszú volt és a megmaradt 0-k száma feleződött. Visszamegyünk az elejére és megint minden második 0-t átírjuk x -re. Az útközben talált x -eken csak jobbra lépünk. Ezt addig ismétljük, amíg lehet. Ha végül nem marad 0, csak az elején az 1-es, akkor elfogadjuk.

8. Az M_1 Turing-gép az $L_1 \subseteq \{0, 1\}^*$, az M_2 Turing-gép az $L_2 \subseteq \{0, 1\}^*$ nyelvet fogadja el, a gépeknek egy-egy szalagja van. Ezek segítségével vázzon egy olyan (akár több szalagos) Turing-gépet, ami az $L_1 \cap L_2$ nyelvet fogadja el!

Megoldás: Az ötlet, hogy egymás után futtatjuk a két gépet. Legyen az M gépnek 3 szalagja, a 2. szalagon az M_1 , a 3. szalagon az M_2 működését követjük majd. Az állapotok az M_1 és az M_2 állapotai és még néhány $:$. Az M_2 elfogadó állapotai lesznek az elfogadó állapotok.

Először lemásoljuk a bemenetet a 2. és 3. szalagra (mindkettőn elé rakunk egy X -et is, X egy új karakter, nem szerepel az M_i gépek szalag ábécéjében).

Ez után lépünk M_1 eredeti kezdő állapotába, és az M_1 szabályait figyelembe véve futtatjuk az M_1 gépet (csak a 2. szalagot használja, a többin helyben marad, nem változtat). Az M_1 -ben az eredetileg elfogadó állapotoknál a hiányzó átmeneteket irányítsuk az M_2 gép eredeti kezdő állapotába, innen az M_2 szabályaival tudunk továbblépni (csak a 3. szalagon változtatva). Az így vázolt Turing-gép csak akkor fogad el egy szót, ha M_1 elfogadó állapotban akadt volna el (ekkor lépünk át az M_2 -részbe) és M_2 is elfogadna, azaz amikor a bemeneti szó az $L_1 \cap L_2$ egy eleme.

Megjegyzés: Ha az M_1 nem áll meg az adott bemeneten, akkor az M gépünk sem fog megállni, és az M_2 -re nem is kerül benne sor. De ez ebben az esetben nem baj, mert a szó biztos nincs a két nyelv metszetében (már L_1 -ben sincs benne).

9. Oldjuk meg a 6. feladatot összeadás helyett szorzás esetére is.

Megoldás: Az iskolában tanult algoritmust kell megvalósítani. A lépésszám $O(n^2)$.