Turing-gépek

1. Készítsen veremautomatát az $S \to \mathtt{a}S\mathtt{a} \mid \mathtt{b}S\mathtt{b} \mid \mathtt{aa} \mid \mathtt{bb} \mid \mathtt{a} \mid \mathtt{b}$ nyelvtanból és adjon meg az $\mathtt{ababa}$ szóhoz egy elfogadó számítást (ha van ilyen)!

   Megoldás: Csak az előadáson tanult sémát kell követni: $Q = \{q_0, q, q_e\}$, q_0 a kezdő állapot, $F = \{q_e\}$, $\Gamma = \Sigma \cup V \cup \{Z\}$. Az állapotátmenetek:

$$\begin{aligned}
\delta(q_0, \varepsilon, Z) &= \{(q, SZ)\}\\
\delta(q, \mathtt{a}, \mathtt{a}) &= \{(q, \varepsilon)\}\\
\delta(q, \mathtt{b}, \mathtt{b}) &= \{(q, \varepsilon)\}\\
\delta(q, \varepsilon, S) &= \{(q, \mathtt{a}S\mathtt{a}), (q, \mathtt{b}S\mathtt{b}), (q, \mathtt{aa}), (q, \mathtt{bb}), (q, \mathtt{a}), (q, \mathtt{b})\}\\
\delta(q, \varepsilon, Z) &= \{(q_e, Z)\}
\end{aligned}$$

   Egy elfogadó számítás konfiguráció sorozata:

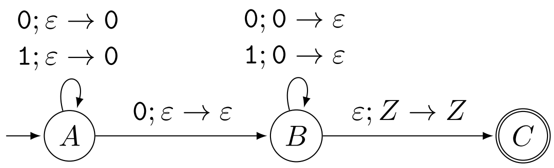
$$(q_0, \mathtt{ababa}, Z) \Rightarrow (q, \mathtt{ababa}, SZ) \Rightarrow (q, \mathtt{ababa}, \mathtt{a}S\mathtt{a}Z) \Rightarrow (q, \mathtt{baba}, S\mathtt{a}Z) \Rightarrow (q, \mathtt{baba}, \mathtt{b}S\mathtt{ba}Z) \Rightarrow$$
$$(q, \mathtt{aba}, S\mathtt{ba}Z) \Rightarrow (q, \mathtt{aba}, \mathtt{aba}Z) \Rightarrow (q, \mathtt{ba}, \mathtt{ba}Z) \Rightarrow (q, \mathtt{a}, \mathtt{a}Z) \Rightarrow (q, \varepsilon, Z) \Rightarrow (q_e, \varepsilon, Z)$$

2. Álljon L azokból a $\{\mathtt{0}, \mathtt{1}\}$ feletti sorozatokból, melyekben a középső karakter $\mathtt{0}$. Igazolja, hogy L környezetfüggetlen nyelv!

   Megoldás: Két lehetőségünk is van: megadhatunk egy környezetfüggetlen nyelvtant a nyelvhez vagy egy veremautomatát.

   A nyelvtan ötlete, hogy ha $w \in L$ legalább 3 hosszú, akkor az első és utolsó karakterét levágva a maradék is L-ben lesz (és ez a két karakter tetszőleges). Ha 3-nál rövidebb, akkor meg csak $w = \mathtt{0}$ lehet. Egy ezt tükröző nyelvtan: $S \to \mathtt{0}S\mathtt{0} \mid \mathtt{0}S\mathtt{1} \mid \mathtt{1}S\mathtt{0} \mid \mathtt{1}S\mathtt{1} \mid \mathtt{0}$.

   A veremautomata meg működhet úgy, hogy $\mathtt{0}$-t rak a verembe, amíg el nem éri a közepét (ezt nem kell tudni, mikor következik be, erre jó a nemdeterminizmus!), ha ez $\mathtt{0}$, akkor a továbbiakban kiszedegeti a $\mathtt{0}$-kat a veremből. Akkor fogad el, ha a szó végénél a veremből is épp kifogytak a $\mathtt{0}$ karakterek.



3. Legyen L_r egy tetszőleges reguláris nyelv és legyen L_c egy tetszőleges környezetfüggetlen nyelv.
   (a) Mutasson olyan példát, amikor $L_r \cap L_c$ nem reguláris!
   (b) Igazolja, hogy $L_r \cap L_c$ mindig környezetfüggetlen!
   (c) Mutasson olyan példát, amikor L_1 és L_2 is környezetfüggetlen, de $L_1 \cap L_2$ nem az!

   Megoldás: (a) Ha $L_r = \{\mathtt{a}, \mathtt{b}\}^*$ és $L_c = \{a^n b^n : n \geq 0\}$, akkor $L_r \cap L_c = L_c$, ami nem reguláris.

   (b) Legyen $M_1 = (Q_1, \Sigma, q_1, F_1, \delta_1)$ egy DVA az L_r nyelvhez, és $M_2 = (Q_2.\Sigma, \Gamma, q_2, Z_0, F_2, \delta_2)$ egy veremautomata az L_c nyelvhez. A kettőből elkészíthetjük azt az M veremautomatát, amelynek állapothalmaza $Q_1 \times Q_2$, kezdőállapota (q_1, q_2), elfogadó állapotainak halmaza $F_1 \times F_2$, az átmeneteiben az állapot első koordinátájában δ_1 szerint lép, a másodikban δ_2 szerint. Amennyiben a veremautomata nem olvas a bemenetről, akkor az állapot első koordinátája nem változik (M_1 nem lép). Ha a veremautomata nem tud lépni, akkor M számítása elakad. A veremben mindig történjen az, ami az M_2 megfelelő lépésében történik.

   Ez az automata lényegében párhuzamosan futtatja az M_1 és M_2 automatát, akkor fogad el, ha mindkettő elfogadó.

   (c) Az $L_1 = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^k : n \geq 0\}$ és $L_2 = \{\mathtt{a}^n\mathtt{b}^k\mathtt{c}^n : n \geq 0\}$ nyelvek környezetfüggetlenek (lásd 3. gyak. 9(d)), a metszetük az $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n : n \geq 0\}$ nyelv, ami nem CF.
   (A (b)-ben vázolt konstrukció az automatákra most azért nem működik, mert két vermet kellene eggyel szimulálni.)

4. Legyen $\Sigma = \{\mathtt{1}\}$, az 1-szalagos Turing-gép átmeneti függvénye $\delta(q_0, \mathtt{1}) = (q_1, \mathtt{1}, J)$, $\delta(q_0, *) = (q_2, *, J)$, $\delta(q_1, \mathtt{1}) = (q_3, \mathtt{1}, J)$, $\delta(q_3, \mathtt{1}) = (q_0, \mathtt{1}, J)$, 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: $\{1^k : k \equiv 2 \pmod 3\}$.

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

5. 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	H	X	J	q_1
	1	$*$	1	H	X	J	q_1
	$*$	$*$	$*$	H	$*$	H	q_5
q_1	0	$*$	0	J	0	J	q_1
	1	$*$	1	J	1	J	q_1
	$*$	$*$	$*$	H	$*$	B	q_2
q_2	$*$	0	$*$	H	0	B	q_2
	$*$	1	$*$	H	1	B	q_2
	$*$	X	$*$	B	X	J	q_3
q_3	0	0	0	H	0	J	q_4
	1	1	1	H	1	J	q_4
q_4	0	0	0	B	0	H	q_3
	0	1	0	B	1	H	q_3
	1	0	1	B	0	H	q_3
	1	1	1	B	1	H	q_3
	0	$*$	0	H	$*$	H	q_5
	1	$*$	1	H	$*$	H	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.

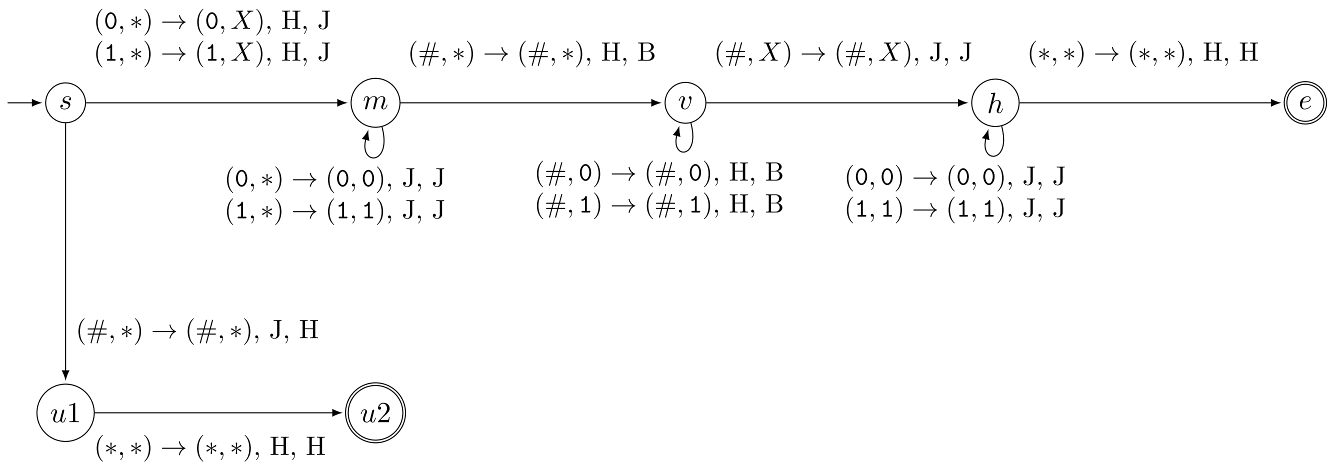
Az üres szón érdemi munka nélkül egyből átjutunk az elfogadó állapotba.

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

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

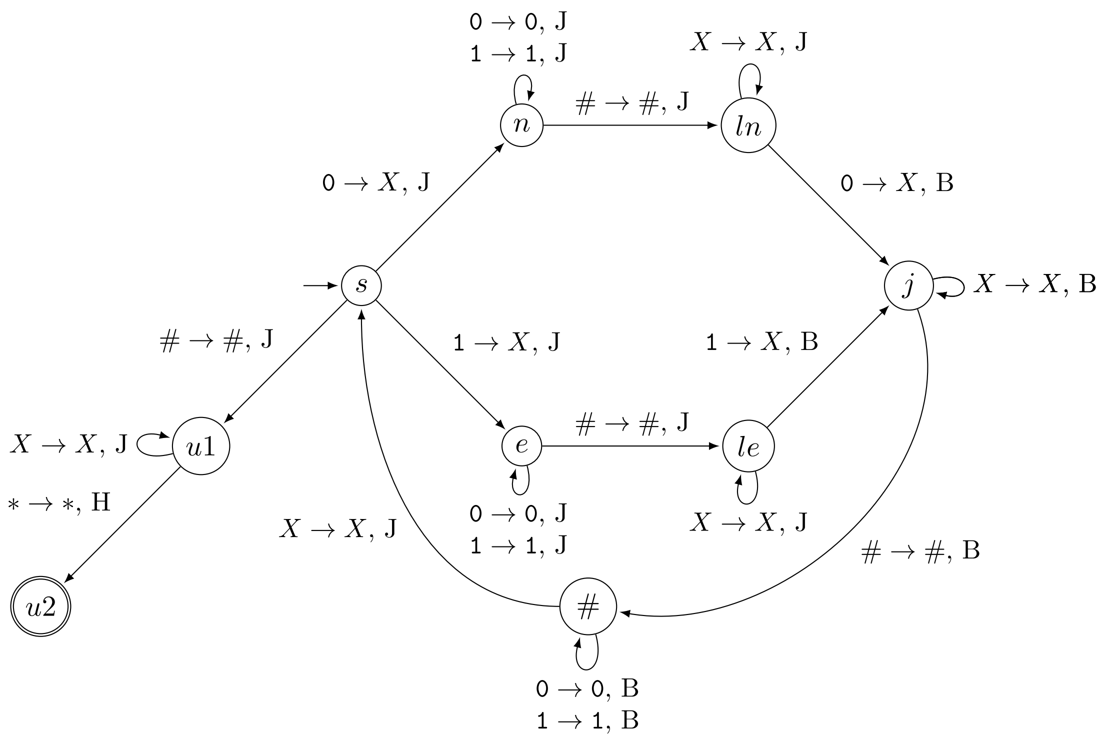
Megoldás: Lássunk előbb egy 2 szalagos gépet! Az ötlet, hogy a 2. szalagra lemásoljuk a szó elejét (az m=másolás állapotban). A # jelhez érve átlépünk egy újabb v állapotba, amivel visszamegyünk a 2. szalag elejére (v=vissza állapot), és onnan kezdve majd ezt hasonlítjuk az 1. szalagon levő szó második feléhez (h állapot). Arra most is figyelni kell, hogy a 2. szalag elejére tegyünk egy jelet ($s \to m$), 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 átlépkedünk a szó első felében még megmaradt `0` és `1` karaktereken. 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, másrészt az s állapotba kevesebb, mint n-szer térünk vissza (hiszen a nem X karakterek száma minden körben eggyel csökken). Ezért a lépésszám összesen $O(n^2)$.

7. Vázoljon Turing-gépet az alábbi nyelvekhez! Nem szükséges precízen megadni az átmeneteket, elegendő a működés elvét (részletesen) leírni.
   (a) $\{\mathtt{a}^n\mathtt{b}^{2n} \mid n \geq 1\}$
   (b) $\{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k : i + j = k \text{ és } i, j, k \geq 1\}$
   (c) $\{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k : i \cdot j = k \text{ és } i, j, k \geq 1\}$

Megoldás: (a) Csináljuk 2 szalaggal: Először, a korábbi példák szerint az első szalagon helyben maradva a második elejére írjunk 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, $\delta(A1, \mathtt{a}, *) = (A2, \mathtt{a}, \mathtt{a}, H, J)$ és $\delta(A2, \mathtt{a}, *) = (A1, \mathtt{a}, \mathtt{a}, J, J)$.

Az első b érkezésekor a $\delta(A1, \mathtt{b}, *) = (B, \mathtt{b}, *, H, B)$ 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 másodiknál 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.

8. Legyen $\Sigma = \{\mathtt{0}, \mathtt{1}, +, =\}$. Vázoljon egy Turing-gépet ahhoz a nyelvhez, amelyik az olyan $x{+}y{=}z$ alakú szavakból áll, ahol $x, y, z \in \{\mathtt{0}, \mathtt{1}\}^*$ 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 összeadát 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.)

9. Az M_1 Turing-gép az $L_1 \subseteq \{\mathtt{0}, \mathtt{1}\}^*$, az M_2 Turing-gép az $L_2 \subseteq \{\mathtt{0}, \mathtt{1}\}^*$ nyelvet fogadja el, a gépeknek egy-egy szalagja van. Ezek segítségével vázoljon 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).