

Számítási modellek

7: Turing gépek

Algoritmus modellek

- Az 1930-as évektől megnőtt az igény az algoritmusok matematikai modelljének létrehozására.
- Több kísérlet:
 - Kurt Gödel: rekurzív függvények
 - Alonso Church: λ -kalkulus
 - Alan Turing: Turing gép
- Az 1930-as évek második felétől számos tétel született, amelyek ezeknek a modelleknek az azonos számítási erejét állapították meg.
- Később sok más számítási modellnél bebizonyosodott, hogy számítási erejük megegyezik a Turing gépekkel. Például:
 - 0-típusú grammatikák
 - veremautomata 2 vagy több veremmel
 - RAM
 - C, Java, Python, stb...

Church-Turing tézis

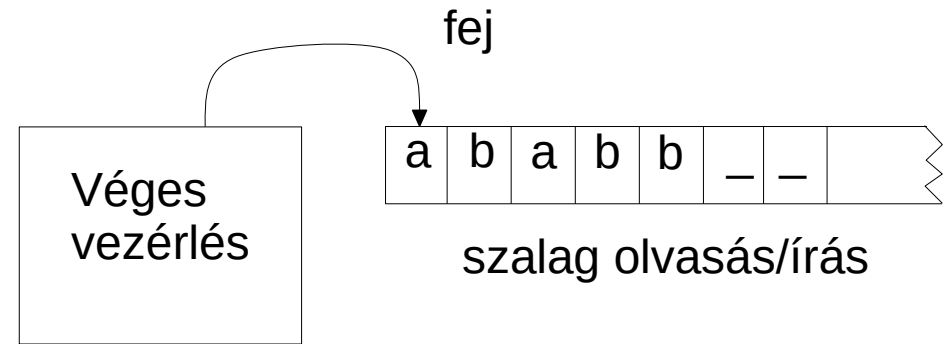
- Nem ismerünk olyan számítási modellt, amelyről tudnánk, hogy erősebb a Turing gépnél, és a több modellre bizonyított, hogy gyengébb, vagy ekvivalens.

A 30-as években megfogalmazásra került a következő:

- **Church-Turing tézis:** Minden formalizálható probléma, ami megoldható algoritmussal, az megoldható Turing géppel is.
- **NEM TÉTEL!**

Turing Gépek (TG)

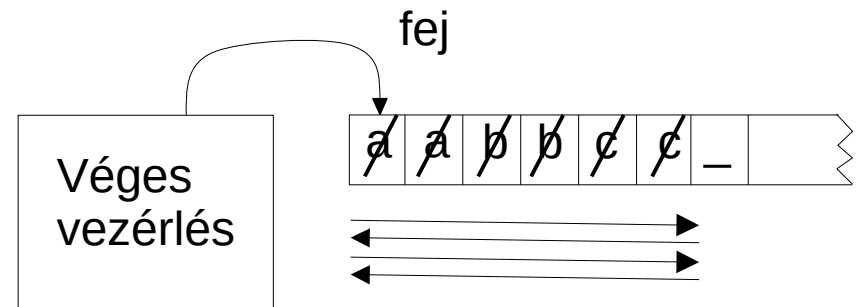
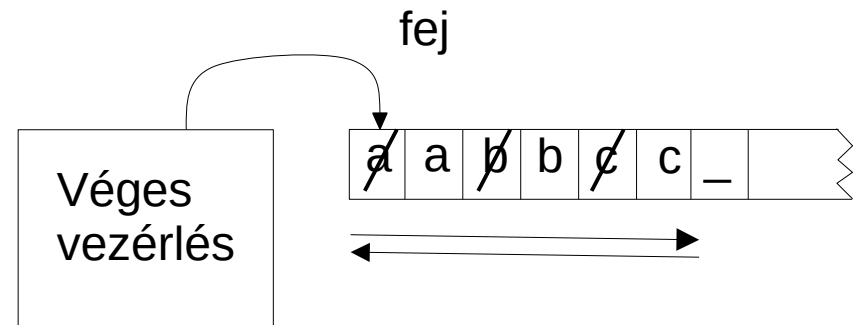
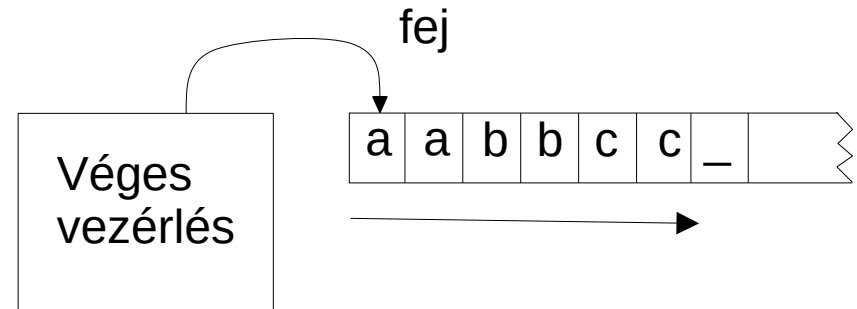
- Alan Turing vezette be 1936-ban
- A fej tud írni és olvasni
- A fej kétirányú (balra vagy jobbra mozgatható)
- A szalag végtelen (jobbra)
- Az inputot végtelen sok „_”
üres karakter követi
- Bármikor elfogadhatja vagy elutasíthatja (nem csak az input végén)



Turing Gép (TG)

TG példa: TG $L = \{ a^n b^n c^n \mid n \geq 0 \}$ felsmeréséhez.

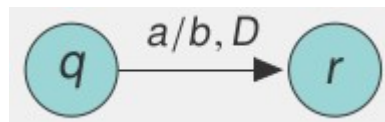
- 1 Olvasás (scan) jobbra, amíg el nem érjük $_$ -t közben ellenőrizzük, hogy az input $\{a,b,c\}^*$ -ban van-e. Ha nem, elutasítjuk.
- 2 A fejet visszavisszük a szalag bal végére.
- 3 Olvasás (scan) jobbra, áthúzzunk egy a -t, egy b -t, és egy c -t.
- 4 Ha minden szimbólum közül az utolsót áthúztuk, fogadjuk el.
- 5 Ha valamelyik szimbólum közül az utolsót áthúztuk, de más szimbólum közül nem, utasítsuk el.
- 6 Ha áthúzatlan szimbólumok maradnak, visszamegyünk a szalag bal végére és ismételjük a 3 ponttól.



TG - Formális Definíció

- Egy **Turing gép (TG)** egy 7-es $M=(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, ahol Q, Σ, Γ véges halmazok
 - Q : állapotok halmaza
 - Σ : input ábécé (az üres szimbólum $_ \notin \Sigma$)
 - Γ : szalag ábécé ($\Sigma \subseteq \Gamma, _ \in \Gamma$)
 - $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ átmenet függvény ($L = \text{Left}, R = \text{Right}$)
 - determinisztikus
 - $q_0 \in Q$: kezdő állapot
 - $q_{accept} \in Q$: elfogadó állapot
 - $q_{reject} \in Q$: elutasító állapot, $q_{reject} \neq q_{accept}$.

- Átmenetdiagram:



megfelel $\delta(q, a) = (r, b, D)$ átmenetnek
($q, r \in Q, a, b \in \Gamma, D \in \{L, R\}$)

TG - Számítás

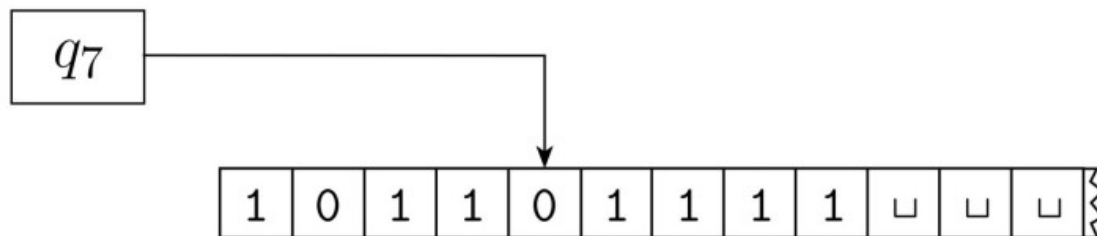
- Az $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ TG a következőképp végzi a számítást.
- Az inputszó: $w = w_1w_2\dots w_n \in \Sigma^*$.
 - az inputszalag első n celláján helyezkedik el,
 - a többi cella üres (a $_$ szimbólumot tartalmazza).
- A legelső $_$ szimbólum jelzi az input végét.
- Az író-olvasó fej az inputszalag legbaloldalibb celláján áll.
- A számítás az átmenet függvény szerint történik.
 - Ha a fej elérte a szalag bal oldali végét, akkor egy helyben marad, függetlenül attól, hogy az átmenet függvény bal oldalra történő elmozdulást jelzett (L).
- A számítás befejeződik, ha M q_{accept} or q_{reject} állapotba lép. Ekkor M megáll.
- Ha egyik eset sem következik be, akkor M sohasem áll meg (loop).

TG - Elfogadás, elutasítás

- Egy M TG 3 lehetséges kimenete egy w inputtal:
 - Elfogadja (q_{accept} állapotba lép)
 - Elutasítja megállással (q_{reject} állapotba lép)
 - Elutasítja úgy, hogy sohasem áll meg (“loop”).

TG - Konfiguráció

- A TG **konfigurációja** egy uqv alakú szó, ahol $u, v \in \Gamma^*$ és $q \in Q$.
 - uv : a szalag aktuális tartalma
 - q : aktuális állapot
 - a fej v első szimbólumán áll
 - v utolsó szimbólumát csak $_$ szimbólumok követik a szalagon.
- Példa: $1011q_701111$



TG - Konfiguráció átmenet

- $C_1 \vdash C_2$, ha C_1 konfigurációból C_2 konfiguráció **egy lépésben** (közvetlenül) **elérhető**:
Legyen $a, b, c \in \Gamma$ és $u, v \in \Gamma^*$. Legyen $q_i, q_j \in Q$.
 - $uaq_i b v \vdash uq_j a c v$ ha $\delta(q_i, b) = (q_j, c, L)$,
 - $uaq_i b v \vdash uacq_j v$ ha $\delta(q_i, b) = (q_j, c, R)$.
- Speciális esetek:
 - A szalag bal oldali végén
 - $q_i b v \vdash q_j c v$ ha $\delta(q_i, b) = (q_j, c, L)$,
(megakadályozza, hogy a TG leessen a szalag bal végén)
 - $q_i b v \vdash cq_j v$ ha $\delta(q_i, b) = (q_j, c, R)$.
 - Az input jobb oldali végén
 - uaq_i ekvivalens a $uaq_i _$ konfigurációval.

TG - Elfogadás, Elutasítás, Megállás

- Az M TG **kezdőkonfigurációja** w inputtal: q_0w .
- **Elfogadó konfiguráció:** a konfiguráció állapota q_{accept} .
- **Elutasító konfiguráció:** a konfiguráció állapota q_{reject} .
- Elfogadó és elutasító konfigurációk **megállási konfigurációk**
 - Nem vezetnek további konfigurációkhoz.
- δ megadható mint: $\delta: Q' \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$, ahol $Q' = Q \setminus \{q_{accept}, q_{reject}\}$.

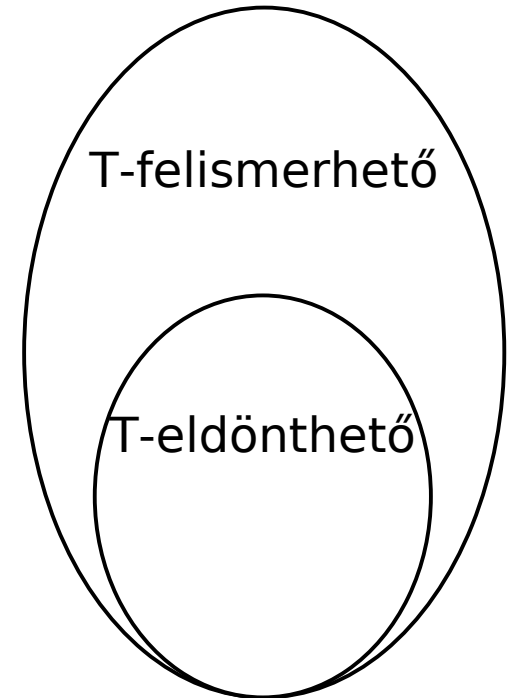
TG - elfogadott nyelv

- Egy M TG a w inputszót **elfogadja**, ha létezik olyan C_1, C_2, \dots, C_k konfiguráció-sorozat, ahol
 1. C_1 az M kezdő konfigurációja w inputtal,
 2. $C_i \vdash C_{i+1}$, $1 \leq i \leq k - 1$ és
 3. C_k egy elfogadó konfiguráció.
- Azon szavak halmazát, melyet M elfogad/felismer, az M által **elfogadott/felismert nyelv**nek nevezzük és $L(M)$ -mel jelöljük, azaz
 - $L(M) = \{ w \mid M \text{ elfogadja } w\text{-t} \}$.

TG - Felismerés, Eldöntés

- Egy M TG **felismeri** L -t, ha $L = L(M)$.
- Egy L nyelv **Turing-felismerhető**, ha $L = L(M)$ valamely M TG-re.
- Turing-felismerhető nyelvek a **rekurzívan felsorolható** nyelvek.

- Egy M TG egy **eldöntő**, ha M minden inputtal megáll.
- M **eldönti** L -t, ha $L = L(M)$ és M egy eldöntő.
- Egy L nyelv **Turing-eldönthető**, ha $L = L(M)$ valamely M eldöntő TG-re.
- Turing-eldönthető nyelvek a **rekurzív** nyelvek.



Eldöntő TG Példa

- Példa: TG M , amely eldönti $L = \{0^{2^n} \mid n \geq 0\}$ -t.

- M a w inputtal:

1. Olvassuk (scan) a szalagot balról jobbra és húzzunk át minden második 0-t.

2. Ha az 1. pontban a szalag egyetlen 0-t tartalmazott, elfogadjuk.

3. Ha az 1. pontban a szalag több mint egy 0-t tartalmazott és a 0-k száma páratlan volt, elutasítjuk.

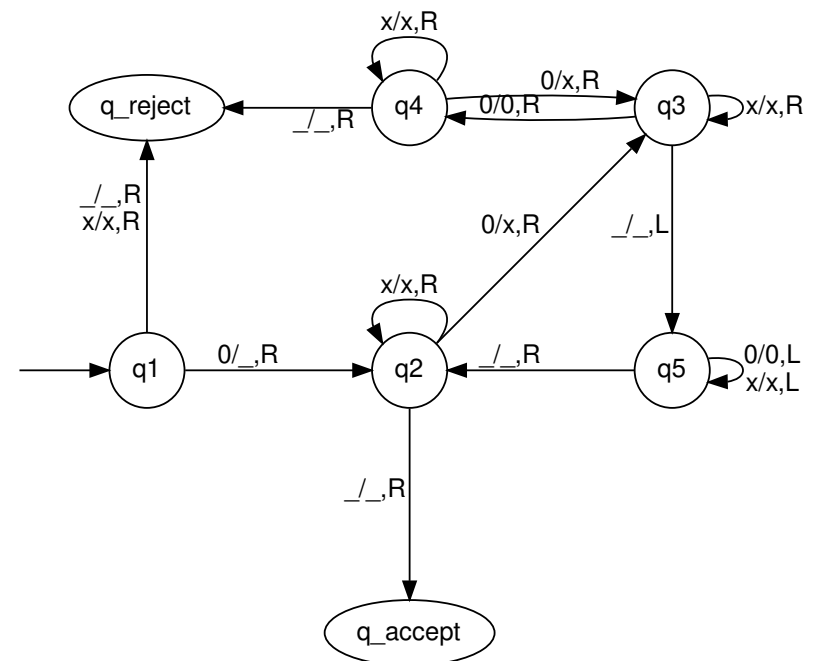
4. Vigyük vissza a fejet a szalag bal végéhez.

5. Go to 1.

- Formális leírás:

$M = (Q, \Sigma, \Gamma, \delta, q_1, q_{accept}, q_{reject})$:

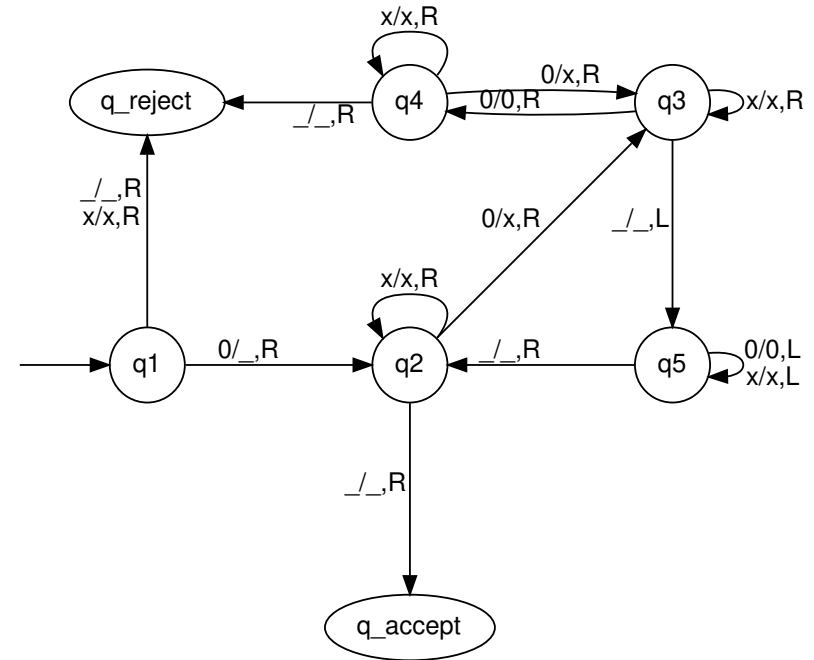
- $Q = \{q_1, q_2, q_3, q_4, q_5, q_{accept}, q_{reject}\}$,
- $\Sigma = \{0\}$, $\Gamma = \{0, x, _ \}$.
- δ az állapot átmenet diagrammal adott:



Eldöntő TG Példa

- Példa: TG M , amely eldönti

$$L = \{0^{2^n} \mid n \geq 0\}.$$



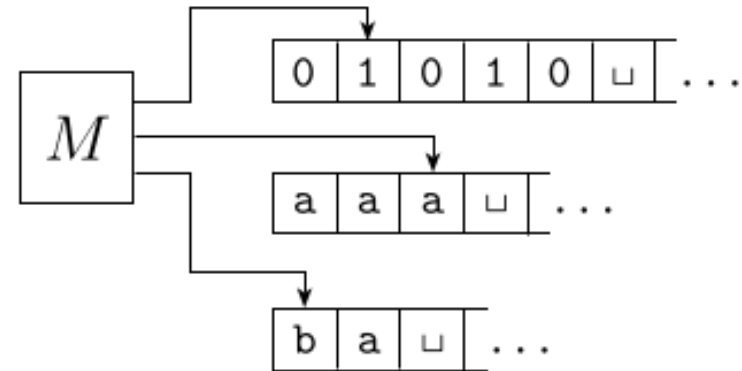
- M futása a 0000 inputtal:

$q_1 0000$	$\sqcup q_5 x 0 x \sqcup$	$\sqcup x q_5 x x \sqcup$
$\sqcup q_2 000$	$q_5 \sqcup x 0 x \sqcup$	$\sqcup q_5 x x x \sqcup$
$\sqcup x q_3 00$	$\sqcup q_2 x 0 x \sqcup$	$q_5 \sqcup x x x \sqcup$
$\sqcup x 0 q_4 0$	$\sqcup x q_2 0 x \sqcup$	$\sqcup q_2 x x x \sqcup$
$\sqcup x 0 x q_3 \sqcup$	$\sqcup x x q_3 x \sqcup$	$\sqcup x q_2 x x \sqcup$
$\sqcup x 0 q_5 x \sqcup$	$\sqcup x x x q_3 \sqcup$	$\sqcup x x q_2 x \sqcup$
$\sqcup x q_5 0 x \sqcup$	$\sqcup x x q_5 x \sqcup$	$\sqcup x x x q_2 \sqcup$
		$\sqcup x x x \sqcup q_{\text{accept}}$

TG Változatok - Többszalagos TG

- Többszalagos TG:

- Minden szalag saját író-olvasó fejjel rendelkezik.
- Kezdetben az input az 1. szalagon van, a többi szalag üres.
- Az átmenet függvény lehetővé teszi a szalagokról való egyidejű olvasást, a szalagokra történő egyidejű írást, és az író-olvasó fejek egyidejű léptetése:



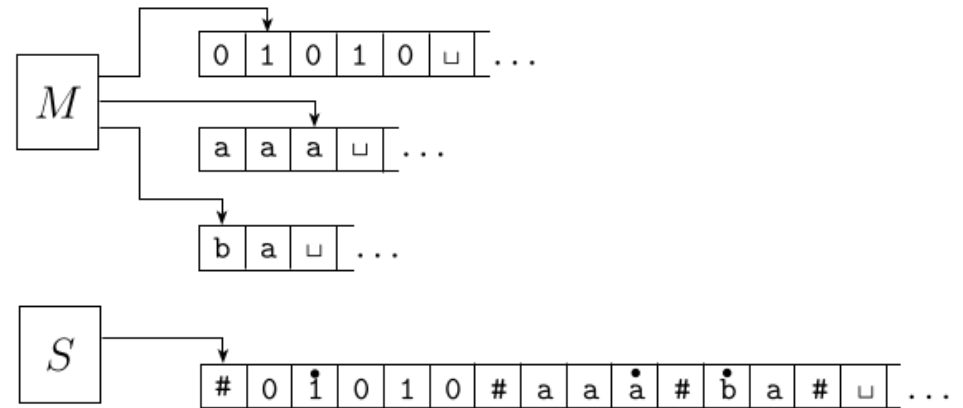
- $\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R, S\}^k$,
ahol k a szalagok száma,
S: "stay put" a fej helyben marad. Helyettesíthető egy jobbra lép, majd egy balra lép átmenettel.
- azaz $\delta(q_i, a_1, \dots, a_k) = (q_j, b_1, \dots, b_k, L, R, \dots, L)$

Egyszalagos, Többszalagos TG

Tétel: Minden többszalagos TG-hez konstruálható azzal ekvivalens egyszalagos TG.

Biz.: Egy többszalagos M TG átalakítása egy ekvivalens egyszalagos S TG-pé.

- S szimulálja M -et úgy, hogy S egyetlen szalagján tárolja az M gép $k \geq 2$ darab szalagjának tartalmát blokkokban. A blokkok egy új $\#$ szimbólummal vannak elválasztva.

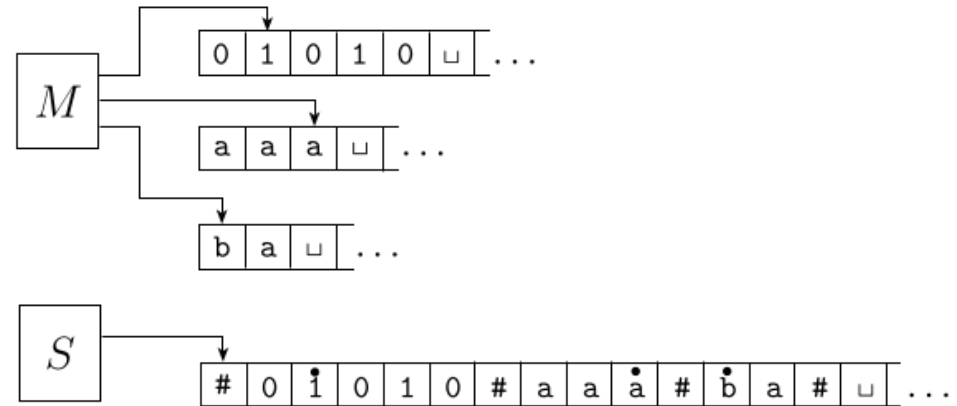


- Az M gép k szalagjának tartalma S egy szalagján: $\#w\#_ \# _ \# _ \dots _ \#$, ahol w az M első szalagjának tartalma.
- A fejek pozícióit pontozott szimbólumokkal tartjuk nyilván (a pontozott szimbólumok újonnan bevezetett szimbólumok).

Egyszalagos, Többszalagos TG

Biz. (folyt.):

- S működése:
- 1. M minden lépéséhez:
 - a: Pásztázzuk végig (scan) a szalagot, hogy megtaláljuk a pontozott szimbólumokat.
 - b: Pásztázzuk végig (scan) újra, hogy aktualizáljunk az M gép δ -jának megfelelően.
 - c: Sifteljünk, hogy helyet csináljunk, ha kell.
- 2. Elfogadjuk/elutasítjuk ha M azt teszi.



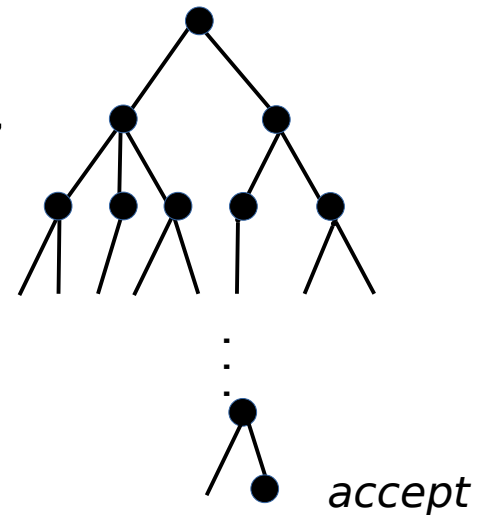
Nemdeterminisztikus TG

- Egy **nemdeterminisztikus TG** (NTG) hasonló a determinisztikus TG-hez, kivéve az átmeneti függvényét $\delta : Q \times \Gamma \rightarrow P(Q \times \Gamma \times \{L, R\})$.

Tétel: Minden N NTG-hez van azzal ekvivalens determinisztikus TG M .

Biz.: Ötlet:

- Egy NTM N számítása w inputtal reprezentálható egy fával (számítási fa)
 - A csomópontok megfelelnek a konfigurációknak,
 - A gyökér megfelel a kezdő konfigurációnak.
- M próbálja az N nemdeterminisztikus számításának összes lehetséges ágát. (BFS)
- Ha M elér egy elfogadó állapotot egy ágon, M elfogad és megáll. Különben, M nem áll meg. \square



Nemdeterminisztikus TG

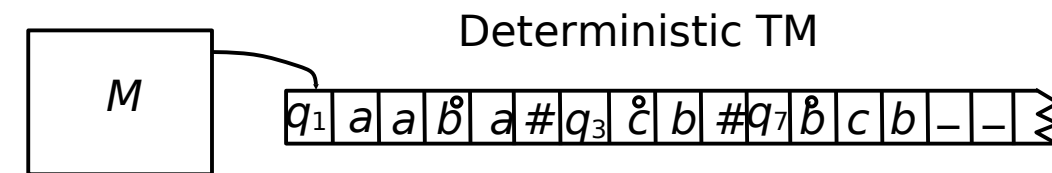
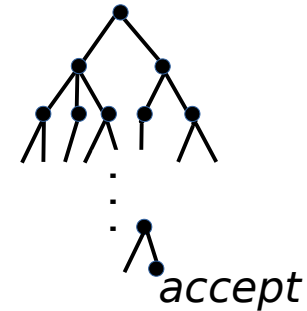
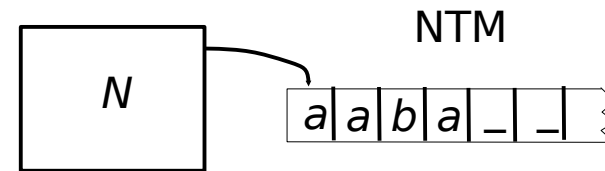
Tétel: Minden N NTG-hez van azzal ekvivalens determinisztikus TG M .

Biz.:

- M szimulálja N -t úgy, hogy minden egyes szál szalagját külön „blokkban” tárolja a szalagon.

A blokkok egy új $\#$ szimbólummal vannak elválasztva.

- Minden blokkban tárolja a fej pozícióját és
- a szálhoz tartozó állapotot.
- Ha egy szál elágazik, M lemásolja a blokkot.
- Ha egy szál elfogad, M elfogad. \square

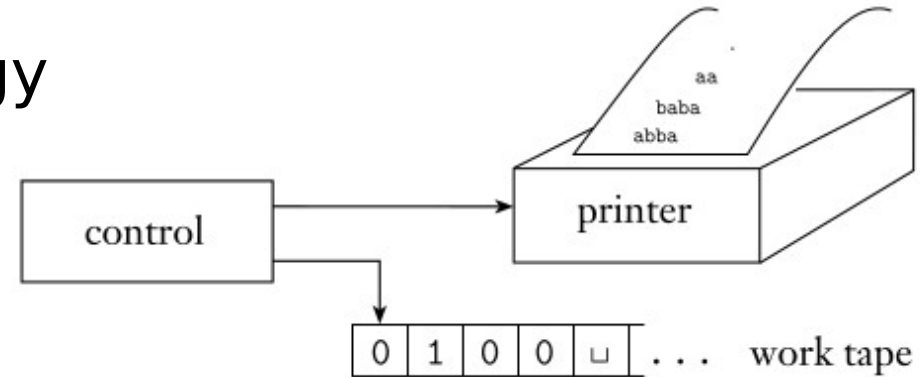


Következmények

- **Korollár:** Egy nyelv Turing-felismerhető, akkor és csak akkor, ha van azt felismerő többszalagos TG.
- **Korollár:** Egy nyelv Turing-felismerhető, akkor és csak akkor, ha van azt felismerő nemdeterminisztikus TG.
- **Megjegyzés:** Az előző tétel bizonyítását módosíthatjuk úgy, hogy ha N a nemdeterminisztikus számítás minden ágán megáll, akkor M is megáll. Egy NTG N egy **eldöntő** ha N minden inputra minden ágon megáll.
- **Korollár:** Egy nyelv akkor és csak akkor eldönthető, ha van azt eldöntő nemdeterminisztikus TG.

Turing Enumerátor

- Egy **Turing Enumerátor** egy determinisztikus TG egy nyomtatóval.
- Egy üres szalaggal kezd és ki tudja nyomtatni a w_1, w_2, w_3, \dots szavakat. Lehetséges, hogy örökké fut.
- A felsorolt **nyelv** azon **szavak halmaza, melyeket kinyomtat**. (Egy generátor, nem egy felismerő.)
- E enumerátorhoz: $L(E) = \{w \mid E \text{ kinyomtatja } w\text{-t}\}$.



Turing Enumerátor

Tétel: Egy L nyelv akkor és csak akkor Turing-felismerhető, ha $L=L(E)$ valamely E Turing enumerátorra.

Biz.:

- (\leftarrow) E enumerátor konvertálása ekvivalens M TG-re.
 - M működése w inputtal:
 - Szimuláljuk E -t (üres inputtal).
 - Amikor E kiír egy x szót, ellenőrizzük, hogy $x=w$.
 - Elfogadjuk, ha $=$, máskülönben folytatjuk a szimulálást.
- (\rightarrow) TM M konvertálása ekvivalens E enumerátorra.
 - $E =$ Szimuláljuk M -et minden $w_i \in \Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, \dots\}$ szóra
 - Amikor M elfogadja w_i -t, kiírjuk w_i -t.
 - Folytassuk a következő w_i -vel.
 - Probléma: Mi van, ha M sohasem áll meg w -vel?
 - Megoldás: Szimuláljuk M TG i lépését w_1, w_2, \dots, w_i szavakkal, $i=1, 2, \dots$
Kiírjuk azokat w_j -ket, $j \leq i$, amelyeket elfogad M . □

TG és 0. Típusú Grammatikák

Tétel: Minden 0. típusú G grammatikához megadható egy NTG, amely felismeri az $L(G)$ -t.

Biz.:

- Legyen M egy TG 3 szalaggal,
 - az első szalag az M inputját tartalmazza,
 - a második szalag tartalmazza G szabályait,
 - a harmadik szalagon mindig van egy α sztring (kezdetben a G kezdőszimbóluma).
- 1. M nemdeterminisztikusan választ egy $p \rightarrow q$ szabályt és egy pozíciót α -ban.
- 2. Ha a pozíció p -vel kezdődik, azaz $\alpha = xpy$, akkor kicseréli ki p -t q -ra, az új α sztring xqy lesz.
- 3. Ha az 1. és 3. szalag tartalma egyenlő, akkor M megáll q_{accept} állapottal. Máskülönben, ismételjük 1-től.
- Világos, $L(M) = L(G)$ □

Korollár: Az előző tétel alapján egy determinisztikus TG is megadható.

TG és 0. Típusú Grammatikák

Tétel: Minden $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ determinisztikus TG-hez megadható egy G grammatika, amely $L(M)$ nyelvet generálja.

Biz.:

- G mondatai M konfigurációit kódolják. G az ellenkező irányba halad. Nemdeterminisztikusan létrehoz egy elfogadó konfigurációt, majd abból megpróbál egy kezdő konfigurációt levezetni.
- Legyen $G = ((\Gamma \setminus \Sigma) \cup Q \cup \{S, A, \triangleright, \triangleleft\}, \Sigma, P, S)$, ahol P :

$$1) S \rightarrow \triangleright A q_{accept} A \triangleleft$$

$$2) A \rightarrow aA \mid \varepsilon \quad (\forall a \in \Gamma)$$

$$3) bq' \rightarrow qa, \text{ ha } \delta(q, a) = (q', b, R)$$

$$4) q'b \rightarrow qa, \text{ ha } \delta(q, a) = (q', b, S)$$

$$5) q'cb \rightarrow cqa, \text{ if } \delta(q, a) = (q', b, L) \quad (\forall c \in \Gamma)$$

$$6) _ \triangleleft \rightarrow \triangleleft, \triangleleft \rightarrow \varepsilon, \triangleright _ \rightarrow \triangleright, \triangleright q_0 \rightarrow \varepsilon$$

TG és 0. Típusú Grammatikák

Biz. (folyt.):

$$1) S \rightarrow \triangleright A q_{\text{accept}} A \triangleleft$$

$$2) A \rightarrow aA \mid \varepsilon \quad (\forall a \in \Gamma)$$

$$3) bq' \rightarrow qa, \text{ ha } \delta(q, a) = (q', b, R)$$

$$4) q'b \rightarrow qa, \text{ ha } \delta(q, a) = (q', b, S)$$

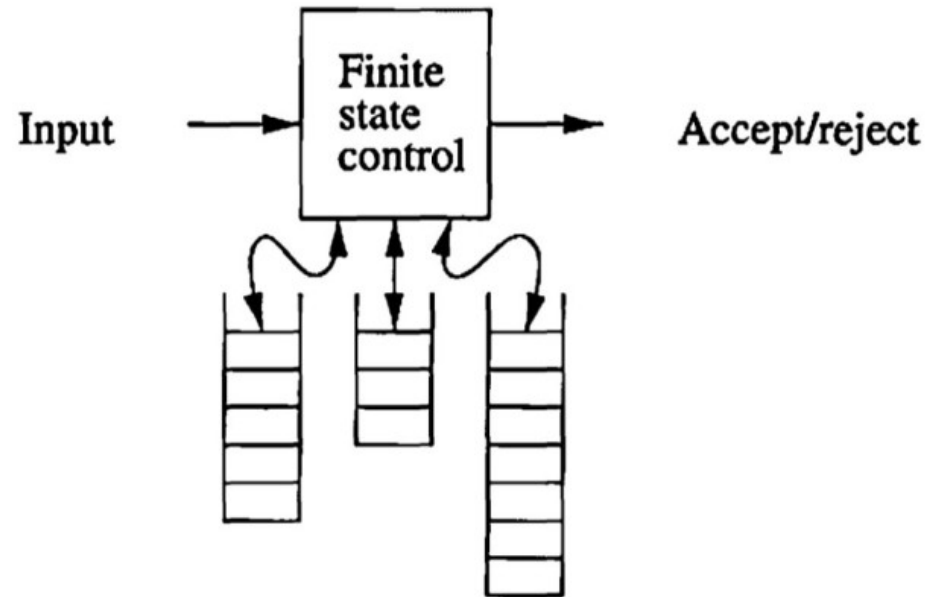
$$5) q'cb \rightarrow cqa, \text{ if } \delta(q, a) = (q', b, L) \quad (\forall c \in \Gamma)$$

$$6) _ \triangleleft \rightarrow \triangleleft, \triangleleft \rightarrow \varepsilon, \triangleright _ \rightarrow \triangleright, \triangleright q_0 \rightarrow \varepsilon$$

- 1)-2) létrehoz egy elfogadó konfigurációt.
- 3)-5) a konfiguráció átmeneteket fordított irányban szimulálja. Azaz, ha pl. $\alpha cqa\beta \vdash \alpha q'cb\beta$ egy $\delta(q, a) = (q', b, L)$ szabály szerint, akkor G -ben az 5-ös pont szerint, $q'cb$ átíródhat cqa -ra.
- 6) Ha a mondatforma egy kezdőkonfiguráció (esetleg néhány extra $_$ karakterrel), akkor ezek a szabályok törlik a felesleges szimbólumokat.
- Indukcióval a levezetés hossza alapján megmutatható, hogy $q_0 w \vdash^* \alpha q_{\text{accept}} \beta$ a.cs.a. $S \Rightarrow^* \triangleright \alpha q_{\text{accept}} \beta \triangleleft \Rightarrow^* \triangleright _ i q_0 w _ j \triangleleft \Rightarrow^* w$. □

Többvermű gépek

- Egy ***k*-vermű gép** egy determinisztikus veremautomata k veremmel.
- Állapotok véges halmaza.
- Véges input ábécé.
- Véges verem ábécé, ugyanaz minden veremre.
- A többvermű gép átmenete függ:
 - az állapottól,
 - az olvasott input szimbólumtól,
 - minden verem tetején lévő veremszimbólumtól.



Többvermű gépek

- A többvermű gép egy **állapot-átmenet** során:
 - állapotot vált,
 - az egyes verem tetején lévő legfelső veremszimbólumot valahány (a 0-t is beleértve) veremszimbólummal helyettesíti.
- Konfiguráció- (állapot-)átmenet:
 - $\delta(q, a, X_1, \dots, X_k) = (p, y_1, \dots, y_k)$.
 - q állapotban X_i szimbólummal az i -edik verem tetején, $i = 1, \dots, k$, a szimbólumot olvas,
 - p állapotba megy és kicseréli az X_i szimbólumot az i -edik verem tetején y_i -re, minden $i = 1, \dots, k$ -ra.
- A többvermű gép a végállapotába jutva fogadja el az egyes szavakat.
- Feltesszük, hogy az input végén (nem része annak) van egy speciális \$ szimbólum, amelyet **végjelzőnek** (endmarker) is hívunk. A \$ jelzi az input összes betűjének a feldolgozását.

Többvermű gép vs. TG

Tétel: Ha egy L nyelv felismerhető egy k -vermű géppel, akkor adható egy TG, ami felismeri L -t.

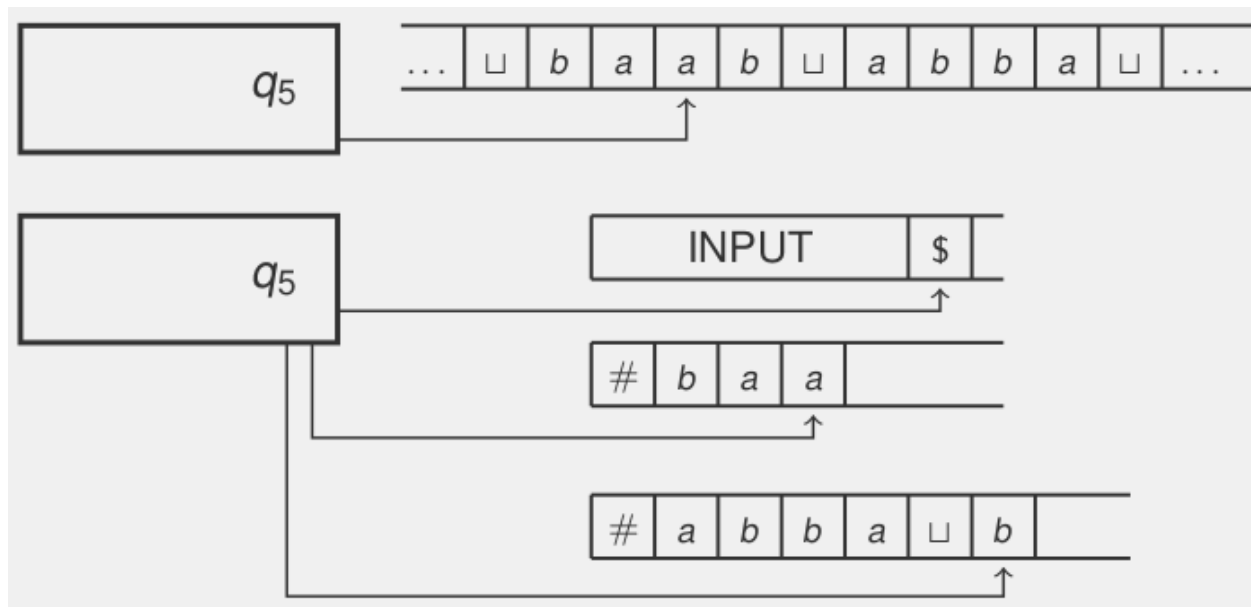
Biz.: Egy k -vermű gépet egy $(k+1)$ szalagos TG-pel könnyen tudunk szimulálni:

- az első szalagon tároljuk az inputot,
- a többi k szalag a k verem tartalmát tárolja.
- Ha egy lépésben j betűt írunk az egyik veremre, akkor azt szimulálhatjuk a TG-pel j lépésekben, betűről betűre.



Többvermű gép vs. TG

- **Tétel:** Ha egy L nyelv felismerhető egy M TG-el, akkor adható egy 2-vermű gép S , amely felismeri L -t.
- **Biz.:** Ötlet: Az egyik verem a TM fejtől balra lévő szimbólumokat tárolja, a másik verem a fejtől jobbra lévő szimbólumokat.



Többvermű gép vs. TG

Biz. (folyt.):

- Kezdetben, az S veremei a $\#$ verem szimbólummal indulnak
 - $\#$ nem eleme M szalagábécéjének.
 - $\#$ csak a verem aljának jelzésére szolgál.
- Tegyük fel, hogy S bemenetén ott van a w szó.
- S a w szót az első vermébe másolja. S befejezi a másolást, ha $\$$ -t olvasta ($\$$ jelzi az input végét).
- S egyenként kiveszi a szimbólumokat az első veremből és a második verembe helyezi (ϵ -átmenetekkel). Ekkor az első verem üres lesz, a második verem w -t tartalmazza, w első betűje lesz a verem tetején.
- S az M kezdeti állapotának megfelelő állapotba lép.

Többvermű gép vs. TG

Biz. (folyt.):

- S az M átmenetét a következőképp szimulálja:
- Feltesszük, hogy S ismeri M állapotát, amelyet q -val jelölünk. Ehhez M minden q állapotához S -nek saját megfelelő állapota van.
- S ismeri az X szimbólumot, amelyet éppen olvas M : X az S második veremének teteje. Kivétel, ha a második verem csak $\#$ -t (veremvég szimbólum) tartalmaz, ekkor M egy $_$ üres szimbólumra ért.
- Így S ismeri M következő átmenetét. S a következő állapotába lép M következő állapotának megfelelően.

Többvermű gép vs. TG

Biz. (folyt.):

- Elég csak a balra és a jobbra lépéssel foglalkoznunk.
- Ha M egy X szimbólumot Y -ra cserél és jobbra lép, akkor S az Y -t az 1. verembe helyezi, és eltávolítja X -et a 2. verem tetejéről, azaz Y az M fejétől balra van. Két speciális eset van:
 - Ha a 2. verem csak a $\#$ veremalja szimbólumot tartalmazza és így $X = _$, akkor a 2. verem tartalma nem változik, (M még egy $_$ felé lép jobbra.)
 - Ha $Y = _$ és az 1. verem tetején $\#$ van, akkor az 1. verem tartalma változatlan marad. (M fejétől balra már csak $_$ szimbólumok vannak.)

Többvermű gép vs. TG

Biz. (folyt.):

- Ha M az X szimbólumot Y -ra cseréli, és balra mozog, akkor S eltávolítja az 1. verem legfelső szimbólumát – nevezzük Z -nek –, és a 2. verem tetején lévő X -t ZY -ra cseréli.
 - Azaz, az M fejtől korábban balra lévő szimbólum (Z) lesz a fej alatt. Kivétel az az eset, amikor Z a $\#$ veremalja szimbólum. Ekkor $_Y$ kerül a 2. verem tetejére, és semmit nem távolít el az 1. veremből.
- S elfogad, ha M új állapota elfogadó állapot, egyébként S az M következő lépését szimulálja. □

References

- Michael Sipser: Introduction to the Theory of Computation. 3rd edition, 2012.