

A digitális számítás elmélete

Előadás: kedd 8:00-10:00, 319-es terem

Burcsi Péter

e-mail: bupe@inf.elte.hu

Gyakorlat: Hudoba Péter

Kérdések:

az előadások szünetében, utána vagy emailben
vizsga előtt igény szerint konzultáció

Követelmények

- Az órák kötelezőek, 70%-ban jelen kell lenni
- A félév végi jegy a gyakorlati jegy és a kollokviumi jegy átlaga, a gyakorlati jegy nagyobb súllyal (55-45).
- A gyakorlati jegyet az alábbiak szerint számítjuk:
0.6 évfolyam ZH jegye + 0.4 órai kis ZH-k átlaga
- Kollokviumra csak a gyakorlaton legalább elégségest elérő hallgató mehet, ellenkező esetben a félév végi aláírás nem adható.
- A gyakorlatok során 3 **ZH lesz (március közepe, április közepe, május eleje)**, plusz egy
- **évfolyam ZH:** az utolsó előadás ideje alatt.
- Az egyik röpz h lehet 1-es (de a nagy zh, és 1-nél több kicsi nem)

Gyakorlat

Javító ZH megírására nincs mód, igazolt távollét esetén lehet csak legfeljebb egy pót ZH-t írni.

A kollokvium tárgya:

minden, ami az előadáson elhangzott, definíciók, tételek és bizonyítás, valamint a megjegyzések és azok belátása.

A kollokvium írásbeli beugró és szóbeli részből áll.

Irodalom

Korábbi évek: wiki

Tankönyv

Az előadás nagyrészt az alábbi könyvre épül:

Sipser, M.: Introduction to the Theory of Computation. PWS Publishing Co.,
1997. ISBN/ISSN: 0-534-94728-X

További irodalom:

Hopcroft, Motwani, Ullman: Introduction to Automata Theory, Languages, and
Computation

Irodalom

A **magyar nyelvű irodalomból** az alábbi, pl. a Műszaki Könyvesboltban, ill. a BME jegyzetboltban kapható tankönyvek használhatók:

C.H. Papadimitriou: Számítási bonyolultság

Demetrovics János, Jordan Denev, Radiszlav Pavlov: A számítástudomány matematika alapjai, Nemzeti Tankönyvkiadó, Bp. 1999, 4 és 5 fejezetek (Automaták és formális nyelvek, Turing gépek)

Rónyai L., Ivanyos G., Szabó R : Algoritmusok, Typotex, 1999, Budapest. 7 és 8 fejezetek (Turing gépek, Az NP nyelvosztály)

Bach Iván: Számítástechnikai nyelvészet, Műegyetemi Kiadó, Bp. 1997.

Bach Iván: Formális nyelvek, Typotex Kiadó, 2001

(Automaták és formális nyelvek, Turing gépek)

Kapható pl. a Műszaki Könyvesboltban, ill. a BME jegyzetboltban

A témák

- formális nyelvek
- automaták
- mely feladatok oldhatók meg automatákon, ill. számítógéppel
- a feladatok nehézsége, a megoldó módszerek idő és hely igénye

automaták, nyelvek

- a számítás matematikai modelljei
- *véges automata*: szövegfeldolgozás, fordítók, hw tervezés
- *context-free nyelvek*: programozási nyelvek
- *Turing gépek*

kiszámíthatóság elmélete

- egyes problémák nem oldhatók meg számítógéppel (Kurt Gödel, Alan Turing, Alonzo Church, 1920-30)
- melyek igen, melyek nem?
- új elméleti modellek újfajta számítógépekhez

bonyolultság

- miért – eddig nem sikerült
- ha egy probléma tudottan nehéz:
 - helyettesítsük egy könnyebbel
 - elégedjünk meg közelítő megoldással
 - elégedjünk meg egy általában gyors módszerrel
 - váltsunk számítási paradigmát (pl. véletlen)
- kriptográfia
- könnyű (pl. rendezés) és nehéz (pl. órarend) problémák

A digitális számítás elmélete

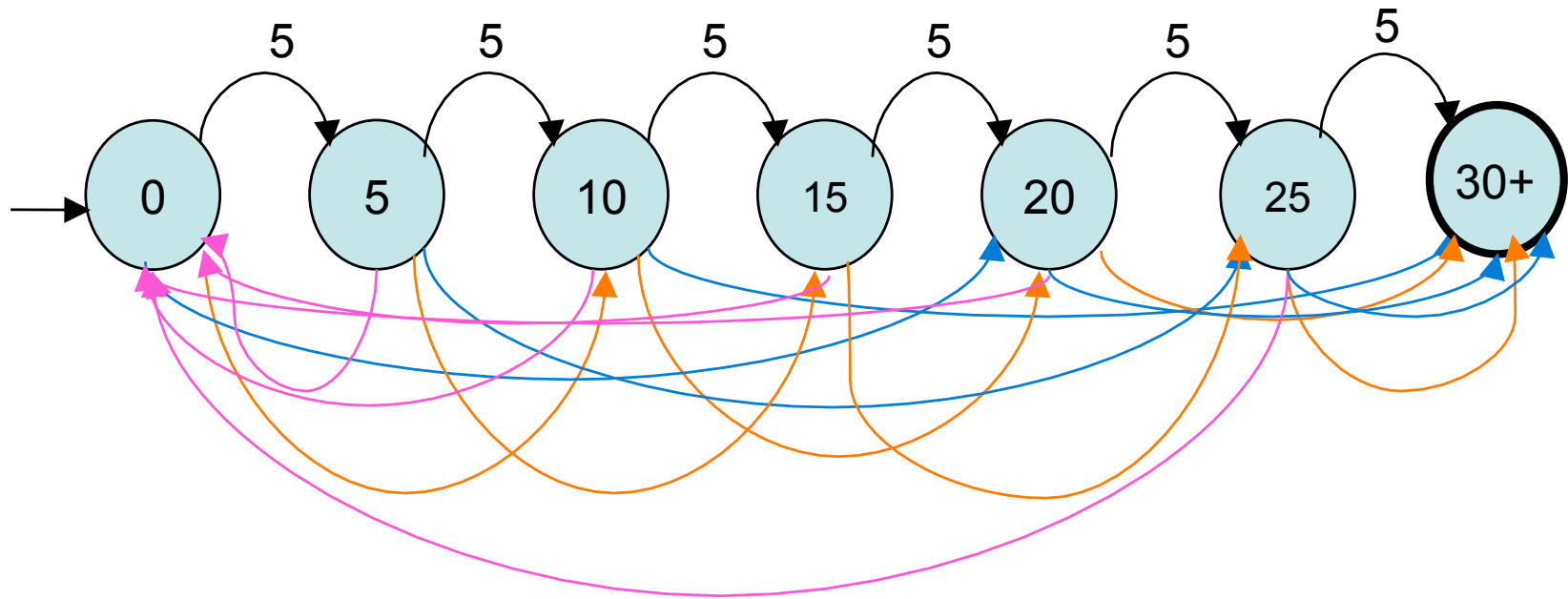
1. előadás 2017. febr. 14.

Determinisztikus véges automaták

Példa: Pénzautomata

be: 5, 10, 20 Ft, VISSZA

válasz: csoki ha legalább 30 Ft-ot bedobtunk, különben VISSZA esetén pénz vissza

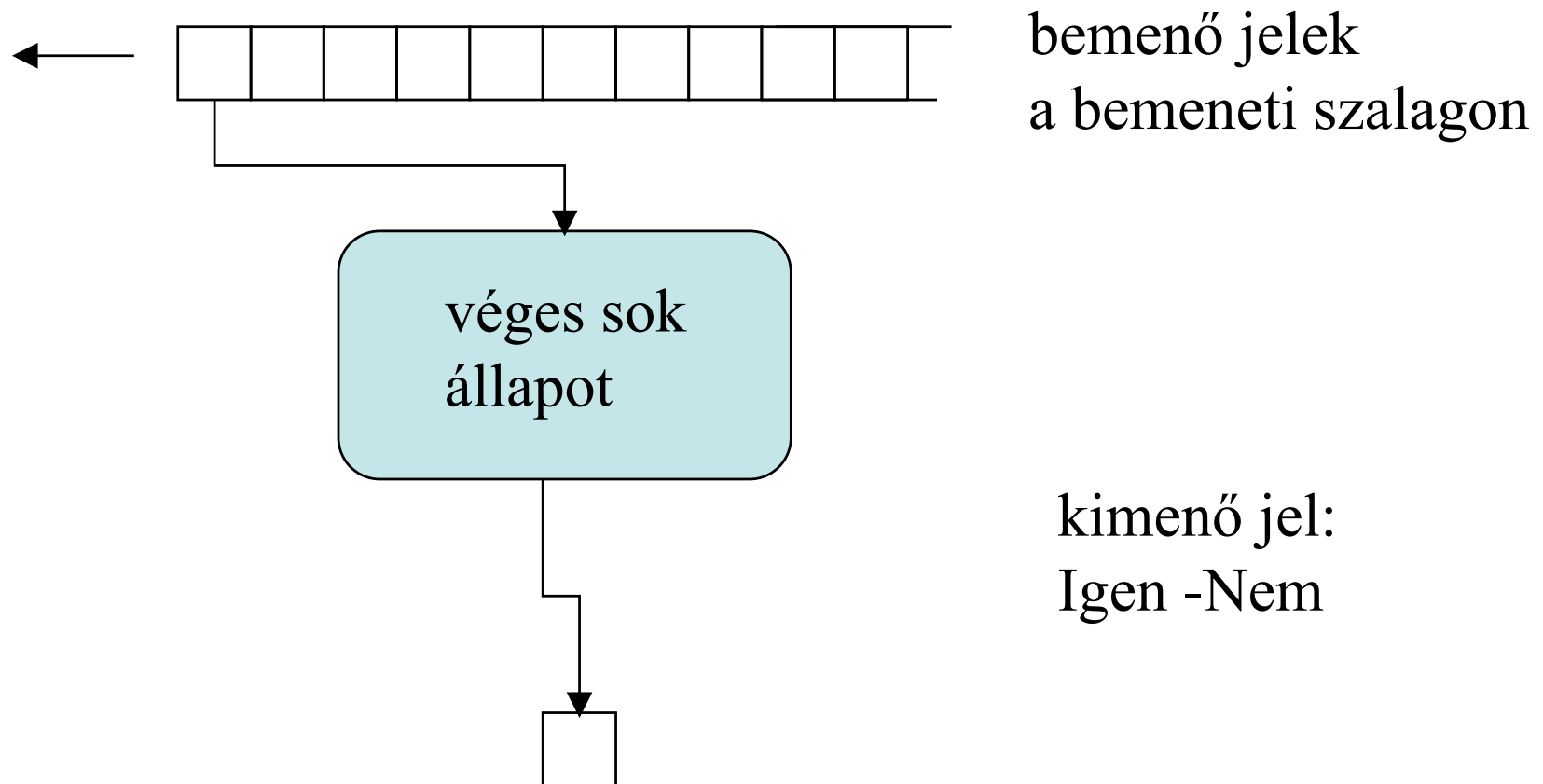


állapot, kezdő és végállapot

állapot átmenet

bemenő jelek

Determinisztikus véges automaták



Determinisztikus véges automaták

Definíció: **Determinisztikus véges automata (dva vagy va)**

$M = (Q, \Sigma, \delta, q_0, F)$ ahol:

1. Q , egy véges halmaz az **állapotok** halmaza
2. Σ , egy véges halmaz az **abc**
3. $\delta: Q \times \Sigma \rightarrow Q$ az **állapotátmenet függvény**
4. $q_0 \in Q$ a **kezdeti állapot**
5. $F \subseteq Q$ a **végállapotok halmaza**

Determinisztikus véges automaták

Definíció: **Felismert (elfogadott) jelsorozat**

$M=(Q, \Sigma, \delta, q_0, F)$ egy dfa, $w = x_1 x_2 \dots x_n$ Σ -beli jelek sorozata.

M felismeri w -t, ha $\exists r_0 r_1 \dots r_n$ Q -beli állapotok sorozata, hogy:

1. $r_0 = q_0$
2. $\delta(r_i, x_{i+1}) = r_{i+1} \quad i = 1, \dots, n-1$
3. $r_n \in F$

Definíció: az M dfa **felismeri az L nyelvet**, ha

$$L = \{w \mid M \text{ felismeri } w\text{-t}\}.$$

Jelölés: Az M által felismert nyelv: $L(M)$

Nyelvek reprezentációja

Definíciók: Σ tetszőleges jelek egy nemüres, véges halmaza, az **abc**. Σ^* jelöli az összes, Σ -beli jelekből álló véges **sorozatok** (**szavak**) halmazát.

$$\Sigma^* = \{w = a_1 \dots a_n \mid a_i \in \Sigma, i = 1, \dots, n, n \geq 0\}$$

$w = a_1 \dots a_n$ szó **hossza** n .

Speciálisan, ε jelöli a 0 hosszú sorozatot, az **üres szót**.

A bináris abc, I a 0 és 1 jeleket tartalmazza.

Példa: $\Sigma = \{\text{if, then, else, for, do, a, b, c} \}$

if a then b else c, do a, do do, aabba

Nyelvek reprezentációja

Definíció: L egy **formális nyelv** Σ felett, ha $L \subseteq \Sigma^*$.

Egy nyelv véges módon megadható:

- a) Elemeit felsorolva – ha azok száma véges.
- b) $L = \{w \in \Sigma^* \mid w \text{ P tulajdonságú}\}$
- c) $L = L(M)$ ahol M egy automata, pl. dva.
- d) Operátorokkal
- e) Generáló nyelvtannal

Megj: Nem lehet az összes Σ feletti nyelvet végesen reprezentálni!

Reguláris operátorok

Definíció: Reguláris operátorok egy Σ abc feletti nyelveken

- **Egyesítés (unió):** $A \cup B = \{x \mid x \in A \text{ vagy } x \in B\}$
- **Szorzat (illesztés, konkatenáció):** $A \circ B = \{xy \mid x \in A, y \in B\}$,

ahol

$x = x_1x_2\dots x_n$, $y = y_1y_2\dots y_m$ esetén $xy = x_1x_2\dots x_ny_1y_2\dots y_m$

- **Iterálás (Kleene csillag):** $A^* = \{w_1w_2\dots w_k \mid w_i \in A \text{ } i = 1, \dots, k, k \geq 0\}$
 1. megj. $\varepsilon \in A^*$,
 2. megj. $L = \Sigma$ esetén $L^* = \Sigma^*$

Operátorok nyelveken

Definíció: További operátorok egy Σ abc feletti nyelveken

- **Metszet:** $A \cap B = \{w \mid w \in A \text{ és } w \in B\}$
- **Különbség:** $A - B = \{w \mid w \in A \text{ és } w \notin B\}$, $A^C = \Sigma^* - A$
- **Tükörkép:** $A^R = \{x^R \mid x \in A\}$ ahol $x = x_1x_2\dots x_n$ esetén $x^R = x_nx_{n-1}\dots x_1$

P1: $L = \{w \in \{0,1\}^* \mid w\text{-ben 2 vagy 3 db 1-es szerepel úgy, hogy az első és a második nem szomszédosak}\}$

$$\{0\}^* \circ \{1\} \circ \{0\} \circ \{0\}^* \circ \{1\} \circ \{0\}^* \circ (\{1\} \circ \{0\}^*) \cup \{e\}$$

Reguláris kifejezések

Definíció: Egy Σ abc feletti reguláris kifejezések:

1. Σ minden eleme.
2. ε
3. $\emptyset$
4. Ha α és β reguláris kifejezések, akkor $(\alpha\beta)$ is az.
5. Ha α és β reguláris kifejezések, akkor $(\alpha\cup\beta)$ is az.
6. Ha α reguláris kifejezés, akkor α^* is az.

Minden reguláris kifejezés reprezentál egy nyelvet. Pl:

reg. kif.

nyelv

$(a^*bc)^*ab$

$(\{a\}^*\circ\{b\}\circ\{c\})^*\circ\{a\}\circ\{b\}$

Reguláris kifejezések

Tétel: Legyen $\mathcal{L}$ azon nyelvek halmaza, melyek felismerhetők determinisztikus véges automatával.

$\mathcal{L}$ zárt a reguláris operátorokra nézve.

Azaz ha $A_1 \in \mathcal{L}$ és $A_2 \in \mathcal{L}$, akkor:

1. $A_1 \cup A_2 \in \mathcal{L}$.
2. $A_1 \circ A_2 \in \mathcal{L}$.
3. $A_1^* \in \mathcal{L}$.

Reguláris kifejezések

Bizonyítás, az 1. állításé:

Legyen $A_1 = L(M_1)$, $A_2 = L(M_2)$,

$$M_1 = (Q_1, \Sigma_1, \delta_1, q_1, F) \quad M_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$$

Megkonstruáljuk $M = (Q, \Sigma, \delta, q_0, F)$ -t, hogy $L(M) = A_1 \cup A_2$

1. $\Sigma = \Sigma_1 = \Sigma_2$ (különben $\Sigma = \Sigma_1 \cup \Sigma_2$)
2. $Q = Q_1 \times Q_2 = \{ (r_1, r_2) \mid r_1 \in Q_1 \text{ és } r_2 \in Q_2 \}$.
3. $\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$.
4. $q_0 = (q_1, q_2)$
5. $F = \{ (r_1, r_2) \mid r_1 \in F_1 \text{ vagy } r_2 \in F_2 \} = (F_1 \times Q_2) \cup (Q_1 \times F_2)$.

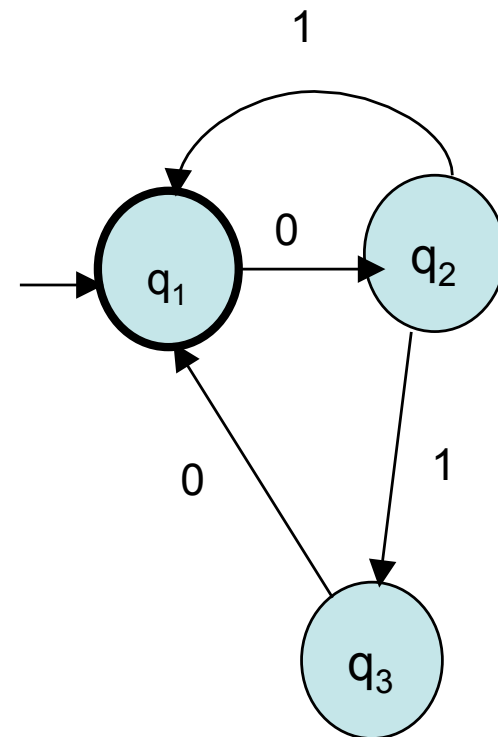
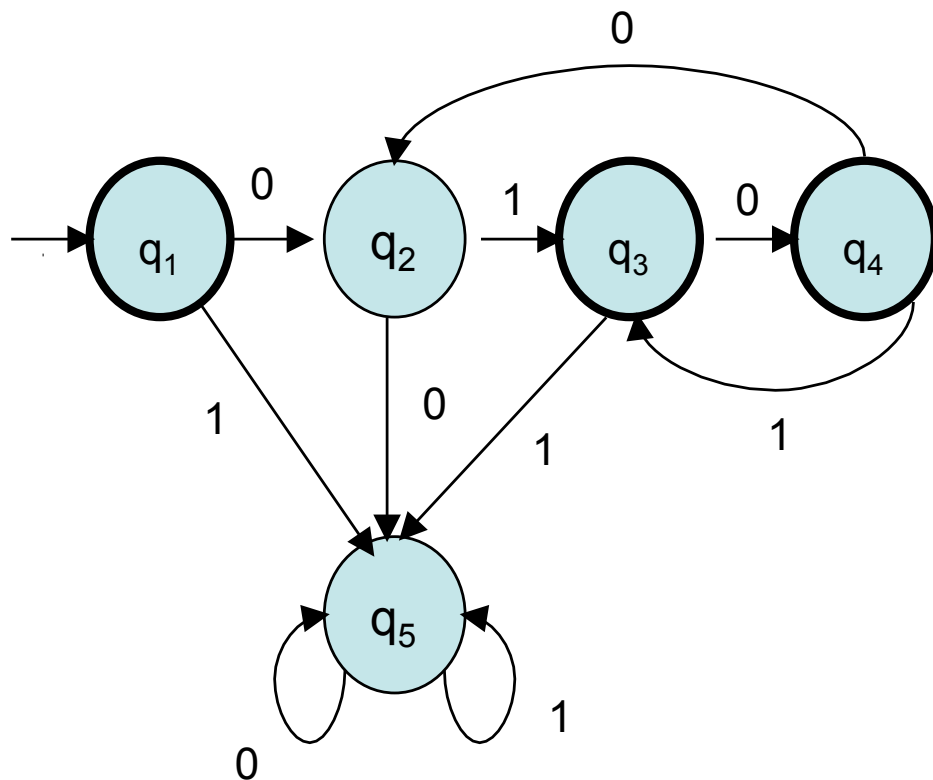
Reguláris kifejezések

Bizonyítás, az 2. és 3. állításé:

Kerülővel, nemdeterminisztikus véges automatákkal.

Nemdeterminisztikus véges automaták

$$L = (01 \cup 010)^*$$



Nemdeterminisztikus véges automaták

Definíció: **Nemdeterminisztikus véges automata (nva)**

$N = (Q, \Sigma, \delta, q_0, F)$ ahol:

1. Q , egy véges halmaz az **állapotok** halmaza
2. Σ , egy véges halmaz az **abc**
3. $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$ az **állapotátmenet fv**
4. $q_0 \in Q$ a **kezdeti állapot**
5. $F \subseteq Q$ a **végállapotok halmaza**

Jelölés: $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

Nemdeterminisztikus véges automaták

Definíció: Felismert (elfogadott) jelsorozat

$N=(Q, \Sigma, \delta, q_0, F)$ egy nva, $w = w_1 w_2 \dots w_n \Sigma_\epsilon$ -beli jelek sorozata.

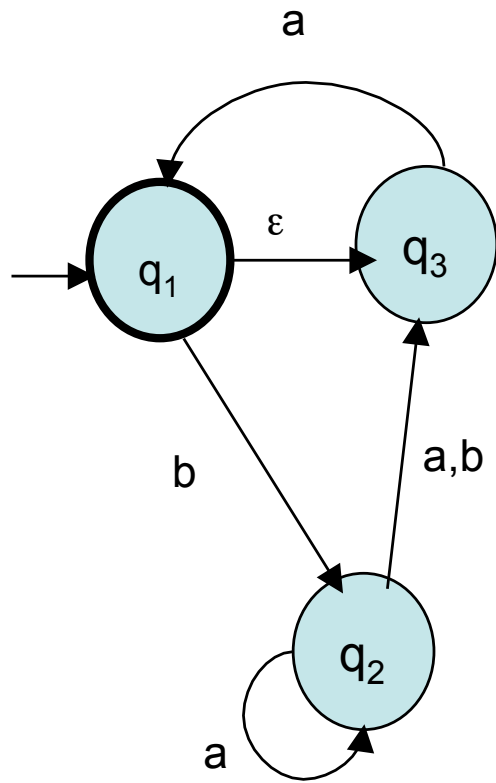
N felismeri w -t, ha $\exists r_0 r_1 \dots r_n$ Q -beli állapotok sorozata, hogy:

1. $r_0 = q_0$
2. $r_{i+1} \in \delta(r_i, w_{i+1}) \quad i = 0, \dots, n-1$
3. $r_n \in F$

Definíció: Az N nva **felismeri az L nyelvet**, ha

$$L = \{w \mid N \text{ felismeri } w\text{-t}\}.$$

Nemdeterminisztikus véges automaták



	a	b	ϵ
q_1	$\emptyset$	$\{q_2\}$	$\{q_3\}$
q_2	$\{q_2, q_3\}$	$\{q_3\}$	$\emptyset$
q_3	$\{q_1\}$	$\emptyset$	$\emptyset$

baba

~~aba~~

~~bab~~

abba

Nemdeterminisztikus véges automaták

Definíció: M_1 és M_2 **ekvivalensek**, ha

$$L(M_1) = L(M_2).$$

Tétel: Minden $N = (Q, \Sigma, \delta, q_0, F)$ nva-hoz létezik vele ekvivalens M dva.

Biz: _Megkonstruáljuk az $M=(Q', \Sigma, \delta', q_0', F')$ va-t.

a) Ha N gráfja nem tartalmaz ε címkét.

1. $Q' = \mathcal{P}(Q)$

2. $\delta'(R, a) = \{ q \in Q \mid \exists r \in R, \text{ hogy } q \in \delta(r, a) \} = \bigcup_{r \in R} \delta(r, a)$

3. $q_0' = \{q_0\}$

4. $F' = \{ R \in Q' \mid \exists r \in R, \text{ hogy } r \in F \}$

Nemdeterminisztikus véges automaták

b) Ha N gráfja tartalmaz ε címkét.

Jelölés: Ha $R \subseteq Q$, $E(R) = \{ q \in Q \mid q \text{ elérhető } R\text{-ből } 0 \text{ vagy több } \varepsilon \text{ él mentén haladva} \}$

1. Ld. a)

2. $\delta'(R, a) = \{ q \in Q \mid \exists r \in R, \text{ hogy}$

$$q \in E(\delta(r, a)) \} = \bigcup_{r \in R} E(\delta(r, a))$$

3. $q_0' = E(\{q_0\})$

4. Ld. a)

M valóban ekvivalens N -nel.

Nemdeterminisztikus véges automaták

Tétel: Legyen $\mathcal{L}$ azon nyelvek halmaza, melyek felismerhetők determinisztikus véges automatával.

$\mathcal{L}$ zárt a reguláris operátorokra nézve.

Azaz ha $A_1 \in \mathcal{L}$ és $A_2 \in \mathcal{L}$, akkor:

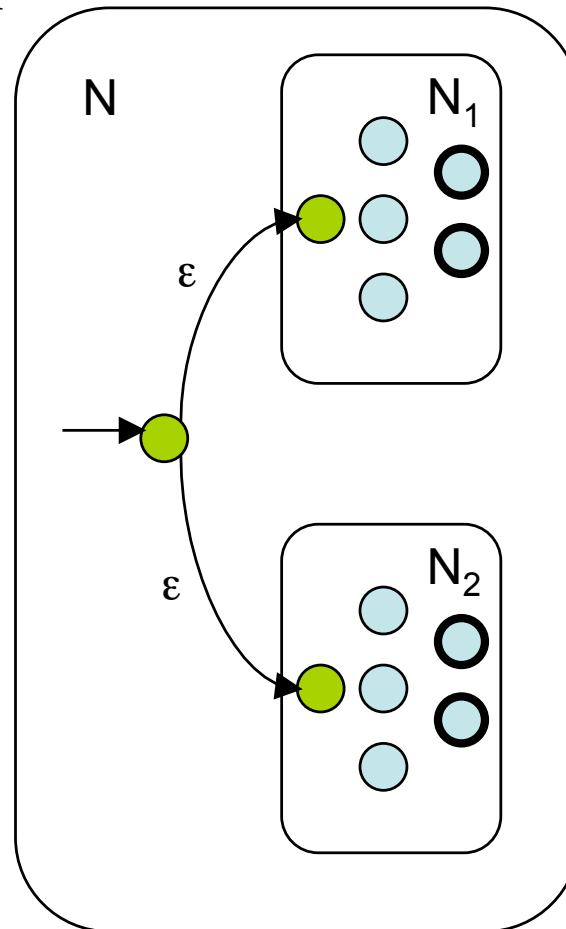
1. $A_1 \cup A_2 \in \mathcal{L}$.
2. $A_1 \circ A_2 \in \mathcal{L}$.
3. $A_1^* \in \mathcal{L}$.

Nemdeterminisztikus véges automaták

Bizonyítás: Nva-kat használhatunk va helyett!

$$L(N_1) = A_1 \quad L(N_2) = A_2 \quad L(N) = A$$

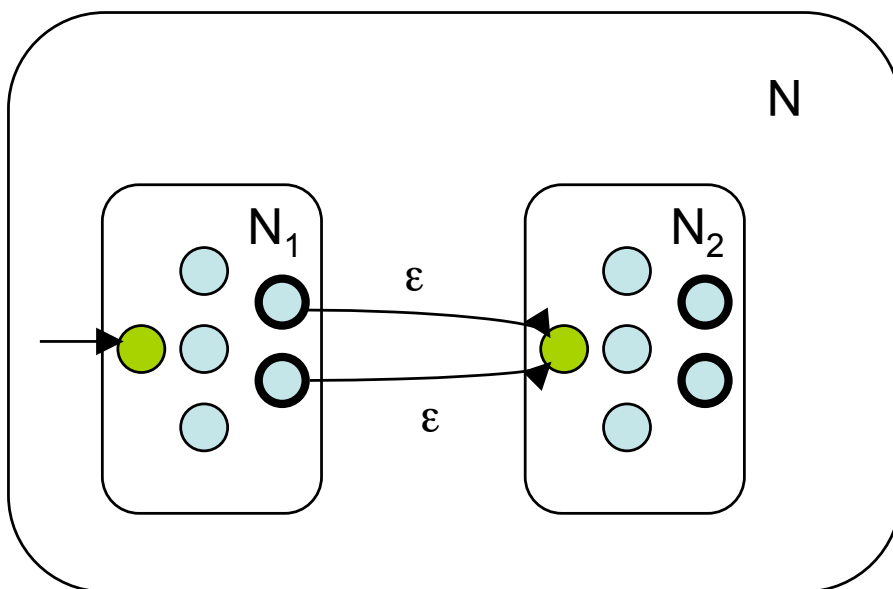
1. $A = A_1 \cup A_2$



Nemdeterminisztikus véges automaták

Bizonyítás:

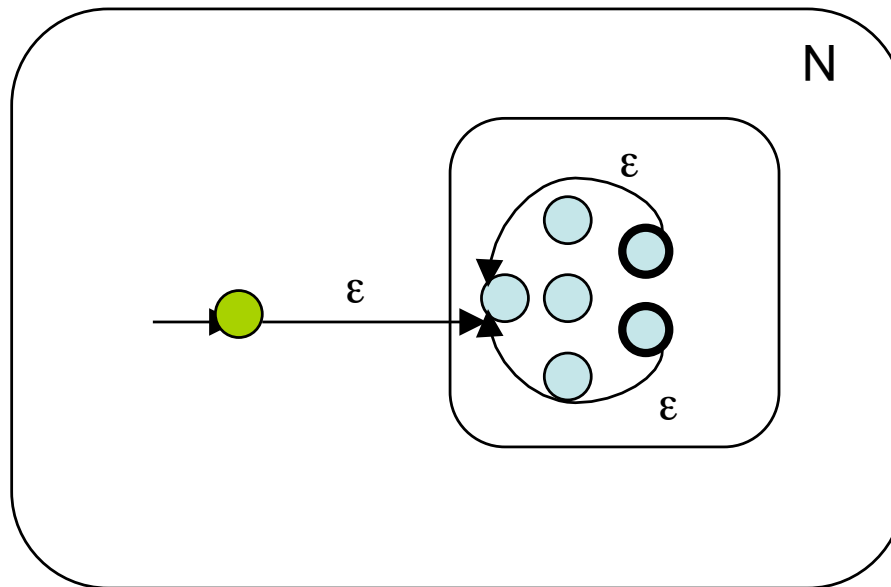
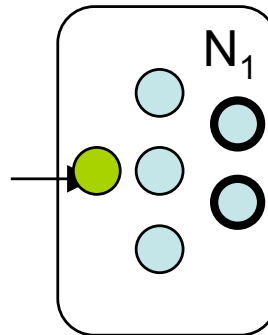
2. $A = A_1 \circ A_2$



Nemdeterminisztikus véges automaták

Bizonyítás:

3. $A = A_1^*$

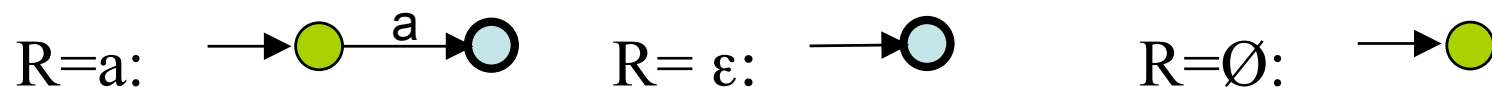


$\mathcal{L}_R$ és $\mathcal{L}_D$ ekvivalenciája

Következmény: Az alábbi, valamely adott Σ feletti nyelvosztályok megegyeznek:

1. $\mathcal{L}_R$ a reguláris nyelvek halmaza
(reguláris műveletekkel / kifejezésekkel megadható nyelvek).
2. $\mathcal{L}_N$ az nva-k által felismert nyelvek.
3. $\mathcal{L}_D$ a dva által felismert nyelvek.

Bizonyítás:



A korábbi két tételből következik, hogy:

$$\mathcal{L}_R \subseteq \mathcal{L}_N \quad \text{és} \quad \mathcal{L}_N = \mathcal{L}_D$$

$\mathcal{L}_R$ és $\mathcal{L}_D$ ekvivalenciája

Megmutatjuk, hogy $\mathcal{L}_D \subseteq \mathcal{L}_R$ is igaz.

Ehhez bevezetjük az **általánosított nemdeterminisztikus véges automata** fogalmát, a következő specialitásokkal:

- az élek reguláris kifejezésekkel címkézettek
- a kezdőállapotból minden más állapotba megy él, de bele egy sem
- egyetlen, a kezdőállapottól különböző végállapot van
- a közbülső állapotok mindegyikéből megy egyetlen él minden közbülső állapotba, így saját magába is

$\mathcal{L}_R$ és $\mathcal{L}_D$ ekvivalenciája

Definíció: **Általánosított nemdeterminisztikus véges automata**
(ánva)

$N = (Q, \Sigma, \delta, q_{\text{kezdő}}, q_{\text{vég}})$ ahol:

1. Q , egy véges halmaz az **állapotok** halmaza
2. Σ egy véges halmaz, az **abc**
3. $\delta: (Q - \{q_{\text{vég}}\}) \times (Q - \{q_{\text{kezdő}}\}) \rightarrow \mathcal{R}$ az **állapotátmenet fv**,
ahol $\mathcal{R}$ a Σ feletti reguláris kifejezések halmaza
4. $q_{\text{kezdő}} \in Q$ a **kezdeti állapot**
5. $q_{\text{vég}} \in Q$ a **végállapot**

$\mathcal{L}_R$ és $\mathcal{L}_D$ ekvivalenciája

Definíció: Felismert (elfogadott) jelsorozat

$N=(Q, \Sigma, \delta, q_{\text{kezdő}}, q_{\text{vég}})$ egy ánya. N felismeri a $w \in \Sigma^*$ sortozatot,

ha $w = w_1 w_2 \dots w_k$ ahol $w_i \in \Sigma^*$ $i = 1, \dots, k$, és $\exists r_0 r_1, \dots, r_k$

Q -beli állapotok sorozata, hogy:

1. $r_0 = q_{\text{kezdő}}$
2. $r_k = q_{\text{vég}}$
3. $w_i \in L(R_i)$, ahol $R_i = \delta(r_{i-1}, r_i)$ $i = 1, \dots, k$

Definíció: N ánya **felismeri az L nyelvet**, ha

$$L = \{w \mid N \text{ felismeri } w\}.$$

$\mathcal{L}_R$ és $\mathcal{L}_D$ ekvivalenciája

A bizonyítás során megkonstruálandó ánya-kat a következő séma szerint használjuk:

- M a kiindulási k állapotú dva
- A_{k+2} egy M -mel ekvivalens, $k+2$ állapotú ánya
- A_{k+1} egy A_{k+2} -mel ekvivalens, $k+1$ állapotú ánya
- ...
- A_2 egy A_m -mal ekvivalens, $k+1$ állapotú ánya
- az A_2 egyetlen élén szereplő reguláris kifejezés amit kerestünk

$\mathcal{L}_R$ és $\mathcal{L}_D$ ekvivalenciája

Bizonyítás: Megmutatjuk, hogy $\mathcal{L}_D \subseteq \mathcal{L}_R$ is igaz.

1. $M = (Q, \Sigma, \delta, q_0, F)$ a kiindulási k állapotú dva. A_{k+2} egy M -mel ekvivalens, $k+2$ állapotú ánya, melyet úgy kapunk, hogy:
 - hozzáadunk Q -hoz egyetlen új $q_{\text{vég}}$ állapotot, melybe minden F -beli állapotból ε címkéjű él vezet,
 - hozzáadunk Q -hoz egyetlen új $q_{\text{kezdő}}$ állapotot, melyből q_0 -ba ε címkéjű él vezet,
 - a többszörös éleket egy újjal helyettesítjük, melyre a kihagyott éleken szereplő szimbólumok unióját írjuk címkeként
 - Az olyan állapotok közé, melyek között nincs M -ben él, beírunk egy $\emptyset$ -zal címkézett élt.

$\mathcal{L}_R$ és $\mathcal{L}_D$ ekvivalenciája

2. Ezután az alábbi KONVERTÁL rekurzív eljárást A_{k+2} -ra alkalmazva, előállítunk egy reguláris kifejezést.

KONVERTÁL(G)

1. Legyen k G állapotainak a száma.
2. Ha $k=2$, visszaadjuk a G egyetlen élén szereplő reguláris kifejezést.
3. Ha $k>2$, legyen $q' \in Q - \{q_{\text{kezdő}}, q_{\text{vég}}\}$ tetszőleges. Legyen:
 $G' = (Q', \Sigma, \delta', q_{\text{kezdő}}, q_{\text{vég}})$, ahol $Q' = Q - \{q'\}$
és $\delta'(q_i, q_j) = (R_1) (R_2)^* (R_3) \cup (R_4)$, ahol:
 $R_1 = \delta(q_i, q')$ $R_2 = \delta(q', q')$ $R_3 = \delta(q', q_j)$ $R_4 = \delta(q_i, q_j)$
4. Adjuk vissza KONVERTÁL(G')-t.

$\mathcal{L}_R$ és $\mathcal{L}_D$ ekvivalenciája

2. Megmutatjuk, hogy KONVERTÁL(G) ekvivalens G-vel.
 $k = |Q|$ szerinti teljes indukciót használunk.

1. $k=2$ -ra igaz.

2. Indukciós feltevés: Igaz $k-1$ állapotú ánya esetén.

3. Megmutatjuk, hogy igaz k állapotú ánya esetén is.

Ehhez megmutatjuk, hogy $L(G) = L(G')$

a) $L(G) \subseteq L(G')$

Legyen $w \in L(G)$, és

$q_{\text{kezdő}}, q_1, q_2, \dots, q_{\text{vég}}$ felismerő állapotok sorozata.

Ha q' nem szerepel, akkor ez a sorozat Q' -beli is, így
 $w \in L(G')$.

$\mathcal{L}_R$ és $\mathcal{L}_D$ ekvivalenciája

Ha ..., q_i , q' , ... q' , q_j , ... szerepel, akkor

$w = \dots, w_i, w_1', \dots w_s', w_j, \dots$ és

$w_i \in \delta(q_i, q') = R_1,$

$w_t' \in \delta(q', q') = R_2, \quad t=1, \dots, s,$

$w_j \in \delta(q', q_j) = R_3$

Tehát ..., q_i , q_j , ... sorozat elfogadó állapotok sorozata

Q' -ben, mert $\delta'(q_i, q_j) = (R_1)(R_2)^*(R_3) \cup (R_4)$, így $w \in L(G)$.

b) $L(G) \subseteq L(G')$

Legyen $w \in L(G')$, és $q_{\text{kezdő}}, q_1, q_2, \dots, q_{\text{vég}}$ egy felismerő állapotátmenet sorozat.

A fentihez hasonló megfontolással látható, hogy $w \in L(G)$ is.

$\mathcal{L}_R$ és $\mathcal{L}_D$ ekvivalenciája

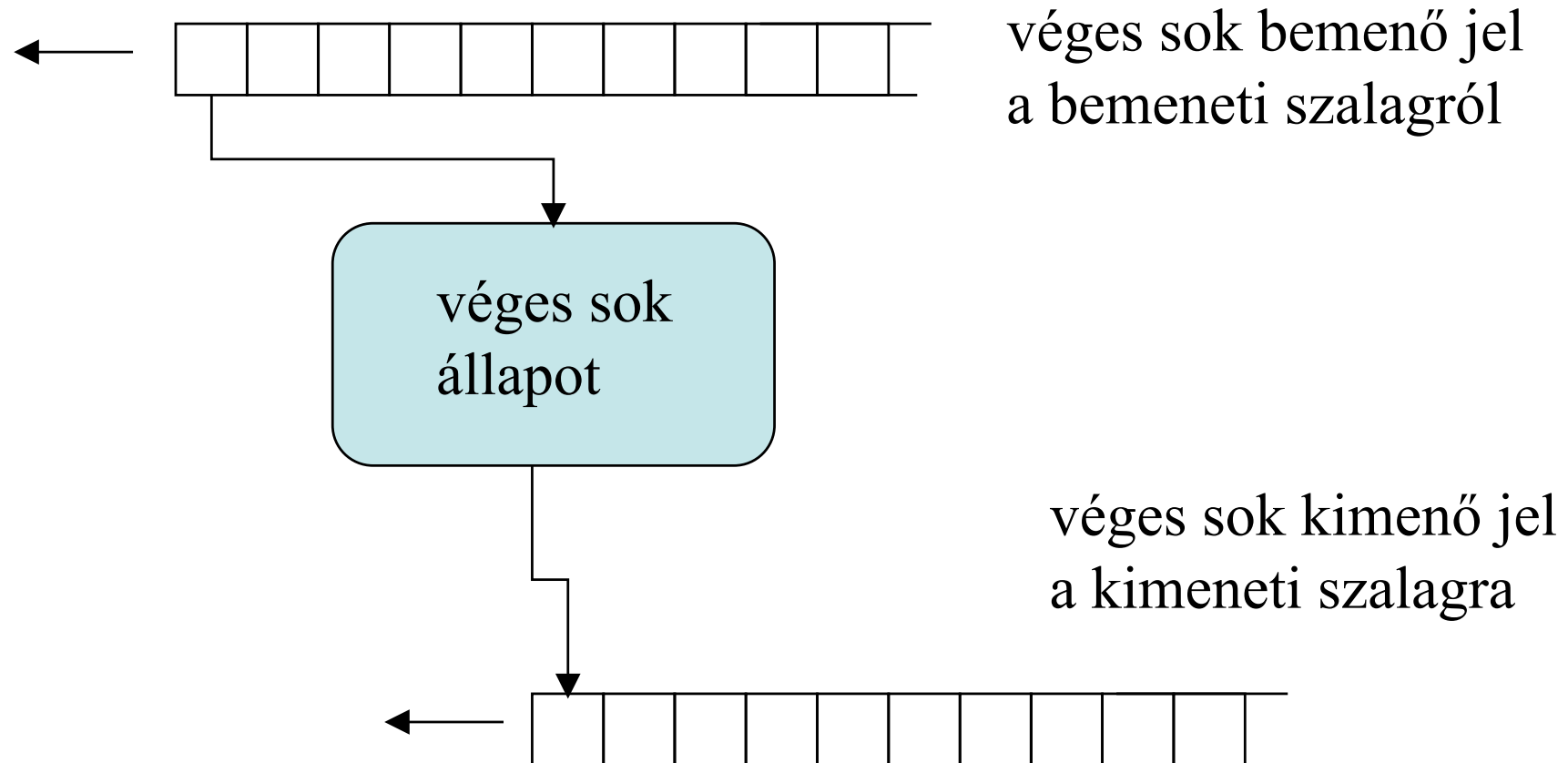
Mivel:

KONVERTÁL(G') ekvivalens G -vel, az indukciós feltevés miatt,

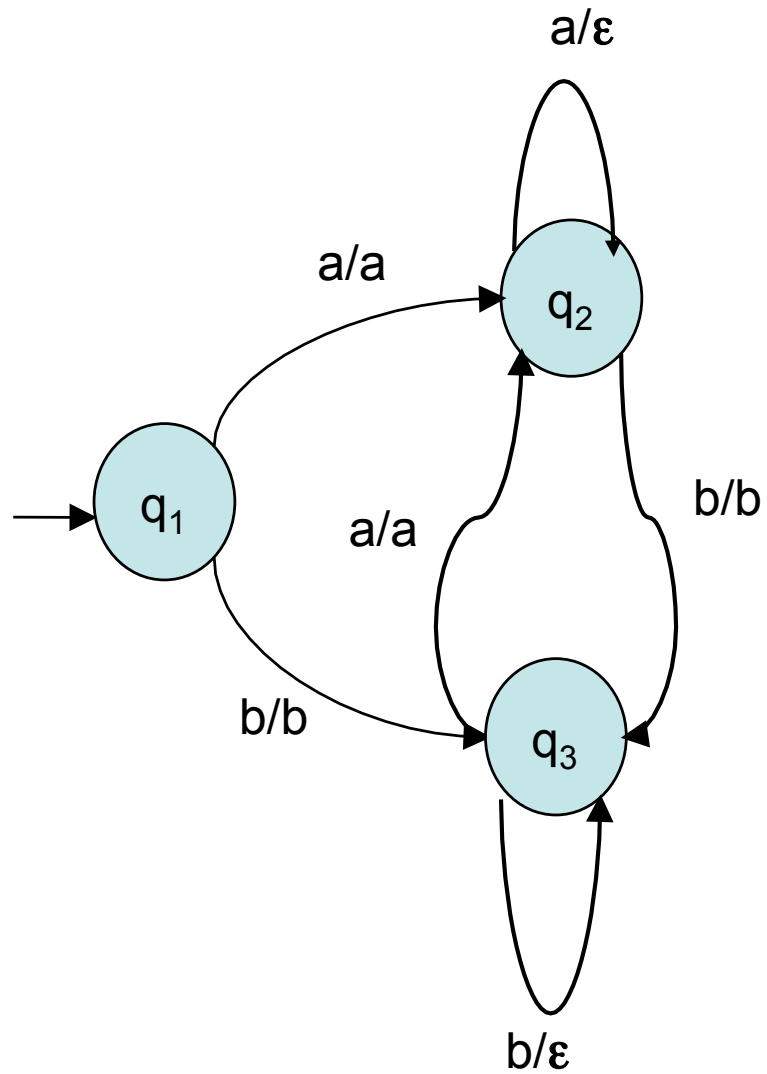
és $L(G) = L(G')$,

ezért KONVERTÁL(G) ekvivalens G -vel.

Véges átalakítók



Véges átalakítók



aaaaa

a

aba

aba

aaaabbaba

ababa

Véges átalakítók

Definíció: Véges átalakító (fordító)

$M = (Q, \Sigma, \Gamma, \delta, q_0)$ ahol:

1. Q , egy véges halmaz az **állapotok** halmaza
2. Σ egy véges halmaz, a **bemeneti abc**
3. Γ egy véges halmaz, a **kimeneti abc**
4. $\delta: Q \times \Sigma \rightarrow Q \times \Gamma$ az **állapotátmenet függvény**
5. $q_0 \in Q$ a **kezdeti állapot**

Miről volt szó eddig?

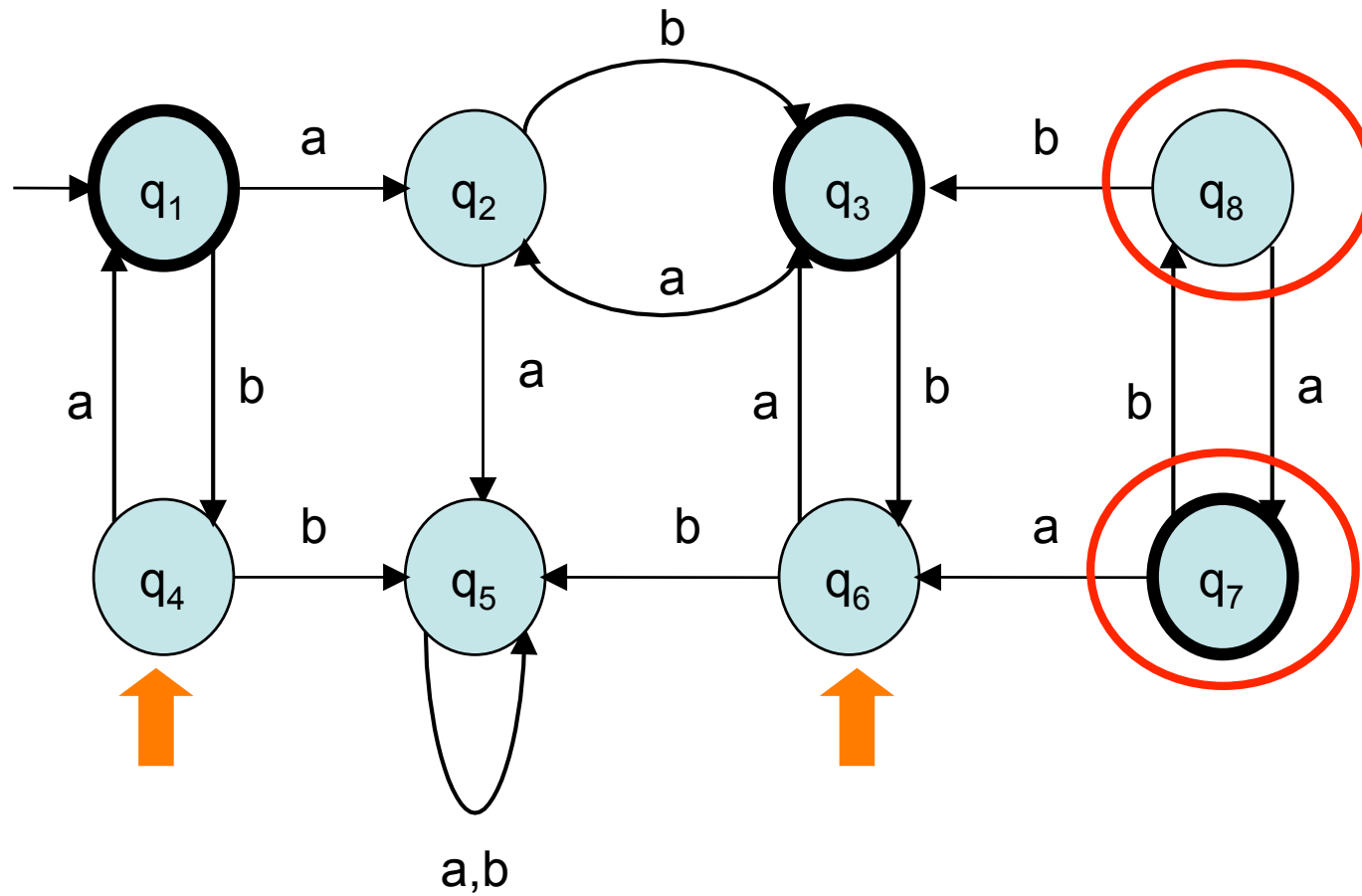
- formális nyelv fogalma
- műveletek formális nyelvekkel
- reguláris nyelvek
- determinisztikus véges automata
- nemdeterminisztikus va
- dva, ndva által felismert nyelv
- $\mathcal{L}_N = \mathcal{L}_D$
- $\mathcal{L}_R = \mathcal{L}_D$
- véges fordítók

A digitális számítás elmélete

2. előadás 2017. febr. 21.

- Egy reguláris nyelvet felismerő minimális dva
- A dva nem elég: vannak nemreguláris nyelvek
- Reguláris nyelvek jellemzése a ‘pumpáló lemmával’

Minimális dva



$$L(M) = (ab \cup ba)^*$$

Minimális dva

Definíció: Legyen $L \subseteq \Sigma^*$ egy nyelv. $x, y \in \Sigma^*$ **L-re nézve ekvivalensek**, ha $\forall z \in \Sigma^*$ esetén igaz, hogy:
 $xz \in L$ akkor és csak akkor (csakkor), ha $yz \in L$.

Ezt a relációt $\approx_L$ -vel jelöljük.

Megj: $\approx_L$ **ekvivalencia reláció** Σ^* -on, mely **ekvivalencia osztályokra** bontja Σ^* -t.

Egy $x \in \Sigma^*$ ekvivalencia osztályát $[x]$ jelöli.

Minimális dva

P1: $L = (ab \cup ba)^*$ esetén a $\Sigma^* \approx_L$ szerinti ekvivalencia osztályai:

1. $[\varepsilon] = L$
2. $[a] = La$
3. $[b] = Lb$
4. $[aa] = L(aa \cup bb)\Sigma^*$

Ugyanis igaz, hogy:

- a) a fenti halmazok ekvivalencia osztályok
- b) különböznek
- c) együtt kiadják Σ^* -t

Minimális dva

a) a halmazok $\approx_L$ szerinti ekvivalencia osztályok, azaz $\forall z \in \Sigma^*$
 esetén: $xz \in L$ csakkor, ha $yz \in L$, ahol $L = (ab \cup ba)^*$

1. $[\varepsilon] = L$, $x = \varepsilon$, $y \in L$

$\varepsilon z = z \in L$ csakkor, ha $yz \in L$

2. $[a] = La$, $x = a$, $y \in La$

$az \in L$ csakkor, ha $yz \in L$

$a(b(..)(..) \dots (..))$ $(..) (..) \dots (..)a(b(..)(..) \dots (..))$

3. mint 2

4. $[aa] = L(aa \cup bb)\Sigma^*$, $x = aa$, $y \in L(aa \cup bb)\Sigma^*$

$aa z \in L$ semmilyen z -re, $yz \in L$ semmilyen z -re

Minimális dva

Definíció: Legyen $M = (Q, \Sigma, \delta, q_0, F)$ dva.

$(r, x) \in Q \times \Sigma^*$, **M egy konfigurációja**

Definíció: Legyen $M = (Q, \Sigma, \delta, q_0, F)$ dva, (r, x) egy konfiguráció.

(r, x) **rákövetkezője** az (r', x') konfiguráció, ha:

- $ax' = x$, ahol $a \in \Sigma$,
- $\delta(r, a) = r'$.

Jelölés: $(r, x) \vdash_M (r', x')$.

Definíció: $(r, x) \vdash_M^* (r', x')$, **(r, x) –ből (r', x') elérhető**, ha

$(r, x) \vdash_M (r_1, y_1) \vdash_M \dots (r_k, y_k) \vdash_M (r', x')$ valamely $k \geq 0$.

Minimális dva

Definíció: Legyen $M = (Q, \Sigma, \delta, q_0, F)$ dva. $x, y \in \Sigma^*$ **M-re nézve**

ekvivalensek, ha $\exists q \in Q$ hogy

$$(q_0, x) \vdash_M^* (q, \varepsilon) \text{ és } (q_0, y) \vdash_M^* (q, \varepsilon)$$

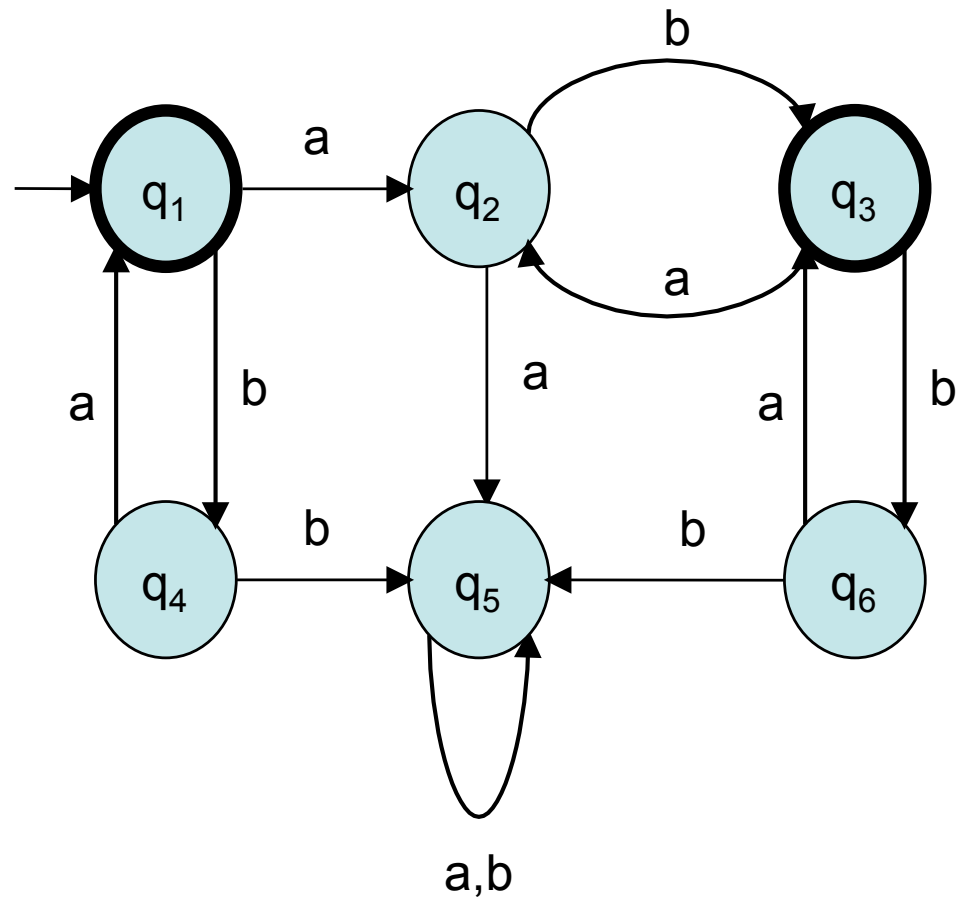
Ezt a relációt $\sim_M$ -vel jelöljük.

Megj: $\sim_M$ **ekvivalencia reláció** Σ^* -on.

Jelölés: Minden q elérhető állapothoz egy-egy Σ^* -részhalmaza
ekvivalencia osztály tartozik, ezt $[q]$ jelöli.

Minimális dva

1. $[q_1] = (ba)^*$
2. $[q_2] = La$
3. $[q_3] = abL$
4. $[q_4] = b(ab)^*$
5. $[q_5] = L(bb \cup aa) \Sigma^*$
6. $[q_6] = abLb$



Belátása hf.

Minimális dva

Tétel: Legyen $M = (Q, \Sigma, \delta, q_0, F)$ dva. Ekkor $\forall x, y \in \Sigma^*$ esetén:

$$x \sim_M y \Rightarrow x \approx_{L(M)} y$$

Biz: $x \in \Sigma^*$ esetén legyen $q(x)$ az az állapot, melyre

$$(q_0, x) \vdash_M^* (q(x), \varepsilon)$$

Így $x, z \in \Sigma^*$ esetén $xz \in L(M)$ akkor és csak akkor, ha

$$(q(x), z) \vdash_M^* (q', \varepsilon), \text{ ahol } q' \in F.$$

De $x \sim_M y \Rightarrow q(x) = q(y)$, tehát

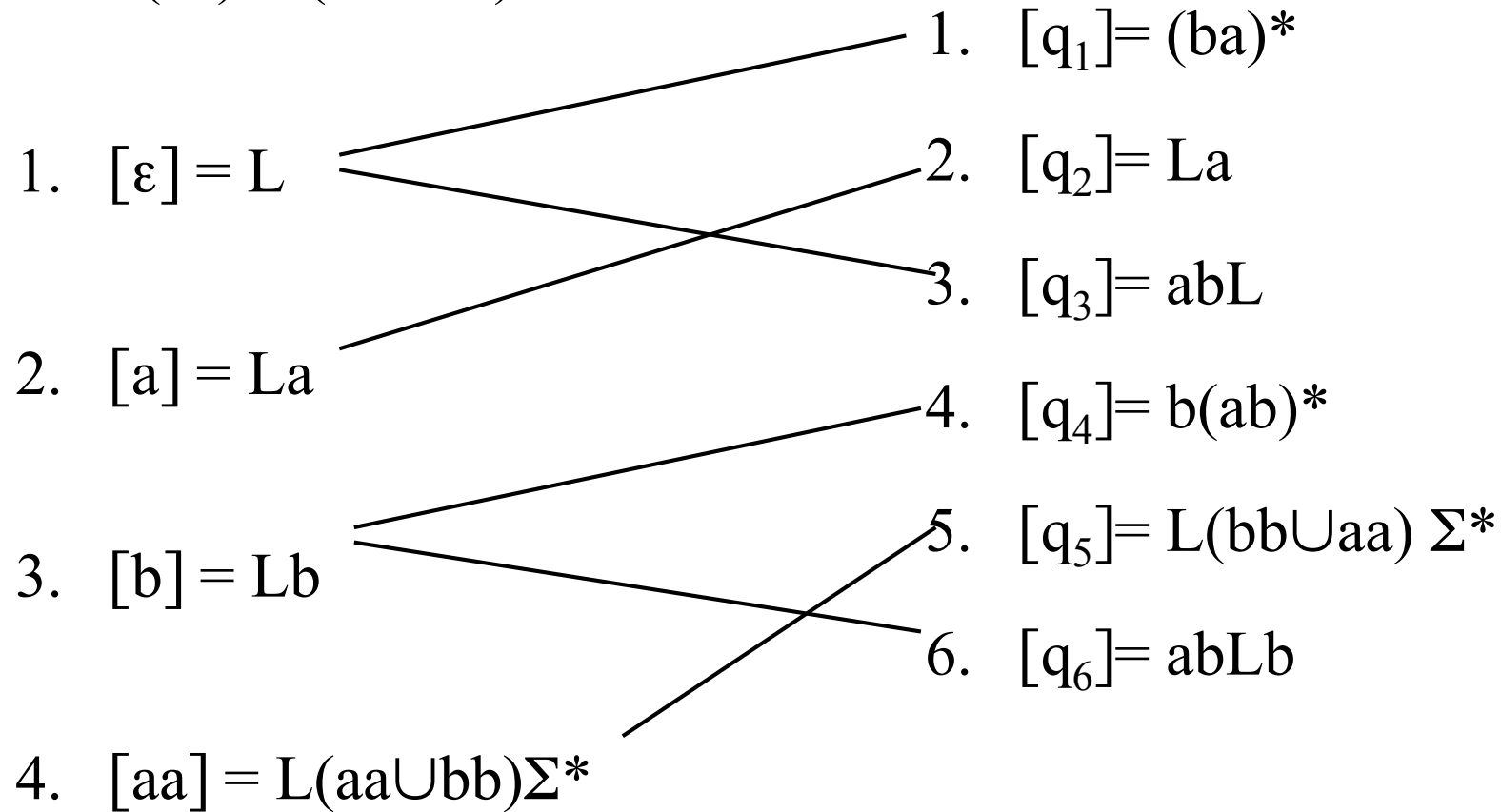
$xz \in L(M)$ akkor és csak akkor, ha $yz \in L(M) \forall z \in \Sigma^*$ esetén.

Azaz $x \approx_{L(M)} y$.

Minimális dva

$\sim_M$ finomítása $\approx_{L(M)}$ -nek

P1: $L(M) = (ab \cup ba)^*$



Minimális dva

Tétel: Ha $L \subseteq \Sigma^*$ egy reguláris nyelv, akkor

$\exists M = (Q, \Sigma, \delta, q_0, F)$ dva, hogy $L(M) = L$, és állapotainak száma, n éppen az $\approx_L$ reláció által meghatározott ekvivalencia osztályok száma. M , az állapotainak számozásától eltekintve, egyértelmű.

Biz: Legyen

- $Q = \{ [x] \mid x \in \Sigma^* \}$
- $q_0 = [\varepsilon]$
- $F = \{ [x] \mid x \in L \}$
- $\delta([x], a) = [xa]$

Minimális dva

- $Q = \{ [x] \mid x \in \Sigma^* \}$ véges, ugyanis mivel L reguláris, $\exists M'$ dva, hogy $L = L(M')$. De láttuk, hogy $\sim_{M'}$ finomítása $\approx_L$ -nek, így $\approx_L$ -nek maximum annyi ekvivalencia osztálya van, mint $\sim_{M'}$ -nek, ami pedig véges.

- $\delta([x], a) = [xa]$ jól definiált, azaz ha $x' \in [x]$, akkor $[xa] = [x'a]$

De $xa \approx_L x'a$ csak akkor ha $x \approx_L x'$

- Minden $x, y \in \Sigma^* : ([x], y) \vdash_M^* ([xy], \varepsilon)$

Biz.: y hossza szerinti teljes indukcióval.

$y = \varepsilon$ esetén trivi.

Ind. feltevés: igaz max. n hosszú y -okra.

Ekkor $y = y'a$ -t tekintve és az indukciós feltevést

alkalmazva: $([x], y'a) \vdash_M^* ([xy'], a) \vdash_M ([xy'a], \varepsilon) = ([xy], \varepsilon)$.

Minimális dva

Megmutatjuk, hogy $L=L(M)$.

Minden $y \in \Sigma^*$ esetén:

$$y \in L(M) \iff ([\varepsilon], y) \vdash_M^* (q, \varepsilon), \text{ ahol } q \in F \iff [y] \in F \iff y \in L.$$

Köv: $L \subseteq \Sigma^*$ reguláris csakkor, ha az $\approx_L$ reláció által meghatározott ekvivalencia osztályok száma véges.

Melyik nyelv nem reguláris?

- $A_k = \{ 1^{nk} \mid n \geq 0 \} \in \mathcal{L}_R$
- $B = \{ 1^n \mid n \text{ prím} \}$
- $C = \{ w \mid \text{azonos számú 0 és 1 } w\text{-ben} \}$
- $D = \{ w \mid \text{azonos számú 01 és 10 részsorozat } w\text{-ben} \} \in \mathcal{L}_R \quad \text{HF!}$

Reguláris nyelvek jellemzése

Tétel (Pumpáló lemma, Bar-Hillel lemma):

Ha $L \subseteq \Sigma^*$ egy reguláris nyelv, akkor $\exists p \in \mathbb{N}$ szám, hogy

$\forall s \in L$ esetén, ha s hossza leglább p , akkor s felbontható

$s = xyz$ alakba úgy, hogy:

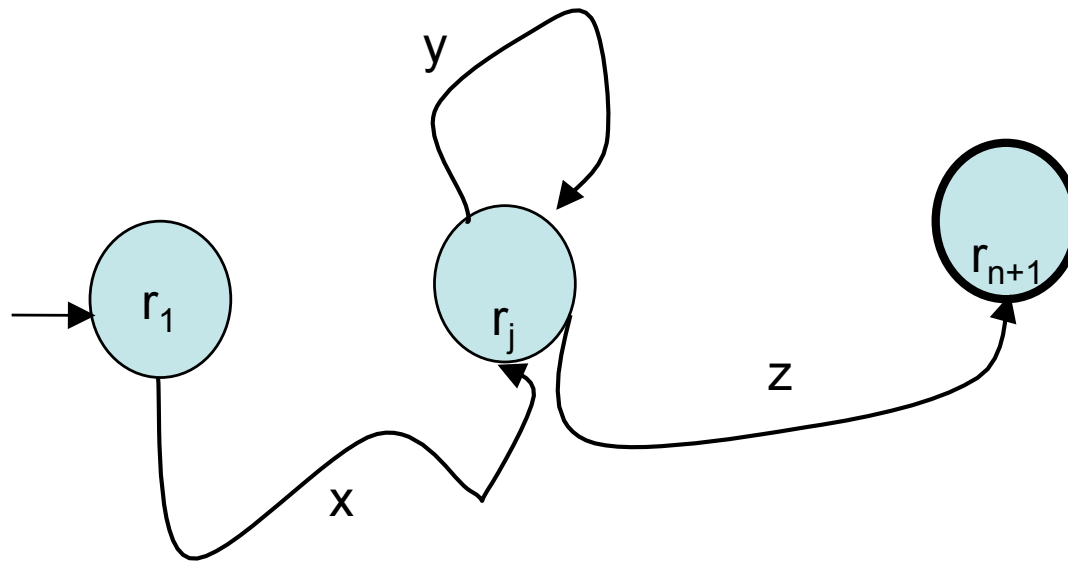
1. $xy^n z \in L$ minden $n \geq 0$ esetén,
2. $|y| > 0$,
3. $|xy| \leq p$.

Biz:

- $p = |Q|$ legyen.
- Legyen $s \in L$, $s = s_1 s_2 \dots s_n$, $p \leq n$. Az s -et elfogadó állapotok sorozata legyen $r = r_1 r_2 \dots r_{n+1}$, ekkor $p < n+1$. Tehát r tagjai között kell lennie legalább két azonosnak, a **skatulya elv** miatt. Jelölje ezeket r_j és r_k . Feltehetjük, hogy $k \leq p+1$.

Reguláris nyelvek jellemzése

- $X = s_1 s_2 \dots s_{j-1}, \quad y = s_j s_{j+1} \dots s_{k-1}, \quad Z = s_k s_{k+1} \dots s_n.$



1. $xy^n z \in L$ minden $n \geq 0$ esetén valóban,
2. $|y| > 0$, mert $j \neq k$,
3. $|xy| = k-1 \leq p$, mert $k \leq p+1$.

A pumpáló lemma használata

Pl: $C = \{ w \mid \text{azonos számú 0 és 1 } w\text{-ben} \}$ **nem** reguláris.

Biz:

A pumpáló lemmát használjuk, indirekten bizonyítunk.

Tfel, hogy C reguláris, ekkor p jelölje a pumpáló hosszt.

Legyen $s = 0^p 1^p$. Mivel $|s| \geq p$, a pumpáló lemma szerint $s = xyz$,
hogy

- $xy^n z \in L$ minden $n \geq 0$ esetén (1. feltétel),
- és $|xy| \leq p$. (3. feltétel)

Ekkor xy , így y is csupa 0-ból áll, és $|y| \geq 1$, tehát $xy^2 z \notin L$.

Azaz az indirekt feltevés hamis, az eredeti állítás az igaz.

Miről volt szó ma?

- Egy reguláris nyelvet felismerő minimális dva
- A dva nem elég: vannak nemreguláris nyelvek
- Reguláris nyelvek jellemzése a ‘pumpáló lemmával’
- A pumpáló lemma használata indirekt bizonyítással

A digitális számítás elmélete

3. előadás február 28.

- Grammatikák
- Környezetfüggetlen grammatikák és nyelvek
- Levezetés, levezetési fa, többértelmű CF nyelvek
- Chomsky-féle normálalak
- Környezetfüggetlen nyelvek jellemzése a ‘pumpáló lemmával’
- Chomsky-féle nyelvosztályok
- Programozási nyelvek szintaxisának megadása

Helyes angol mondatok nyelvtana

<SENTENCE> → <NOUN-PH> <VERB-PH>

<NOUN-PH> → <ARTICLE> <NOUN>

<NOUN-PH> → <ARTICLE> <NOUN> <PREP-PH>

<VERB-PH> → <VERB>

<VERB-PH> → <VERB> <NOUN-PH>

<VERB-PH> → <VERB> <NOUN-PH> <PREP-PH>

<PREP-PH> → <PREP> <ARTICLE> <NOUN>

- <ARTICLE> → a | the
- <NOUN> → boy | girl | flower | binocular
- <VERB> → sees | likes | touches
- <PREP> → with

Helyes angol mondatok nyelvtana

the boy sees

<ART> <NOUN> <VERB>

<NOUN-PH> < VERB-PH >

the boy sees a girl

<ART> <NOUN>

<ART> <NOUN> <VERB> <NOUN-PH>

<NOUN-PH> <VERB-PH>

the flower likes a boy

the flower likes the flower

Helyes angol mondatok nyelvtana

the boy	sees	the girl	with the binoculars
<ARTICLE> <NOUN>	<VERB>	<NOUN-PH>	<PREP-PH>
<NOUN-PH>		<VERB-PH>	

the boy	sees	the girl	with the binoculars
<ARTICLE> <NOUN>	<VERB>	<ART.> <NOUN>	<PREP-PH>
<ARTICLE> <NOUN>	<VERB>		<NOUN-PH>
<NOUN-PH>			<VERB-PH>

Formális nyelvek nyelvtana

A $\{0^i 1^i \mid i > 0\}$ nemreguláris nyelv nyelvtana

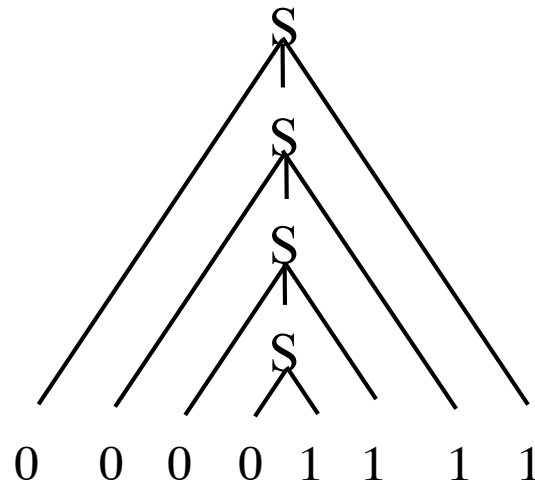
$S \rightarrow 0 S 1$

$S \rightarrow 0 1$

levezetési szabályok

terminálisok (a nyelv abc-je)

grammatikai változók
vagy nemterminális jelek



a **kezdő szimbólum**: egy
kitüntetett nemterminális
jel

Generatív grammatikák

Definíció: Egy $G = (V, \Sigma, R, S)$ négyes generatív grammatika, ahol:

- V egy véges halmaz, a **grammatikai változók vagy nemterminálisok** halmaza,
- Σ egy véges halmaz, diszjunkt V -vel, a **terminálisok halmaza**,
- R egy véges halmaz, a **levezetési szabályok halmaza**, ahol egy levezetési szabály alakja

$$\alpha \rightarrow \beta$$

ahol α és β terminálisok és nemterminálisok tetszőleges sorozata, de α -ban kell legyen egy nemterminális.

- $S \in V$ a **kezdő- vagy mondatszimbólum**.

Generatív grammatikák

Definíció: Egy $G = (V, \Sigma, R, S)$ grammatika alapján a $\kappa \in (V \cup \Sigma)^*$ szóból **közvetlenül levezethető** a $\mu \in (V \cup \Sigma)^*$ szó, ha léteznek olyan $\gamma, \delta \in (V \cup \Sigma)^*$ szavak és $\alpha \rightarrow \beta$ szabály R -ben, hogy:

$$\kappa = \gamma \alpha \delta, \quad \mu = \gamma \beta \delta.$$

Jelölés:

$$\kappa \Rightarrow_G \mu \text{ vagy } \kappa \Rightarrow \mu$$

Generatív grammatikák

Definíció: Tetszőleges u és v $(V \cup \Sigma)^*$ -beli szavak esetén, azt mondjuk, hogy **v levezethető az u -ból** a $G = (V, \Sigma, R, S)$ grammatika alapján, ha $u = v$, vagy léteznek $n \geq 0$ és $u = w_1, \dots, w_n = v$ $(V \cup \Sigma)^*$ -beli szavak úgy, hogy

$$w_1 \Rightarrow_G w_2 \Rightarrow_G \dots \Rightarrow_G w_n.$$

Jelölés:

$$u \Rightarrow_G^* v \text{ vagy } u \Rightarrow^* v$$

Generatív grammatikák

Definíció: A $G = (V, \Sigma, R, S)$ grammatika **generálja az $L \subseteq \Sigma^*$ nyelvet**, ha $L = L(G)$, ahol

$$L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^* w \}$$

a G grammatika által generált nyelv.

Definíció: A G és G' grammatikák **ekvivalensek**, ha $L(G) = L(G')$

Rekurzív grammatikák

Definíció: A $G = (V, \Sigma, R, S)$ grammatika **rekurzív**, ha van olyan

$A \in V$, és $\gamma, \delta \in (V \cup \Sigma)^*$ szavak hogy

$$A \Rightarrow^* \gamma A \delta$$

Ekkor az A egy **rekurzív nemterminálisa** a grammatikának.

Megj:

Nemrekurzív grammatikák véges nyelvet generálnak.

Kérdés: Milyen nyelvet generálnak a rekurzív grammatikák?

Környezetfüggetlen (Context Free, CF) grammatikák

Definíció: A $G = (V, \Sigma, R, S)$ grammatika **környezetfüggetlen**, ha a levezetési szabályainak bal oldalán egyetlen nemterminális szerepel, azaz a szabályok

$$A \rightarrow \beta$$

alakúak, ahol $A \in V$, $\beta \in (V \cup \Sigma)^*$

1. Megj: Egy szabályaival megadott CF grammatika nemterminálisait nagybetűkkel, a terminálisokat kisbetűkkel jelöljük.

2. Megj: A kezdő szimbólum az első szabály bal oldalán szerepel (megállapodás).

3. Megj: $A \rightarrow \beta \mid \gamma \mid \delta$ forma gazdaságos szabályösszevonás.

Környezetfüggetlen (Context Free) grammatikák

1. Példa: $G = (\{S\}, \{a, b\}, R, S)$, és

$$R = \{ S \rightarrow aSb \mid SS \mid \varepsilon \}$$

~~baba~~

~~aba~~

~~bab~~

~~abba~~

abab

aaabbb

aababb

()()

((()))

((()()))

Környezetfüggetlen (Context Free) grammatikák

2. Példa: $G = (V, \Sigma, R, E)$, ahol:

$$V = \{ E, T, F \} \quad \Sigma = \{ (,), +, *, \text{id} \}$$

$$R = \{ E \rightarrow E + T \quad \text{expression} \quad R1$$

$$T \rightarrow T * F \quad \text{term} \quad R3$$

$$T \rightarrow F \quad R4$$

$$F \rightarrow (E) \quad \text{factor} \quad R5$$

$$F \rightarrow \text{id} \} \quad R6$$

Környezetfüggetlen (Context Free) grammatikák

$(id * id + id) * (id + id)$ kifejezése levezetése

$$E \Rightarrow T \Rightarrow T * F \Rightarrow T * (E) \Rightarrow T * (E+T) \Rightarrow$$

$$T * (T+T) \Rightarrow T * (F+T) \Rightarrow T * (id+T) \Rightarrow$$

$$T * (id+F) \Rightarrow T * (id+id) \Rightarrow F * (id+id) \Rightarrow$$

$$(E) * (id+id) \Rightarrow (E+T) * (id+id) \Rightarrow (E+F) * (id+id) \Rightarrow$$

$$(E+id) * (id+id) \Rightarrow (T+id) * (id+id) \Rightarrow (T * F+id) * (id+id) \Rightarrow$$

$$(F * F+id) * (id+id) \Rightarrow (F * id+id) * (id+id) \Rightarrow (id * id+id) * (id+id)$$

$E \rightarrow E+T$	R1
$E \rightarrow T$	R2
$T \rightarrow T * F$	R3
$T \rightarrow F$	R4
$F \rightarrow (E)$	R5
$F \rightarrow id$	R6

Egy reguláris nyelv környezetfüggetlen-e?

Tétel: Igen.

Biz: Ha L reguláris, akkor van őt felismerő M dva. Készítünk egy éppen L -et generáló CF grammatikát.

M minden q_i állapotának megfeleltetünk egy R_i nemterminális szimbólumot, a kezdőszimbólum R_0 .

$R_i \rightarrow aR_k$ szabály, ha $\delta(r_i, a) = r_k$

$R_i \rightarrow \varepsilon$ szabály, ha r_i végállapot M -ben.

Az így definiált grammatika éppen az L -beli szavakat generálja valóban.

Megj.: A kapott grammatika egy ún. jobbreguláris grammatika.

Egyértelmű-e mindig egy CF grammatika
szerinti levezetés?

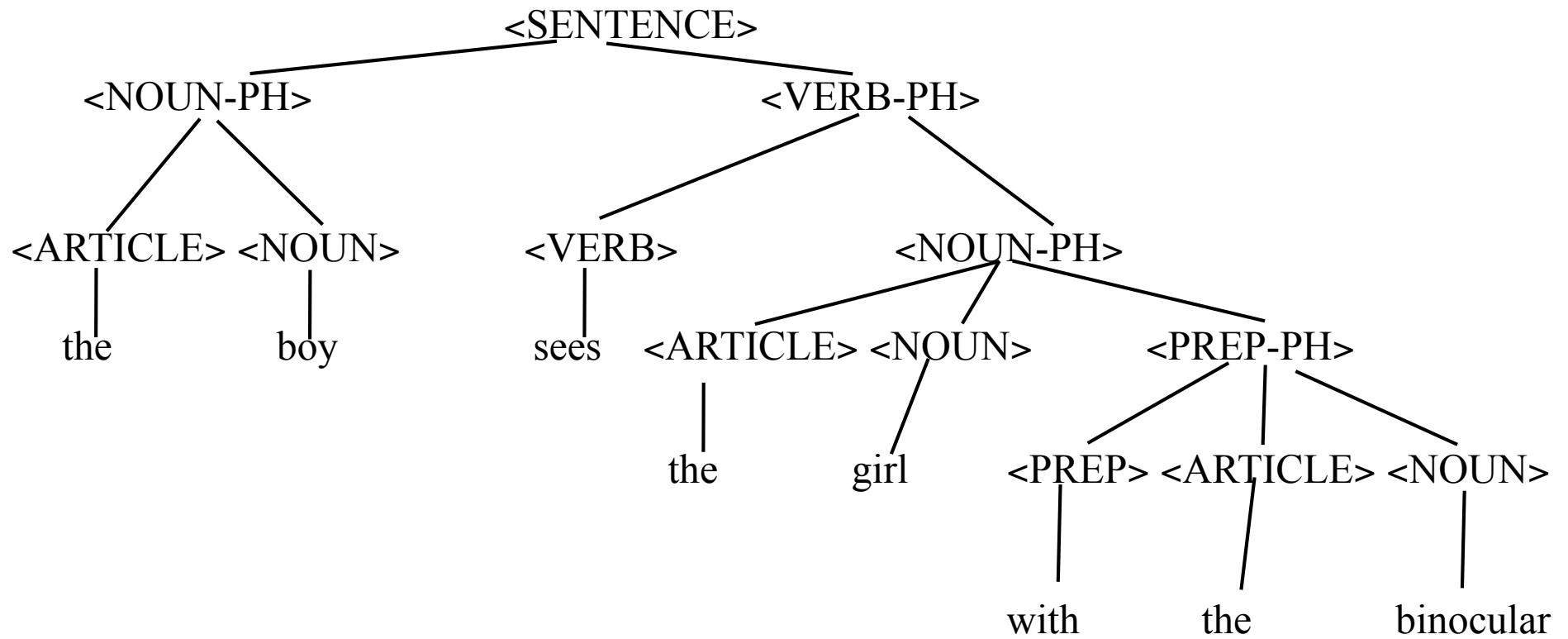
Válasz: Nem mindig, mert a szabályok alkalmazásának sorrendje
tetszőleges lehet.

Levezetési fa

Definíció: Egy CF G grammatika szerinti levezetés olyan fával ábrázolható, melynek **gyökere** a kezdőszimbólum, **levelei** balról jobbra a levezetett szó betűi, további **csomópontjai nemterminálisok**, és az elágazások egy-egy alkalmazott szabálynak felelnek meg.
Ez a **levezetési fa**.

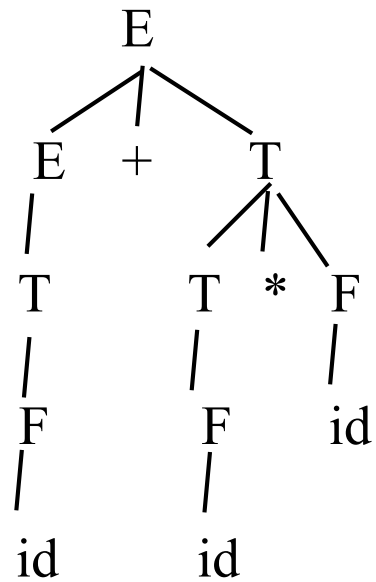
Levezetési fa

1. Pl:



Levezetési fa

2. Pl: $\text{id} + \text{id} * \text{id}$



$E \rightarrow E + T$	R1
$E \rightarrow T$	R2
$T \rightarrow T * F$	R3
$T \rightarrow F$	R4
$F \rightarrow (E)$	R5
$F \rightarrow \text{id}$	R6

Környezetfüggetlen (Context Free) grammatikák

Definíció: Egy **környezetfüggetlen grammatika szerinti levezetés baloldali**, ha a levezetés egyes lépéseiben a legbaloldalibb nemtermimálist helyettesítjük be.

Definíció: Egy nyelvtan **egyértelmű**, ha minden generált szavának egyetlen baloldali levezetése van. Különben **többértelmű**.

Megj: Minden baloldali levezetés meghatároz egy levezetési fát. Tehát egyértelmű nyelvtan esetén minden generált szóhoz egyetlen levezetési fa létezik.

Definíció: Egy CF nyelv **eredendően többértelmű**, ha nem generálható egyértelmű grammatikával.

Egyértelmű-e mindig egy CF grammatika szerinti *baloldali* levezetés?

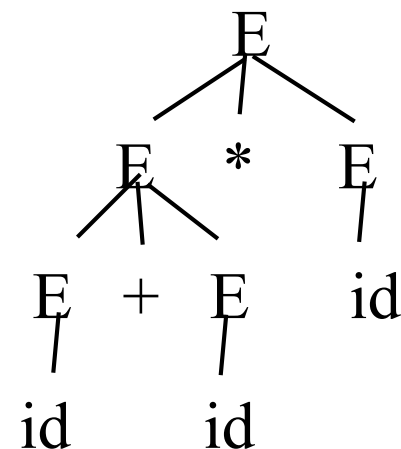
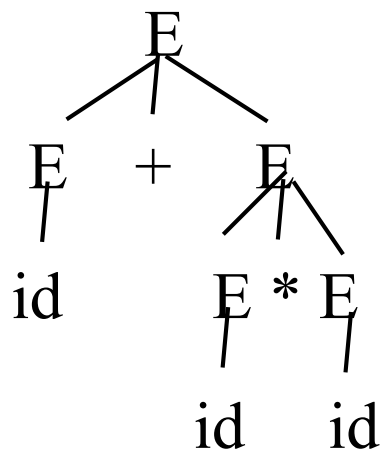
Válasz: Nem mindig.

1. Pl: the boy sees the girl with the flower

2. Pl: $R' = \{E \rightarrow E+E, E \rightarrow E * E, E \rightarrow (E), E \rightarrow id\}$

$E \rightarrow E+T$
 $E \rightarrow T$
 $T \rightarrow T * F$
 $T \rightarrow F$
 $F \rightarrow (E)$
 $F \rightarrow id$

$id + id * id$ két baloldali levezetése:



Chomsky-féle normál alak (CNF)

Definíció: A $G = (V, \Sigma, R, S)$ CF grammatika **Chomsky-féle normál alakban** adott, ha a szabályok az alábbi alakúak:

$$A \rightarrow BC$$

$$A \rightarrow a$$

$$S \rightarrow \varepsilon$$

ahol $A, B, C \in V$, $a \in \Sigma$, B és C nem kezdőszimbólumok.

Tétel: Tetszőleges CF nyelvhez megadható őr generáló grammatika Chomsky-féle normál alakban.

Chomsky-féle normálalak (CNF)

Bizonyítás:

1. $S_0 \rightarrow S$ új kezdő szimbólum
2. Minden $A \rightarrow \varepsilon$ szabályra:
 - a) töröljük $A \rightarrow \varepsilon$ szabályt, ahol A nem kezdősz.
 - b) $B \rightarrow \alpha A \beta$ szabályokhoz adjunk újabbakat úgy, hogy töröljük A -t, minden lehetséges módon!
pl: $B \rightarrow \alpha A \beta A \gamma$ esetén a köv. 3 szabályt iktatjuk be:
 $B \rightarrow \alpha \beta A \gamma$ $B \rightarrow \alpha A \beta \gamma$ $B \rightarrow \alpha \beta \gamma$
 - c) $B \rightarrow A$ esetén a $B \rightarrow \varepsilon$ szabályt csak akkor iktatjuk be, ha korábban a)-ban nem töröltünk már ilyen szabályt.
3. Minden $A \rightarrow C$ szabályra:
 - a) töröljük $A \rightarrow C$ szabályt
 - b) ha van $C \rightarrow \alpha$ szabály, iktassuk be az $A \rightarrow \alpha$ -t, kivéve ha azt már valamikor töröltük a) folyamán.

Chomsky-féle normálalak (CNF)

4. Végül konvertáljuk a megmaradt és beszűrt szabályokat max 2 hosszú jobboldalas alakra:

a) $A \rightarrow \alpha_1 \alpha_2 \dots \alpha_k$ ahol $k \geq 3$, α_i terminális vagy nemterminális.

Helyettesítjük ezt új nemterminálisok és szabályok beiktatásával:

$$A \rightarrow \alpha_1 A_1, \quad A_1 \rightarrow \alpha_2 A_2, \quad A_2 \rightarrow \alpha_3 A_3, \quad \dots \quad A_{k-2} \rightarrow \alpha_{k-1} \alpha_k$$

b) Minden a)-ban hozzáadott, valamint legalább 2 hosszú jobb oldalú szabályban helyettesítsük α_i terminálist U_i nemterminálissal, és iktassuk be a

$U_i \rightarrow \alpha_i$ szabályt.

CF nyelvek jellemzése pumpáló lemmával

Tétel: (Pumpáló lemma, Bar-Hillel lemma):

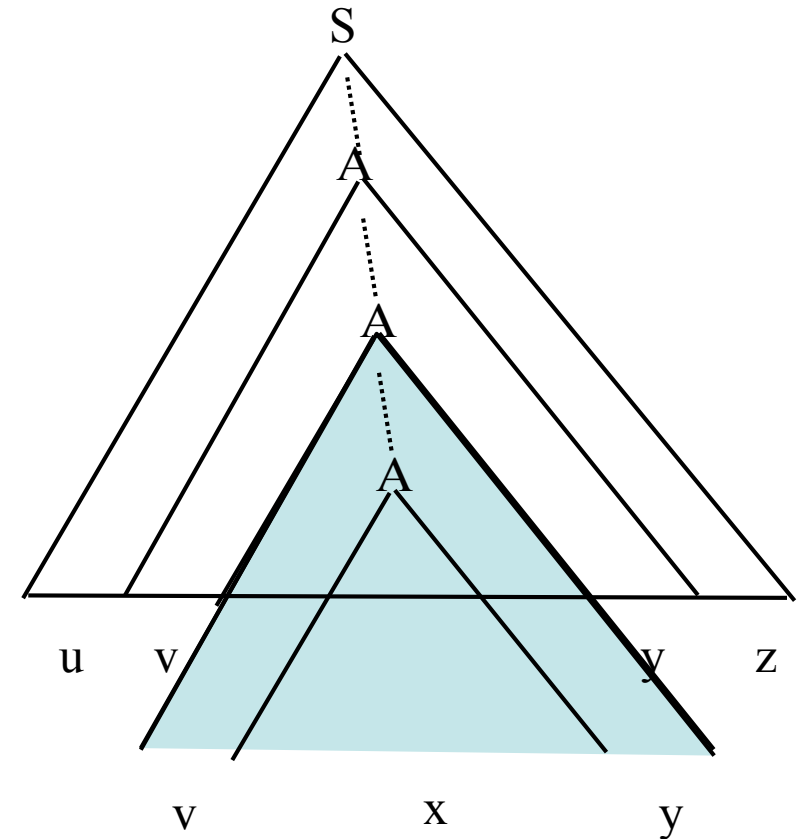
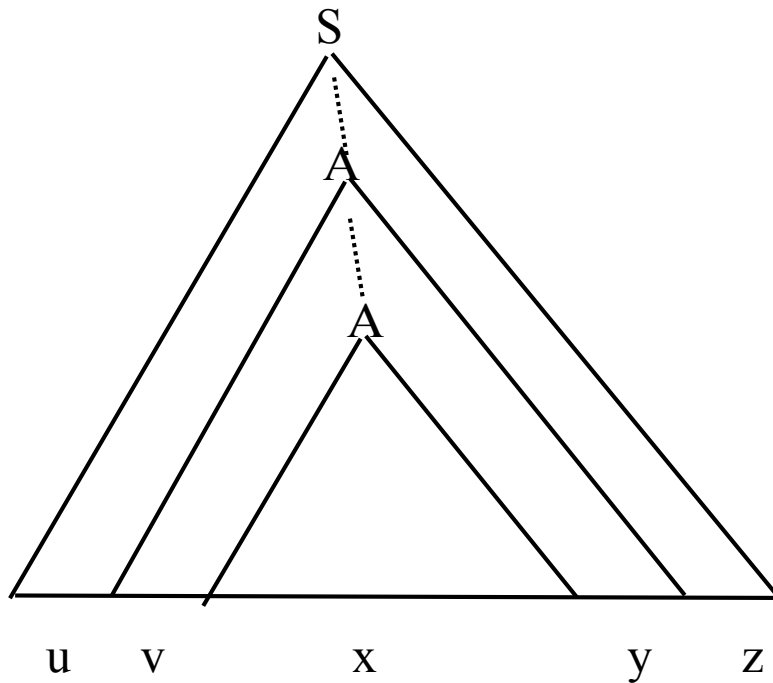
Ha $L \subseteq \Sigma^*$ egy CFL, akkor $\exists p \in \mathbb{N}$ szám, hogy

$\forall s \in L$ esetén, ha s hossza legalább p , akkor s felbontható $s = uvxyz$ alakba úgy, hogy:

1. $uv^nxy^nz \in L$ minden $n \geq 0$ esetén,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

CF nyelvek jellemzése pumpáló lemmával

Biz: Ötlet: s jó hosszú L-beli sorozat, ekkor levezetésében 2-szer szerepel egy A



CF nyelvek jellemzése pumpáló lemmával

Bizonyítás:

Legyen G egy, az adott L -t generáló CFG, és b legyen a szabályok jobb oldalán szereplő szimbólumok max. száma, feltételezzük hogy $b \geq 2$. Tetszőleges m mélységű levezetési fában max. b^m levél szerepel.

Legyen $p = b^{|V|+2}$, ahol $|V|$ a G -beli változók száma. Mivel $b \geq 2$, így $p > b^{|V|+1}$, ezért ha s L -beli legalább p hosszú sorozat, akkor minden levezetési fája legalább $|V|+2$ mély kell hogy legyen.

Legyen s . L -beli legalább p hosszú sorozat, legyen τ s egy legkevesebb csomópontú levezetési fája. τ legalább $|V|+2$ mély, ezért a leghosszabb út a gyökértől egy levélig is legalább $|V|+2$. Ez az út legalább $|V|+1$ változót tartalmaz, ekkor a galambdúc elv miatt van olyan változó, mely legalább 2-szer szerepel az út végső, $|V|+1$ hosszú szakaszán. Legyen ez A .

A pumpáló lemma használata

Pl: $L = \{ a^n b^n c^n \mid 0 \leq n \}$ **nem** CFL.

Biz: A pumpáló lemmát használjuk, indirekten bizonyítunk.

Tfel indirekten, hogy L CFL, ekkor p jelölje a pumpáló hosszt.
Legyen $s = a^p b^p c^p$. Ekkor $s = uvxyz$.

1. Ha v is és y is egy-egy fajta terminálist tartalmaznak, akkor uv^2xy^2z nem tartalmaz mindhárom terminálisból ugyanannyit, így uv^2xy^2z nem L -beli.
2. Ha pl. v tartalmaz több terminálist, akkor v^2 -ben azok sorrendje lesz nem megfelelő, így uv^2xy^2z nem L -beli.

A fenti esetek valamelyike fenn kell hogy álljon, de azok mind ellentmondásra vezettek, így az indirekt feltevés hamis.

Chomsky-féle nyelvosztályok

A nyelvet generáló grammatika szabályaira tett megkötések alapján:

3-as nyelvosztály:

reguláris (R)

$A \rightarrow a, A \rightarrow aB$ alakú szabályok

2-es nyelvosztály:

környezetfüggetlen (CF)

$A \rightarrow \alpha$ alakú szabályok

1-es nyelvosztály (k:

környezetfüggő (CS)

$\beta A \gamma \rightarrow \beta \alpha \gamma$ alakú szabályok

$\alpha \rightarrow \beta$ alakú szabályok ahol $|\alpha| \leq |\beta|$ nem csökkenő

megj: ezek ekvivalensek!

0-ás nyelvosztály:

tetszőleges alakú szabályok

1.Megj: egyre bővebb nyelvosztályok

2. Megj: nyelv jellemzése az öt generáló grammatika alapján

Backus-Naur Form (BNF)

programozási nyelvek definiálására

- meta-nyelv, eredetileg 59-60-ból, az Algol és Fortran definiálására
- Turing-díj Backusnak
- változatok ma is programozási nyelvek megadására
- <http://cs.wpi.edu/~kal/PLT/PLT2.1.2.html>
- <http://cui.unige.ch/db-research/Enseignement/analyseinfo/BNFweb.html>
- ha egy szintaxis BNF formában adott, vannak eszközök, melyek automatikusan generálnak egy elemzőt, mely a szintaktikus helyességet ellenőrzi, és
levezetési fát produkál, sőt le is fordítja a programot
<http://www.thefreecountry.com/developercity/constructcompilers.shtml>

Miről volt szó ?

- Grammatikák
- Környezetfüggetlen grammatikák és nyelvek
- Levezetés, levezetési fa, többértelmű CF nyelvek
- Chomsky-féle normálalak
- Környezetfüggetlen nyelvek jellemzése a ‘pumpáló lemmával’
- Chomsky-féle nyelvosztályok
- BNF, Programozási nyelvek szintaxisának megadása

A digitális számítás elmélete

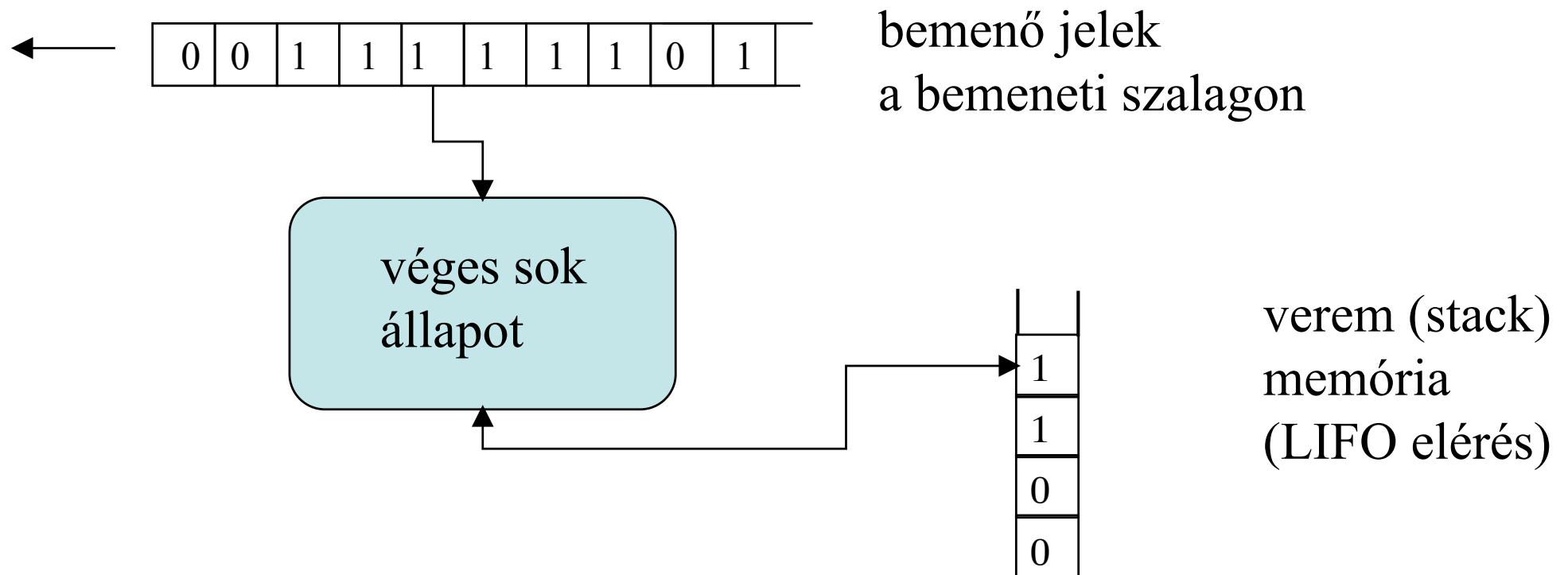
4. előadás márc. 7.

- Verem automata
- Verem automaták által felismert nyelvek éppen a CF nyelvek
- Operátorok CF nyelveken.
- CF nyelv és R (reguláris) nyelv metszetre CF nyelv
- Nem determinisztikus verem automata, nem determinisztikus CF nyelvek.

Verem automaták

- DVA nem alkalmas CF nyelv felismerésére ...
- Milyen módosítás szükséges?

Pl: $\{w w^R \mid w \in \Sigma^*\}$, $\{0^n 1^n \mid 0 \leq n\}$ CF nyelvek felismeréséhez



Verem automaták

Definíció: Verem automata

$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ ahol:

1. Q , egy véges halmaz az **állapotok** halmaza
2. Σ , egy véges halmaz a **bemeneti abc**
3. Γ , egy véges halmaz a **verem abc**
4. $\delta: Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow \mathcal{P}(Q \times \Gamma_\varepsilon)$ az **állapotátmenet függvény**
5. $q_0 \in Q$ a **kezdeti állapot**
6. $F \subseteq Q$ a **végállapotok halmaza**

Jelölés: $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$, $\Gamma_\varepsilon = \Gamma \cup \{\varepsilon\}$ ('nem olvasunk-írunk')

Verem automaták

Definíció: Felismert (elfogadott) jelsorozat

$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ egy verem automata, $w = w_1 w_2 \dots w_n \Sigma_\varepsilon$ -beli jelek sorozata.

M felismeri w -t, ha $\exists$:

$r_0 r_1 \dots r_n$ Q -beli állapotok sorozata, és

$s_0 s_1 \dots s_n \Gamma_\varepsilon$ -feletti szavak sorozata hogy:

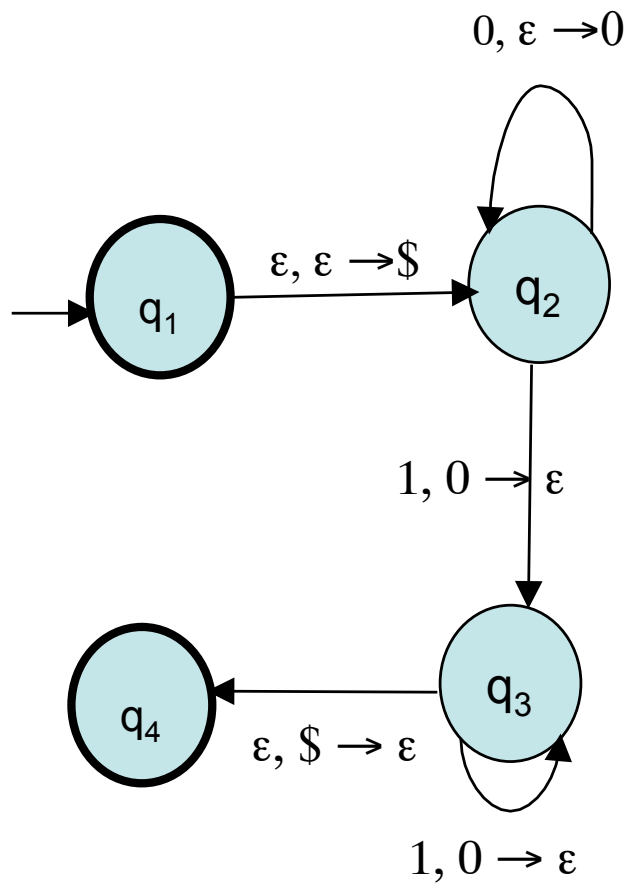
1. $r_0 = q_0$ és $s_0 = \varepsilon$
2. $(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$ $i = 0, \dots, n-1$, ahol:
 $s_i = at$ és $s_{i+1} = bt$ valamely $a, b \in \Gamma_\varepsilon$ és $t \in \Gamma^*$
3. $r_n \in F$

Definíció: M verem automata **felismeri az A nyelvet**, ha

$A = \{w \mid M \text{ felismeri } w\text{-t}\}.$

Verem automaták

$$\Sigma = \{0, 1\}, \Gamma = \{0, \$\}$$



Σ_ϵ	0			1			ϵ		
Γ_ϵ	0	\$	ϵ	0	\$	ϵ	0	\$	ϵ
q_1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{(q_2, \$)\}$
q_2	$\emptyset$	$\emptyset$	$\{(q_2, 0)\}$	$\{(q_3, \epsilon)\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q_3	$\emptyset$	$\emptyset$	$\emptyset$	$\{(q_3, \epsilon)\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{(q_4, \epsilon)\}$	$\emptyset$
q_4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

01

~~010~~

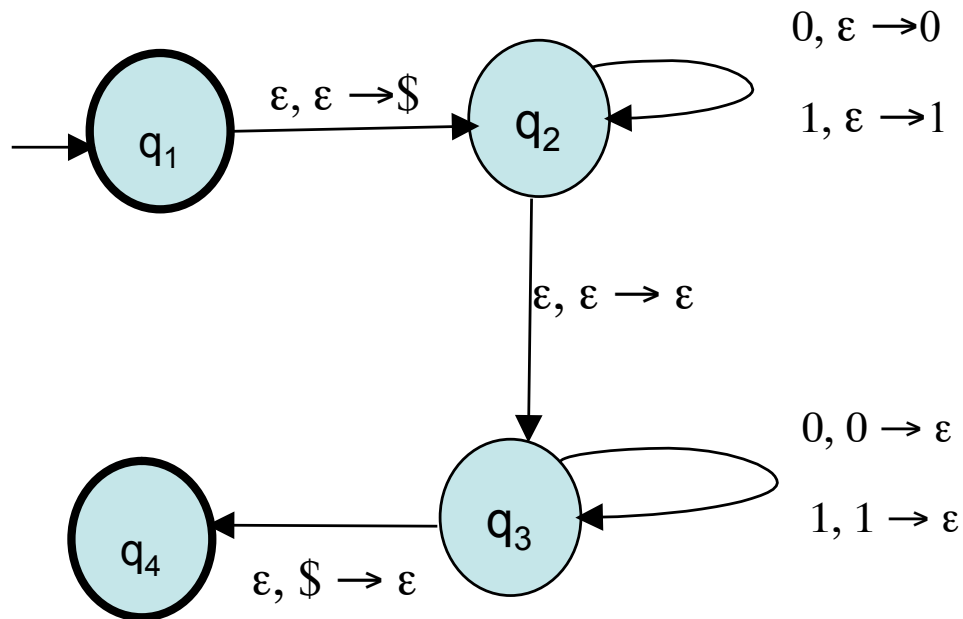
~~1100~~

0011

$\{ 0^n 1^n \mid 0 \leq n \}$

Verem automaták

$$\Sigma = \{0, 1\}, \Gamma = \{0, 1\}$$



00
~~01~~
~~10~~
11
~~000~~
~~100~~
~~010~~
~~0101~~
0110

$$\{ ww^R \mid w \in \{0,1\}^* \}$$

Verem automaták

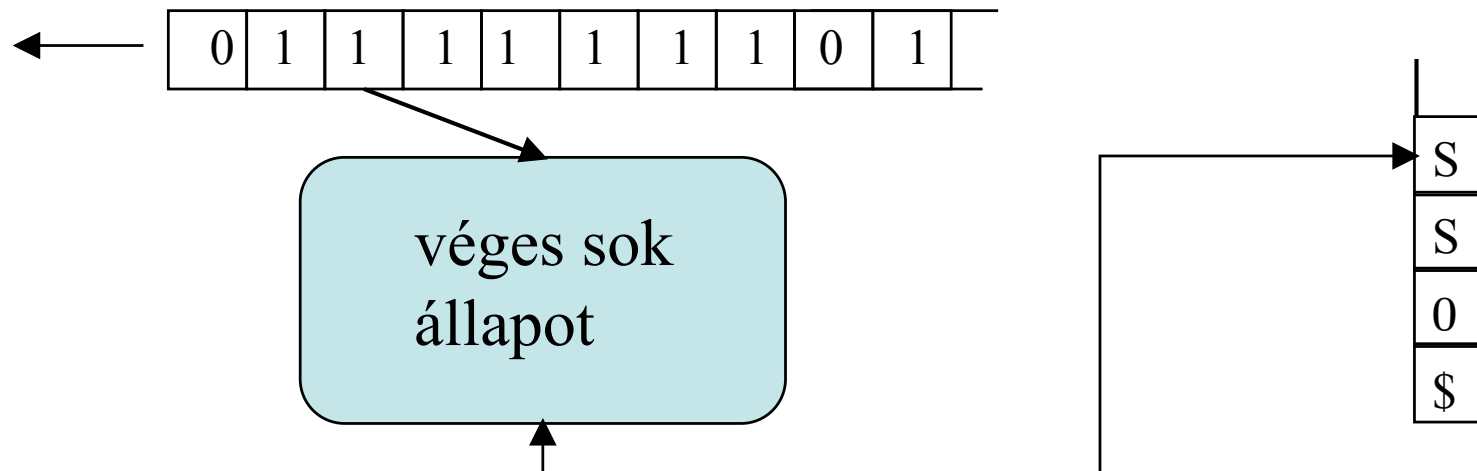
Tétel: Minden L CF nyelvhez létezik őt felismerő verem automata.

Biz: Legyen egy, az L -et generáló CF grammatika G .

Megkonstruálunk egy M verem automatát, amely minden bemeneti jelsorozatról eldönti, hogy levezethető-e a G grammatika szerint.

Ötlet: M a G baloldali levezetéseit szimulálja.

Pl: $S \rightarrow 01SS0$ szabály alkalmazása



Verem automaták

1. Általánosítás: $(r, u) \in \delta(q, a, s)$, ahol $u = u_1 u_2 \dots u_k$.

Szimulálás az eredeti modellel: beiktatunk új $q_1, q_2, \dots, q_{k-1}$ közbülső állapotokat, melyekre:

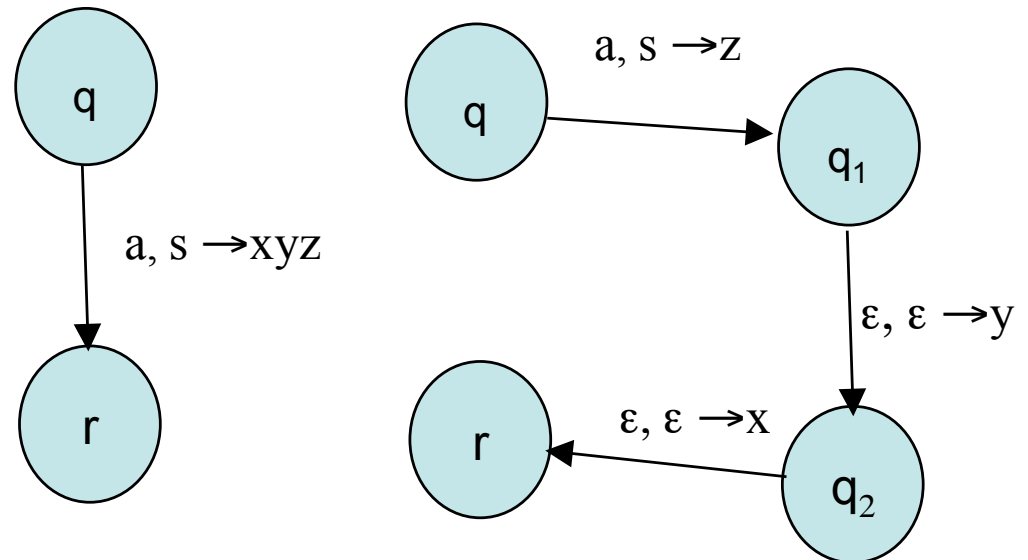
$$\delta(q, a, s) = \{(q_1, u_k)\}$$

$$\delta(q_1, \epsilon, \epsilon) = \{(q_2, u_{k-1})\}$$

$$\delta(q_2, \epsilon, \epsilon) = \{(q_3, u_{k-2})\}$$

...

$$\delta(q_{k-1}, \epsilon, \epsilon) = \{(r, u_1)\}$$



Verem automaták

Tehát használhatjuk azt a rövidítést, hogy $(r, u) \in \delta(q, a, s)$, ami azt jelenti, hogy a q állapotból az a jel és s verembeli jel olvasására az automata az r állapotba kerül, és az u sorozatot betette a verembe.

2. M állapotainak halmaza, $Q = \{q_{\text{kezdő}}, q_{\text{vég}}, q_{\text{ciklus}}\}$.

- $\delta(q_{\text{kezdő}}, \varepsilon, \varepsilon) = \{(q_{\text{ciklus}}, S\$)\}$ **inicializálás**

- **a) a verem legfelső eleme nemterminális**

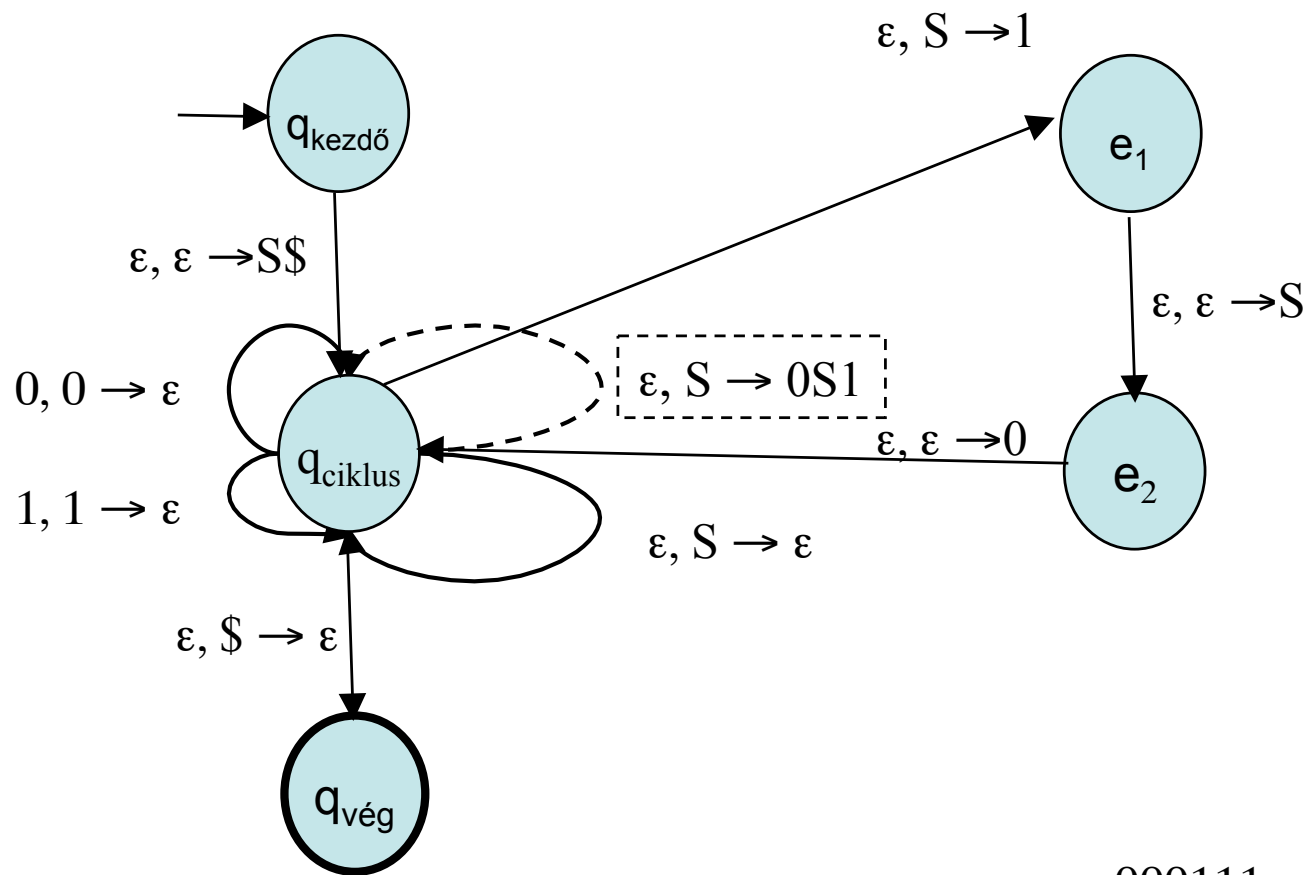
$$\delta(q_{\text{ciklus}}, \varepsilon, A) = \{(q_{\text{ciklus}}, w) \mid A \rightarrow w \text{ szabálya } G\text{-nek}\}$$

- b) a verem legfelső eleme terminális: $\delta(q_{\text{ciklus}}, a, a) = \{(q_{\text{ciklus}}, \varepsilon)\}$**

- c) a verem legfelső eleme $\$$: $\delta(q_{\text{ciklus}}, \varepsilon, \$) = \{(q_{\text{vég}}, \varepsilon)\}$**

Verem automaták

Példa: $S \rightarrow 0S1 \mid \varepsilon$



000111

ε

Verem automaták

Tétel: Minden M verem automatához létezik G CF grammatika, mely éppen az M által felismert nyelvet generálja.

Biz: Feltesszük, hogy M -re igaz, hogy:

1. Egyetlen végállapota van, ez $q_{\text{vég}}$.
2. A végállapotot üres veremmel éri el M .
3. Minden egyes állapotátmenet vagy **betesz** egy szimbólumot a verembe, vagy **kivesz** egyet onnan.

Egy adott M módosítható úgy, hogy a feltételek teljesüljenek.

Verem automaták

Olyan nyelvtant kreálunk, amely minden p és q állapotpárhoz tartalmaz egy A_{pq} változót, melyből minden olyan szó levezethető, amely az M automatát p -ből q -ba viszi úgy, hogy p -ben és q -ban is üres a verem.

Egy ilyen p -t q -ba vivő állapotátmenet-sorozat *betesz* lépéssel kell hogy kezdődjön, és *kivesz* lépéssel kell hogy végződjön.

Legyen t az első lépéssel betett szimbólum. Két esetet lehet:

- a) A t végig a veremben maradt, csak az utolsó *kivesz* távolította el.

Ezt egy $A_{pq} \rightarrow aA_{rs}b$ szabállyal szimuláljuk, ahol a az első, b pedig az utolsó lépésben a bemenetről olvasott szimbólum, r a p -t követő, s a q -t megelőző állapot az állapotátmenet sorozatban.

Verem automaták

b) A t már hamarabb kikerült a veremből valamely r állapotba érkezte: $A_{pq} \rightarrow A_{pr}A_{rq}$.

Legyen $M=(Q, \Sigma, \Gamma, \delta, q_0, \{q_{\text{vég}}\})$. G grammatika nemterminálisai legyenek $\{A_{pq} \mid p, q \in Q\}$, és legyen a kezdőszimbólum $A_{q_0q_{\text{vég}}}$

Megadjuk G szabályait:

- a) $A_{pq} \rightarrow aA_{rs}b$ ahol $a, b \in \Sigma_\varepsilon$, $p, q, r, s \in Q$, és $\exists t \in \Gamma$, hogy:
 $(r, t) \in \delta(p, a, \varepsilon)$ és $(q, \varepsilon) \in \delta(s, b, t)$.
- b) $A_{pq} \rightarrow A_{pr}A_{rq}$ minden $p, q, r \in Q$ esetén.
- c) $A_{pp} \rightarrow \varepsilon$ minden $p \in Q$ esetén.

Megmutatjuk, hogy $x \in L(G) \Leftrightarrow x \in L(M)$.

Verem automaták

1. Lemma: Ha x levezethető A_{pq} -ból, akkor M a p állapotból üres veremmel indulva, az x -et beolvasva el tud jutni a q állapotba úgy, hogy a verem ismét üres.

Biz: a levezetések lépésszáma, n szerinti teljes indukcióval.

Alap eset: $n=1$

Az egyetlen levezetési szabály jobb oldalán csak terminális szerepel.

Ilyen szabály csak $A_{pp} \rightarrow \varepsilon$ lehet.

Ekkor valóban, a p állapotból a p állapot elérhető, és üres kiindulási veremmel üres vermet eredményez.

Verem automaták

Indukciós feltevés: legfeljebb k levezetési szabály alkalmazása esetén igaz a lemma állítása.

Megmutatjuk, hogy ekkor $k+1$ -re is igaz.

Legyen $A_{pq} \Rightarrow^*_G x$ egy $k+1$ hosszú levezetés. Az első lépés, a szabályok alakja alapján, kétféle lehet:

a) $A_{pq} \rightarrow aA_{rs}b$, ekkor $x=ayb$, ahol y levezethető A_{rs} -ből k lépésben.

Ekkor az ind. feltevés szerint r -ből s állapot elérhető az y inputot beolvasva úgy, hogy az r és az s állapotban is a verem üres.

Másrészt a szabály definíciója szerint $\exists t \in \Gamma$, hogy:

$(r,t) \in \delta(p, a, \varepsilon)$ és $(q, \varepsilon) \in \delta(s, b, t)$. Azaz t -t betéve, majd az utolsó lépésben kivéve, a lemma állítása igaz $A_{pq} \Rightarrow^*_G x$ -re.

Verem automaták

b) $A_{pq} \rightarrow A_{pr}A_{rq}$, ekkor $x=yz$, ahol y levezethető A_{pr} -ből és z levezethető A_{rq} -ból legfeljebb k lépésben.

Ekkor az ind. feltevés szerint p -ből az r állapot elérhető az y inputot beolvasva, illetve r -ből a q állapot elérhető az z inputot beolvasva úgy, hogy a p, r és q állapotban is a verem üres.

Azaz p -ből a q állapot elérhető az x inputot beolvasva úgy, hogy a p és q állapotban a verem üres, a lemma állítása igaz $A_{pq} \Rightarrow^*_G x$ -re.

Verem automaták

2. Lemma: Ha M valamely p állapotából és üres veremből x -et beolvasva, van olyan állapotátmenetek sorozata, hogy az automata q állapotba kerül, és a verem ismét üres, akkor x levezethető a grammatika A_{pq} nemterminálisából.

Biz: az állapotátmenetek száma, n szerinti teljes indukcióval.

Alap eset: $n=0$

Ekkor nincs beolvasás, tehát $x = \epsilon$ kell hogy legyen. Valóban szerepel

G szabályai között A_{pp} $\rightarrow \epsilon$, ekkor az állítás igaz.

Verem automaták

Indukciós feltevés: legfeljebb k hosszú állapotátmenet sorozatra igaz a lemma állítása.

Megmutatjuk, hogy ekkor $k+1$ -re is igaz.

Legyen x -hez egy megfelelő $k+1$ hosszú állapotátmenet-sorozat első ill. utolsó állapota p ill. q . E sorozathoz tartozó vermeket tekintve, két eset lehetséges:

a) üres verem nem szerepel közbülső helyen.

Ekkor az első betett elem azonos kell hogy legyen a legvégén kivett elemmel, legyen ez t . Legyen a az elsőnek, b az utolsónak beolvasott elem, r a második, s az utolsó előtti állapot. Mivel $(r,t) \in \delta(p, a, \varepsilon)$ és $(q, \varepsilon) \in \delta(s, b, t)$, így a $A_{pq} \rightarrow aA_{rs}b$ szabály szerepel G -ben. Legyen $x=ayb$. Ekkor az r -ből s -be jutó, y -t beolvasó átmenet-sorozatra alkalmazható az ind. feltevés, azaz y levezethető A_{rs} -ből. De ekkor $A_{pq} \Rightarrow^*_G x$ is igaz.

Verem automaták

- b) Legyen r az a közbülső állapot, amikor a verem kiürül. Ekkor a p -ből r -be illetve r -ből q -ba vivő állapotátmenetek sorozata legfeljebb k hosszú. Legyen $x=yz$, ahol y az r állapotig beolvasott, z pedig a megmaradó részsorozata x -nek. Az ind. feltevés miatt $A_{pr} \Rightarrow^*_G y$ és $A_{rq} \Rightarrow^*_G z$. Mivel az $A_{pq} \rightarrow A_{pr}A_{rq}$ szabály szerepel G -ben, így ennek alkalmazásával x levezethető A_{pq} -ből.

Verem automaták

-

A fenti eredményeket alkalmazva az $A_{q_0q_{\text{vég}}}$ kezdőszimbólumra azt kapjuk, hogy x levezethető $A_{q_0q_{\text{vég}}}$ -ből csakkor, ha M a kezdő állapotából, üres veremmel x -et beolvasva, van állapotátmenetek olyan sorozata, hogy az automata a vég állapotba kerül, és a verem ismét üres.

Tehát x -et felismeri M csakkor, ha x -et generálja a G grammatika.

Ezzel a tétel állítását bebizonyítottuk.

Műveletek CF nyelveken

Tétel: Legyen $\mathcal{L}$ a CF nyelvek halmaza.

$\mathcal{L}$ zárt a reguláris operátorokra nézve.

Azaz ha $A_1 \in \mathcal{L}$ és $A_2 \in \mathcal{L}$, akkor:

1. $A_1 \cup A_2 \in \mathcal{L}$.
2. $A_1 \circ A_2 \in \mathcal{L}$.
3. $A_1^* \in \mathcal{L}$.

Műveletek CF nyelveken

Bizonyítás:

Legyen $G_1 = (V_1, \Sigma_1, R_1, S_1)$ és $G_2 = (V_2, \Sigma_2, R_2, S_2)$ A_1 ill. A_2 $\mathcal{L}$ –beli nyelveket generáló grammatikák. Az általánosság megszorítása nélkül feltehetjük, hogy V_1 és V_2 diszjunktak.

1. Unió esete

Legyen $G = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, R, S)$, ahol

$R = R_1 \cup R_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\}$ Ekkor $L(G) = L(G_1) \cup L(G_2)$

2. Szorzat esete

Legyen $G = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, R, S)$, ahol

$R = R_1 \cup R_2 \cup \{S \rightarrow S_1 S_2\}$ Ekkor $L(G) = L(G_1) \circ L(G_2) = A_1 \circ A_2$

Műveletek CF nyelveken

3. Iterált esete

Legyen $G = (V_1 \cup \{S\}, \Sigma_1, R, S)$, ahol

$R = R_1 \cup \{S \rightarrow \varepsilon, S \rightarrow SS_1\}$ Ekkor $L(G) = L(G_1)^* = A_1^*$

Műveletek CF nyelveken

Tétel: Legyen $\mathcal{L}$ a CF nyelvek halmaza.

$\mathcal{L}$ **nem** zárt a metszetre nézve.

$\mathcal{L}$ **nem** zárt a komplementerre nézve.

Bizonyítás:

$A_1 = \{a^n b^n c^m \mid n, m \geq 0\}$, $A_2 = \{a^n b^m c^m \mid n, m \geq 0\}$ mindketten CF nyelvek. $A_1 \cap A_2 = \{a^n b^n c^n \mid n \geq 0\}$ viszont nem CF (beláttuk).

A De Morgan azonosság szerint: $A_1 \cap A_2 = (A_1^c \cup A_2^c)^c$, így a komplementerre zártság maga után vonná a $\cap$ -re zártságot.

Műveletek CF nyelveken

Tétel: Ha A_1 CF nyelv, A_2 reguláris nyelv, akkor $A_1 \cap A_2$ CF nyelv.

Bizonyítás: Hasonlít a reguláris nyelvek esetében használtra. Legyen

$M_1 = (Q_1, \Sigma, \Gamma, \delta_1, q_1, F_1)$ verem automata, hogy $A_1 = L(M_1)$ és

$M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ dva, hogy $A_2 = L(M_2)$. Legyen

$M = (Q_1 \times Q_2, \Sigma, \Gamma, \delta, (q_1, q_2), F_1 \times F_2)$ verem automata, ahol:

$\delta: Q_1 \times Q_2 \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow \mathcal{P}(Q_1 \times Q_2 \times \Gamma_\varepsilon)$ állapotátmenet fv legyen:

$$\delta(p, q, a, b) = \{ r, (\delta_2(q, a), s) \mid (r, s) \in \delta_1(p, a, b) \}$$

M valóban $A_1 \cap A_2$ -t ismeri fel (hf).

Determinisztikus CF nyelvek

Def: Az $M=(Q, \Sigma, \Gamma, \delta, q, F)$ verem automata determinisztikus, ha a

$\delta: Q \times \Sigma_{\varepsilon} \times \Gamma_{\varepsilon} \rightarrow \mathcal{P}(Q \times \Gamma_{\varepsilon})$ állapotátmenet fv olyan, hogy minden q, a, b esetén $\delta(q, a, b)$ maximum egyelemű halmaz.

Def: Az $L \subseteq \Sigma^*$ nyelv determinisztikus CF, ha az $L\{\$$ nyelvhez van

őt generáló determinisztikus verem automata, ahol

$\$$ nem Σ -beli szimbólum, mely a nyelv szavainak végét jelöli.

Pl: $L = a^* \cup \{a^n b^n \mid n \geq 1\}$

Determinisztikus CF nyelvek

Tétel: A determinisztikus CF nyelvek osztálya zárt a reguláris operátorokra, valamint a komplementerképzésre, és a metszetképzésre.

Biz: nem tárgyaljuk

Tétel: A determinisztikus CF nyelvek osztálya szűkebb, mint CF nyelvek osztálya, azaz van CF nyelv, ami nem determinisztikus CF nyelv.

Pl: $L = \{a^k b^m c^n \mid k \neq m \text{ vagy } m \neq n\}$ CF, de nem determinisztikus CF.

Biz: A fenti tételt alkalmazzuk és keresünk egy alkalmas reguláris nyelvet.

Miről volt szó ?

- Verem automata
- Verem automaták által felismert nyelvek = CF nyelvek
- Operátorok CF nyelveken.
- CF nyelv és R (reguláris) nyelv metszetre CF nyelv
- Nem determinisztikus verem automata, nem determinisztikus CF nyelvek.

A digitális számítás elmélete

6. előadás március 21.

- Turing gép variációk és ekvivalenciájuk az alap géppel

Múlt óra:

- fej helyben maradhat
- több szalag
- nem determinisztikus állapotátmenet

Mai óra:

- felsoroló (enumerator)

- Függvényszámítás Turing géppel, Hilbert 10-edik problémája, Church-Turing tézis.

Turing gép variációk: nem determinisztikus TG

Ismétlés: Minden nemdeterminisztikus Turing-géphez van vele ekvivalens determinisztikus Turing-gép.

Következmény: Egy L nyelv pontosan akkor **Turing felismerhető**, ha van ND TG mely felismeri.

Def: Egy N ND TG eldöntő, ha mindig megáll q_i -ben vagy q_n -ben (minden ágon).

Tétel: Egy L nyelv **Turing eldönthető** akkor és csak akkor, ha van ND TG, mely eldönti.

Turing gép variációk: felsoroló TG

Definíció: Felsoroló Turing gép

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{elfogad}}, q_{\text{elutasít}})$ ahol:

1. ...
2. nincsenek bemeneti sorozatok, csak munkaszalag
3. ...
4. $\delta: Q \times \Gamma \rightarrow (Q \times \Gamma \times \{J, B\} \times \Sigma^*)$ az **állapotátmenet függvény**,
ahol a Σ^* -beli sorozatok a **felsorolt (kinyomtatott) sorozatok**.
5. ...
6. ...
7. ...

Megj: Az összes felsorolt elemek (egyesekek esetleg többször is) alkotják a TG által felsorolt nyelvet.

Turing gép variációk: felsoroló TG

Tétel: Egy L nyelvet felsoroló E TG-hez létezik egy N TG, mely éppen L -et ismeri fel, és viszont.

Biz: E -hez N

Legyen N egy kétszalagos TG, ahol az első szalagra a bemeneti sorozat kerül, a másodikra íródik mindig az E által felsorolt új sorozat. N elfogadja a w inputot, ha az egy, a második szalagra kiírt sorozattal azonos. N -hez van S ekvivalens TG.

Turing gép variációk: felsoroló TG

M-hez E

Legyen A az S által felismert nyelv. Legyen $s_1, s_2, \dots, s_n, \dots$ az összes lehetséges Σ^* -beli sorozat. E a következő képen működik:

1. $i=1,2,3,\dots$ ra ismételi:
2. Szimulálja M -met i lépésig, az $s_1, s_2, \dots, s_i$ szavakon.
3. Ha elfogadó állapotba ér M , kinyomtatja a megfelelő s_j szót.

Turing gép variációk: további példák

- két irányban végtelen szalag
- 2 dimenziós szalag
- több fej
- több elutasító és elfogadó állapot
- RAM (random access machine)
- Kétvermes automata, kétszámlálós gép, ...

A fentiek mind ekvivalensek az alap TG-ekkel. Egy adott fajta TG-hez konstruálható vele ekvivalens alap TG.

Függvényyszámítás Turing géppel

Definíció: $x \in \Sigma^*$ kezdeti bemeneti sorozatra az M Turing gép

outputja az $y \in \Sigma^*$ sorozat, ha M megáll az x inputra, és megállásakor a szalagon éppen y szerepel (az üres szimbólumok előtt), valamint a fej ismét az első pozíción áll.

Ezt a tényt $y = M(x)$ jelöli.

Definíció: Az M által kiszámított fv az a

$f_M: \Sigma^* \rightarrow \Sigma^*$ parciálisan értelmezett fv, mely minden x -en értelmezett, melyre van y , hogy $y = M(x)$.

Definíció: Egy $f: \Sigma^* \rightarrow \Sigma^*$ parciális függvény **parciálisan rekurzív**,

ha van M TG, hogy $f = f_M$.

Ha ezen felül f minden Σ^* -ra értelmezve van, akkor f **rekurzív**.

Függvényyszámítás Turing géppel

Megj: Tetszőleges n természetes szám egyértelműen felírható mint

$\mathcal{B} = 0 \cup 1(0 \cup 1)^*$ eleme, ezt $\text{bin}(n) = w$, illetve $\text{num}(w) = n$ jelöli.

Minden, a természetes számokon értelmezett, és természetes szám értékű fv tekinthető egy $f: \mathcal{B} \rightarrow \mathcal{B}$ függvénynek.

Továbbá minden, k -dimenziós $\mathbb{N}^k \rightarrow \mathbb{N}$ függvény tekinthető egy $f: \mathcal{B}^k \rightarrow \mathcal{B}$ függvénynek.

Definíció: $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{elfogad}}, q_{\text{elutasít}})$ TG, $0, 1, ; \in \Sigma$ M

kiszámítja az $f: \mathcal{B}^k \rightarrow \mathcal{B}$ függvényt, ha tetszőleges $w_1, \dots, w_k \in \mathcal{B}$

esetén a $w = w_1 ; \dots ; w_k$ sorozatra $M(w) = u$, ahol

$u = f(w_1, \dots, w_k)$.

Church-Turing tézis

Algoritmus: véges sok lépéssel kiszámít egy fv-t, eldönt egy kérdést.

Megvalósítható-e minden algoritmus Turing géppel?

Church-Turing tézis: Ami algoritmussal kiszámítható/eldönthető, az

Turing kiszámítható/eldönthető. Nevezetesen:

- Egy $f : \Sigma^* \rightarrow \Sigma^*$ parciális fv algoritmussal kiszámítható $\Leftrightarrow$ f parciálisan rekurzív.
- Egy $f : \Sigma^* \rightarrow \Sigma^*$ teljes fv algoritmussal kiszámítható $\Leftrightarrow f$ rekurzív.
- Egy L nyelvre a nyelvbe való tartozás problémája algoritmussal eldönthető $\Leftrightarrow$ a nyelv rekurzív.

Church-Turing tézis

Tudunk-e eldönthetetlen problémákról?

Hilbert 10 problémája: Diofantikus egyenletek általános

megoldhatósága: adjunk algoritmust, mely tetszőleges d . e -re eldönti, hogy az megoldható-e.

$f(x_1, \dots, x_n) = 0$ egész megoldását keressük,

ahol $f(x_1, \dots, x_n)$ egész együtthatós, n változós polinom.

Általában: NINCS algoritmus (Matijaszevics, 1970), H. 10 eldönthetetlen.

Church-Turing tézis

1. Megj: H10 Turing-felismerhető!

Az összes szám n -est sorra véve teszteljük, hogy megoldás-e. Ha van mo, az véges sok lépésben kiderül. De ha nincs, az nem.

2. Megj: Egyváltozós polinomok esetében a probléma eldönthető, mivel a lehetséges gyökök korlátozottak:

$$\pm k \frac{c_{\max}}{c_1}$$

(k : a tagok száma, $c_{\max}$: a legnagyobb abszolút értékű eh., c_1 : a legnagyobb kitevőjű tag eh.-ja.)

Miről volt szó ?

- Turing gép variációk, ekvivalenciájuk az alap géppel
 - fej helyben maradhat
 - több szalag
 - nem determinisztikus állapotátmenet
 - felsoroló (enumerator)
 - RAM
- Függvényszámítás Turing géppel
- Hilbert 10-dik problémája
- Church-Turing tézis

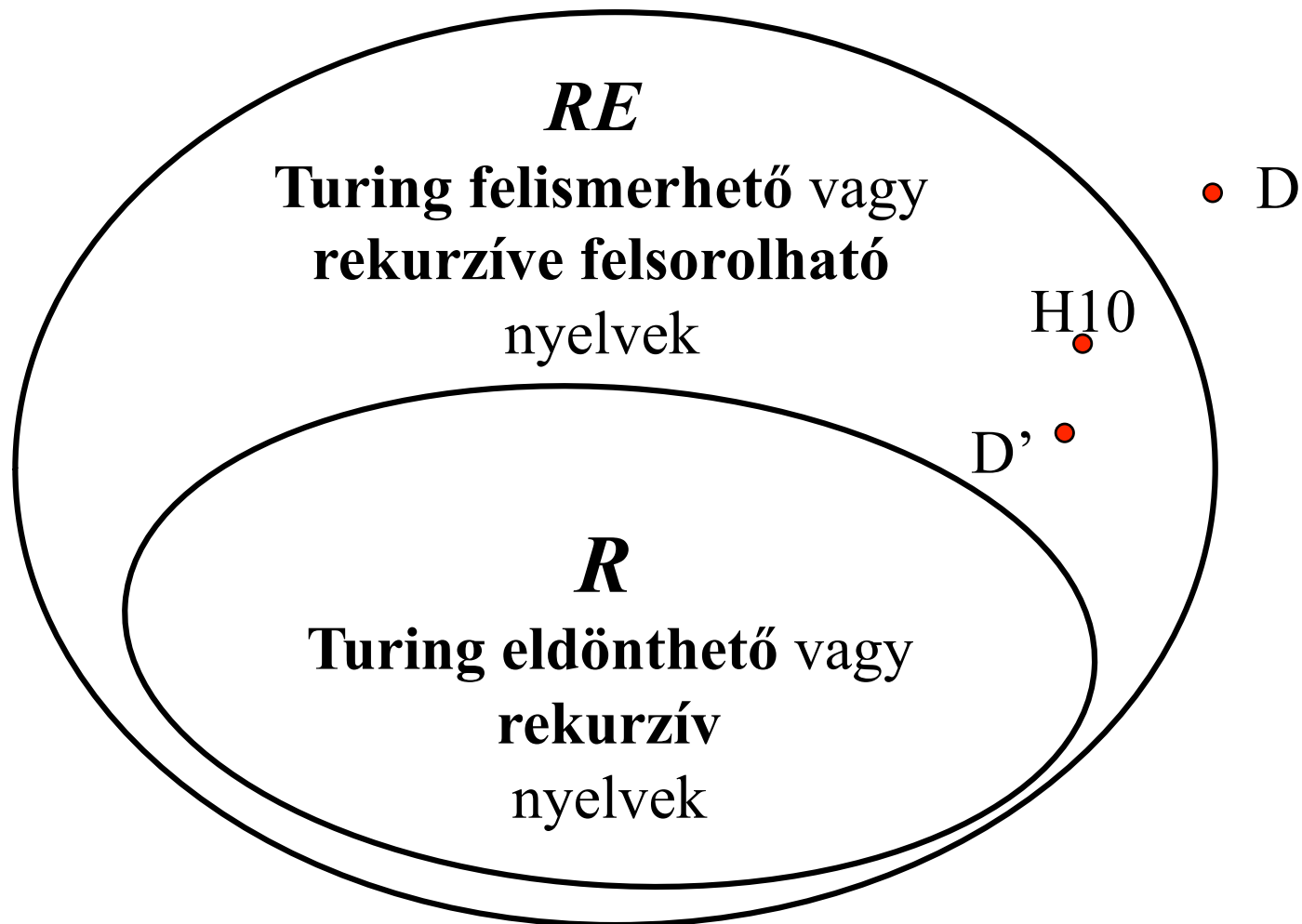
A digitális számítás elmélete

7. előadás március 21.

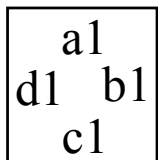
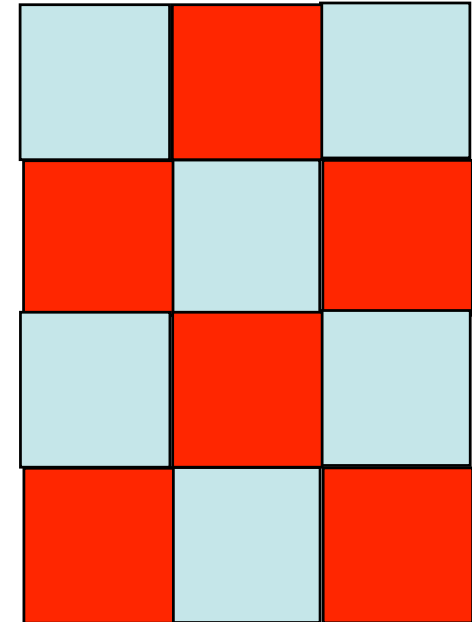
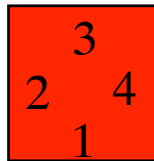
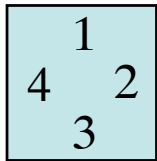
- További példák nem Turing-eldönthető problémákra
 - dominó
 - kongruens számok ... nem tudjuk
- Turing gép megadása kóddal
- Van nem Turing-felismerhető nyelv
 - számossági meggondolás
 - konkrét nyelv megkonstruálása
- Az univerzális Turing gép
- A megállási probléma

Nyelvek felismerhetősége – problémák eldönthetősége

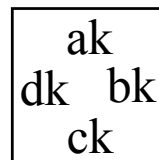
Az összes nyelv



A Dominóprobléma



...



k féle dominóval lefedhető-e a sík

$$F = a_1 * b_1 * c_1 * d_1 \# a_2 * b_2 * c_2 * d_2 \# \dots a_k * b_k * c_k * d_k \#$$

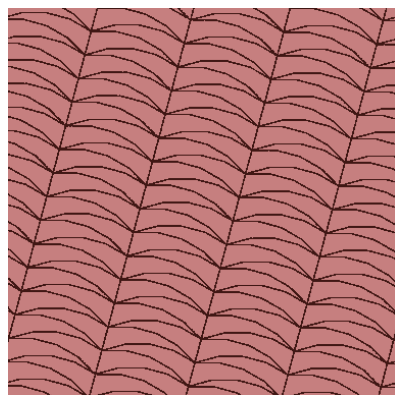
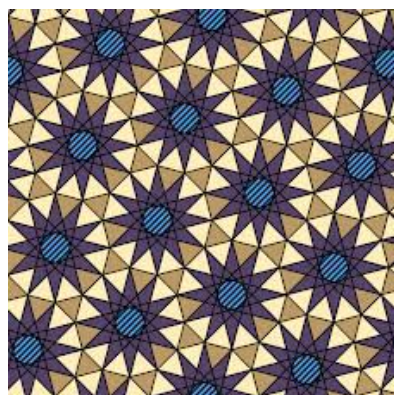
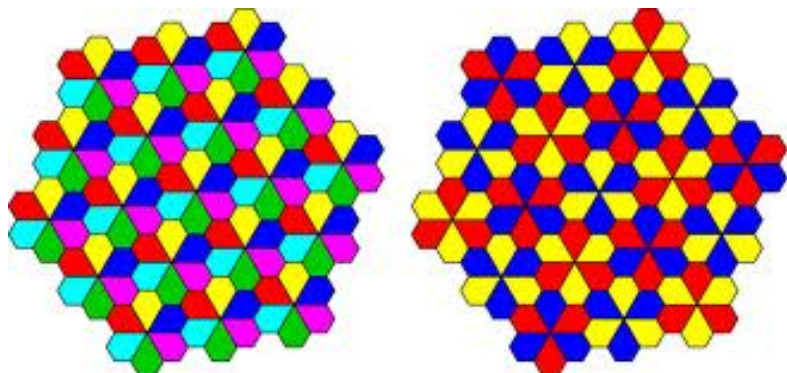
$$D = \{F \in \{0, 1, \dots, 9, *, \#\}^* \mid \text{a sík lefedhető az } F \text{ féle dominókkal}\}$$

D nem eldönthető, és nem is felismerhető, azaz

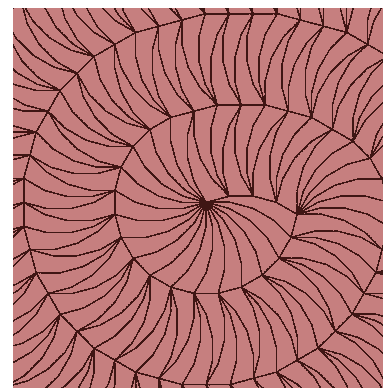
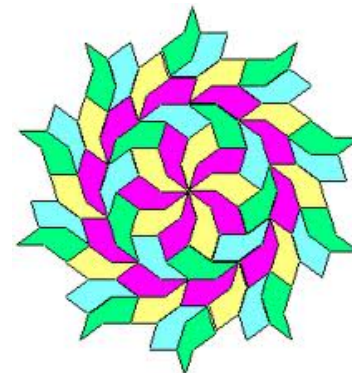
D nem rekurzív, és nem is rekurzíve felsorolható.

Periodikusság

- Periodikus:

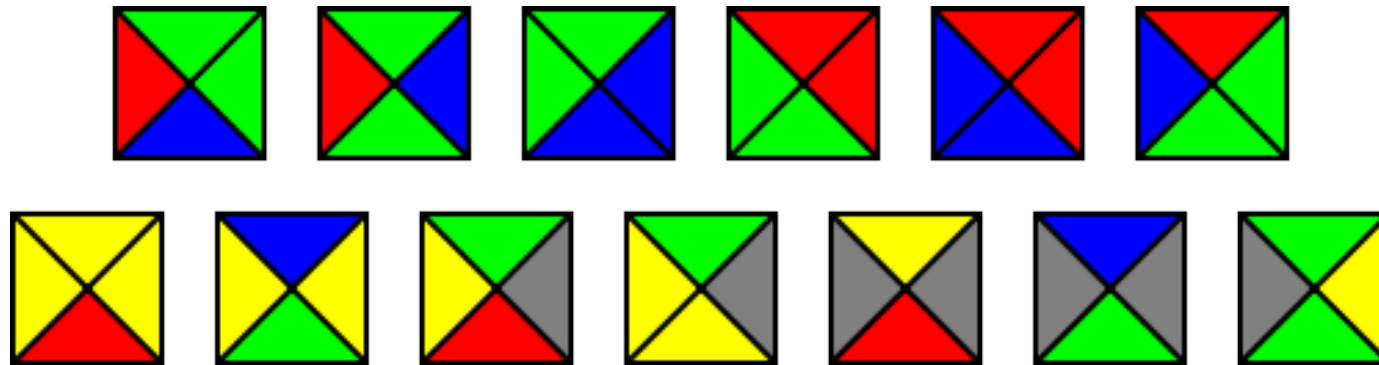


Nem periodikus:



Dominó probléma

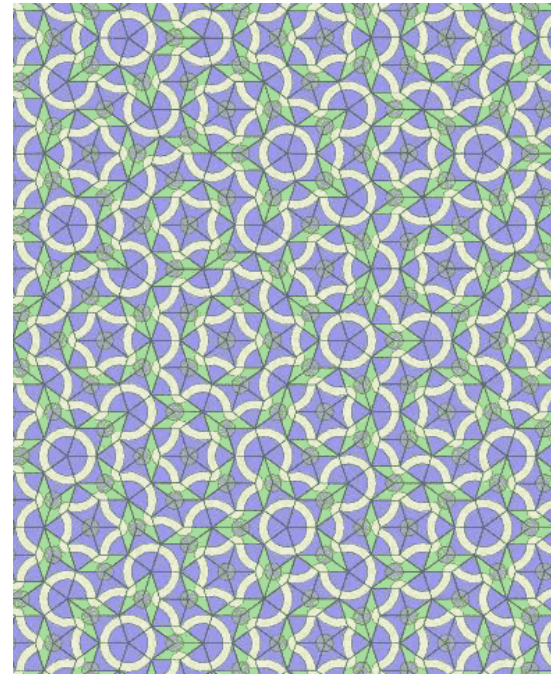
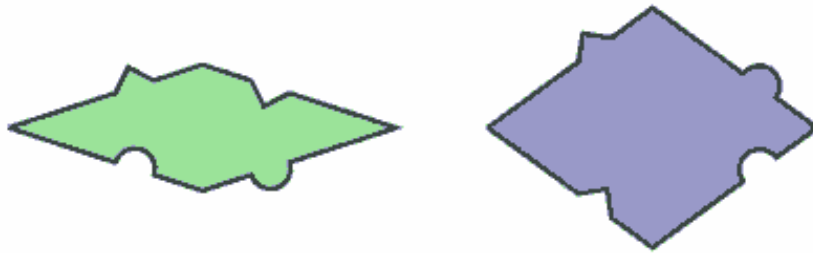
- Adott néhány csempefajta, oldalak színei illeszkedjenek.



- Bármekkora fürdőszobára elég?
 - Ha nem elég, akkor az kiderül (végigpróbálva)
 - Ha van periodikus, kiderül (végigpróbálva)

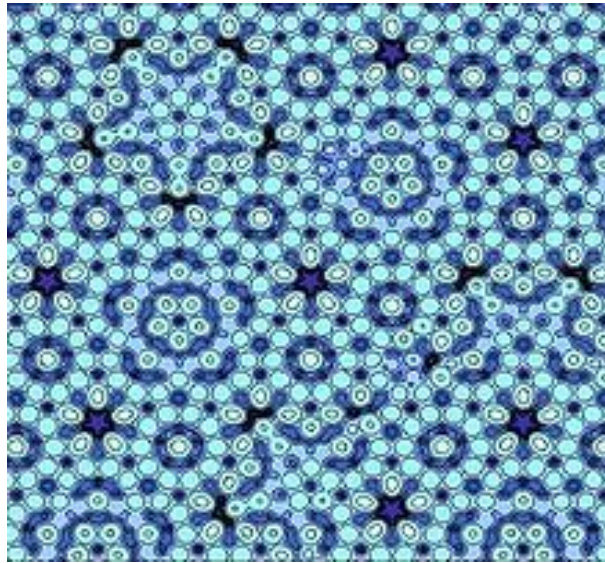
Kvázikristályok

- Léteznie kell olyan csempéknek, amikkel csak NEM periodikusan csempézhetünk.
- Penrose



Kvázikristályok

- Három dimenzióban: Al-Pd-Mn kristály, nem periodikus



- Schechtman: Nobel-díj 2011.

Egy felismerhető verzió (PCP)

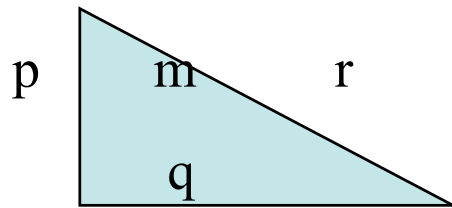
- $\begin{array}{|c|} \hline a_1 \\ \hline b_1 \\ \hline \end{array}, \quad \dots, \quad \begin{array}{|c|} \hline a_k \\ \hline b_k \\ \hline \end{array}, \quad a_i, b_i \in \{0,1\}\text{-felelti nem üres szavak.}$

$$F = a_1 * b_1 \# a_2 * b_2 \# \dots a_k * b_k \#$$

$$D' = \{F \in \{0,1,*,\#\}^* \mid \text{van olyan } i_1, \dots, i_n \text{ indexsorozat, hogy} \\ a_{i_1} \dots a_{i_n} = b_{i_1} \dots b_{i_n}\}.$$

Nyitott kérdés: a kongruens számok felismerése

Definíció: Egy m pozitív szám **kongruens**, ha van olyan derékszögű háromszög, melynek területe m , és oldalai racionális számok.



$$p^2 + q^2 = r^2$$

$$pq = 2m$$

Pl. 5 kongruens: $3/2$, $20/3$, $41/6$

6 kongruens 3,4,5

Nem ismert, hogy a K (a kongruens számok problémája) eldönthető-e.

Megj: K rekurzíve felsorolható: Mivel a rac. számok megszámlálhatók, az azokból alkotott párok is azok. Tehát ha van olyan p, q hogy $pq = 2m$ akkor az kiderül véges sok lépés után. Majd egy ilyen párra az is véges sok lépésben kiderül, ha az a Pythagorasz tételnek eleget tesz valamely r rac. számot választva.

Nyelvek felismerhetősége – problémák eldönthetősége

Tétel: Van olyan $L' \subseteq \Sigma^* = \{0,1\}^*$ nyelv, amely

nem Turing felismerhető, azaz nem rekurzíve felsorolható.

Biz: Egy TG leírható véges jelsorozattal, pl. a definíciós megadással.

Az összes TG felsorolható, mivel k hosszú jelsorozattal megadott TG véges sok van. Legyen $M_1, M_2, \dots, M_n, \dots$ egy felsorolásuk (lásd később részletesen).

Az M_i TG egyértelműen meghatározza az általa felismert L_i nyelvet,

így a Turing felismerhető nyelvek is felsorolhatók: $L_1, L_2, \dots, L_n, \dots$

Azaz a **Turing felismerhető nyelvek megszámlálhatóan sokan** vannak.

Másrészt: Az **összes Σ feletti nyelv számossága** éppen a $\{0,1\}^*$ lehetséges részhalmazainak a számossága, ami **több, mint megszámlálható**. (Biz. középiskola, ill. Cantor módszer.)

Egy nem Turing felismerhető nyelv

Feltesszük az $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{elfogad}}, q_{\text{elutasít}})$ TG-ről, hogy:

- $Q = \{0, 1, \dots, q\}$, 0 a kezdő, q az elfogadó, q-1 az elutasító állapot
- $\Sigma = \{0, 1\}$ bemeneti abc,
- $\Gamma = \{0, 1, \dots, s\}$ szalag abc, s felel meg az üres jelnek, _-nek
- $J=1, B=0$

Ekkor M egyértelműen megadható egy $\{0, 1\}^*$ -beli w szóval, ez M **kódja**. Másrészt egy $\{0, 1\}^*$ -beli szóhoz legfeljebb egy M_w tartozik, aminek a jellegzetességei (állapotok, állapotátmenet fv, abc-k) megadhatók w alapján.

Tehát a TG-eket tekinthetjük speciális $\{0, 1\}^*$ -beli szavaknak.

Egy nem Turing felismerhető nyelv

Pl: Egy M TG megadható a köv. módon:

$$q\#s\# q_0\#x_0\#q_0' \# x_0' \# m_0'\#q_1\#x_1\#q_1' \# x_1' \# m_1' \# \dots \\ \dots q_r\#x_r \#q_r' \#x_r' \#m_r'$$

$$\text{ahol } \delta(q_i, x_j) = (q_i', x_j', m_j')$$

Írjuk fel az egész számokat binárisan a fenti sorozatban.

Az így kapott sorozatban alkalmazzuk a következő helyettesítést:

$$\# = 11, 1 = 01, 0 = 00.$$

Az egyértelműség érdekében az állapotátmenetek kódjait (azaz a

$$q_i\#x_i \#q_i' \#x_i' \#m_i' \text{ -nek megfelelő sorozatokat})$$

lexikografikusan soroljuk fel.

Ezzel M-et egy $\{0, 1\}^*$ -beli w szóval írtuk fel.

Egy nem Turing felismerhető nyelv

Jelölje M_w a w által kódolt Turing-gépet.

Megjegyzés: a $\{0, 1\}^*$ -beli szavak felsorolhatók.

Pl. a **kanonikusan**: Az üres szót, majd az 1, 2, ..., k , ...

hosszúságúakat soroljuk fel. Az azonos hosszú sorozatokat

lexikografikusan soroljuk fel, így: ε , 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, ...

Egy nem Turing felismerhető nyelv: L_d

Def: Diagonális nyelv: azon TG kódja, mely TG-ek nem fogadják el a saját kódjukat

$$L_d = \{ w \in \{0, 1\}^*, \text{ az } M_w \text{ gép létezik, és } w \notin L_{M_w} \}$$

Tétel: L_d nem rekurzíve felsorolható nyelv.

Biz: Indirekten, t. fel, hogy L_d rekurzíve felsorolható. Azaz van N TG, mely éppen L_d szavait ismeri fel, $L_d = L_N$. Azaz N felismeri w-t, ha w egy olyan M_w TG-t kódol, mely nem fogadja el a saját kódját.

Mi a helyzet N-nel? Legyen N kódja s. Felismeri-e N s-et, azaz $s \in L_d$ teljesül-e?

Ha feltesszük hogy igen, akkor a def. szerint $s \notin L_d$, ELLENTMONDÁS!!

Ha feltesszük hogy nem, akkor a def. szerint $s \in L_d$, ELLENTMONDÁS!!

Tehát mindkét esetben képtelenségre jutottunk, így az indirekt feltevésünk hamis.

Az univerzális Turing gép

Képes szimulálni tetszőleges, a w kódjával adott M_w TG-et.

Tétel: Van olyan U 3 szalagos TG, melyre teljesül, hogy ha

$w, s \in \{0, 1\}^*$, és M_w létezik, akkor az U gép pontosan akkor fogadja el (utasítja el, kerül végtelen ciklusba) az $w\#s$ bemenetet, amikor M_w az s bemenetet elfogadja (elutasítja, ciklusba kerül).

Bizonyítás: A 3 szalag szerepe

$w\#s$	nem változik, M_w működését adja meg
s	M_w szalagjának a szimulálása
0	M_w pillanatnyi állapota

Az univerzális Turing gép

U működése:

0. Ellenőrzi, hogy w TG-t ír-e le, ha nem, elutasít. Ha igen, beírja s -t a 2., és 0 -t (a kezdőállapot kódját) a 3. szalagra.

1. Az első szalag alapján meghatározza, hogy M_w mit lépne, ha a 3. szalagon megadott állapotban lenne, és az olvasófej alatt a 2. szalagon szereplő jelet látná.

U megteszi M_w lépését, és megfelelően módosítja a 2. és 3. szalag tartalmát.

U megáll elfogadó ill. elutasító állapotban, ha az M_w szerinti új állapot éppen elfogadó ill. elutasító. Ha nem, akkor az 1. lépést ismétli.

Az univerzális nyelv, L_u

Kérdés: Eldönthető-e, hogy egy adott M gép elfogad-e egy adott $s \in \{0, 1\}^*$ szót? ('Szuper' algoritmust keresnénk.)

Def: Univerzális nyelv

$$L_u = \{ w\#s \in \{0, 1\}^*, \quad M_w \text{ létezik,} \quad \text{és } s \in L_{M_w} \}$$

Megj: L_u éppen az univerzális TG nyelve,

azaz $L_u = L_U$, ahol U egy univ. TG.

Tehát L_u rekurzíve felsorolható.

Az univerzális nyelv, L_u

Tétel: (Turing, 1936) L_u nem rekurzív nyelv.

Biz: Indirekten, feltesszük hogy L_u rekurzív nyelv. Ekkor van N TG, mely eldönti L_u -t, azaz minden inputra megáll, és $L_u = L_N$.

Legyen N' a következők szerint működő TG:

1. Ha a $w \in \{0, 1\}^*$ bemenet nem TG kódja, elutasítja.
2. Ha a w bemenet egy TG kódja, akkor elindítja N -t az $w\#w$ bemenettel. Ha ez a szimuláció elfogadó állapotban megáll, akkor N' elutasító állapotban végez, és fordítva, elutasító N állapot esetén a N' végállapota elfogadó.

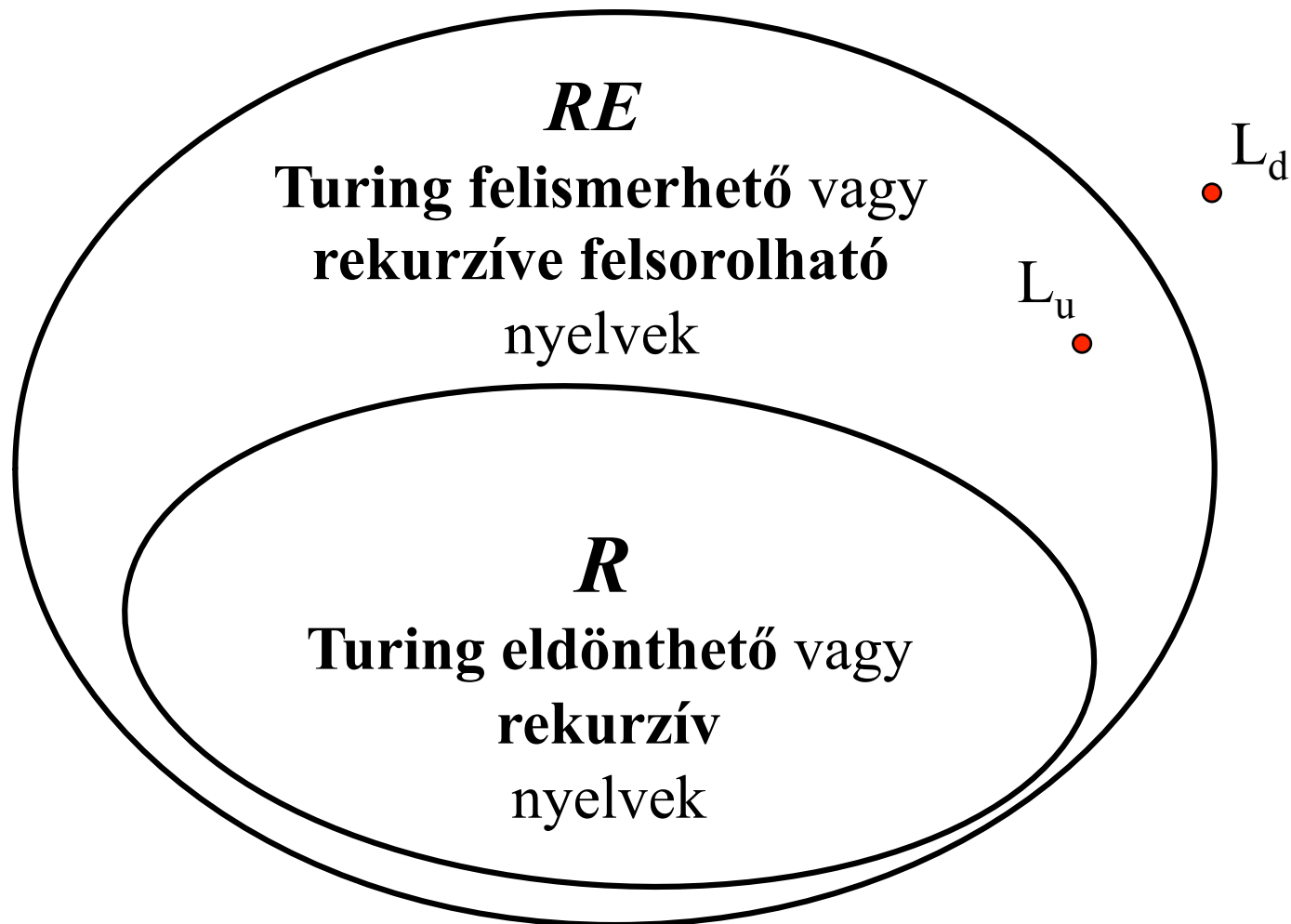
Az univerzális nyelv, L_u

N' minden w bemenetre megáll, az ind. feltevés miatt.

N' pontosan akkor fogadja el w -t, ha M_w létezik, és $w \notin L_M$.

Tehát az $L_{N'}$ a nyelv éppen az L_d nyelv, amelyről láttuk, hogy nem rekurzíve felsorolható. Tehát ellentmondásra jutottunk, az ind. feltevésünk hamis kell hogy legyen, amiből következik, hogy az eredeti állítás igaz.

Nyelvek felismerhetősége – problémák eldönthetősége



A megállási probléma

Kérdés: Eldönthető-e, hogy egy adott M TG gép megáll-e minden bemenetre? ('Nincs-e végtelen ciklus a programunkban'?)

Def: A megállási problémához (halting problem) tartozó L_h nyelv

$$L_h = \{w\#s \in \{0, 1\}^*, M_w \text{ létezik, és } s \text{ bemenettel elindítva megáll}\}$$

Tétel: L_h rekurzíve felsorolható, de nem rekurzív nyelv.

Biz: Legyen U egy univ. TG. Ez L_h szavaira meg fog állni. Legyen U' az a TG, mely azonos U-val, kivéve hogy U elutasít állapotát elfogadra változtattuk. Ekkor $L_h = L_{U'}$, azaz L_h rek. felsorolható.

A megállási probléma

Biz: Azt indirekte mutatjuk meg, hogy L_h nem rekurzív.

Ind. felt: L_h rekurzív, azaz van olyan M TG, mely mindig megáll, és

$$L_h = L_M$$

Legyen M' az a TG, mely először M -et futattja, majd ha M elfogadó állapotban végzett, U -t.

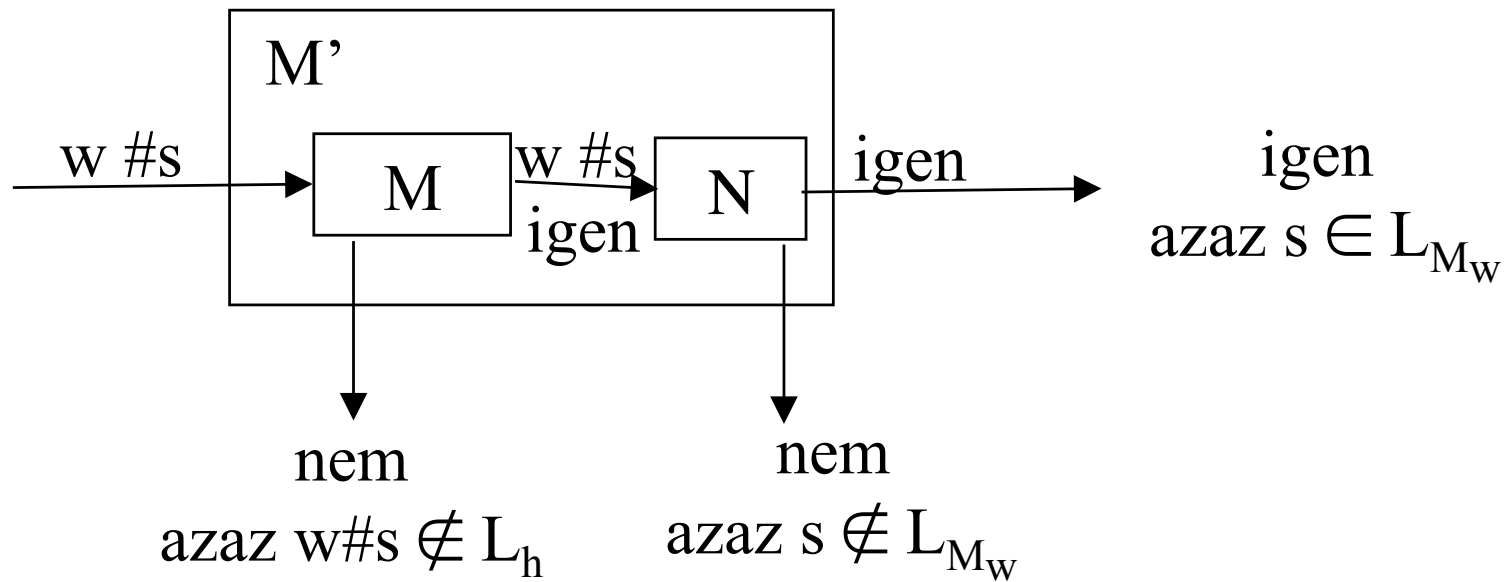
M' mindig megáll az ind. felt. miatt, és M' éppen L_u -t ismeri fel.

De L_u -ról láttuk, hogy nem rekurzív. Az ellentmondás oka az ind.

feltevés, az hamis tehát, azaz a tétel második részének állítása is igaz.

A megállási probléma

Ábra:



A megállási probléma üres inputra

Kérdés: Eldönthető-e, hogy egy adott M TG gép megáll-e az üres bemenetre?

Def: $L_\varepsilon = \{w \in \{0, 1\}^*, M_w \text{ létezik, és az } \varepsilon \text{ (üres) bemenettel elindítva megáll}\}$

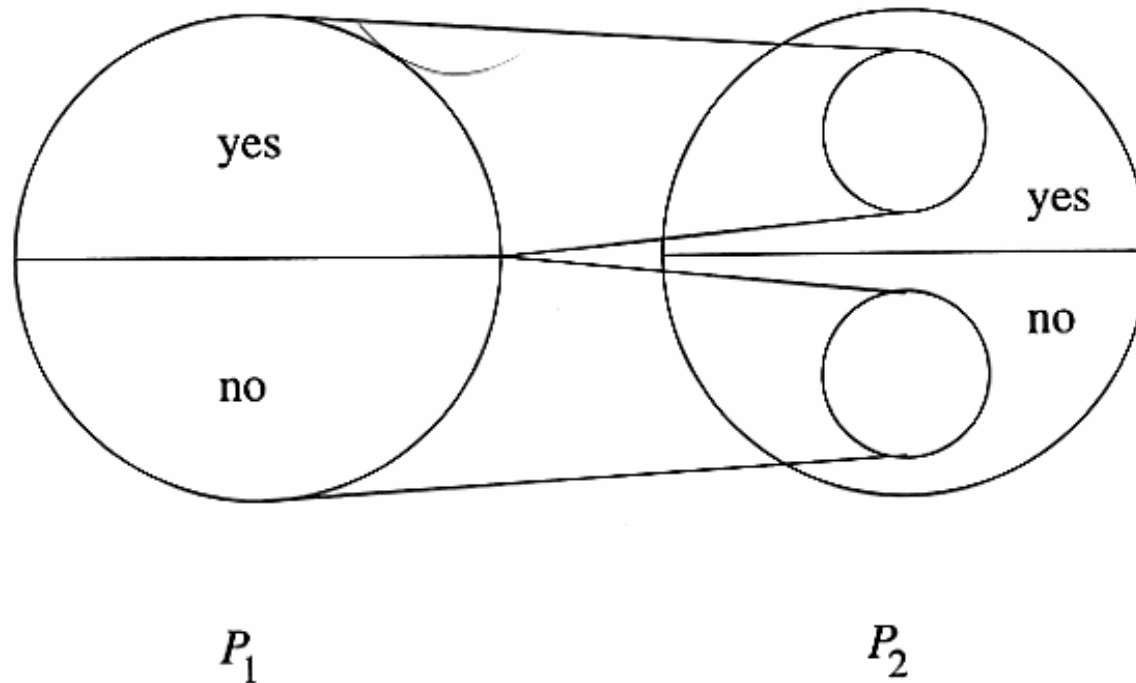
Tétel: L_ε rekurzíve felsorolható.

Biz: A rek. felsorolhatóságot bizonyítja, hogy egy $w \in L_\varepsilon$ esetén az L_h megállási nyelvet felismerő automatát a $w\#\varepsilon = w\#$ inputtal futtatva, ez a gép éppen L_ε -t ismeri fel.

Tétel: L_ε nem eldönthető. Bizonyítás: **visszavezetéssel.**

Visszavezetések

P_1 visszavezetése P_2 -re ($P_1 \leq P_2$):



Def. $L_1 \subseteq \Sigma^*$ visszavezethető az $L_2 \subseteq \Delta^*$ nyelvre, ha van olyan Turing-géppel kiszámítható $f: \Sigma^* \rightarrow \Delta^*$ függvény, hogy minden $w \in \Sigma^*$ szóra teljesül, hogy $w \in L_1 \Leftrightarrow f(w) \in L_2$.

Visszavezetések

Tétel: Ha $L_1 \leq L_2$ és

- L_2 eldönthető (rekurzív), akkor L_1 is eldönthető.
- L_2 Turing-felismerhető (rekurzívan felsorolható), akkor L_1 is Turing-felismerhető.

Biz.: pl. a visszavezetést kiszámoló és az L_2 -t eldöntő Turing-gép „kompozíciója” az L_1 nyelvet dönti el.

Hogyan lehet ezt eldönthetetlenség bizonyítására használni?

Ha $L_1 \leq L_2$ és L_1 nem eldönthető, akkor L_2 sem lehet eldönthető.

A megállási probléma eldönthetlensége újra

Tétel: L_h eldönthetetlen.

Biz.: Megmutatjuk, hogy $L_u \leq L_h$. Tetszőleges M Turing-géphez és s bemenethez konstruáljuk meg az M' Turing-gépet a következő módon.

M' annyiban különbözik M -től, hogy amikor M elutasító állapotba lép, akkor M' végtelen ciklusba lép. Minden további esetben az M' másolja M működését. Legyen w és w' rendre az M és az M' kódja. Nyilvánvaló, hogy az $f(w)=w'$ fgv. kiszámítható egy Turing-géppel. Továbbá:

$$w\#s \in L_u \Leftrightarrow w'\#s \in L_h.$$

L_ε eldönthetlensége újra

Tétel: L_ε eldönthetetlen.

Biz.: Megmutatjuk, hogy $L_h \leq L_\varepsilon$.

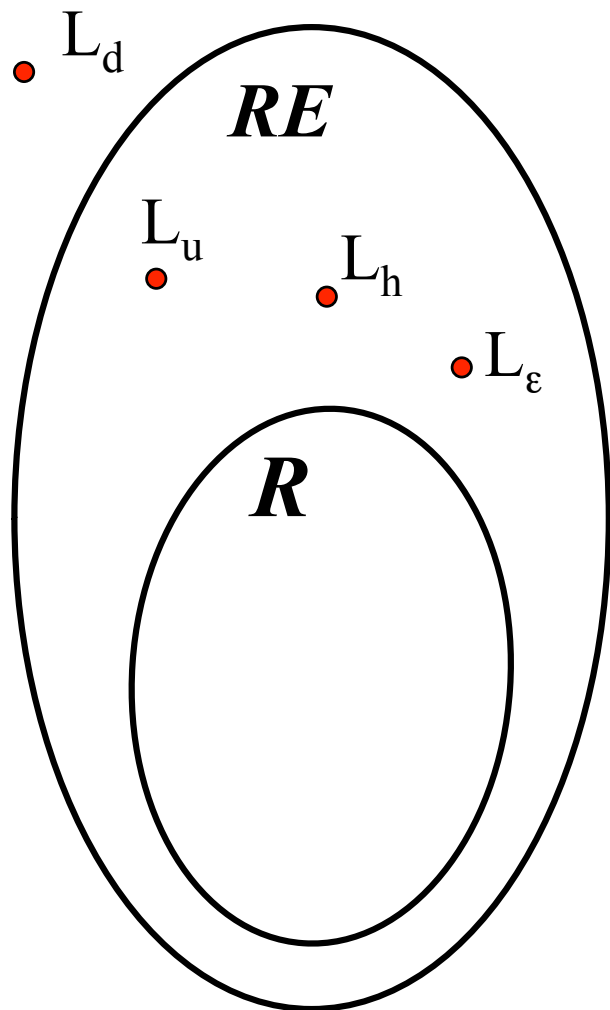
Adott M Turing-géphez és s bemenethez legyen M' a következő Turing-gép.

M' tetszőleges u bemeneten a következőt teszi. M' megvizsgálja u -t, és ha u nem az üres szó, akkor M' elfogadja a bemenetet. Ha u az üres szó, akkor M felmásolja a szalagjára az s szót és elkezdi szimulálni az M -et az s szón. Legyen rendre w és w' az M és az M' kódja. Az $f(w)=w'$ fgv. kiszámítható egy Turing-géppel. Továbbá:

$$w\#s \in L_u \Leftrightarrow w' \in L_h.$$

Áttekintés

Bizonyítások:



$L_d \notin RE$ Indirekte, közvetlenül Cantor diag.
elv.

$L_u \in R$ -ből következne, hogy $L_d \in RE$

$L_h \in R$ -ből következne, hogy $L_u \in R$

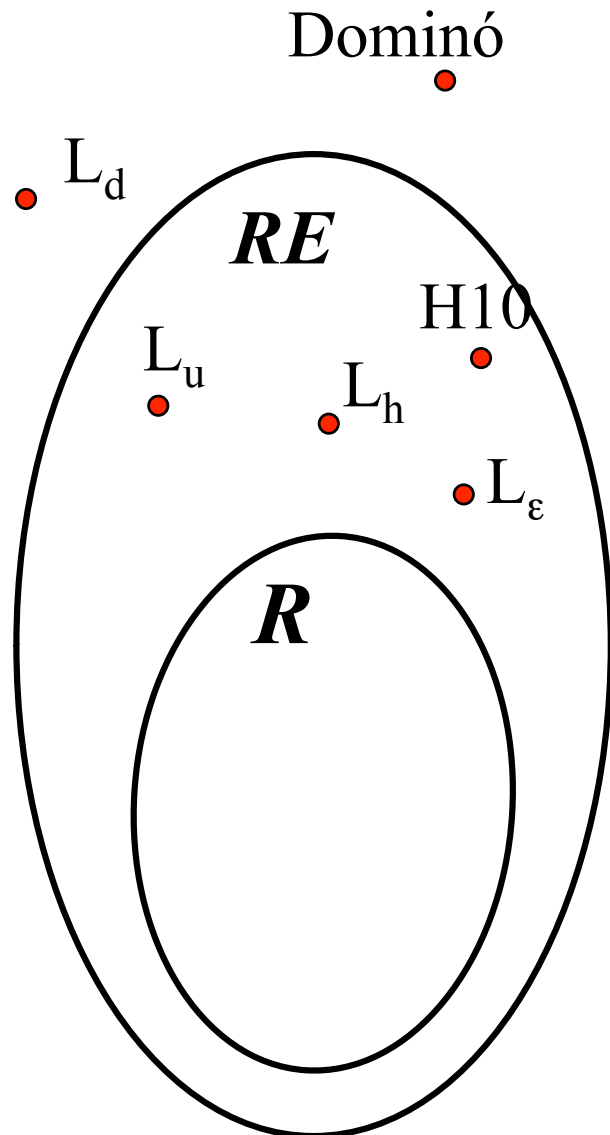
$L_\varepsilon \in R$ -ből következne, hogy $L_h \in R$

A digitális számítás elmélete

8. előadás március 28.

- Turing gépek és nyelvtanok
- A nyelvosztályok áttekintése
- Turing gépek és a természetes számokon értelmezett függvények

Áttekintés



Bizonyítások:

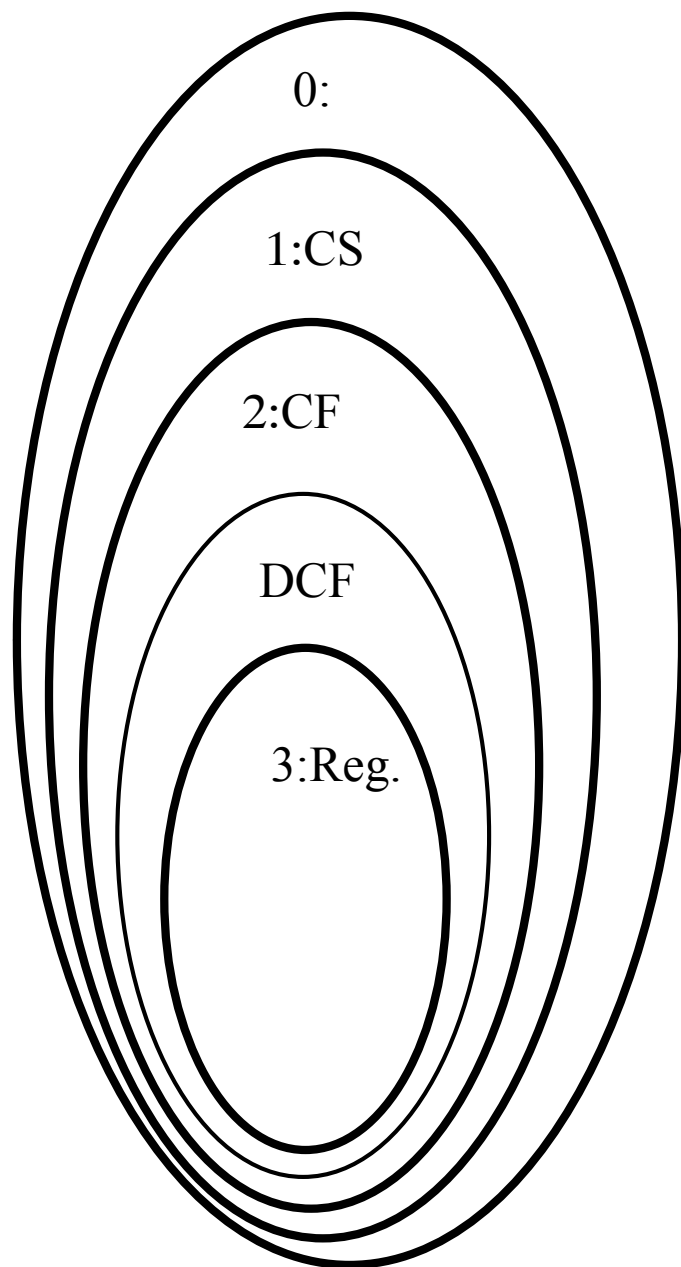
$L_d \notin RE$ Indirekte, közvetlenül Cantor diag.
elv.

$L_u \in R$ -ből következne, hogy $L_d \in RE$

$L_h \in R$ -ből következne, hogy $L_u \in R$

$L_\epsilon \in R$ -ből következne, hogy $L_h \in R$

Grammatikával generálható nyelvek (ism)



Generatív grammatikák (ism)

Definíció: Egy $G = (V, \Sigma, R, S)$ négyes generatív grammatika, ahol:

- V egy véges halmaz, a **grammatikai változók vagy nemterminálisok** halmaza,
- Σ egy véges halmaz, diszjunkt V -vel, a **terminálisok halmaza**,
- R egy véges halmaz, a **levezetési szabályok halmaza**, ahol egy levezetési szabály alakja

$$\alpha \rightarrow \beta$$

ahol α és β terminálisok és nemterminálisok sorozata,

- $S \in V$ a **kezdő- vagy mondatszimbólum**.

Generatív grammatikák (ism)

Definíció: A $G = (V, \Sigma, R, S)$ grammatika **generálja az** $L \subseteq \Sigma^*$
nyelvet, ha $L = \{ w \in \Sigma^* \mid S \Rightarrow^* w \}$

Jelölés:

$L(G)$ a G **grammatika által generált nyelv.**

Chomsky-féle nyelvosztályok (ism)

A nyelvet generáló grammatika szabályaira tett megkötések alapján:

3-as nyelvosztály:

reguláris (Reg)

$A \rightarrow a$, $A \rightarrow aB$ alakú szabályok

2-es nyelvosztály:

környezetfüggetlen (CF)

$A \rightarrow \alpha$ alakú szabályok

1-es nyelvosztály

környezetfüggő (CS)

$\beta A \gamma \rightarrow \beta \alpha \gamma$ alakú szabályok

$\alpha \rightarrow \beta$ alakú szabályok ahol $|\alpha| \leq |\beta|$ nem csökkenő

megj: ezek ekvivalensek!

0-ás nyelvosztály:

tetszőleges alakú szabályok

Rekurzíve felsorolható és nyelvtannal generálható nyelvek kapcsolata

Tétel: A nyelvtannal generálható nyelvek osztálya megegyezik a rekurzíve felsorolható nyelvek osztályával. Azaz L nyelv akkor és csak akkor generálható grammatikával, ha van a nyelvet felismerő Turing -gép.

Biz: $\Rightarrow$ irány (a Turing-Church tézisből is következik/azt támasztja alá)

Megmutatjuk, hogy ha L grammatikával generálható, akkor van M TG, hogy $L(M)=L$.

Legyen $G = (V, \Sigma, R, S)$ grammatika mely L -et generálja.

A grammatika szerinti levezetés egy-egy lépését meghatározza, hogy mely szabályt alkalmazzuk, és annak bal oldalát az éppen aktuális mondatszerű forma hányadik elemére illesztjük.

Rekurzíve felsorolható és nyelvtannal generálható nyelvek kapcsolata

R	nem változik, a szabályok kódolt alakja
w	a bemeneti szó
S	a k lépéssel levezethető mondatszerű formák
	munkaszalag, a k-1 lépés eredménye

Négyszalagos, ‘univerzális’, nyelvtan szerinti levezetést szimuláló

TG-et határozunk meg.

M működése:

1. Átmásolja a 3. szalagot a 4-re, majd törli 3 tartalmát.
2. A 4. szalagon szereplő minden egyes mondatszerű formára:
 - a) Minden egyes $\alpha \rightarrow \beta$ szabályra megkeresi mindazon helyeket a mondatszerű formában, ahol α előfordul.
 - b) Minden egyes ilyen esetben a mondatszerű formát úgy toldja hozzá a 3. szalagon szereplő sorozatokhoz, hogy elválasztójel után a mondatszerű forma megfelelő helyén α helyett β szerepel.
3. Megvizsgálja, hogy a 4. szalagon szerepel-e w -vel azonos mondatszerű forma. Ha igen, elfogadó állapotban megáll, ha nem, 1-re lép.

M éppen a G grammatika által generált szavakat fogja elfogadni.

Rekurzíve felsorolható és nyelvtannal generálható nyelvek kapcsolata

Biz: $\Leftarrow$ irány (vázlat)

Egy $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{elfogad}}, q_{\text{elutasít}})$ TG-hez megkonstruáljuk azt a $G = (V, \Sigma, R, S)$ grammatikát, amely éppen $L(M)$ -et generálja.

Feltesszük, hogy ha M megáll, akkor a szalag üres. (Ezek nem megszorító feltevések, mivel egy tetszőleges TG-hez megadható vele ekvivalens ilyen TG.)

M működése során egy pillanatnyi konfigurációt megadunk egy jelsorozattal: $u p v$, ahol p egy állapot, uv a szalagon levő sorozat a végtelen sok $_$ jel előtt, a fej v első elemén áll.

Például, az $x a p b y$ konfiguráció rákövetkezője az $x a c q y$, ha $\delta(p, b) = (q, c, J)$ (a fej jobbra lépett)

Rekurzíve felsorolható és nyelvtannal generálható nyelvek kapcsolata

A G grammatika V változói Q és Σ szimbólumai, valamint S és határ szimbólum.

A szabályok M működését visszafelé írják le:

1. Ha $\delta(p, a) = (q, b, J)$, akkor $b \ q \rightarrow p \ a$
2. Ha $\delta(p, a) = (q, b, B)$, akkor $q \ b \rightarrow p \ a$
3. Szabályok az input elé és után beszúrt üres ill. határ szimbólumok törlésére

Az egyetlen elfogadó állapot felel meg a kezdő szimbólumnak:

$S \rightarrow q_{\text{elfogad}} _$

Ha M egy inputra az elfogadó állapotba érkezik, akkor az input levezethető a fenti szabályokkal, és szó levezetésnek megfelel M konfigurációinak egy sorozata, amely a szót mint inputot a végállapotba viszi. Tehát G éppen $L(M)$ -et generálja.

Környezetfüggő nyelvtannal generálható nyelvek felismerése TG-pel

Mi a helyzet az 1-es (CS) nyelvosztállyal?

Def: **Lineárisan korlátozott TG** egy olyan, speciális TG, mely a szalagnak csak a bemeneti által elfoglalt részét használja.

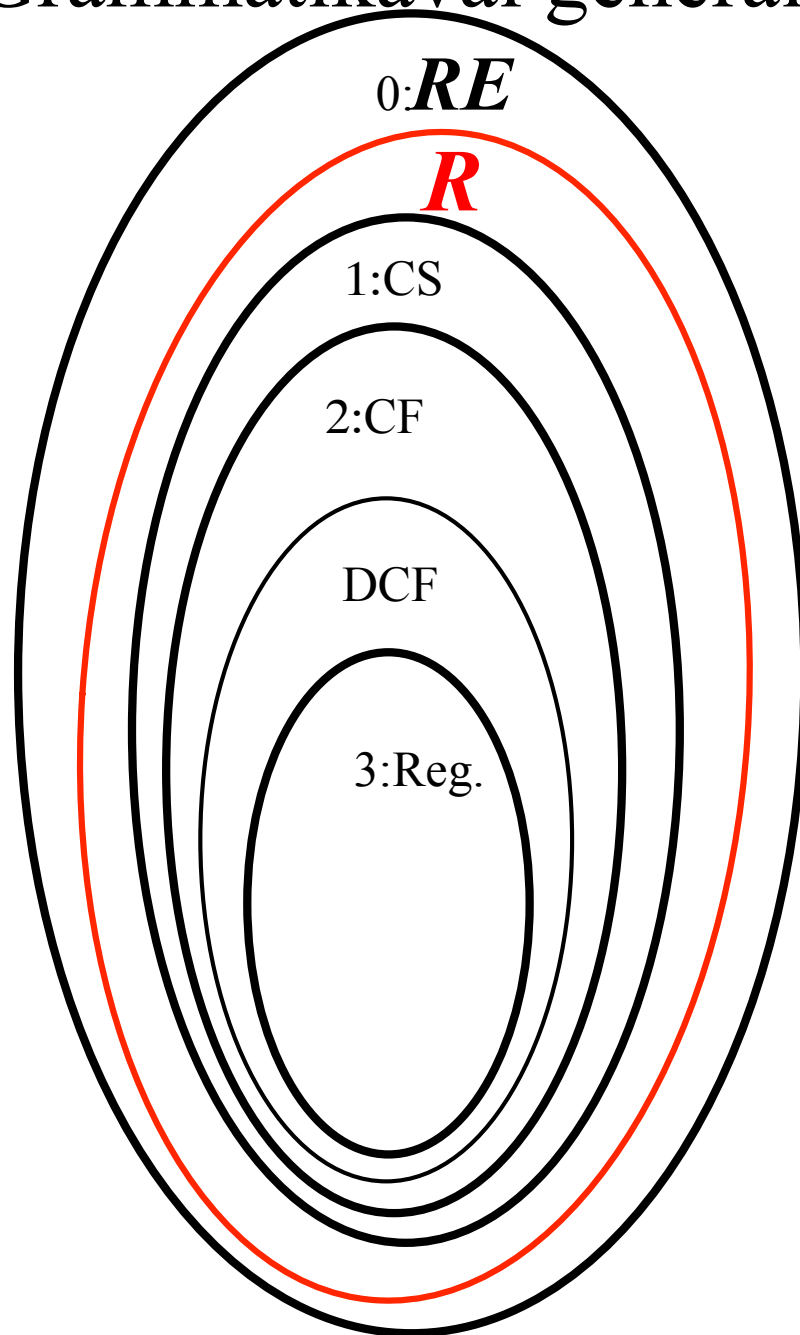
Tétel: A környezetfüggő nyelvtannal generálható nyelvek osztálya (CS) megegyezik a lineárisan korlátozott TG-pel felismerhető nyelvek osztályával.

Biz: nem részletezzük

Tétel: A környezetfüggő nyelvtannal generálható nyelvek osztálya (CS) valódi részhalmaza a rekurzív nyelvek osztályának.

Biz: nem részletezzük

Grammatikával generálható nyelvek összevetése



Numerikus függvények

A nemnegatív egészeket N jelöli. N^k -en értelmezett, N értékű függvényeket vizsgálunk.

$$\text{Pl. } f(n, m) = m n^2 + 3 m^{2m+17}$$

A fv értékét úgy számítjuk ki, hogy a kifejezéseit visszavezetjük alap kifejezésekre (pl. szorzás) és alacsonyabb kitevőjű kifejezésekre,

$$\text{pl: } m^n = m m^{n-1}$$

Numerikus függvények

Def: Az **alap függvények** a következők:

- a) $\mathbb{N} \rightarrow \mathbb{N}$ rákövetkező fv: $\text{succ}(\mathbf{n}) = n+1.$
- b) $\mathbb{N}^k \rightarrow \mathbb{N}$ ($k \geq 0$) k -változós zero fv: $\text{zero}_k(\mathbf{n}_1, \dots, \mathbf{n}_k) = 0.$
- c) $\mathbb{N}^k \rightarrow \mathbb{N}$ ($k \geq 0$) k -változós j -dik identitás fv: $\text{id}_{k,j}(\mathbf{n}_1, \dots, \mathbf{n}_k) = n_j.$

Bonyolultabb függvényeket a következő módokon kombinálunk:

Def: Legyen $k, m \geq 0$, és adottak a $g: \mathbb{N}^k \rightarrow \mathbb{N}$ illetve

$h_1, \dots, h_k: \mathbb{N}^m \rightarrow \mathbb{N}$ függvények. Ekkor **g kompozíciója a**

$h_1, \dots, h_k$ függvényekkel az az $f: \mathbb{N}^m \rightarrow \mathbb{N}$ fv, melyre:

$$f(n_1, \dots, n_m) = g(h_1(n_1, \dots, n_m), \dots, h_k(n_1, \dots, n_m)).$$

Numerikus függvények

Def: Legyen $k \geq 0$, és adottak a $g: \mathbb{N}^k \rightarrow \mathbb{N}$ illetve

$h: \mathbb{N}^{k+2} \rightarrow \mathbb{N}$ függvények. Ekkor **g és h által rekurzívan definiált függvény** az az $f: \mathbb{N}^{k+1} \rightarrow \mathbb{N}$ fv, melyre:

$$f(n_1, \dots, n_k, \mathbf{0}) = g(n_1, \dots, n_k).$$

$$f(n_1, \dots, n_k, \mathbf{m+1}) = h(n_1, \dots, n_k, m, f(n_1, \dots, n_k, \mathbf{m})).$$

Def: A **primitív rekurzív függvények** az alap függvények, és a belőlük tetszőleges, véges számú kompozícióval és rekurzióval származtatott függvények.

Numerikus függvények

1. pl: $\text{plus}(n,m) = n+m$ függvény rekurzív definíciója:

$$\text{plus}(n,0)=n, \text{plus}(n,m+1)=\text{plus}(n,m)+1.$$

Primitív rekurzív függvényként:

$$k=1, \quad g: \mathbb{N} \rightarrow \mathbb{N}, \quad g(n) = \text{id}_{1,1}(n) = n,$$

$$h: \mathbb{N}^3 \rightarrow \mathbb{N}, \quad h(n,m,p) = \text{succ}(\text{id}_{3,3}(n,m,p)).$$

Ekkor g és h által rekurzívan definiált $\text{plus}: \mathbb{N}^2 \rightarrow \mathbb{N}$ fv, melyre:

$$\text{plus}(n, 0) = g(n) = \text{id}_{1,1}(n) = n.$$

$$\begin{aligned} \text{plus}(n, m+1) &= h(n, m, \text{plus}(n, m)) = \text{succ}(\text{id}_{3,3}(n, m, \text{plus}(n, m))) \\ &= \text{succ}(\text{plus}(n, m)). \end{aligned}$$

Numerikus függvények

2. Pl: $\text{mult}(n,m)=nm$ függvény rekurzív definíciója:

$$\text{mult}(n, 0)=0, \text{mult}(n, m+1)=\text{plus}(n, \text{mult}(m,n)).$$

Primitív rekurzív függvényként:

$$\text{mult}(n,0) = \text{zero}(n),$$

$$\text{mult}(n,m+1) = h(n,m,\text{mult}(n,m))=\text{plus}(n,\text{mult}(m,n))$$

3. Pl: $\text{exp}(n, 0) = \text{succ}(\text{zero}(n)).$

$$\text{exp}(n, m+1) = \text{mult}(n, \text{exp}(n,m)).$$

Numerikus függvények

Tétel: A primitív rekurzív függvények számossága megszámlálhatóan végtelen.

Biz: A primitív rekurzív függvények megadhatók az alap függvények, a kompozíció és rekurzió, valamint zárójelek és term. számok szimbólumaiból álló véges sorozatokkal, ilyenből megsz. végtelen van.

Tétel: Nem minden kiszámítható fv. primitív rekurzív.

Biz: Cantor-féle átlós módszerrel, unáris fv-ekre

Soroljuk fel az összes primitív rekurzív unáris függvényt,

$f_1, f_2, \dots, f_n, \dots$

Legyen g az az unáris fv, melyre $g(n) = f_n(n) + 1$

Numerikus függvények

A g fv kiszámítható, és különbözik minden f_n -től, ugyanis $g(n) \neq f_n(n)$.

Példa totális, kiszámítható nem primitív rekurzív függvényre:

az Ackermann függvény.

$A(m,n) =$

$n+1$, ha $m=0$,

$A(m-1,1)$, ha $m>0$ és $n=0$,

$A(m-1,A(m,n-1))$, ha $m>0$ és $n>0$.

Numerikus függvények

Def: Legyen $k \geq 0$. A $g: \mathbb{N}^{k+1} \rightarrow \mathbb{N}$ függvény **minimalizációja** az az $f: \mathbb{N}^k \rightarrow \mathbb{N}$ függvény, melyet a következő módon értelmezünk:

$f(n_1, \dots, n_k) =$ a legkisebb m , melyre $g(n_1, \dots, n_k, m) = 1$,

ha van ilyen m

$f(n_1, \dots, n_k) = 0$ különben.

Numerikus függvények

Megj: A $g: \mathbb{N}^{k+1} \rightarrow \mathbb{N}$ függvény minimalizációja jól definiált, de nincs módszer a kiszámítására.

$m := 0,$

while $g(n_1, \dots, n_k, m) \neq 1$ do $m := m + 1,$

output m .

Nem algoritmus, mivel nem feltétlenül áll meg.

Def: Legyen $k \geq 0$. A $g: \mathbb{N}^{k+1} \rightarrow \mathbb{N}$ függvény **minimalizálható**, ha a fenti eljárás mindig megáll, azaz minden $n_1, \dots, n_k \in \mathbb{N}$ esetén van olyan $m \in \mathbb{N}$, hogy $g(n_1, \dots, n_k, m) = 1$.

Def: Egy f fv **μ -rekurzív**, ha előállítható az alap függvényekből a kompozíció, a rekurzió és a minimalizáció operátorokkal, az utóbbit minimalizálható függvényre alkalmazva.

Numerikus függvények

Tétel: Az $f: \mathbb{N}^k \rightarrow \mathbb{N}$ függvény μ -rekurzív akkor és csak akkor, ha rekurzív, azaz kiszámítható Turing géppel.

Biz: nem részletezzük

A digitális számítás elmélete

9. előadás április 4.

- Függvények aszimptotikus összehasonlítása
- Algoritmusok (TG-ek) idő és tárigénye
- A P osztály
- Tár-idő tétel

Hogyan értelmezzük algoritmusok hatékonyságát?

- Eldönthető nyelv (feladat) felismerésére van-e praktikusán használható algoritmus?

Milyen fajta feladatok vannak?

Eldöntendő kérdés Pl. prímszám-e az adott szám?

– nyelvbe tartozás

Hozzárendelés Pl. lnko, Hamilton-kör keresés

– fv kiszámítás

- Egy problémára (pl. szám prím voltának eldöntése, utazó ügynök) adott algoritmusokat hogyan értékeljük, hasonlítsuk össze?

Algoritmusok összehasonlítása

1. Időigény (lépésszám) illetve tárigény szerinti összehasonlítás
2. A feladat nagyságának (a megadás hosszának) függvényében, a nagy feladatokra koncentrálnak (asszimptotikus viselkedés)
3. A legrosszabb esetet tekintve (worst case)
4. Azonos fajta TG-pel és be és kimeneti reprezentációval megvalósított algoritmusok (ezt majd még pontosítjuk)

Algoritmusok összehasonlítása

Pl: T. fel hogy a gép 10^{12} művelet/sec sebességű. A és B algoritmust hasonlítjuk össze.

időigény	1 perc alatt	1 óra alatt
A $3000 n^5$	$n = 114$	$n = 260$
B 2^n	$n = 45$	$n = 52$

A 'nagy ordó' jelölés függvények asszimptotikus viselkedésének jellemzésére

Def: $f, g: \mathbb{N} \rightarrow \mathbb{R}^+$ függvények. $f(n) = O(g(n))$, ha található

c és n_0 pozitív egészek, hogy

$$f(n) \leq c g(n) \text{ ha } n \geq n_0.$$

Azt mondjuk, hogy $g(n)$ **asszimptotikus felső korlátja** $f(n)$ -nek.

<u>Pl:</u>	$f(n)$	$g(n)$
	$6n^3 + 3000n^2 + 10^{100}$	n^3
	$n^{100} + 2^n$	2^n
	$\log n + n$	n
	$\log_2 n + \log_3 n$	$\log n$ nem számít az alap

TG időkorlátja

Jelölés: Egy adott M TG esetén:

$T_M(n)$ jelöli M maximális lépésszámát (számolási idejét) az n hosszúságú bemenetek esetén.

$S_M(n)$ jelöli az M által maximálisan elolvasott munkaszalag jelek számát n hosszúságú bemenetek esetén.

Megj: Ha M nem áll meg valamely n hosszú bemenetre, akkor $T_M(n)$ végtelen. Ekkor $S_M(n)$ -t is végtelennek tekintjük.

Definíció: Legyen $t: \mathbb{N} \rightarrow \mathbb{N}$ függvény olyan, hogy $t(n) \geq n$ minden n -re. Az M TG **$t(n)$ időkorlátos**, ha az n hosszú bemenetek esetén legfeljebb $t(n)$ lépést tesz meg, azaz $T_M(n) \leq t(n)$.

Definíció: $\text{TIME}(t(n)) = \{L \mid L \text{ felismerhető egy } O(t(n)) \text{ időkorlátos } M \text{ TG-pel.}\}.$

TG tárkorlátja

Definíció: Legyen $s: \mathbb{N} \rightarrow \mathbb{N}$ függvény olyan, hogy $s(n) \geq \log_2 n$ minden n -re. Az M TG **$s(n)$ tárkorlátos**, ha az n hosszú bemenetek esetén legfeljebb $s(n)$ db cellát használ a munkaszalagon, azaz $S_M(n) \leq s(n)$.

Definíció: $\text{SPACE}(s(n)) = \{L \mid L \text{ felismerhető egy } O(s(n)) \text{ tárkorlátos } M \text{ TG-pel.}\}.$

TG időkorlátja

P1: $L = \{0^k 1^k \mid k \geq 0\}$. Egy L -et eldöntő TG:

1. Keres 0 jelet 1 után, ha van elutasít, ha nincs, visszamegy a szalag elejére.
2. Ha van még 0 és 1 a szalagon, ismételd:
Menj a szalag elejére, és húzd át a legelső 0-t és 1-t.
3. Ha maradt áthúzatlan 0 vagy 1, elutasít, különben **elfogad**.

Időigény: $O(n^2)$

Megj: 2-szalaggal $O(n)$ lépésben eldönthető a nyelv

A **P** osztály

P1: $\text{TIME}(n)$ a lineáris időben felismerhető nyelvek osztálya.

$\text{TIME}(n^2)$ a négyzetes időben felismerhető nyelvek osztálya.

Definíció: $\mathbf{P} = \bigcup_{k \geq 1} \text{TIME}(n^k)$ a polinom időben
egyszalagos TG-pel eldönthető nyelvek osztálya.

1. Megj: Valamely, az alappal ekvivalens speciális TG modellt
használva, ha az polinom időben szimulálható alap TG-pel, akkor
 $\mathbf{P}$ invariáns arra nézve, hogy az alap vagy a speciális TG-et
használjuk számítási modellként.

2. Megj: $\mathbf{P}$ a praktikus számítógéppel megoldható problémáknak
felel meg. Általában találunk $\mathbf{P}$ -beli problémákra gyors
(négyzetes, köbös,...) algoritmust

Példák P-beli problémákra

Ismert polinomiális algoritmus az alábbi problémákra:

PATH = $\{ \langle G, s, t \rangle \mid G \text{ irányított gráf, és van benne irányított út } s\text{-ből } t\text{-be} \}$.

RELRPIME = $\{ \langle x, y \rangle \mid x \text{ és } y \text{ relatív prímek} \}$.

GRAPH-2-COLOR = $\{ G \mid G \text{ gráf csúcsai kiszínezhetők 2 színnel} \}$.

EULER-CYCLE = $\{ G \mid G \text{ gráfban van Euler kör} \}$.

PRIME = $\{ x \mid x \text{ prím} \}$. **2002-es eredmény!**

Példák mindmáig nem **P**-beli problémákra

Nem sikerült eddig polinomiális algoritmus találni pl. az alábbi problémákra:

HAMILTON-CYCLE = $\{G \mid G \text{ gráfban van Hamilton kör}\}$.

GRAPH-3-COLOR = $\{G \mid G \text{ gráf csúcsai kiszínezhetők 3 színnel}\}$.

TG-ek szimulációjának idő és tárigénye

Láttuk, hogy speciális TG-ek (többszalagos, RAM, ... nem determinisztikus) szimulálhatók alap TG-pel. Ehhez több idő illetve tár kell, mennyivel? Mi az egyszerűbb gépeken a hatékonyságromlás mértéke?

Tétel: Az N k -szalagos TG-hez van olyan egyszalagos M TG, hogy $L(M) = L(N)$, és ha N $f(n)$ időigényű, akkor M $O(f(n)^2)$ időigényű.

Biz: nem részletezzük, a korábbi konstrukcióhoz hasonló

TG-ek szimulációjának idő és tárigénye

Tétel: Az N TG-hez van olyan többszalagos M TG, hogy egy alkalmas n_0 számon túl

a) tetszőleges $\varepsilon > 0$ esetén $T_M(n) \leq n(1 + \varepsilon)$, ha $T_N(n) = O(n)$

b) tetszőleges $c > 0$ esetén $T_M(n) \leq c T_N(n)$, ha $\lim T_N(n)/n = \infty$

Biz: nem részletezzük.

Megj: A fenti tétel magyarázza, hogy miért tekinthettünk el a konstans szorzótól az időigény definiálásakor.

Tár-idő tétel

Tétel: Ha $L \in \text{SPACE}(s(n))$, akkor van olyan L -től függő c konstans, hogy $L \in \text{TIME}(2^{s(n)})$, .

Biz: (vázlat) Legyen M egy L -et $O(s(n))$ tárkorláttal eldöntő Turing-gép és legyen w az M egy tetszőleges n hosszú bemenete. Ekkor a kezdőkonfigurációból elérhető konfigurációk száma $cn2^{O(s(n))} = 2^{O(s(n))}$, ahol c egy a szalagábécétől függő konstans.

Mivel M $2^{O(s(n))}$ lépése után konfiguráció ismétlődés áll fenn, egyes konstrukcióval elérhető, hogy az M -et szimuláló N Turing-gép $2^{O(s(n))}$ lépés után észrevegye ha végtelen ciklusba esik. Következik, hogy N az L -et ismeri fel és $2^{O(s(n))}$ időigényű, ami bizonyítja a tétel állítását.

Nyelvosztályok komplementere

Jelölés: L I feletti nyelv esetén $L^C = I^* \setminus L$ az L komplementere.

Def: $\mathcal{L}$ nyelvosztály esetén $\text{co}\mathcal{L} = \{L \mid L^C \in \mathcal{L}\}$ az $\mathcal{L}$ nyelvosztály komplementere.

Tétel:

a) $\text{co}(\text{co}\mathcal{L}) = \mathcal{L}$

b) Ha $\mathcal{A} \subseteq \mathcal{B}$ nyelvosztályok, akkor $\text{co}\mathcal{A} \subseteq \text{co}\mathcal{B}$.

Biz:

a) $(L^C)^C = L$ alapján.

b) $L \in \text{co}\mathcal{A} \Rightarrow L^C \in \mathcal{A} \Rightarrow L^C \in \mathcal{B} \Rightarrow L \in \text{co}\mathcal{B}$

Tár-idő tételek komplementek nyelvosztályokra

Tétel: $\text{TIME}(t(n)) = \text{coTIME}(t(n))$, azaz

$L \in \text{TIME}(t(n))$ akkor és csak akkor, ha $L^c \in \text{TIME}(t(n))$.

Biz: Az L -et eldöntő M TG-ben az elfogadó és elutasító állapotokat felcserélve, egy azonos futásidejű, L^c -t eldöntő TG-et kapunk, tehát $L^c \in \text{TIME}(t(n))$, azaz $\text{coTIME}(t(n)) \subseteq \text{TIME}(t(n))$.

Másrészt $\text{TIME}(t(n)) = \text{co}(\text{coTIME}(t(n))) \subseteq \text{coTIME}(t(n))$.

Tehát $\text{TIME}(t(n)) = \text{coTIME}(t(n))$.

Tár-idő tételek komplement nyelvosztályokra

Tétel: $\text{SPACE}(s(n)) = \text{coSPACE}(s(n))$, azaz

$L \in \text{SPACE}(s(n))$ akkor és csak akkor, ha $L^c \in \text{SPACE}(s(n))$.

Biz: Legyen $L \in \text{SPACE}(s(n))$. A tár-idő tételbeli N gép minden inputra megáll, $O(s(n))$ tárkorlátos.

Ebben felcserélve az elfogadó és elutasító állapotokat, adódik

hogy $L^c \in \text{SPACE}(s(n))$, azaz

$\text{coSPACE}(s(n)) \subseteq \text{SPACE}(s(n))$.

A másik irány mint előbb.

Nevezetes nyelvosztályok

Definíció: **EXPTIME** = $\bigcup_{k \geq 1} \text{TIME}(2^{n^k})$ az exponenciális időben felismerhető nyelvek osztálya.

Definíció: **PSPACE** = $\bigcup_{k \geq 1} \text{SPACE}(n^k)$ a polinom tárban felismerhető nyelvek osztálya.

Tétel: **P** $\subseteq$ **PSPACE** $\subseteq$ **EXPTIME**

Biz: Ha M TG $t(n)$ időkorlátos, akkor $t(n)$ tárkorlátos is (lépésenként max. egy jelet ír a munkaszalagra).

Tehát $\text{TIME}(n^k) \subseteq \text{SPACE}(n^k)$, és így

$$\mathbf{P} = \bigcup_{k \geq 1} \text{TIME}(n^k) \subseteq \bigcup_{k \geq 1} \text{SPACE}(n^k) = \mathbf{PSPACE}$$

Nevezetes nyelvosztályok

Ha $L \in \text{PSPACE}$, akkor valamely k -ra $L \in \text{SPACE}(n^k)$

A tár-idő tétel szerint ekkor hogy

$$L \in \text{TIME}(2^{n^k}) \subseteq \text{EXPTIME}.$$

Tétel: $\text{EXPTIME} \subset \mathbf{R}$ (=Rekurzív nyelvek osztálya)

Biz: $\subseteq$ következik EXPTIME időkorlátosságából.

A valódi tartalmazáshoz mutatunk egy L nyelvet, mely rekurzív, de
nem **EXPTIME**–beli,

Nevezetes nyelvosztályok

$L = \{ w \in \{0, 1\}^*, \text{ az } M_w \text{ gép létezik, és legfeljebb } 2^{|w|} \text{ lépésben elutasítja } w\text{-t, ahol } |w| \text{ a szó hossza.} \}$

1. L -nek végtelen sok eleme van, mivel egy M_w -t kibővíthetünk a kiinduló helyzetből soha nem elérhető állapotokkal. Egy ilyen y leírás esetén M_y éppen úgy fog működni, mint M_w , tehát az időkorlátja is azonos.
2. Megmutatjuk, hogy L semmilyen n -re nincs benne $\text{TIME}(2^{2^{n-1}})$ -ban. Ez elég is, mivel $\text{EXPTIME} \subseteq \text{TIME}(2^{2^{n-1}})$.

Nevezetes nyelvostályok

Indirekte feltesszük, hogy L felismerhető $c2^{2^{n-1}}$ időkorlátos M TG-pel.

Legyen n_0 olyan szám, hogy $n \geq n_0$ esetén $c2^{2^{n-1}} < 2^{2^n}$

Legyen w egy n_0 -nál hosszabb szó, melyre M_w létezik és úgy viselkedik mint M . Most megmutatjuk, hogy mind $w \in L$, mind a $w \notin L$ feltételezés ellentmondásra vezet.

Ha $w \in L$, akkor M_w elfogadja w -t $c2^{2^{n-1}} < 2^{2^n}$ lépésben, de ekkor L def. szerint $w \notin L$

Ha $w \notin L$, akkor M_w nem utasítja el w -t $c2^{2^{n-1}} < 2^{2^n}$ lépésben, de megáll, ezért elfogadja w -t, amiből $w \in L$ következne.

A digitális számítás elmélete

10. előadás április 25.

- NP nyelvosztály
- NP nyelvhez van polinomiális időben ellenőrizhető tanú
- NP-teljes problémák

Nemdeterminisztikus TG (ism)

Definíció: Nemdeterminisztikus **Turing gép (ND TG):**

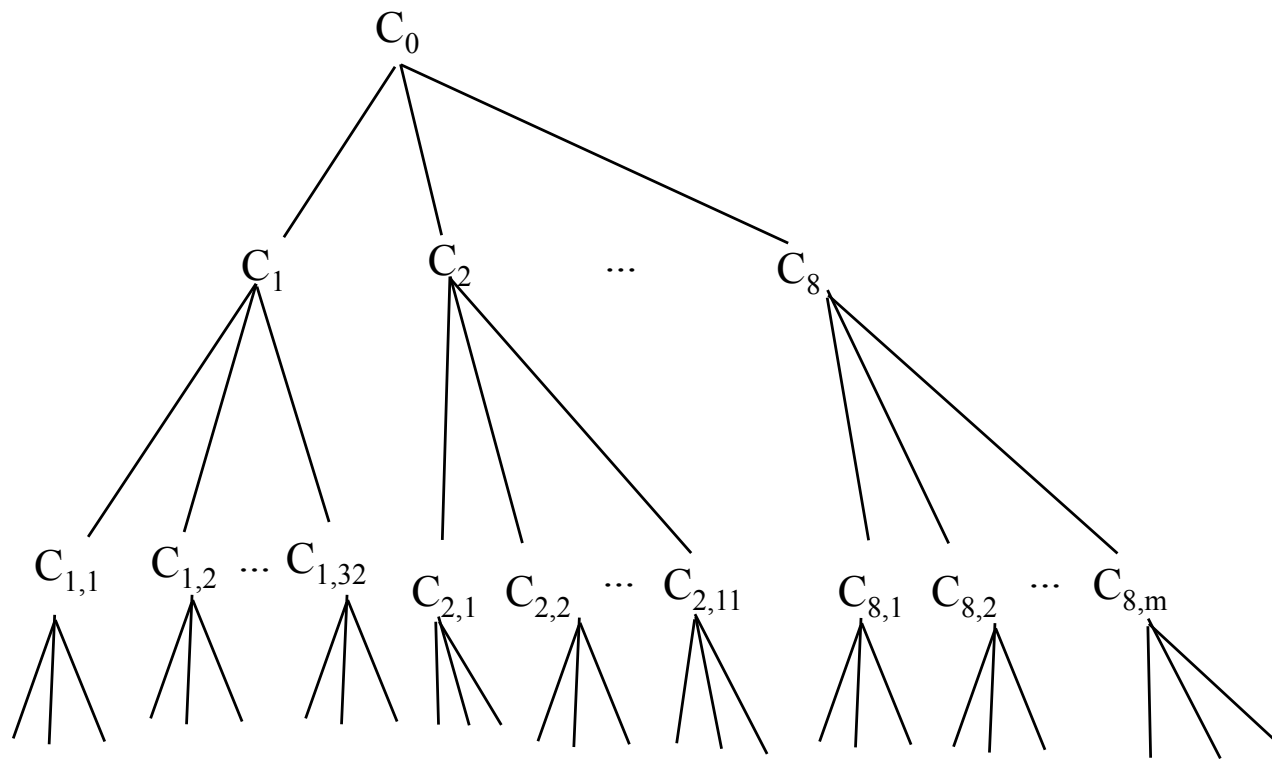
$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{elfogad}}, q_{\text{elutasít}})$ ahol:

1. ...
2. ...
3. ...
4. $\delta: Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{J, B\})$ az **állapotátmenet függvény**
5. ...
6. ...
7. ...

Megj: A ND TG működését fa írja le. Ha egy ágon az elért állapot elfogadó, akkor a ND TG elfogadta a kezdeti sorozatot.

Turing gép variációk: nemdeterminisztikus TG

Tétel: Minden M NDTG-hez van vele ekvivalens S TG. (volt)



Pl: 2,1,3 címke
melyik konfigurációnak felel meg?

NDTG időkorlátja

Definíció: Legyen $t: \mathbb{N} \rightarrow \mathbb{N}$ függvény olyan, hogy $t(n) \geq n$ minden n -re. Az M NDTG **$t(n)$ időkorlátos**, ha az n hosszú bemenetek esetén M minden számítási út mentén legfeljebb $t(n)$ lépést megtéve megáll. Azaz n hosszú bemenetek esetén legfeljebb $t(n) + 1$ mély a számítási fa.

Megj: Nem ismert olyan fizikai megvalósítás, mely $t(n)$ -nel arányos időben szimulálná egy $t(n)$ időkorlátos NDTG működését.

Definíció: $\text{NTIME}(t(n)) = \{L \subseteq I^* \mid L \text{ felismerhető egy } O(t(n)) \text{ időkorlátos } M \text{ NDTG-pel.}\}.$

Az **NP** nyelv osztály

Definíció: **NP** = $\bigcup_{k \geq 1} \text{NTIME}(n^k)$ a polinom időben
egyszalagos NDTG-pel eldönthető nyelvek osztálya.

1. Megj: **P** $\subseteq$ **NP**

mivel egy $t(n)$ időkorlátos TG tekinthető egy speciális, $t(n)$
időkorlátos NDTG-nek, így $\text{TIME}(n^k) \subseteq \text{NTIME}(n^k)$

2. Megj: **P** $\subset$ **NP** vagy **P** = **NP** ?

Nyitott kérdés, 1M \$ díj!

Az **NP** nyelv osztály

Tétel: $\mathbf{P} \subseteq \mathbf{NP} \cap \mathbf{coNP}$

Biz: $\mathbf{P} \subseteq \mathbf{NP}$ láttuk.

De a co operátor tartalmazás megőrző volta miatt ekkor

$$\mathbf{coP} \subseteq \mathbf{coNP}$$

Láttuk hogy $\mathbf{coP} = \mathbf{P}$, tehát $\mathbf{P} \subseteq \mathbf{coNP}$ is teljesül.

NP -be tartozás megmutatása bizonyítékkal

Állítás: $x \in L$

Tanú, bizonyíték: y

A bizonyíték:

- nem hosszú,
- hatékonyan ellenőrizhető!

P1.

Merlin lovagjai és udvarhölgyei - egy párosítás

Hamilton kör van-e a gráfban? – az összes csúcs egy sorozata

G gráf síkbarajzolható-e? – egy síkba lerajzolt gráf

NP –beli nyelvhez tartozás jó bizonyítéka

Egy NP-beli nyelv komplementere nem mindig NP-beli.

Pl:

Merlin lovagjai és udvarhölgyei nem házasíthatók össze

tanú adható: lovagok és hölgyek megfelelő részhalmazai

Nincs Hamilton kör a gráfban

tanú nem adható:

G gráf nem síkba rajzolható

tanu adható: valamelyik Kuratowski-gráf mint G részgráfja

Nem ismert, hogy $\text{co NP} = \text{NP}$ teljesül-e.

NP -be tartozás megmutatása bizonyítékkal

Def: Legyen L egy Σ feletti nyelv. L polinomiálisan bizonyítható, ha van olyan V algoritmus, hogy

$$L = \{w \mid V \text{ elfogadja } \langle w, c \rangle \text{ t valamilyen } \Sigma^*\text{-beli } c\text{-re}\},$$

és V polinom idejű w hosszát tekintve.

Tétel: NP azokat a nyelveket tartalmazza, amelyek polinomiálisan bizonyíthatók.

NP –teljes problémák

Egy L NP-beli nyelv NP-teljes, ha ő a „legnehezebb” az NP-beli nyelvek közül.

1. Megj: ‘Elég’ az NP-teljes problémákat vizsgálni.
2. Megj: Egy NP-teljes problémára nem érdemes polinom idejű algoritmust keresni (mivel az a feltételezés, hogy $P \subset NP$).

Nevezetes **NP** –teljes problémák

Def: A és B Σ^* feletti nyelvek.

A polinom időben visszavezethető B-re, ha van

$f: \Sigma^* \rightarrow \Sigma^*$ polinomiális időben kiszámítható fv, hogy

$w \in A \Leftrightarrow f(w) \in B$.

Ezt $A \leq_p B$ jelöli.

Tétel: Ha $A \leq_p B$ és $B \in P$, akkor $A \in P$.

Biz: Legyen M az a TG, mely polinom időben felismeri B-t.

Ekkor a következő N TG A-t ismeri fel polinom időben:

Nevezetes **NP** –teljes problémák

w input esetén N:

- Kiszámítja $f(w)$ -t,
- M-et futtatja $f(w)$ -n, és outputként M outputját adja.

Def: L **NP-teljes**, ha

- L NP-beli, és
- tetszőleges A NP-beli probléma polinom időben visszavezethető L-re.

Nevezetes **NP** –teljes problémák

Def: Bool változók igaz (1) vagy hamis (0) értéket vehetnek fel.

Def: Bool kifejezések:

és	vagy	nem
$0 \wedge 0 = 0$	$0 \vee 0 = 0$	$\neg 0 = 1$
$0 \wedge 1 = 0$	$0 \vee 1 = 1$	$\neg 1 = 0$
$1 \wedge 0 = 0$	$1 \vee 0 = 1$	
$1 \wedge 1 = 1$	$1 \vee 1 = 1$	

Pl: $\Phi = (\neg x \wedge y) \vee (x \wedge \neg z)$

$\text{SAT} = \{\Phi \mid \Phi \text{ kielégíthető Bool kifejezés}\},$

Nevezetes **NP** –teljes problémák

Tétel: Ha B NP-teljes és $B \in P$, akkor $NP = P$.

Biz: definíciókból köv.

Tétel: Ha B NP-teljes, C NP-beli, és $B \leq_p C$, akkor C is NP-teljes.

Biz: definíciókból köv.

Nevezetes **NP** –teljes problémák

Tétel: (Cook-Levin) $\text{SAT} \in \text{P}$ akkor és csak akkor, ha $\text{NP} = \text{P}$.

Másképp fogalmazva, SAT NP-teljes.

Biz: nem részletezzük

Def: $3\text{SAT} = \{ \Phi \mid$

$$\Phi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \dots \wedge (a_k \vee b_k \vee c_k)$$

valamely k -ra, ahol tetszőleges a_i, b_i, c_i egy Boole-változó vagy annak negáltja

Φ kielégíthető Boole-kifejezés}.

Tétel: 3SAT NP-teljes.

Biz: SAT polinom időben visszavezethető 3SAT-ra. (Nem részletezzük)

Nevezetes **NP** –teljes problémák

Tétel: VERTEX COVER (lefogható-e G minden éle legfeljebb k ponttal) NP-teljes.

Tétel: k -COLORABILITY (kiszínezhető-e G minden éle legfeljebb k színnel) NP-teljes $k \geq 3$ esetén.

Tétel: HAMPATH NP-teljes.

Biz: 3SAT polinom időben visszavezethető HAMPATH-ra.

Savitch tétele

Def. **NPSPACE** = $\cup_{k \geq 1} \text{NSPACE}(n^k)$, ahol

$\text{NSPACE}(s(n)) = \{L \mid L \text{ felismerhető egy } O(s(n)) \text{ tárkorlátos } M \text{ nemdeterminisztikus TG-pel.}\}.$

Tétel. Ha $f(n) \geq O(\log_2(n))$, akkor

$$\text{NSPACE}(f(n)) \subseteq \boxed{\text{D}}\text{SPACE}((f(n))^2).$$

Biz. (Vázlat) Egyrészt, Savitch megmutatta, hogy a $\text{PATH} \in \text{TIME}((\log_2(n))^2)$.

Savitch tétele

Másrészt, legyen N egy $f(n)$ tárigényű nemdet. TG. Az N konfigurációs gráfja egy n hosszú bemeneten $2^{O(f(n))}$ méretű, és ebben a gráfban kell egy determinisztikus TG-nek egy utat keresni a kezdőkonfigurációból egy elfogadó konfigurációba.

Savitch tétele alapján ez elvégezhető $(\log_2(2^{O(f(n))}))^2$ azaz $O((f(n))^2)$ tárral, ami bizonyítja az állítást.

Következmény. PSPACE=NPSPACE

Emlékeztetőül: az a sejtés, hogy $P \subset NP$.

Problémaosztályok tartalmazása

Tétel: $PSPACE \subseteq EXPTIME$

Biz (vázlat):

Legyen $L \in PSPACE$, és N egy L -et $f(n)$ tárral felismerő TG.

Korábban láttuk, hogy N konfigurációs gráfja egy n hosszú

bemeneten $2^{O(f(n))}$ méretű. Ebben a gráfban kell egy M

determinisztikus TG-nek egy elfogadó konfigurációt keresni.

Mivel a PATH probléma P -beli, ez megoldható $2^{O(f(n))}$ időben.

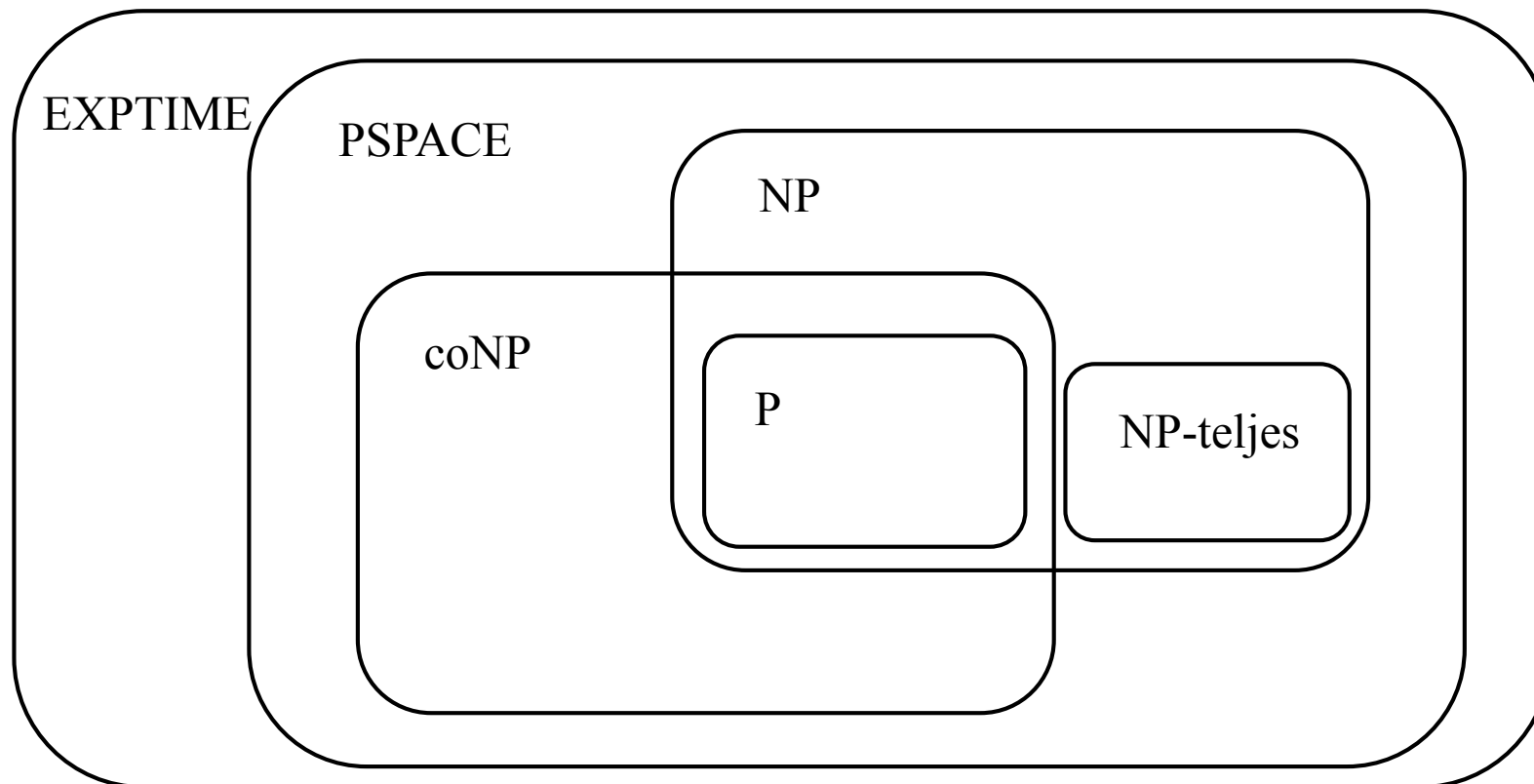
Mivel $f(n)$ egy polinom függvény kapjuk, hogy $L \in EXPTIME$.

Problémaosztályok tartalmazása

Következmény: $P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME$.

Nem ismert, hogy a tartalmazások valódiak-e.

Ismert, hogy $P \subset EXPTIME$. Tehát a fenti láncban legalább az egyik helyen valódi a tartalmazás, sejtés: mindegyik valódi.



PSPACE-teljes problémák

Def. Egy L nyelv PSPACE-teljes, ha PSPACE-beli és minden további PSPACE-beli nyelv polinom időben visszavezethető L -re.

Def. $QBF = \{ \langle \varphi \rangle : \varphi \text{ igaz prenex alakú zárt Boole-formula} \}$

Példa $\exists x_1 \forall x_2 \exists x_3 [(x_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (\bar{x}_2 \vee \bar{x}_3)] \in QBF$
 $\exists x \forall y [(x \vee y) \wedge (\bar{x} \vee \bar{y})] \notin QBF$

Tétel QBF PSPACE teljes.

PSPACE-teljes problémák

Megjegyzés

- QBF akkor is PSPACE-teljes, ha a formula magja knf, a kvantorok alternálnak, és az első kvantor $\exists$ (esetleg az utolsó is $\exists$).
- Ekkor QBF felfogható kétszemélyes játékként.

Adott $\varphi = \exists x_1 \forall x_2 \exists x_3 \dots Q x_n \psi$

- ◆ az 1. játékos választja az $x_1, x_3, \dots$ változók értékét és célja ψ igazzá tétele,
- ◆ a 2. játékos választja felváltva az $x_2, x_4, \dots$ értékét, célja ψ hamissá tétele.

$\psi \in \text{QBF} \Leftrightarrow$ az 1. játékosnak nyerőstratégiája van.

PSPACE-teljes problémák

FÖLDRAJZI JÁTÉK:

Adott egy G irányított gráf egy p kezdőcsúcsból. Két játékos p -ből indulva felváltva egy olyan út csúcsait nevezi meg, mely nem halad át már meglátogatott csúcson. Az a játékos veszít, aki nem tudja folytatni. A probléma annak eldöntése, hogy az első játékosnak van-e nyerő stratégiája.

Tétel A FÖLDRAJZI JÁTÉK PSPACE-teljes.

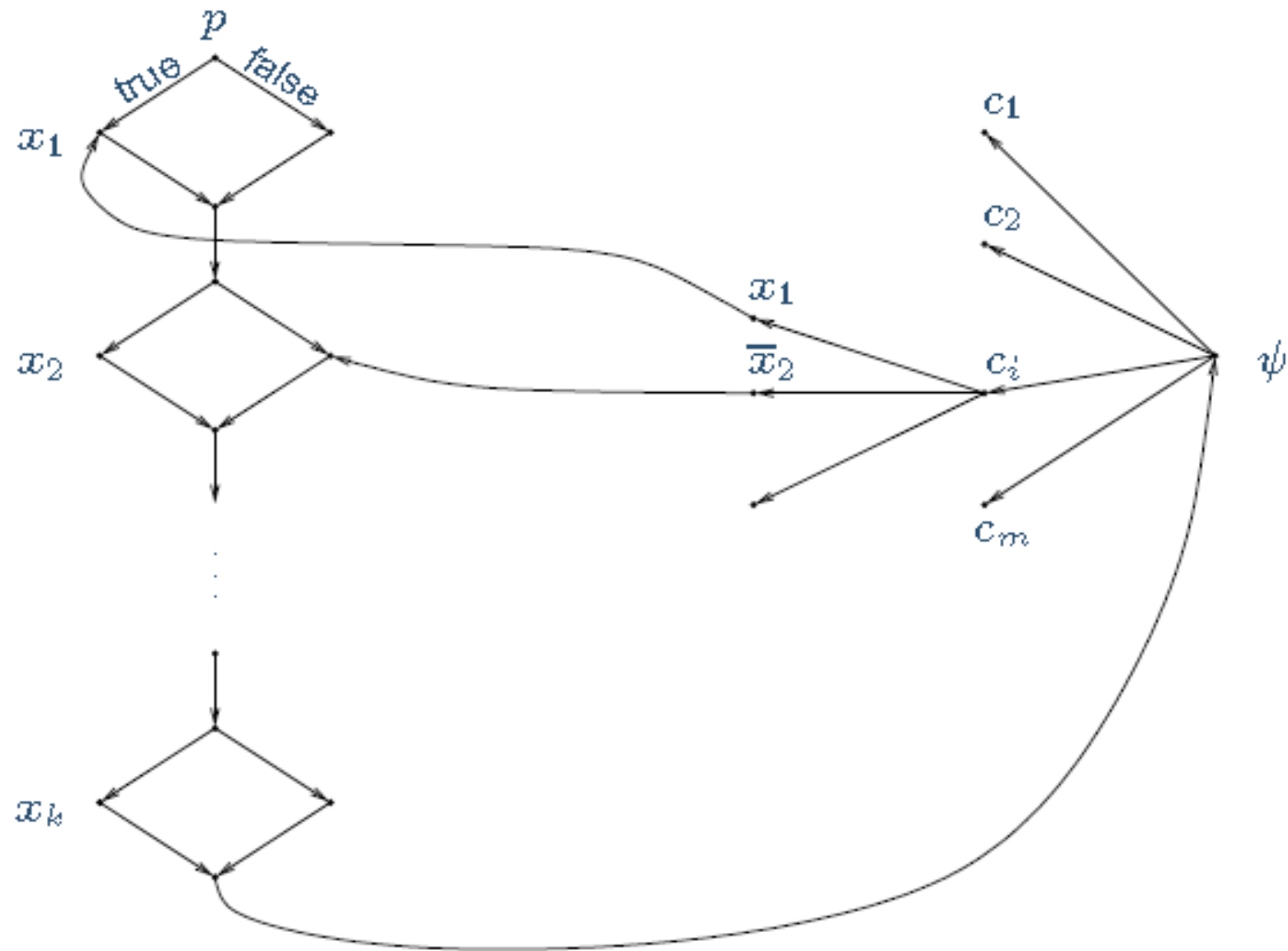
Bizonyítás Az, hogy PSPACE-ben van, a QBF-hez hasonlóan igazolható. Azt, hogy PSPACE-nehéz, a QBF-ből való visszavezetéssel igazoljuk.

Legyen adva a φ formula

$$\varphi = \exists x_1 \forall x_2 \exists x_3 \dots \exists x_k \psi \quad k \text{ páratlan}$$

$$\psi = c_1 \wedge \dots \wedge c_m \quad \text{knf.}$$

PSPACE-teljes problémák



Irodalom

10. előadás dec.5.

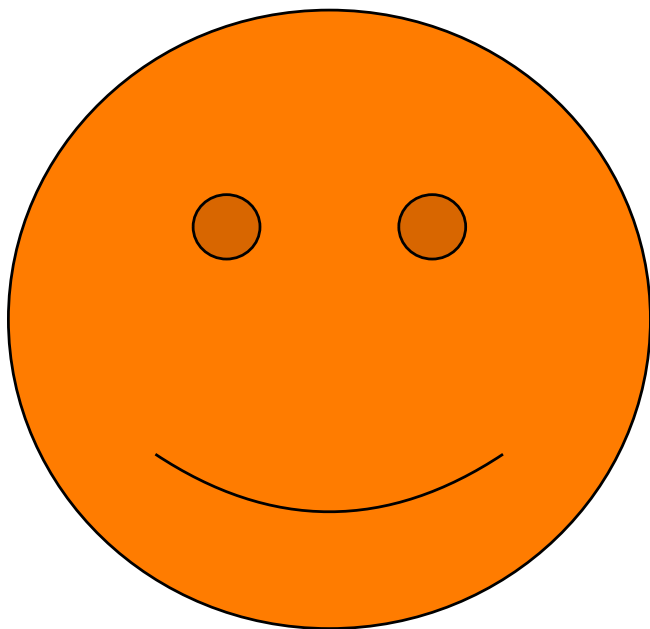
Rónyai-Ivanyos-Szabó: Algoritmusok, 255-265, 268-279. oldalak

A digitális számítás elmélete

11. előadás május 3.

- Információtartalom algoritmikus alapokon
- Kolmogorov bonyolultság
- Kolmogorov véletlen sorozatok

Információtartalom – intuitív példák



Információtartalom – intuitív példák

Melyik sorozatnak a legtöbb ill. legkevesebb az információtartalma?

- 1111111111111111
- 1001011110110101
- 1010101010101010

Információtartalom – intuitív példák

Adjuk meg röviden az alábbi számokat:

- 8192 2^{13} 13=1101₂
+..

- 832040 F(30) F(n) = F(n-1) + F(n-2)

$$\frac{\Pi}{2} = \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdot \dots \cdot \frac{2n}{2n-1} \cdot \frac{2n}{2n+1} \dots$$

- 1415926535 8979323846

$$\frac{\Pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots + \frac{(-1)^n}{2n+1} + \dots$$

Információtartalom – intuitív példák

Rövid megadás

- természetes nyelven: értelmezési probléma, pl:
 - max. 100 írógép karakterrel megadható számok
 - ezen legnagyobbikának rákövetkezője?
- binárisan: jó, hossz azonos a bin. szám hosszával
- **Turing gépet használva**

Összenyomhatóság

Def: $I = \{0, 1\}$, M egy olyan TG, melynek bemeneti és szalag abc-je I .

M az $f_M: I^* \rightarrow I^*$ parciális függvényt számolja ki.

Ekkor egy I^* -beli x szó esetén

$$C_M(x) = \begin{cases} \min \{ |y| : y \in I^*, f_M(y) = x \} & \text{ha van ilyen } y \\ \infty & \text{különben} \end{cases}$$

$C_M(x)$ azt adja meg, hogy M milyen mértékben nyomja össze x -et.

Megj: $C_M(x)$ függ M -től! $C_{M_1}(x) = \infty$ $C_{M_2}(x) = 0$ alkalmas M_i -re.

Összenyomhatóság

Tétel: (Invariancia tétel)

Legyen U egy univerzális TG. Ekkor tetszőleges M TG-hez létezik $c_M \in \mathbb{Z}^+$ szám, hogy minden I^* -beli x szó esetén

$$C_U(x) \leq C_M(x) + c_M.$$

Biz: Legyen y egy legrövidebb szó, melyre $f_M(y) = x$.

Legyen M kódja $w \in I^*$.

Ekkor U a $w\#y$ bemenettel x -et adja meg, azaz $f_U(w\#y) = x$.

$$\text{Tehát } C_U(x) \leq |w\#y| = |w\#| + |y| = |w\#| + C_M(x) = \underbrace{c_M}_{\text{}} + C_M(x)$$

Összenyomhatóság

Megj: (Invariancia tétel következménye)

Legyen U_1 és U_2 univerzális TG-k. Ekkor létezik olyan

$c = c_{U_1, U_2} \in \mathbb{Z}^+$ szám, hogy minden I^* -beli x szó esetén

$$| C_{U_1}(x) - C_{U_2}(x) | \leq c.$$

Biz: Az invariancia tételt $U = U_1$, $M = U_2$ és $U = U_2$, $M = U_1$

szereposztással felírva

$$C_{U_1}(x) - C_{U_2}(x) \leq c_{U_2} \text{ és } C_{U_2}(x) - C_{U_1}(x) \leq c_{U_1}$$

$$\text{Tehát } | C_{U_1}(x) - C_{U_2}(x) | \leq c = \max \{ c_{U_1}, c_{U_2} \}.$$

Kolmogorov bonyolultság

Megj: Ha U univerzális TG-k, akkor $C_M(x)$
nem függ **lényegesen** U -tól (max konstans erejéig).

A továbbiakban rögzítünk egy U univ. TG-et.

Def: Egy I^* -beli x szó esetén a szó **Kolmogorov-bonyolultsága**
 $C(x)$, ahol $C(x) = C_U(x)$.

1. Megj: $C(x) \leq |x| + k$ alkalmas, x -től független k számra.

Az invariancia tételből következik, olyan M -et választva,
mely rögtön megáll.

2. Megj: $C(\text{bin}(n)) \leq \log_2 n + c$ alkalmas c -re, ahol $\text{bin}(n)$ az n szám
bináris alakját jelöli

Kolmogorov bonyolultság

Def: Egy I^* -beli x szó **összenyomhatatlan**, ha

$$C(x) \geq |x|.$$

Tétel: Legyen $k \in \mathbb{Z}^+$ szám. Ekkor

- 1) Legfeljebb $2^{k+1} - 1$ olyan x szó van, melyre $C(x) \leq k$.
- 2) Minden n természetes számra van n hosszú, összenyomhatatlan szó.

Biz: 1) Legyen $H_k = \{x \in I^* , C(x) \leq k \}$. H_k -nak legfeljebb annyi eleme van, mint a max. k hosszú I^* -beli szorozatok száma, azaz

$$1 + 2 + 2^2 + 2^3 + \dots + 2^k = 2^{k+1} - 1$$

Kolmogorov bonyolultság

Biz: 2) Az n hosszú szavak száma 2^n .

Legyen $k = n - 1$.

Ekkor 1) szerint legfeljebb $2^{k+1} - 1 = 2^n - 1$ olyan x szó van, melyre $C(x) \leq n - 1$.

Tehát van olyan szó van, mely nincs H_{n-1} -ben, azaz

$C(x) > n - 1$, tehát $C(x) \geq |x| = n$.

Azaz x összenyomhatatlan.

1.Megj: Ha $n > 8$, akkor az n hosszú szavak több mint 99 %-ának Kolmogorov bonyolultsága nagyobb mint $n-8$.

Ui a nem ilyen sorozatok H_{n-8} -beliek, ezek száma legfeljebb $2^{n-7} - 1$, melyek aránya kevesebb mint $2^{n-7} / 2^n = 2^{-7} = 1/128 < 1/100$.

Kolmogorov bonyolultság

Tétel: A Kolmogorov bonyolultságot megadó $C: I^* \rightarrow Z^+$ függvény nem rekurzív.

Biz: Legyen $F: Z^+ \rightarrow I^*$ függvény, hogy $F(m) = x$, ahol x a lexikografikus rendezés szerinti első eleme a $\{z \in I^* \mid C(z) \geq m\}$ halmaznak.

Tegyük fel indirekten, hogy C rekurzív.

Ekkor van olyan N TG, mely kiszámítja $C(z)$ -t.

Kolmogorov bonyolultság

Ekkor legyen M az a TG, mely tetszőleges $\text{bin}(m)$ bemenet esetén:

- lexikografikus sorrendben veszi I^* elemeit,
- N -nel a kiválasztott z -re kiszámítja $C(z)$ -t,
- ha $C(z) \geq m$, megáll és kiírja z -t.

Kolmogorov bonyolultság

Ekkor $m \leq C(F(m))$ F megadása miatt.

De az invariancia tétel szerint van olyan c , hogy

$$C(F(m)) \leq C_M(F(m)) + c$$

Másrészt M definíciója szerint igaz, hogy

$$C_M(F(m)) \leq \log_2 m + 1$$

De az is igaz lesz, hogy

$$m \leq \log_2 m + c + 1$$

Ami viszont nem teljesül elég nagy m -től kezdődően.

Tehát az indirekt feltevés hamis volt, az állítás igaz.

Kolmogorov bonyolultság és a véletlen

Jelölés: I^∞ jelöli az összes végtelen 0-1 sorozatot. Egy $x \in I^\infty$ esetén x_n az első n elemből álló szót jelöli.

Def: Egy $x \in I^\infty$ sorozat **Kolmogorov-véletlen sorozat**, ha

$\lim_{n \rightarrow \infty} C(x_n) / n = 1$, azaz a sorozat kezdőszeletei „aszimptotikusan” majdnem teljesen összenyomhatatlanok.

Tétel: Ha $x \in I^\infty$ egy Kolmogorov-véletlen sorozat, akkor nincs M TG, mely n (bináris) bemenetre x_n -t állítja elő.

Biz: Indirekte, tegyük fel hogy van ilyen M TG. Ekkor $C_M(x_n) \leq \log_2 n + 1$, és az invariancia tétel szerint $C(x_n) \leq \log_2 n + 1 + c_M$:

De ekkor $\lim_{n \rightarrow \infty} C(x_n) / n \leq \lim_{n \rightarrow \infty} (\log_2 n + 1 + c_M) / n = 0$, ellentmondás!

Irodalom

11. előadás május. 7.

Rónyai-Ivanyos-Szabó: Algoritmusok, 229-238 oldalak