Számítási modellek

8: Eldönthetőség, Felismerhetőség

Objektumok kódolása sztringként

- Legyen *O* egy objektum (pl. automata, TG, polinom, gráf, stb.). Ekkor az *O* kódolását egy sztringként <*O*>-val jelöljük.
- Legyen $O_1, O_2,\ldots,O_k$ objektumok egy listája. Ekkor ezek kódolását egyetlen sztringbe $<O_1, O_2,\ldots,O_k>$-val jelöljük.

- Turing gépek leírása:
- Az algoritmusok magyar nyelvű leírását fogjuk használni, amikor TG-eket írunk le, tudva azt, hogy ezeket a leírásokat (elvileg) konvertálni tudjuk állapotokká, átmenetekké, stb.
- M = "w inputtal:

    [Az algoritmus magyar nyelvű leírása]"

Példa

- TG *M*, amely felismeri $L = \{a^k b^k c^k : k \geq 0\}$ -t.
- *M* = "*w* inputtal:

    1) Ellenőrizzük, hogy $w \in a^*b^*c^*$ teljesül-e. Ha nem, akkor utasítsuk el.

    2) Számoljuk meg az *a*-kat, *b*-ket és *c*-ket *w*-ben.

    3) Ha a számuk egyenlő, akkor fogadjuk el, egyébként, utasítsuk el."

- High-level leírás rendben van.
- Nem kell szalagokat, állapotokat, stb. Kezelni.

TG-k kódolása

- Feltesszük, hogy $\Sigma = \{0,1\}$.
- Egy M TG **kódja** (jelölés: $<M>$) a következő:
- Legyen $M = (Q, \{0,1\}, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, ahol
    - $Q = \{p_1,\ldots,p_k\}$, $\Gamma = \{X_1,\ldots,X_m\}$, $D_1 = R$, $D_2 = S$, $D_3 = L$,
    - $k \geq 3$, $p_1 = q_0$, $p_{k-1} = q_{accept}$, $p_k = q_{reject}$,
    - $m \geq 3$, $X_1 = 0$, $X_2 = 1$, $X_3 =$ _.
    - A $\delta(p_i, X_j) = (p_r, X_s, D_t)$ átmet elkódolva: $0^i10^j10^r10^s10^t$.
    - $<M>$: elkódolt átmenetek listája, amelyben az átmenet kódokat 11-gyel választjuk el.
- Megjegyzés: $<M>$ 0-val kezdődik és 0-val végződik és nem tartalmazza az 111-t részsztringként.
- $<M,w> := <M>111w$

Elfogadási probléma determinisztikus véges automatákra (DVA)

- Legyen A_{DVA} = {<*B*,*w*> | *B* egy DVA és *B* elfogadja *w*-t}.

Tétel: A_{DVA} eldönthető.

Biz.: Adunk egy $M_{A\text{-}DVA}$ TG-t, ami elfogadja A_{DVA}- t.

- $M_{A\text{-}DVA}$ = "*s* inputtal:
  - Ellenőrizzük, hogy az s input <*B*,*w*> alakú-e, ahol *B* egy DVA és *w* egy sztring. Ha nem, akkor utasítsuk el.
  - Szimuláljuk a *B* számítását *w*-vel.
  - Ha egy elfogadó állapotban fejeződik be, akkor fogadjuk el. Ha nem, akkor utasítsuk el."

Elfogadási probléma nemdeterminisztikus VA-ra (NVA)

- Legyen A_{NVA} = {<*B*,*w*> | *B* egy NVA és *B* elfogadja *w*-t}.

Tétel: A_{DVA} eldönthető.

Biz.: Adunk egy $M_{A\text{-}NVA}$ TG-t, ami eldönti A_{NFA}-t.

- $M_{A\text{-}NVA}$ = "<*B*,*w*> inputtal:
  - Alakítsuk át a *B* NVA-t ekvivalens *B'* DVA-ra.
  - Indítsuk $M_{A\text{-}DFA}$ TG-t <*B'*,*w*> inputtal. [$M_{A\text{-}DVA}$ eldönti A_{DVA}-t]
  - Fogadjuk el, ha $M_{A\text{-}DVA}$ elfogadja.
  - Utasítsuk el, ha nem."

- Új elem: Használjuk az átalakítást és a korábban elkészített TG-t szubrutinként.

Üresség (emptiness) probléma DVA-ra

- Legyen $E_{DVA} = \{<B> \mid B$ egy DVA és $L(B) = \emptyset\}$.

Tétel: E_{DVA} eldönthető.

Biz.: Adunk egy $M_{E\text{-}DVA}$ TG-t, ami eldönti E_{DVA}-t.

- $M_{E\text{-}DVA}$ = “$<B>$ inputra:
  [Ötlet: Keressünk egy utat a kezdőállapotból egy elfogadó állapotba.]
  - Jelöljük meg a kezdőállapotot.
  - Ismételjük, amíg nem tudtunk új állapotot megjelölni:
    - Jelöljünk meg minden állapotot, amelybe már korábban megjelölt állapotból vezet él (átmenet).
  - Fogadjuk el, ha nincs megjelölt elfogadó állapot.
  - Utasítsuk el, ha valamely elfogadó állapot meg van jelölve.”

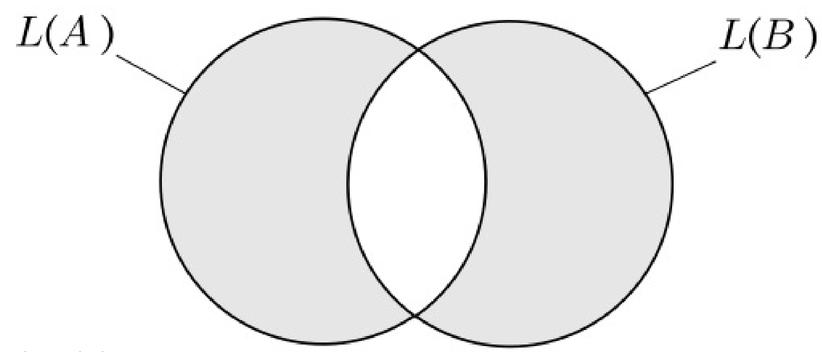
Ekvivalencia probléma DVA-ra

- Legyen $EQ_{\text{DVA}} = \{<A,B> \mid A,B$ DVA-k és $L(A) = L(B)\}$.

Tétel EQ_{DVA} eldönthető.

Biz.: Adunk egy $M_{\text{EQ-DVA}}$ TG-t, ami eldönti EQ_{DVA}-t.

- $M_{\text{EQ-DVA}}$ = "$<A,B>$ inputtal:
  [Ötlet: Készítsünk egy C DVA-t, ami azokat a w szavakat fogadja el, ahol A és B különböznek.]
  - Készítsünk egy C DVA-t, melyre
    $$L(C) = (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B))$$
  - Indítsuk $M_{\text{E-DVA}}$-t C inputtal.
  - Fogadjuk el, ha $M_{\text{E-DVA}}$ elfogad.
  - Utasítsuk el, ha $M_{\text{E-DVA}}$ elutasít."



Elfogadási probléma környezetfüggetlen grammatikákra (CFG)

- Legyen A_{CFG} = {<*G*,*w*> | *G* egy CFG és *G* generálja *w*-t}.

Tétel: A_{CFG} eldönthető.

Biz.: Adunk egy $M_{\text{A-CFG}}$ TG-t, ami eldönti A_{CFG}-t.

- $M_{\text{A-CFG}}$ = "<*G*,*w*> inputra:
  - Alakítsuk át Chomsky normálformára.
  - Próbáljuk ki az összes max(1, 2|*w*| - 1) lépésből álló levezetést.
  - Fogadjuk el, ha valamelyik *w*-t generálja.
  - Utasítsuk el, ha nem.

- Chomsky normálforma (CNF) csak alábbi alakú szabályok vannak:
  - $A \to BC$
  - $B \to b$
  - $S \to \varepsilon$
- Lemma 1: Minden CFG átalakítható CNF-re.
- Lemma 2: Ha *G* CNF-ben van és $w \in L(G)$, akkor *w* minden levezetése max(1, 2|*w*| - 1) lépésből áll.

Üresség probléma CFG-re

- Legyen E_{CFG} = {<G> | G egy CFG és $L(G) = \emptyset$}.

Tétel: E_{CFG} eldönthető.

Biz.: Adunk egy $M_{E\text{-}CFG}$ TG-t, amely eldönti E_{CFG}-t.

- $M_{E\text{-}CFG}$ = “<G> inputtal:
  [Ötlet: dolgozzunk a terminálisokból indulva visszafelé]
    - Jelöljük meg a terminálisok összes előfordulását G-ben.
    - Ismételjük, amíg nem tudunk új szimbólumot megjelölni
        - Jelöljük meg az A nemterminális összes előfordulását, ha $A \to B_1B_2...B_k$ egy szabály és már minden B_i meg van jelölve.
    - Utasítsuk el, ha a kezdőszimbólum meg van jelölve.
    - Fogadjuk el, ha nincs.”

Ekvivalencia probléma CFG-re

- Legyen $EQ_{\mathrm{CFG}} = \{<G,H> \mid A,B$ CFG-k és $L(G) = L(H)\}$.

Tétel: EQ_{CFG} nem eldönthető.

Megjegyzés: CF nyelvek nem zártak a komplementer és a metszet műveletekre.

Biz.: l. Sipser, 5.1. exercise

Nem Turing-felismerhető nyelvek létezése

- Minden $i \geq 1$-re, legyen w_i az i-edik eleme a $\{0,1\}^*$ halmaznak, amely rendezett hossz szerint és lexikografikusan, azaz $\{\varepsilon,0,1,00,01,10,11,000,001,\ldots\}$.
- Jelölje M_i azt a TG-t aminek a kódja w_i (l. 4. fólia: TG-k kódolása. Ha w_i nem a kódja semmilyen TG-nek, akkor M_i egy tetszőleges TG, ami semmit sem fogad el).

Tétel: Létezik olyan nyelv, ami nem Turing-felismerhető.

Biz.:

- Egy TG egy nyelvet ismer fel.
- A TG-k száma megszámlálhatóan végtelen (a TG-k kódolása egy injekció $\{0,1\}^*$-be, amelynek számossága megszámlálhatóan végtelen).
- A nyelvek száma $\{0,1\}$ felett, azaz $|\{L \subseteq \{0,1\}^*\}|$ nem megszámlálhatóan végtelen (continuum kardinalitású). □

Egy nem Turing-felismerhető nyelv: L_d

Tétel: Legyen $L_d = \{w_i : w_i \notin L(M_i)\}$ (L_d a diagonális nyelv).
L_d nem Turing-felismerhető, azaz $L_d \notin RE$.

Biz.: Georg Cantor **diagonalizálási** módszere.

- Tekintsük a T bit táblát, amelyre $T(i,j) = 1 \Leftrightarrow w_j \in L(M_i)$ ($i,j \geq 1$).
- Legyen z a végtelen hosszúságú bitsorozat T átlójában és $\bar{z}$ a bitenkénti komplementere z-nek.
- Minden $i \geq 1$-re, T az i-edik sora az $L(M_i)$ nyelv karakterisztikus vektora.
- $\bar{z}$ az L_d nyelv karakterisztikus vektora.
- Ha L_d felismerhető lenne egy D TG-vel, akkor D karakterisztikus vektora egy sor lenne T-ben.
- $\bar{z}$ különbözik T minden sorától, így L_d különbözik minden RE beli nyelvtől. $\square$

T	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\dots$	$\langle D \rangle$	$\dots$
M_1	$\underline{1}$	0	1		1	
M_2	0	$\underline{1}$	1	$\dots$	0	$\dots$
M_3	1	0	$\underline{0}$		1	
$\vdots$		$\vdots$		$\ddots$		
D	1	0	1		$\underline{?}$	
$\vdots$		$\vdots$				$\ddots$

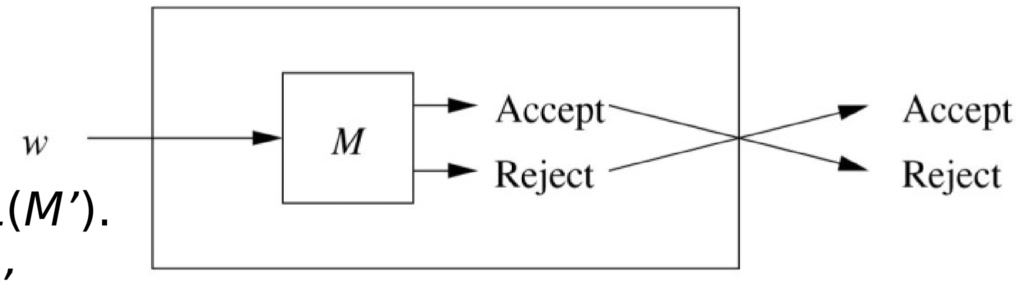
$\bar{z} = 001\dots$

Rekurzív nyelvek *R*

- Egy L nyelv **rekurzív** ($L \in R$), ha $L = L(M)$ valamely M **eldöntő** TG-re.

Tétel: Ha $L \in R$, akkor $\overline{L} \in R$.

Biz.:

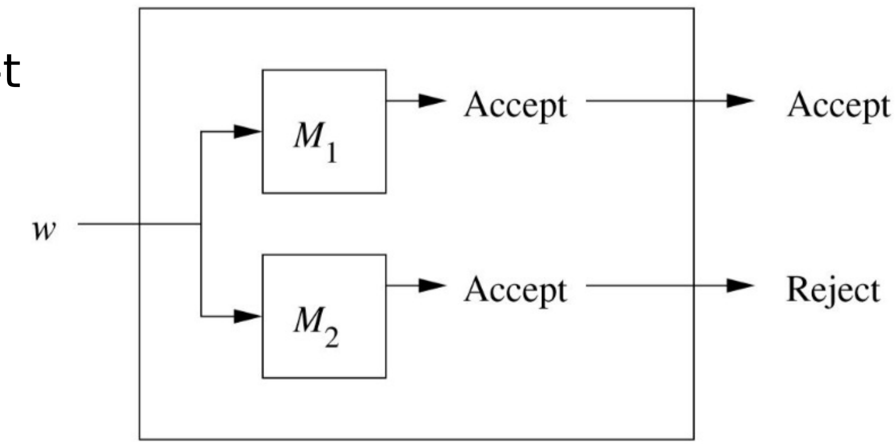


- Legyen $L = L(M)$ valamely M TG-re, amely megáll minden inputtal. Konstruálunk egy M' TG-t, ú.h. $\overline{L} = L(M')$.
  - M elfogadó állapotai lesznek M' elutasító állapotai.
  - M'-nek lesz egy új elfogadó állapota: r (r-ből nincs további átmenet).
  - M minden q nem elfogadó állapotára és a input szimbólumára, ahol nincs átmenet (q,a)-ra M-ben (M elfogadás nélkül megáll), adjunk (q,a)-hoz átmenetet az r állapotba.
- Mivel M megáll inden input szóval, M' is megáll minden input szóval.
- M' pontosan akkor fogad el egy w szót, ha M nem fogadja el w-t. Így az M' által elfogadott nyelv $\overline{L}$. $\square$

Rekurzívan felsorolható nyelvek (*RE*) komplementere

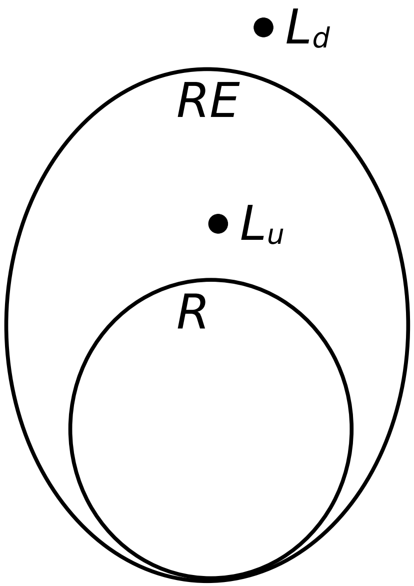
Tétel: Ha $L \in RE$ és $\overline{L} \in RE$, akkor $L \in R$ (és $\overline{L} \in R$).

Biz.:



- Legyen $L = L(M_1)$ és $\overline{L} = L(M_2)$. M_1-t és M_2-t párhuzamosan szimuláljuk egy M TG-vel.
- Legyen M egy 2-szalagos TG.
    - M az 1. szalagon szimulálja M_1-t,
    - M a 2. szalagon szimulálja M_2-t.
    - M állapotai megfelelnek Q_1 x Q_2 állapotpároknak (M_1 és M_2 állapotai)
- Ha M inputja $w \in L$, akkor M_1 elfogadja w-t és megáll. Ekkor M elfogadja w-t és megáll.
- Ha M inputja $w \in \overline{L}$, akkor M_2 elfogadja w-t és megáll. Ekkor M elutasítja w-t és megáll. Így M minden inputtal megáll és $L(M) = L$. $\square$

R és *RE*



- Univerzális nyelv: $L_u = \{<M,w> \mid M$ egy TG és $w \in L(M)\}$.

Tétel: $L_u \in RE \setminus R$.

Biz.:

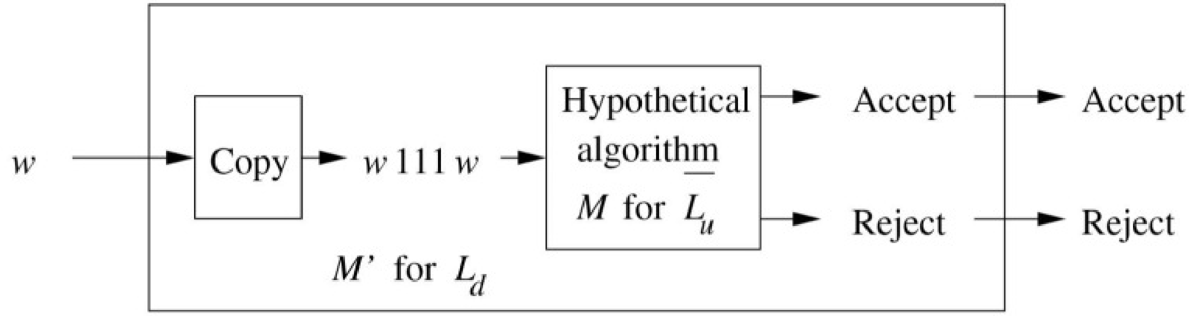
- L_u rekurzívan felsorolható (Turing-felismerhető):
- Konstruálunk egy U TG-t, ami felismeri L_u-t.
  U-t az univerzális TG-nek nevezzük.
- Legyen U egy többszalagos TG, ú.h.
  - 1. szalag tartalmazza az inputot, ami M és w kódolása.
    M és w ezen az előadáson bemutatott kódolását használjuk.
  - 2. szalagot M szalagjának szimulálására használjuk.
    A 2. szalagot w-vel inicializáljuk.
    A 2. szalagon a fejet az első szimulált cellára visszük.
  - 3. szalagot M állapotainak tárolásához használjuk.
    A 3. szalagot M kezdőállapotával inicializáljuk.
  - 4. szalagot munkaszalagként használjuk.

R és *RE*

Biz. (folyt.):

- *M* egy átmenetének szimulálása:
  - *U* megkeresi az 1. szalagon az átmenetet *M* aktuális állapotára (a 3. szalagon tárolva) és *M* aktuális szalagszimbólumára (a 2. szalagon tárolva).
  - Ezután *U* eltárolja az új állapotot a 3. szalagon, *U* megváltoztatja a szalag szimbólumot a 2. szalagon, *U* balra vagy jobbra lépteti *M* író/olvasó fejét a 2. szalagon az átmenet szerint.
  - Ha *M* végállapotába lép, jelezve, hogy *M* elfogadja *w*-t, akkor *U* elfogadja a $<M,w>$-t és megáll.
- Így, $L(U) = L_u$.
- $\Rightarrow L_u \in RE$

R és *RE*



Biz. (folyt.):

- L_u nem rekurzív:
- TF. ellentmondáshoz, hogy L_u rekurzív. Ekkor L_u komplementere $\overline{L_u}$ is az.
- Akkor létezni kell egy M TG-nek, amely eldönti $\overline{L_u}$-t.
- Ekkor át tudjuk alakítani M-et egy M' TG-re, amely elfogadja L_d-t:
    - M' a w inputját $<w,w>$ párrá alakítja.
    - M' szimulálja M-et $<w,w>$ inputtal, feltételezve, hogy az első w egy M_i TG kódolása, a második w pedig egy w_i bináris sztring kódolása.
    - M eldönti $\overline{L_u}$-t, így M pontosan akkor fogadja el $<w,w>$-t ha M_i nem fogadja el w_i-t.
- Így M' pontosan akkor fogadja el w-t ha $w \in L_d$.
  De korábban megmutattuk, hogy nem létezik olyan TG, amely felismeri L_d-t. Következésképpen M nem létezik.
- $\Rightarrow L_u \notin R$. $\square$

Megállási probléma

- Turing eredeti munkájában a TG-pel történő elfogadást / számítást megállással, nem feltétlenül végállapotban való megállással fogalmazta meg.
- Egy M TG-hez legyen $H(M)$ azon w input szavak halmaza, amelyekkel M megáll (végállapotban vagy nem végállapotban).
- A **megállási probléma**:
  $HALT = \{<M,w> \mid w \in H(M)\}$.
- **Tétel**: $HALT \in RE \setminus R$.
  Biz.: L_u-hoz hasonlóan.
- Hasonló érvek használhatók annak megmutatására, hogy szoftver verifikációval kapcsolatos számos gyakorlati probléma nem eldönthető. Például nem eldönthető, hogy egy program végtelen ciklusba kerül-e.

R és $\mathcal{L}_1$

- Egy **lineárisan korlátolt automata (LKA)** egy nemdeterminisztikus TG, melynél
  - az input ábécé Σ tartalmaz két speciális szimbólumot $\triangleright$ (bal végjel/endmarker) és $\triangleleft$ (jobb végjel/endmarker).

$\triangleright$	w_1	w_2	$\ldots$					w_n	$\triangleleft$

  - Az input $\triangleright(\Sigma \setminus \{\triangleright,\triangleleft\})^*\triangleleft$ alakú.
  - $\triangleright$ és $\triangleleft$ nem írhatók felül.
  - A fej nem lehet sem $\triangleright$-tól balra, sem $\triangleleft$-tól jobbra.
  - A fej kezdőpozíciója a jobb szomszédja annak a cellának, ami $\triangleright$-t tartalmazza.
- Egy LKA egy olyan NTG, amely korlátos munkaterülettel rendelkezik.
- Nevét egy vele ekvivalens modellről kapta, amelyben a rendelkezésre álló tár az input hosszának konstansszorosa (lineáris függvénye).

R és $\mathcal{L}_1$

Tétel:

- (1) Minden *G* 1-es típusú grammatikához megadható egy LKA *A*, melyre *L*(*A*) = *L*(*G*).
- (2) Minden *A* LKA-hoz megadható egy *G* 1-es típusú grammatika, melyre *L*(*G*) = *L*(*A*).

Biz.:

- (1) Az előző előadáson láttuk, hogy minden *G* 0. típusú grammatikához lehet konstruálni egy NTG-t, ami *L*(*G*)-t felismeri.
- A konstrukció a 3. szalagján nemdeterminisztikusan szimulált egy *G*-beli levezetést, az iterációk végén ellenőrizte, hogy a 3. szalagon lévő mondatforma megegyezik-e az 1. szalagon lévő *w* inputtal.
- Ha *G* 1-es típusú, azaz hossznemcsökkentőek a szabályai, akkor a 2. szalagon lévő mondatforma hossza sose haladhatja meg |*w*| -t, így ez az NTG egy LKA.

R és $\mathcal{L}_1$

Biz. (folyt.):

- (2) Minden A LKA-hoz megadható egy G 1-es típusú grammatika, melyre $L(G) = L(A)$.
- Módosítjuk egy kicsit az előző előadás konstrukcióját.
- Legyen $\Gamma' := \Gamma \setminus \{\triangleright,\triangleleft\}$ és $G = ((\Gamma \setminus \Sigma) \cup Q \times \Gamma' \cup \{S,A\}, \Sigma, P, S)$.

1) $S \to \triangleright A(q_{accept},a)A\triangleleft \mid \triangleright A(q_{accept},a)\triangleleft \mid \triangleright(q_{accept},a)A\triangleleft \mid \triangleright(q_{accept},a)\triangleleft$ $\quad (\forall\ a \in \Gamma')$

2) $A \to aA \mid a$ $\quad (\forall\ a \in \Gamma')$

3) $b(q',c) \to (q,a)c$ if $(q',b,R) \in \delta(q,a)$ $\quad (\forall\ c \in \Gamma')$

4) $(q',b) \to (q,a)$ if $(q',b,S) \in \delta(q,a)$

5) $(q',c)b \to c(q,a)$ if $(q',b,L) \in \delta(q,a)$ $\quad (\forall\ c \in \Gamma')$

6) $\triangleright(q_0,a) \to \triangleright a$ $\quad (\forall\ a \in \Gamma')$

- 1-2. generálunk egy tetszőleges elfogadó konfigurációt. Mivel A LKA, ezért u elfogadásához elég olyan hosszút, mint u. Ezután a mondatforma hossza fix.
- 3-5. a konfigurációátmeneteket a grammatikában fordított irányban szimuláljuk.

R és $\mathcal{L}_1$

Biz. (folyt.):

1) $S \to \triangleright A(q_{accept},a)A\triangleleft \mid \triangleright A(q_{accept},a)\triangleleft \mid \triangleright(q_{accept},a)A\triangleleft \mid \triangleright(q_{accept},a)\triangleleft$ $(\forall\, a\in\Gamma')$
2) $A \to aA \mid a$ $(\forall\, a\in\Gamma')$
3) $b(q',c) \to (q,a)c$ if $(q',b,R) \in \delta(q,a)$ $(\forall\, c\in\Gamma')$
4) $(q',b) \to (q,a)$ if $(q',b,S) \in \delta(q,a)$
5) $(q',c)b \to c(q,a)$ if $(q',b,L) \in \delta(q,a)$ $(\forall\, c\in\Gamma')$
6) $\triangleright(q_0,a) \to \triangleright a$ $(\forall\, a\in\Gamma')$

- 6. Mivel a grammatika hossz-nemcsökkentő kell legyen technikailag szükségünk van $Q \times \Gamma'$-beli jelekre, ezekből az utolsó lépésig mindig pontosan egy van a mondatformában.
- Minden $a \in \Sigma \setminus \{\triangleright,\triangleleft\}$, $w \in (\Sigma \setminus \{\triangleright,\triangleleft\})^*$-ra vagy $a = \_$, $w = \varepsilon$-ra, megmutatható a levezetés hosszára vonatkozó indukcióval, hogy valamely

  $x \in \Gamma'$, $\alpha,\beta \in (\Gamma')^*$: $\triangleright q_0aw\triangleleft \vdash^* \triangleright\alpha q_{accept}x\beta\triangleleft$ akkor és csak akkor ha
  $S \Rightarrow^* \triangleright\alpha(q_{accept},x)\beta\triangleleft \Rightarrow^* \triangleright(q_0,a)w\triangleleft \Rightarrow \triangleright aw\triangleleft$. $\square$

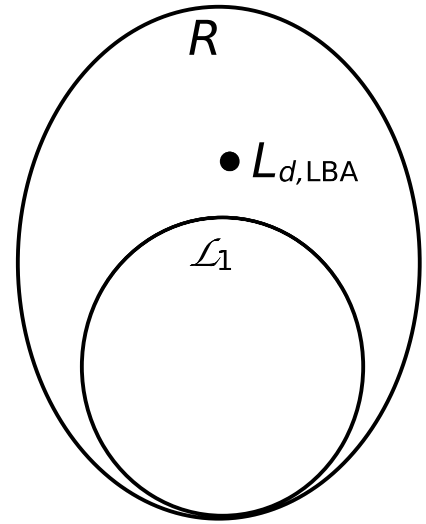
R és $\mathcal{L}_1$

Tétel: Ha *A* egy LKA, akkor *L*(*A*) eldönthető.

Biz.:

- Legyen *w* egy input szó, $|w|=n$. A lineáris korlátoltság miatt a lehetséges konfigurációk száma egy *w* inputra legfeljebb $m(w) = |Q| \cdot n \cdot |\Gamma|^n$.
- Minden számítás, ami hosszabb mint $m(w)$, végtelen cikluhoz vezet.
- Legyen *M'* egy TG, amely <*A*,*w*> inputtal, ahol *A* egy LKA és *w* egy sztring
  1) Szimulálja *A*-t *w*-vel $\leq m(w)+1$ lépésen át
  2) Ha *A* elfogad/elutasít ezen pont előtt, *M'* elfogad/elutasít mint *A*.
  3) Máskülönben elutasít.
- Nyilván $L(M') = L(A)$ és *M'* eldönti $L(A)$-t. □

R és $\mathcal{L}_1$



Tétel: $\mathcal{L}_1 \subset R$.

Biz.:

- Az előző 2 tétel alapján $\mathcal{L}_1 \subseteq R$.
- Legyen $L_{d,\mathrm{LBA}} = \{<A> : A$ egy LBA és $<A> \notin L(A)\}$.
- $L_{d,\mathrm{LBA}}$ eldönthető:
  - LKA A-hoz legyen S egy TG, amely
    - q_{accept} állapotba megy, ha $<A> \notin L(A)$ és
    - q_{reject} állapotba megy, ha $<A> \in L(A)$.

  Mivel $L(A)$ eldönthető, S mindig megáll. $\Rightarrow L_{d,\mathrm{LBA}} \in R$.
- $L_{d,\mathrm{LBA}}$ nem felismerhető LKA-val with LBA ($\Rightarrow L_{d,\mathrm{LBA}} \notin \mathcal{L}_1$)
  - Cantor diagonalizáció módszerével
  - TF. ellentmondáshoz, hogy $L_{d,\mathrm{LBA}}$ felismerhető egy S LKA-val, azaz $L(S)=L_{d,\mathrm{LBA}}$.
    - ha $<S> \in L_{d,\mathrm{LBA}} = L(S)$, akkor S elfogadja $<S>$-t, így $<S> \notin L_{d,\mathrm{LBA}}$, ellentmondás,
    - ha $<S> \notin L_{d,\mathrm{LBA}} = L(S)$, akkor S nem fogadja el $<S>$-t, így $<S> \in L_{d,\mathrm{LBA}}$, ellentmondás. $\square$

References

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