Formális nyelvek és fordítóprogramok

4. Véges automaták és reguláris nyelvek II.

Bodó Zalán

Babeș–Bolyai Tudományegyetem
Matematika és Informatika Kar
Magyar Matematika és Informatika Intézet

Véges automaták és reguláris grammatikák

ε-lépéses automaták

Műveletek véges automatákkal

DVA minimalizálása

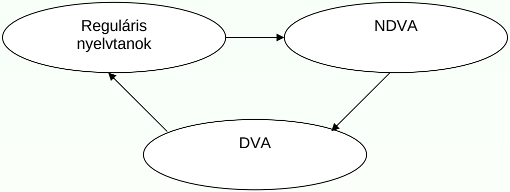
Pumpáló lemma

Véges automaták és reguláris grammatikák

Tétel

A következő három nyelvosztály megegyezik:

- a reguláris nyelvek osztálya
- DVA által felismerhető nyelvek osztálya
- NDVA által felismerhető nyelvek osztálya



Már ismerjük: NDVA $\rightarrow$ DVA

- ismétlés: $G = (N, T, P, S)$ reguláris nyelvtan:
  - P szabályai $A \to aB$ vagy $A \to a$ alakúak, ahol $a \in T$, $B \in N$
  - megengedhető az $S \to \varepsilon$ szabály is, ha S nem szerepel egyetlen szabály jobb oldalán sem
  - $L(G) = \{w \in T \mid S \overset{*}{\underset{G}{\Rightarrow}} w\}$ a G grammatika által generált reguláris nyelv

Tétel

Ha L egy tetszőleges DVA által felismert nyelv, akkor megkonstruálható egy olyan reguláris nyelvtan, amelyik az L nyelvet generálja.

Az algoritmus lépései:

- $A = (Q, \Sigma, E, \{q_0\}, F)$, $L = L(A)$
- $G = (Q, \Sigma, P, q_0)$, vagyis a nemterminális szimbólumoknak az automata állapotait választjuk, a terminálisok az automata ábécéje lesz, a kezdőszimbólum pedig az automata kezdőállapota
- P:
  - ha $(p, a, q) \in E$, akkor felvesszük a $p \to aq$ szabályt
  - ha $(p, a, q) \in E$ és $q \in F$, akkor a fenti szabály mellett felvesszük a $p \to a$ szabályt is

Fontos: ha A felismeri ε-t is, akkor bevezetünk egy új q_0' kezdőszimbólumot, felvesszük az új $q_0' \to \varepsilon$ szabályt, és minden $q_0 \to \alpha$ szabály mellé bevisszük a $q_0' \to \alpha$ szabályt is

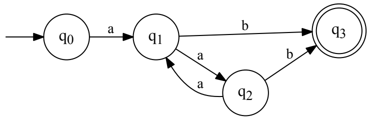
$A = (Q, \Sigma, E, \{q_0\}, F) \to G = (Q, \Sigma, P, q_0)$

ALG 1 DVA $\to$ RG

```
P = ∅
for minden (p, a, q) ∈ E do
    P = P ∪ {p → aq}
    if q ∈ F then
        P = P ∪ {p → a}
    end if
end for
```

+ megjegyzés

1. Példa



$G = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, P, q_0)$
A szabályok: P : {

$$\begin{array}{rcl} q_0 & \to & aq_1 \\ q_1 & \to & aq_2|bq_3|b \\ q_2 & \to & aq_1|bq_3|b \end{array}$$

}
Próba: *aaab*: $q_0 \Rightarrow aq_1 \Rightarrow aaq_2 \Rightarrow aaaq_1 \Rightarrow aaab$

RG $\rightarrow$ NDVA

Tétel

Ha $L = L(G)$ reguláris nyelv, akkor megkonstruálható olyan NDVA, amely az L nyelvet ismeri fel.

Az algoritmus lépései:

- $G = (N, T, P, S)$
- $A = (Q, T, E, \{S\}, F)$, vagyis az automata ábécéje a terminális szimbólumok halmazával fog megegyezni, a kezdőállapot pedig a grammatika kezdőszimbóluma lesz
- $Q = N \cup \{Z\}$, ahol Z egy új szimbólum/állapot
- minden $A \rightarrow aB$ szabályra felvesszük az (A, a, B) átmenetet
- minden $A \rightarrow a$ szabályra felvesszük az (A, a, Z) átmenetet
- $F = \begin{cases} \{Z\}, & \text{ha } G\text{-ben nincs } S \rightarrow \varepsilon \text{ szabály} \\ \{Z, S\}, & \text{ha } G\text{-ben van } S \rightarrow \varepsilon \text{ szabály} \end{cases}$

$$G = (N, T, P, S) \rightarrow A = (Q, T, E, \{S\}, F)$$

ALG 2 RG $\rightarrow$ NDVA

$E = \emptyset$
$Q = N \cup \{Z\}$
for minden $A \rightarrow u$ szabályra **do**
    if u = aB **then**
        $E = E \cup \{(A, a, B)\}$
    end if
    if u = a **then**
        $E = E \cup \{(A, a, Z)\}$
    end if
end for
if $(S \rightarrow \varepsilon) \in P$ **then**
    $F = \{Z\}$
else
    $F = \{Z, S\}$
end if

2. Példa

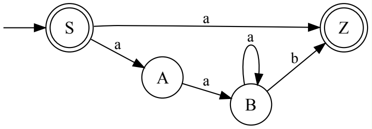
Legyen a köv. G reg. grammatika (szabályhalmaza):

$$\begin{array}{lcl} S & \rightarrow & a|aA|\varepsilon \\ A & \rightarrow & aB \\ B & \rightarrow & aB|b \end{array}$$

$(L(G) = \{a^n x \mid n \geq 0, x =$
ε ha $n \in \{0, 1\}$, ellenkező esetben $x = b\})$
Az átmenettáblázat:

	a	b
S	$\{A, Z\}$	$\emptyset$
A	$\{B\}$	$\emptyset$
B	$\{B\}$	$\{Z\}$
Z	$\emptyset$	$\emptyset$

Az automata:



Reguláris nyelvek – kiegészítés

Ha L_1, L_2 reg. nyelvek, akkor regulárisak a köv. is:

- $L_1 \cup L_2$
- $L_1 L_2,\ L_2 L_1$
- $L_1*,\ L_2*$

Ezen kívül regulárisak köv. konstrukciók is:

- $\overline{L_1},\ \overline{L_2}$
  Biz. Legyen $A = (Q, \Sigma, E, \{q_0\}, F)$ az L_1-et felismerő *teljes determinisztikus* automata. Ekkor $\overline{A} = (Q, \Sigma, E, \{q_0\}, Q \setminus F)$ az $\overline{L_1}$-et fogja felismerni; tehát a grammatika (melyet könnyen megszerkeszthetünk) reguláris lesz
- $L_1 \cap L_2$
  Biz. $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$
- $L_1 \setminus L_2,\ L_2 \setminus L_1$
  Biz. $L_1 \setminus L_2 = L_1 \cap \overline{L_2}$

ε-lépéses automaták

- ε-lépés: az automata nem olvas semmit a bemeneti szalagról, de átmegy egy állapotból a másikba
- $E \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$, $\delta(p, a) = \{q \in Q \mid (p, a, q) \in E\}$

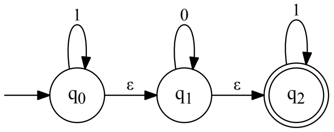
Tétel

Tetszőleges ε-lépéses véges automatához mindig megkonstruálható egy vele ekvivalens NDVA, amely nem tartalmaz ε-lépéseket.

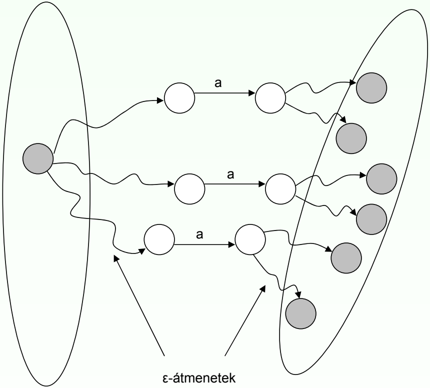
- ε-lépéses automata: $A = (Q, \Sigma, E, I, F)$
- NDVA: $\overline{A} = (Q, \Sigma, \overline{E}, I, \overline{F})$ (az algoritmus az $\overline{E}$ és $\overline{F}$) halmazokat határozza meg)
- **Def.**: $\Lambda(q) =$ azon állapotok halmaza, amelyekbe el lehet jutni q-ból csupa ε-lépéssel (beleértve magát a q-t is). Halmaz esetén:

$$\Lambda(S) = \bigcup_{q \in S} \Lambda(q), \forall S \subseteq Q$$

Példa ε-lépéses véges automatára:



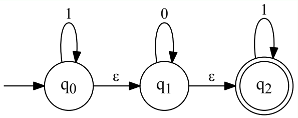
Az algoritmus alapötletének grafikus ábrázolása:



ALG 3 ε-mentesítés

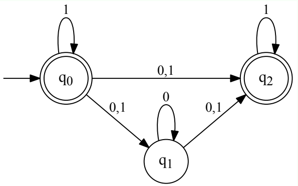
$\overline{F} = F \cup \{q \in I \mid \Lambda(q) \cap F \neq \emptyset\}$
for minden $q \in Q$ **do**
  for minden $a \in \Sigma$ **do**
    $\Delta = \bigcup_{p \in \Lambda(q)} \delta(p, a)$
    $\overline{\delta}(q, a) = \Delta \cup \left(\bigcup_{p \in \Delta} \Lambda(p)\right)$
  end for
end for
$\overline{E} = \{(p, a, q) \mid p, q \in Q, a \in \Sigma, q \in \overline{\delta}(p, a)\}$

3. Példa



$\Lambda(q_0) = \{q_0, q_1, q_2\}$, $\Lambda(q_1) = \{q_1, q_2\}$, $\Lambda(q_2) = \{q_2\}$
$\overline{F} = \{q_2\} \cup \{q_0\} = \{q_0, q_2\}$

$\overline{\delta}$	0	1
q_0	$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$
q_1	$\{q_1, q_2\}$	$\{q_2\}$
q_2	$\emptyset$	$\{q_2\}$



Műveletek véges automatákkal

$A_1 = (Q_1, \Sigma_1, E_1, I_1, F_1)$, $A_2 = (Q_2, \Sigma_2, E_2, I_2, F_2) \Rightarrow$
$A = (Q, \Sigma, E, I, F)$ (ε-lépéses) véges automata

1. **Egyesítés**: $A = A_1 \cup A_2$
   $Q = Q_1 \cup Q_2 \cup \{q_0\}$
   $\Sigma = \Sigma_1 \cup \Sigma_2$
   $I = \{q_0\}$
   $F = F_1 \cup F_2$
   $E = E_1 \cup E_2 \cup \bigcup_{q \in I_1 \cup I_2} \{(q_0, \varepsilon, q)\}$
2. **Szorzat**: $A = A_1 \cdot A_2$
   $Q = Q_1 \cup Q_2$
   $\Sigma = \Sigma_1 \cup \Sigma_2$
   $F = F_2$
   $I = I1$
   $E = E_1 \cup E_2 \cup \bigcup_{p \in F_1, q \in I_2} \{(p, \varepsilon, q)\}$
3. **Iteráció**: A_1*
   $Q = Q_1$
   $\Sigma = \Sigma_1$
   $F = F_1$
   $I = I_1$
   $E = E_1 \cup \bigcup_{p \in I_1, q \in F_1} \{(p, \varepsilon, q), (q, \varepsilon, p)\}$

DVA minimalizálása

Def. Minimális automata

Egy $A = (Q, \Sigma, E, \{q_0\}, F)$ teljes DVA-t minimálisnak nevezünk, ha bármely vele ekvivalens $A' = (Q', \Sigma, E', \{q_0'\}, F')$ teljes DVA esetében teljesül, hogy $|Q| \leq |Q'|$.

Def. Ekvivalens állapotok

Az $A = (Q, \Sigma, E, \{q_0\}, F)$ DVA p és q állapotát ekvivalensnek nevezzük, ha minden w szóra mindkettőből végállapotba jutunk vagy egyikből sem, vagyis

$$p \equiv q, \text{ ha minden } w \in \Sigma^* : \begin{cases} p \xrightarrow{w} r, r \in F \text{ és } q \xrightarrow{w} s, s \in F \text{ VAGY} \\ p \xrightarrow{w} r, r \notin F \text{ és } q \xrightarrow{w} s, s \notin F \end{cases}$$

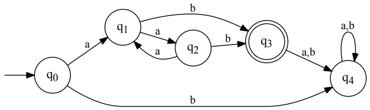
Az algoritmus alapötlete:

- meghatározzunk az ekvivalens állapotpárokat
- az ekvivalenseket összevonjuk
- (az elérhetetlen állapotokat első lépésben kizárjuk – ezek nem játszanak szerepet az algoritmusban; a nem produktívakat hagyjuk – ezekre szükség lehet a teljességhez)

ALG 4

```
Jelöljük meg csillaggal (*) az összes (p, q) állapotpárt, amelyre p ∈ F és q ∉ F vagy fordítva
Minden jelöletlen (p, q) párhoz rendeljünk egy üres listát
for Minden jelöletlen (p, q) párra és minden a ∈ Σ betűre vizsgáljuk meg a (δ(p, a), δ(q, a)) állapotpárokat do
    if az így kapott párok közül valamelyik meg van csillagozva then
        csillagozzuk meg a (p, q) párt is, a már előzőleg ehhez a párhoz rendelt lista elemeivel együtt
    else if a fenti állapotpárok közül egy sincs megcsillagozva then
        írjuk be a (p, q) párt a (δ(p, a), δ(q, a)) párokhoz rendelt lista mindegyikébe, feltéve, hogy δ(p, a) ≠ δ(q, a)
    end if
end for
Vonjuk össze a jelöletlen (ekvivalens) állapotpárokat.
```

4. Példa

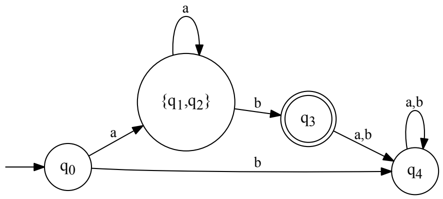


$$
\begin{array}{lllllll}
(q_0, q_1)* & \xrightarrow{a} & (q_1, q_2) & \xrightarrow{a} & (q_2, q_1) & \xrightarrow{a} & (q_1, q_2)\ (\text{kör}) \\
 & & & & & \xrightarrow{b} & (q_3, q_3) \\
 & & & \xrightarrow{b} & (q_3, q_3) & & \\
 & \xrightarrow{b} & (q_4, q_3)* & & & & \\
(q_0, q_2)* & \xrightarrow{a} & (q_1, q_1) & & & & \\
 & \xrightarrow{b} & (q_4, q_3)* & & & & \\
(q_0, q_4)* & \xrightarrow{a} & (q_1, q_4)* & \xrightarrow{a} & (q_2, q_4)* & \xrightarrow{a} & (q_1, q_4)\ (\text{kör}) \\
 & & & & & \xrightarrow{b} & (q_3, q_4)*
\end{array}
$$

Az eredő táblázat (könnyen leolvasható, mit kell/lehet összevonni):

	q_1	q_2	q_3	q_4
				*
			*	*
			*	*
q_0	*	*	*	*

A minimalizált automata:



Pumpáló lemma reguláris nyelvekre

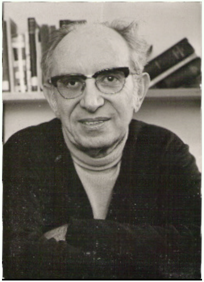
Lemma (Bar-Hillel)

Bármely L reguláris nyelv esetében létezik olyan $n \geq 1$ természetes szám (amely csak L-től függ), hogy L bármely legalább n hosszúságú u szava felírható $u = xyz$ alakban úgy, hogy

1. $|xy| \leq n$,
2. $|y| \geq 1$,
3. $xy^iz \in L$, minden $i = 0, 1, 2, \ldots$ értékre.

A fenti lemmát arra használjuk, hogy bebizonyítsuk egy nyelvről, hogy az nem reguláris.

Yehoshua Bar-Hillel



Yehoshua Bar-Hillel (születési nevén Oscar Westreich, 1915 Bécs–1975 Jeruzsálem) izraeli filozófus, matematikus és nyelvész. A gépi fordítás mint tudományág megteremtője, ő szervezte az első ilyen konferenciát is 1952-ben. Az információ-visszakeresés egyik úttörőjének is tartják.

Egy híres műve:
Bar-Hillel J. *Language and Information. Selected Essays on Their Theory and Application*. Addison-Wesley, Reading, MA, 1964.

5. Példa

Mutassuk meg, hogy az $L = \{a^k b^k \mid k \geq 1\}$ nyelv nem reguláris.

Tegyük fel, hogy L reguláris, és válasszuk a pumpáló lemmában szereplő számot n-nek. Ekkor $|a^n b^n| = 2n > n$, tehát válasszuk az $u = a^n b^n$ szót. Ekkor u felbontható xyz alakban. A lemma 1. és 2. követelményének eleget téve az $u = a^n b^n$ szót a következőképpen bonthatjuk fel:

$$\begin{array}{ccc} x & y & z \\ a^\ell & a^m & a^{n-\ell-m} b^n \end{array}$$

ahol a következő feltételeknek kell teljesülniük: $\ell + m \leq n$, $m \geq 1$ és $\ell \geq 0$.
A 3. pont feltételeit nem tudjuk minden esetben kielégíteni, mivel $xy^i z = a^\ell a^{m \cdot i} a^{n-\ell-m} b^n$ alakú, ami az egyszerűsítések után $a^{n+m(i-1)} b^n$ lesz. Ha i-t pl. 2-nek választjuk, az $a^{n+m} b^n$ szóhoz jutunk, és mivel $m \geq 1$, nem lesz egyenlő számú a és b szimbólumunk.

Ellentmondáshoz jutottunk, tehát a feltevésünk, hogy L reguláris, hamis.