

A δ leképezés elemzése

bevezető paraméterek

δ	K	$V \cup \{\epsilon\}$	W	K	W^*
1.	q_0	ϵ <i>üres verem</i>	Z_0	q_0	Z_0
2.	q_0	1	z	q_1	zxy
3.	q_0 <i>korábban</i>	1 <i>vesztés</i>	z <i>olvasás</i>	q_0	ϵ <i>írás</i>
4.	q_1	1	z	q_1	Z

\rightarrow nem olvasunk, nem írunk végtelen CIKLUS lehet!

$\begin{bmatrix} \delta \\ z \\ z \end{bmatrix}$ *először a z-t \rightarrow 1-gyel bővítjük*
utána olvasunk \rightarrow 2-vel nő

\rightarrow verem méretének bővítése

mindaddig míg 1-et olvasunk be

q_1 -en állunk.

1. sor: Az automata nem olvas az input szalagról, és olvasás előtt után q_0 állapotba kerül, a veremből z_0 -t olvas, és ír (így a veremautomata *nem csinál „semmit”*). Ezt a sort az automata önmagában végtelen sokszor ismételheti (végtelen ciklus).

2. sor: Az automata egy jelet vesz ki a veremből (z), és három jelet tesz vissza (zxy). Ezen sor végrehajtása után *a verem aktuális mérete 2-vel növekszik.*

3. sor: Az automata egy jelet vesz ki a veremből (z), és nem tesz vissza semmit (ϵ). Ezen sor végrehajtása után *a verem* aktuális 1-es *csökken*.

4. sor: Az automata egy jelet vesz ki a veremből (z), és egy jelet tesz vissza (z). Ezen sor végrehajtása után *a verem* aktuális *mérete nem változik*.

A verem automata működésének elemzése

● **Megj.:** A verem automata **nem mindig áll meg**. Előfordulhat, hogy az input szalagról végtelen lépésen keresztül sem olvas, csak veremműveletek végez (olvas a veremből, és ír). Ezen lépések közben csak a verem változik, meg a belső állapot. A veremautomata akkor esik „végtelen ciklusba”, ha a lépéssorozatban előfordul az, hogy kétszer is ugyanazon belső állapotban van, és a verem tetején is ugyanaz a jel áll.

Megj.: Ha egy helyes szót táplálunk az automatába:

1. az automata legfeljebb n lépés után **megáll**, és **elfogadó** állapotba kerül (ha az automata jól választja meg a lépéssorozatot)
- 2. az automata legfeljebb n lépés után **megáll**, de **nem elfogadó** állapotban van (ha az automata nemdeterminisztikus, és rossz útvonalat választott)

3. az automata **megáll** (akár kevesebb, mint n lépésen belül, de akár több mint n lépés után is) **parciális okokból** (ha rossz útvonalat választott)
- 4. az automata **nem áll meg** (nemdeterminisztikus, végtelen ciklusba esik)

Megj.: Ha egy helytelen szót táplálunk az automatába:

5. Az automata legfeljebb n lépés után **megáll**, de **nem elfogadó** állapotban van (nemdeterminisztikus és determinisztikus esetben is)
6. Az automata legfeljebb n lépés után **megáll parciális okokból** (nemdeterminisztikus és determinisztikus esetben is)
7. Az automata **nem áll meg** (ekkor még determinisztikus is lehet)

Számítási kapacitás

● *Mejj.:* Létezik olyan verem automata konstrukció, amelyben nem szerepel az F halmaz. Ezen automata úgy jelzi, hogy felismer egy szót, hogy a működésének végén üres lesz a verem (vagyis a verem tetején levő jel z_0). Az ilyen automaták $G_{z_0}(K, V, W, \delta, q_0, z_0)$ formális hatások.

Tétel: Minden $G_{z_0}(K, V, W, \delta, q_0, z_0)$ **üres veremmel** felismerő verem automatához **konstruálható** olyan $G(K, V, W, \delta, q_0, z_0, F)$ **hagyományos** verem automata, amelyek ekvivalensek, és viszont.

Biz.: Azon múlik, hogy a hagyományos veremautomata felismeréses befejezés megállása előtt ürítse ki a vermet egy veremkiürítő δ lépéssorozattal, melynek során az input szalagról ε -t olvas, a veremből egy jelet olvas, és ε -t tesz vissza.

Amennyiben pedig van egy üres veremmel felismerő verem automatánk, abból úgy tudunk hagyományosat csinálni, hogy a felismerés végén még megvizsgálja, hogy a verem tetején z_0 van-e. Ha igen, egy felismerő állapotba billen, ha nem, akkor egy nem felismerőbe, és csak ekkor áll meg.

Megj.: Az üres veremmel felismerő verem automaták automataosztályának számítása kapacitása megegyezik a hagyományos veremautomaták automataosztály számítási kapacitásával.

Tétel: A verem automaták automataosztályának számítása kapacitása megegyezik a környezetfüggetlen nyelvek osztályával.

Megj.: Vagyis minden 2-es típusú nyelvhez konstruálható olyan verem automata, amely az adott nyelvet felismeri, ill. ha van egy verem automatánk, akkor az általa felismert nyelv legalább 2-es típusú.

Megj.: A veremautomaták felülről kompatibilisek a véges automatákkal, ugyanis ha a veremautomata δ sorai mindig z_0 -t olvas és ír a verembe, és minden lépésnél olvas a szalagról, akkor visszkapjuk a véges automatákat. Viszont ha egy veremautomata felismer egy nyelvet, arról lehet tudni, hogy legalább 2-es típusú, de lehet akár 3-as is!

3.2.5 Példa veremautomata

A nyelv: $S \rightarrow 1S1$, $S \rightarrow 0S0$, $S \rightarrow \varepsilon$, az automata elfogadó állapota a B . Az automata egyúttal üres veremmel is felismerő automata.

K	$V \cup \{\varepsilon\}$	W	K	W_*
q_0	1	z_0	q_0	1
q_0	0	z_0	q_0	0
q_0	0	1	q_0	10
q_0	0	0	q_0	00
q_0	1	1	q_0	11
q_0	1	0	q_0	01
q_0	1	1	A	ε
q_0	0	0	A	ε
A	1	1	A	ε
A	0	0	A	ε
A	ε	z_0	B	z_0
\uparrow	\uparrow	\uparrow	\uparrow	
inver	input.	stack	hol van a stack	

x /
 x / *nem determinisztikus automata*
levezetését látjuk
 x /
 x /

110011
 ↑↑↑↑↑
 kereséjé

← nem lehet elolvasni
 $q_0(z_0)$ a végen azért jelelt:

$q_0(z_0, 1)$

$q_0(z_0, 11)$ $A(z_0)$ → parciális ap

↓ 1-et elolvasunk
 2-t elolvasunk

$q_0(z_0, 110)$

$q_0(z_0, 1100)$

$A(z_0, 11)$

$q_0(z_0, 11001)$

$A(z_0, 1)$

$q_0(z_0, 110011)$

$A(z_0, 1100)$

$A(z_0)$

↓
 nem fogadjuk
 el, mert B az
 elfogadás állapot.

↓
 nem lehet

↓
 utolsó sor alapján
 ELTÖRÁDVA
 A-ban vagyunk, nem olvasunk
 semmit, nem olvasunk semmit →
 állomást B-be.

↓
 $B(z_0)$

↓

$B(z_0)$