Formális nyelvek és fordítóprogramok

7. Veremautomaták és környezetfüggetlen nyelvek

Bodó Zalán

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

Veremautomaták

Def. Veremautomata

Egy nemdeterminisztikus veremautomatát a $V = (Q, \Sigma, W, E, q_0, z_0, F)$ hetessel jelölünk, ahol

- Q az **állapotok** véges és nem üres halmaza
- Σ a **bemeneti ábécé**
- W a **veremábécé**
- $E \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times W \times W^* \times Q$ az **átmenetek** vagy **élek** halmaza
- $q_0 \in Q$ a **kezdőállapot**
- $z_0 \in W$ a **verem kezdőszimbóluma**
- $F \subseteq Q$ a **végállapotok** halmaza

Def. Átmenet

Egy átmenetet a (p, a, z, w, q) ötös jelöl, melyet úgy értelmezünk, hogy az automata

- p állapotban van
- a bemeneti szalagról az a jelet olvassa
- a verem tetején levő szimbólum z
- a z helyére a w szót írja
- a q állapotba megy át

Egy jobb jelölés: $(p, (a, z/w), q)$.

Az **átmenetfüggvény**:

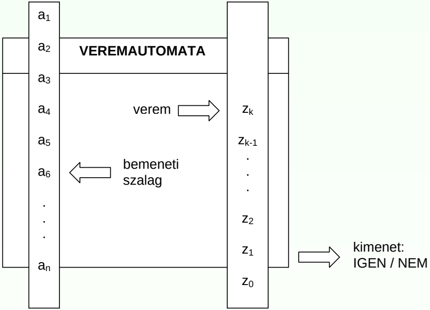
$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times W \to \mathcal{P}(W^* \times Q)$$

Nemdeterminisztikus esetben $\delta(q, a, z) = \{(w_1, p_1), \ldots, (w_k, p_k)\}$

Def. Determinisztikus veremautomata

A veremautomata determinisztikus, ha teljesül, hogy bármely q állapot és z veremszimbólum esetén:

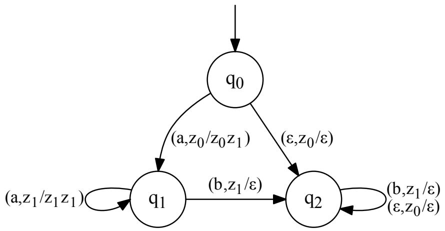
$$|\delta(q,a,z)| \leq 1, \forall a \in \Sigma \cup \{\varepsilon\}$$



ábra: Veremautomata sematikus ábrázolása

1. Példa

Példa veremautomatára:



Az átmenettáblázat:

	a		b	ε
	z_0	z_1	z_1	z_0
q_0	(z_0z_1, q_1)			(ε, q_2)
q_1		(z_1z_1, q_1)	(ε, q_2)	
q_2			(ε, q_2)	(ε, q_2)

Az automata által feismert nyelv (üres veremmel): $\{a^n b^n \mid n \geq 0\}$

Def. Konfiguráció

A veremautomta egy konfigurációjának nevezzük a (q, u, v) hármast, ahol q az aktuális állapot, u a bemeneti szó és v a verem tartalma.

Az automata kétféleképpen válthat konfigurációt, léphet egyik állapotból a másikba:

1. $(q, a_1 a_2 \ldots a_k, x_1 x_2 \ldots x_{m-1} x_m) \Rightarrow (p, a_2 a_3 \ldots a_k, x_1 x_2 \ldots x_{m-1} w)$, ha $(q, (a_1, x_m/w), p) \in E$ (valódi mozgás)
2. $(q, a_1 a_2 \ldots a_k, x_1 x_2 \ldots x_{m-1} x_m) \Rightarrow (p, a_1 a_2 a_3 \ldots a_k, x_1 x_2 \ldots x_{m-1} w)$, ha $(q, (\varepsilon, x_m/w), p) \in E$

A $\Rightarrow$ reláció reflexív, tranzitív lezártját $\overset{*}{\Rightarrow}$-gal jelöljük.

Def.

A V veremautomata **felismeri végállapottal** az u szót, ha létezik a konfigurációknak egy olyan sorozata, melyre teljesülnek a köv. feltételek:

- a sorozat első eleme (q_0, u, z_0),
- a sorozat minden eleméből van átmenet a köv. elembe,
- a sorozat utolsó eleme (p, ε, w), ahol $p \in F$ és $w \in W^*$.

Def.

A V veremautomata **felismeri üres veremmel** az u szót, ha létezik a konfigurációknak egy olyan sorozata, melyre teljesülnek a köv. feltételek:

- a sorozat első eleme (q_0, u, z_0),
- a sorozat minden eleméből van átmenet a köv. elembe,
- a sorozat utolsó eleme $(p, \varepsilon, \varepsilon)$, $p \in Q$.

Veremautomaták átalakítása

Tétel. Üres verem $\leftrightarrow$ végállapot

Egy L nyelv felismerhető egy V_1 automatával üres veremmel akkor és csakis akkor, ha egy V_2 veremautomatával végállapottal.

Bizonyítás. Konstruktív: a köv. két algoritmus megvalósítja az átalakításokat.

1. Üres verem $\rightarrow$ végállapot ($V_1 \rightarrow V_2$):

- átmegyünk egy ε-lépéssel V_1 kezőállapotába úgy, hogy beírjuk V_2 kezdőszimbóluma után V_1 kezdőszimbólumát
- ettől kezdve ugyanúgy működik mint V_1
- minden állapotra V_1-ből: ε-lépéssel töröljük V_2 vermének kezdőszimbólumát és átlépünk V_2 végállapotába

$V_1 = (Q, \Sigma, W, E, q_0, z_0, \emptyset)$

ALG 1 Üres verem $\rightarrow$ végállapot

1: $V_2 = (Q \cup \{p_0, p\}, \Sigma, W \cup \{x\}, E', p_0, x, \{p\})$, $p, p_0 \notin Q$, $x \notin W$

2: $E' = E \cup \{(p_0, (\varepsilon, x/xz_0), q_0)\} \cup \{(q, (\varepsilon, x/\varepsilon), p) \mid q \in Q\}$

---

2. Végállapot $\rightarrow$ üres verem ($V_2 \rightarrow V_1$):

- V_1 ε-lépéssel beírja kezdőszimbóluma mellé V_2 vermének kezdőszimbólumát
- ettől kezdve úgy működik mint V_2
- minden végállapotból ε-lépéssel és minden veremszimbólumra átmegyünk V_1 új állapotába és a veremszimbólumot töröljük
- V_1 új állapotából ε-lépéssel és minden veremszimbólumra átmegyünk ugyanebbe az állapotba és a veremszimbólumot töröljük

$V_2 = (Q, \Sigma, W, E, q_0, z_0, F)$

ALG 2 Végállapot $\rightarrow$ üres verem

1: $V_1 = (Q \cup \{p_0, p\}, \Sigma, W \cup \{x\}, E', p_0, x, \emptyset)$, $p, p_0 \notin Q$, $x \notin W$

2:

$$\begin{aligned} E' &= E \cup \{(p_0, (\varepsilon, x/xz_0), q_0)\} \cup \\ &\quad \{(q, (\varepsilon, z/\varepsilon), p) \mid q \in F, \mathbf{z} \in \mathbf{W}\} \cup \\ &\quad \{(p, (\varepsilon, z/\varepsilon), p) \mid z \in W \cup \{x\}\} \end{aligned}$$

Megjegyzés: Az algoritmust egyszerűsíteni lehet úgy, hogy kihagyjuk az új kezdőállapot és veremszimbólum bevezetését, *mindössze* az utolsó két lépést végézzük el: a végállapotokból ε-lépéssel és minden veremszimbólumra átmegyünk az új állapotba és a veremszimbólumot töröljük + az új állapotból ε-lépéssel és minden veremszimbólumra átmegyünk ugyanebbe az állapotba és a veremszimbólumot töröljük.

Kérdés: A második sorban $z \in W$ vagy $z \in W \cup \{x\}$ kell legyen?

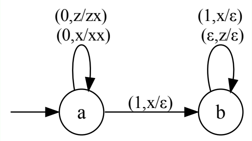
Válasz: Helyesen $z \in W \cup \{x\}$ kell szerepeljen, azaz

$$\begin{aligned} E' \;=\; & E \cup \{(p_0, (\varepsilon, x/xz_0), q_0)\} \cup \\ & \{(q, (\varepsilon, z/\varepsilon), p) \mid q \in F, \mathbf{z} \in \mathbf{W} \cup \{\mathbf{x}\}\} \cup \\ & \{(p, (\varepsilon, z/\varepsilon), p) \mid z \in W \cup \{x\}\} \end{aligned}$$

Kérdés: Miért?

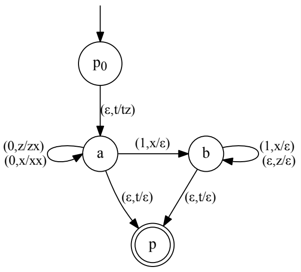
2. Példa

Legyen a következő veremautomata x, z veremszimbólumokkal, ahol z a kezdőszimbólum.



Az automata a $\{0^n1^n \mid n \geq 1\}$ nyelvet ismeri fel **üres veremmel**.

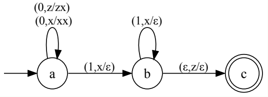
Alkalmazva az algoritmust, az alábbi automatát (t kezdőszimbólummal) kapjuk, amely ugyanezen nyelvet ismeri fel, azonban most **végállapottal**.



(Az $a \to p$ átmenetet akár el is lehet hagyni.)

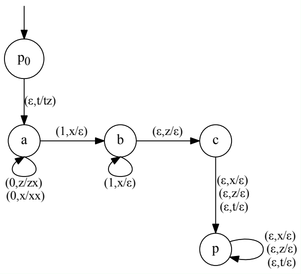
3. Példa

Legyen a következő veremautomata x, z veremszimbólumokkal, ahol z a kezdőszimbólum.



Az automata a $\{0^n1^n \mid n \geq 1\}$ nyelvet ismeri fel **végállapottal**.

Alkalmazva az algoritmust, az alábbi automatát (t kezdőszimbólummal) kapjuk, amely ugyanezen nyelvet ismeri fel, azonban most **üres veremmel**.



(Az első automata végállapottal és egyben üres veremmel ismeri fel a nyelvet, ezért nem lett volna szükséges alkalmazni az algoritmust.)

Veremautomaták és környezetfüggetlen nyelvek

- a veremautomaták és környezetfüggetlen nyelvek kapcsolatát vizsgáljuk
- ismétlés: $G = (N, T, P, S)$ környezetfüggetlen nyelvtan:
  - P szabályai $A \to \beta$ alakúak, $A \in N$, $\beta \in (N \cup T)^+$
  - megengedhető az $S \to \varepsilon$ is, ha S nem szerepel a szabályok jobb oldalán
- $L(G) = \{w \in T \mid S \overset{*}{\underset{G}{\Rightarrow}} w\}$ a G grammatika által generált környezetfüggetlen nyelv

Tétel

Ha G környezetfüggetlen nyelvtan, akkor létezik olyan V nemdeterminisztikus veremautomata, amely **üres veremmel** felismeri az $L(G)$ nyelvet, $L(G) = L(V)$.

$G = (N, T, P, S)$ környezetfüggetlen nyelvtan
$V = (\{q\}, T, N \cup T, E, q, S, \emptyset)$

ALG 3 Környezetfüggetlen nyelvtan $\rightarrow$ veremautomata

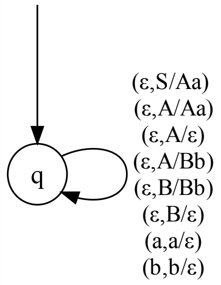
for minden $A \rightarrow \alpha$ szabályra **do**
  Vegyük fel E-be a $(q, (\varepsilon, A/\alpha^{-1}), q)$ átmenetet, ahol α^{-1} az α szimbólumsorozat fordítottját jelenti.
end for
for minden $a \in T$ jelre **do**
  Vegyük fel E-be a $(q, (a, a/\varepsilon), q)$ szabályt.
end for

4. Példa

Tekintsük a következő nyelvtant:

$$\begin{array}{rcl} S & \rightarrow & aA \\ A & \rightarrow & \varepsilon | aA | bB \\ B & \rightarrow & \varepsilon | bB \end{array}$$

A nyelvtan által generált nyelv $\{a^n b^m \mid n \geq 1, m \geq 0\}$.
A kapott egyállapotú automata:



Tétel

Ha V nemdeterminisztikus veremautomata, akkor létezik egy olyan G környezetfüggetlen grammatika, hogy V **üres veremmel** felismeri az $L(G)$ nyelvet, azaz $L(V) = L(G)$.

$V = (Q, \Sigma, W, E, q_0, z_0, \emptyset) \to G = (N, T, P, S)$

ALG 4 Veremautomata $\to$ környezetfüggetlen nyelvtan

$N = \{S\} \cup \{S_{p,z,q} \mid p, q \in Q, z \in W\}$
$T = \Sigma$
for minden $q \in Q$ **do**
    Vegyük fel P-be az $S \to S_{q_0,z_0,q}$ szabályt
end for
for minden $(q, (a, z/z_k \dots z_2 z_1), p) \in E$ **do**
    $\triangleright$ $q \in Q$, $z, z_1, z_2, \dots, z_k \in W$, $a \in \Sigma \cup \{\varepsilon\}$
    for minden $p_1, p_2, \dots, p_k$ állapotra **do**
        Vegyük fel P-be az
$S_{q,z,p_k} \to a S_{p,z_1,p_1} S_{p_1,z_2,p_2} \dots S_{p_{k-1},z_k,p_k}$ szabályt
    end for
end for
for minden $(q, (a, z/\varepsilon), p) \in E$ **do**
    $\triangleright$ $p, q \in Q$, $z \in W$, $a \in \Sigma \cup \{\varepsilon\}$
    Vegyük fel P-be az $S_{q,z,p} \to a$ szabályt
end for

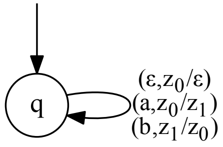
Egy kis magyarázat

- nem csak a helyettesítések hordoznak információt, hanem az állapotok közötti átmenetek is
- ha csak 1 állapota lenne az automatának, alkalmazható lenne az előbbi algoritmus visszafelé:
  - minden veremszimbólumhoz rendelünk egy új nemterminálist
  - szabály bal oldala: a helyettesítés bal oldala
  - szabály jobb oldal: a helyettesítés jobb oldala, kezdve a verem tetején levő szimbólummal; valódi mozgás esetén a jobb oldal az elolvasott szimbólummal kezdődik
- a 4. példában szereplő veremautomatára a következő grammatikát kapjuk:

$$
\begin{aligned}
S &\rightarrow XA \\
A &\rightarrow XA|\varepsilon|YB \\
B &\rightarrow YB|\varepsilon \\
X &\rightarrow a \\
Y &\rightarrow b
\end{aligned}
$$

ahol a köv. hozzárendeléseket használtuk: $S : S$, $a : X$, $b : Y$, $A : A$, $B : B$

Másik példa:



$L(V) = \{(ab)^n \mid n \geq 0\}$
Az előbbi naiv algoritmust használva kapjuk, hogy:

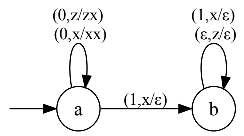
$$\begin{array}{lcl} A & \to & \varepsilon \\ A & \to & aB \\ B & \to & bA \end{array}$$

ahol a köv. megfeleltetéseket használtuk: $z_0 : A$, $z_1 : B$; a kezdőszimbólum az A.

- probléma: az átmenetek is információt hordoznak
- minden veremszimbólumhoz 2 fontos állapotot rendelünk:
  1. amikor a szimbólum megjelenik a verem tetején
  2. amikor a szimbólum kihal utódaiban (amikor a veremben alatta levő szimbólum kerül a verem tetejére)
- jelölés: $S_{q,z,p}$ – nemterminális szimbólum, amit a z veremszimbólumhoz rendeltünk, amely q állapotban kerül a verem tetejére, és p állapotban hal ki
- kétféle szabályunk van ($a \in \Sigma \cup \{\varepsilon\}$):
  1. $\delta(q, a, z) = (z_k z_{k-1} \ldots z_1, p)$
  2. $\delta(q, a, z) = (\varepsilon, p)$
- ezekhez rendre a köv. szabályokat rendelhetjük:
  1. $S_{q,z,p_k} \to a S_{p,z_1,p_1} S_{p_1,z_2,p_2} \ldots S_{p_{k-1},z_k,p_k}$
  2. $S_{q,z,p} \to a$
- a $p_1, p_2, \ldots, p_k$ állapotok nem szükségszerűen különböző állapotokat jelölnek; mivel nem tudunk semmilyen más információt, mint amit már beépítettünk a szabályokba, minden lehetséges kombinációt figyelembe kell vennünk

5. Példa

Tekintsük az alábbi, $\{0^n 1^n \mid n \geq 1\}$ nyelvet felismerő veremautomatát:



Felírjuk a környezetfüggetlen nyelvet, ami azt a nyelvet generálja, amit az automata felismer.

$N = \{S, S_{a,z,a}, S_{a,z,b}, S_{b,z,a}, S_{b,z,b}, S_{a,x,a}, S_{a,x,b}, S_{b,x,a}, S_{b,x,b}\}$

3–5. lépések: $S \rightarrow S_{a,z,a} | S_{a,z,b}$

6–12. lépések:

$$
\begin{array}{lcl}
3. S_{a,z,a} & \rightarrow & 0 S_{a,x,a} S_{a,z,a} \\
4. S_{a,z,b} & \rightarrow & 0 S_{a,x,a} S_{a,z,b} \\
5. S_{a,z,a} & \rightarrow & 0 S_{a,x,b} S_{b,z,a} \\
6. S_{a,z,b} & \rightarrow & 0 S_{a,x,b} S_{b,z,b}
\end{array}
$$

$$
\begin{aligned}
7.S_{a,x,a} &\rightarrow 0S_{a,x,a}S_{a,x,a}\\
8.S_{a,x,b} &\rightarrow 0S_{a,x,a}S_{a,x,b}\\
9.S_{a,x,a} &\rightarrow 0S_{a,x,b}S_{b,x,a}\\
10.S_{a,x,b} &\rightarrow 0S_{a,x,b}S_{b,x,b}
\end{aligned}
$$

13–16. lépések:

$$
\begin{aligned}
11.S_{a,x,b} &\rightarrow 1\\
12.S_{b,x,b} &\rightarrow 1\\
13.S_{b,z,b} &\rightarrow \varepsilon
\end{aligned}
$$

A szabályok számát jelentősen le tudjuk csökkenteni: kizárjuk azokat, melyeknek **nincs értelme**:
Az 5. szabályt kizárhatjuk, mert $S_{b,z,a}$ nem szerepel szabály jobb oldalán.
A 9. szabály hasonlóan, mert $S_{b,x,a}$ nem szerepel szabály jobb oldalán.
A 7. szabályt is, mert mostmár csak ott szerepel $S_{a,x,a}$ jobb oldalon, és ez végtelen ciklust generál.

A 3., 4. és 8. szabályokat is kizárhatjuk, mert csak $S_{a,x,a}$ már nem szerepel jobb oldalon.
Az 1. szabályt is kizárhatjuk, mert $S_{a,z,a}$ nem szerepel jobb oldalon. A következő szabályhalmazt kapjuk (jelölés: $A = S_{a,z,b}$,

$B = S_{a,x,b}$, $C = S_{b,z,b}$, $D = S_{b,x,b}$):

$$\begin{array}{rcl} S & \rightarrow & A \\ A & \rightarrow & 0BC \\ B & \rightarrow & 0BD|1 \\ D & \rightarrow & 1 \\ C & \rightarrow & \varepsilon \end{array}$$

Leegyszerűsítve az alábbi rövid grammatikát kapjuk:

$$\begin{array}{rcl} S & \rightarrow & 0B \\ B & \rightarrow & 0B1|1 \end{array}$$

Könnyen ellenőrizhető, hogy ez a grammatika a $\{0^n1^n \mid n \geq 1\}$ nyelvet generálja.

Környezetfüggetlen nyelvek

Def. Levezetési fa v. szintaxisfa

Egy $G = (N, T, P, S)$ környezetfüggetlen nyelvtan **levezetési fájának** vagy **szintaxisfájának** nevezzük azt a fát, amelyben a gyökér az S kezdőszimbólummal van címkézve, minden belső csúcs címkéje egy nemterminális szimbólum, minden levél pedig egy terminális szimbólummal. Teljesül továbbá az a feltétel, hogy minden $A \to X_1 X_2 \ldots X_k$ szabályra A belső csúcs a fában, amely k közvetlen leszármazottal rendelkezik, melyek rendre: $X_1, X_2, \ldots, X_k$.

A szintaxisfa eredménye az a T feletti szót, melyet úgy kapunk, hogy balról jobbra összeolvassuk a fa leveleinek címkéit.

Def. Legbaloldalibb levezetés

Az $\alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_n$ levezetést **legbaloldalibb levezetésnek** nevezzük, ha minden $i = 1, \ldots, n-1$ esetén van olyan $u_i \in T^*$, $\beta_i \in (N \cup T)^*$ és $(A_i \to \gamma_i) \in P$ szabály, hogy teljesülnek az

$$\alpha_i = u_i A_i \beta_i \text{ és } \alpha_{i+1} = u_i \gamma_i \beta_i (= u_{i+1} A_{i+1} \beta_{i+1})$$

összefüggések.

Def. Egyértelmű grammatika

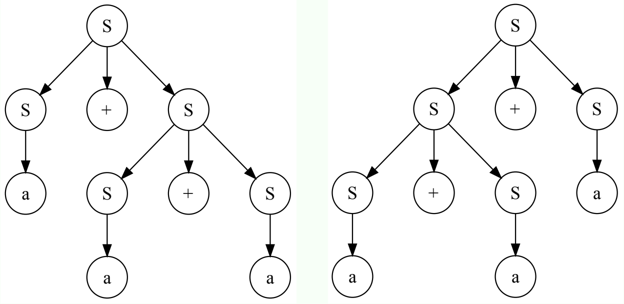
Egy G környezetfüggetlen grammatikát **egyértelműnek** nevezünk, ha minden szónak $L(G)$-ből csak egyetlen legbaloldalibb levezetése van. Ellenkező esetben a grammatika **nem egyértelmű**.

6. Példa

Tekintsük a köv. egyszerű környezetfüggetlen grammatikát: $G_1 = (\{S\}, \{+, a\}, P, S)$, $P = \{S \to S + S \mid a\}$. Észrevehető hogy az $a + a + a$ szó kétféleképpen is levezethető:

1. $S \Rightarrow S + S \Rightarrow a + S \Rightarrow a + S + S \Rightarrow a + a + S \Rightarrow a + a + a$
2. $S \Rightarrow S + S \Rightarrow S + S + S \Rightarrow a + S + S \Rightarrow a + a + S \Rightarrow a + a + a$

Következésképpen a grammatika nem egyértelmű.
A két szintaxisfa:



7. Példa

Tekintsük a köv. környezetfüggetlen grammatikát: $G_2 = (\{S\}, \{+, a\}, P, S)$, $P = \{S \to a \mid a + S\}$. Észrevehető, hogy ugyanazt a nyelvet generálja, mint az előző G_1 grammatika, $L(G_2) = L(G_1) = \{a(+a)^n \mid n \geq 0\}$.
Ebben az esetben viszont G_2 egyértelmű, mivel minden szónak csak *egy* legbaloldalibb levezetése van.

Pumpáló lemma környezetfüggetlen nyelvekre

Lemma (Bar-Hillel)

Tetszőleges L környezetfüggetlen nyelvhez megadható egy olyan n természetes szám (amely csak L-től függ) úgy, hogy minden L-beli, legalább n hosszúságú z szót fel lehet írni $uvwxy$ alakban, és igazak a következők:

1. $|w| \geq 1$,
2. $|vx| \geq 1$,
3. $|vwx| \leq n$,
4. uv^iwx^iy eleme L-nek $\forall i \in \{0, 1, 2, \ldots\}$.

8. Példa

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

Feltételezzük, hogy L k.független, és legyen n a tételben szereplő szám. Ekkor válasszuk pl. a $t = a^n b^n c^n$ szót, ami megfelel, mert $|t| = 3n > n$. Ekkor t felbontható $uvwxy$ alakban. A köv. 5 esetet különböztethetjük meg (azt véve figyelembe, hogy vwx milyen betűket tartalmazhat):

	u	vwx	y
1.	a^k	a^ℓ	$a^{n-k-\ell} b^n c^n$
2.	a^k	$a^{n-k} b^\ell$	$b^{n-\ell} c^n$
3.	$a^n b^k$	b^ℓ	$b^{n-k-\ell} c^n$
4.	$a^n b^k$	$b^{n-k} c^\ell$	$c^{n-\ell}$
5.	$a^n b^n c^k$	c^ℓ	$c^{n-k-\ell}$

Könnyen észrevehető, hogy a 4. feltétel nem lesz igaz, mert nem lesz mindig azonos számú a, b és c betű az $uv^i wx^i y$ szóban.