

Átlátható veremautomaták és nyelvek

Visibly pushdown automata and languages

Németh L. Zoltán

Számítástudomány Alapjai Tanszék
SZTE, Informatikai Tanszékcsoport

2007. április 23.

- 1 Motiváció: modellellenőrzés
- 2 Átlátható veremautomaták definíciója, példák
- 3 Zártsági tulajdonságok, determinalizáció, eldöntési kérdések
- 4 Ekvivalens jellemzések (MSO, reg. fanyelvek, nyelvtanok)
- 5 Átfogalmazás: egymásbaágyazott szavak
- 6 További eredmények, kutatási területek (nested words)

Klasszikus modellellenőrzés

- **Bemenet:** Modell: M , specifikáció: S
- **Kimenet:**
Igen, M teljesíti az S specifikációt, vagy
Nem, és egy ellenpélda.

Klasszikus modellellenőrzés

- **Bemenet:** Modell: M , specifikáció: S
- **Kimenet:**
Igen, M teljesíti az S specifikációt, vagy
Nem, és egy ellenpélda.
- **Klasszikus modellellenőrzés:**
 - mind M , mind S **reguláris** nyelvvel
(vagy ω -reguláris nyelvvel) írható le.
 - algoritmus: $M \cap \bar{S} \stackrel{?}{=} \emptyset$

Klasszikus modellellenőrzés

- **Bemenet:** Modell: M , specifikáció: S
- **Kimenet:**
Igen, M teljesíti az S specifikációt, vagy
Nem, és egy ellenpélda.
- **Klasszikus modellellenőrzés:**
 - mind M , mind S **reguláris** nyelvvel
(vagy ω -reguláris nyelvvel) írható le.
 - algoritmus: $M \cap \bar{S} \stackrel{?}{=} \emptyset$
- **Néhány tipikus specifikáció:**
 - Safety properties: pl. holtponmentesség
 - Liveness properties:
 - minden kérésre végül válasz érkezik,
 - a kezdőállapot mindig elérhető

Pushdown modellek ellenőrzése

- A vezérlés (control flow) leírásához veremre van szükség a (rekurzív) függvényhívások tárolásához, így az M modellt **környezetfüggetlen** nyelv írja le.

Pushdown modellek ellenőrzése

- A vezérlés (control flow) leírásához veremre van szükség a (rekurzív) függvényhívások tárolásához, így az M modellt **környezetfüggetlen** nyelv írja le.
- Ha M **környezetfüggetlen**, de S **reguláris**, az $M \cap \bar{S} \stackrel{?}{=} \emptyset$ ellenőrzés lehetséges.
 - $\bar{S}$ reguláris, $M \cap \bar{S}$ környezetfüggetlen
 - környezetfüggetlen nyelvek üressége eldönthető.

Pushdown modellek ellenőrzése

- A vezérlés (control flow) leírásához veremre van szükség a (rekurzív) függvényhívások tárolásához, így az M modellt **környezetfüggetlen** nyelv írja le.
- Ha M **környezetfüggetlen**, de S **reguláris**, az $M \cap \bar{S} \stackrel{?}{=} \emptyset$ ellenőrzés lehetséges.
 - $\bar{S}$ reguláris, $M \cap \bar{S}$ környezetfüggetlen
 - környezetfüggetlen nyelvek üressége eldönthető.
- De, ha M **környezetfüggetlen** és S **is környezetfüggetlen**, akkor **nincs általános algoritmus**:
 - a környezetfüggetlen nyelvek nem zártak a metszetképzésre
 - két környezetfüggetlen nyelv
 - metszetének üressége,
 - vagy tartalmazása **eldönthetetlen**.

- Pedig környezetfüggetlen modellek bizonyos környezetfüggetlen tulajdonságai nagyon is érdekesek. Például:
 - Hoare-stílusú pre/post feltételek:
 - parciális helyesség:** „Ha φ fennáll egy P eljárás meghívásakor, akkor visszatéréskor ψ teljesül.
 - totális helyesség:** „Ha φ fennáll egy P eljárás meghívásakor, akkor P *befejeződik* és visszatéréskor ψ teljesül.
 - Hozzáférés vezérlés (access control): a P eljárás csak akkor futtatható, ha R (root) a hívási láncban megelőzi (legalább egy példánya a veremben van P hívásakor)

Ezeknek a tulajdonságoknak az ellenőrzéséhez a metódus hívások és visszatérések egyeztetése, vagy a visszatérés nélküli hívások kiszűrése kell -> CFL nyelvek

Környezetfüggetlen specifikációk ellenőrzése

- Bár a feladat általánosan megoldhatatlan.
- Számos eszköz képes környezetfüggetlen modellek **speciális** környezetfüggetlen tulajdonságainak ellenőrzésére.

Környezetfüggetlen specifikációk ellenőrzése

- Bár a feladat általánosan megoldhatatlan.
- Számos eszköz képes környezetfüggetlen modellek **speciális** környezetfüggetlen tulajdonságainak ellenőrzésére.
- Ezekben közös vonás, hogy mind M mind S saját veremmel rendelkezik, de a két verem működése *szinkronizált*.
- Különítsük el az
 - eljárás híváskor (call)
 - visszatéréskor (return) megfigyelhető viselkedéseket.
 - egyéb műveletkor (local)
- Általánosításuk: **átlátható veremautomaták és nyelvek**
- Ha mind M -et mind S -et átlátható veremautomatával adjuk meg a modellellenőrzés kivitelezhető lesz.

Az ábécé felosztása

Rögzítsük a Σ ábécé felosztását három diszjunkt halmazra:

$\hat{\Sigma} = \langle \Sigma_c, \Sigma_r, \Sigma_\ell \rangle$, ahol

- Σ_c - „call”, hívási ábécé
- Σ_r - „return”, visszatérési ábécé
- Σ_ℓ - „local”, lokális ábécé

Egy $\hat{\Sigma}$ feletti átlátható veremautomata, egy olyan veremautomata, mely

- Σ_c -beli betű olvasásakor egy szimbólumot tesz a verembe,
- Σ_r -beli betű olvasásakor egy szimbólumot vesz ki veremből,
- Σ_ℓ -beli betű olvasásakor nem nyúl a veremhez.

Az átlátható veremautomaták definíciója II

Az átlátható veremautomaták (visibly pushdown automata, VPA) definíciója

Egy $\widehat{\Sigma}$ feletti átlátható veremautomata a következő:

$M = (Q, Q_{in}, \Gamma, \delta, Q_F)$, ahol

- Q az állapotok véges halmaza,
- Γ a veremábécé, $\perp \in \Gamma$ a veremkezdő szimbólum,
- $Q_{in} \subseteq Q$ a kezdőállapotok halmaza,
- $Q_F \subseteq Q$ a végállapotok halmaza
- $\delta = \langle \delta_c, \delta_r, \delta_\ell \rangle$, ahol

Az átlátható veremautomaták definíciója II

Az átlátható veremautomaták (visibly pushdown automata, VPA) definíciója

Egy $\hat{\Sigma}$ feletti átlátható veremautomata a következő:

$M = (Q, Q_{in}, \Gamma, \delta, Q_F)$, ahol

- Q az állapotok véges halmaza,
- Γ a veremábécé, $\perp \in \Gamma$ a veremkezdő szimbólum,
- $Q_{in} \subseteq Q$ a kezdőállapotok halmaza,
- $Q_F \subseteq Q$ a végállapotok halmaza
- $\delta = \langle \delta_c, \delta_r, \delta_\ell \rangle$, ahol

[push] $\delta_c \subseteq (Q \times \Sigma_c) \times (Q \times \Gamma - \{\perp\})$

Az átlátható veremautomaták definíciója II

Az átlátható veremautomaták (visibly pushdown automata, VPA) definíciója

Egy $\hat{\Sigma}$ feletti átlátható veremautomata a következő:

$M = (Q, Q_{in}, \Gamma, \delta, Q_F)$, ahol

- Q az állapotok véges halmaza,
- Γ a veremábécé, $\perp \in \Gamma$ a veremkezdő szimbólum,
- $Q_{in} \subseteq Q$ a kezdőállapotok halmaza,
- $Q_F \subseteq Q$ a végállapotok halmaza
- $\delta = \langle \delta_c, \delta_r, \delta_\ell \rangle$, ahol

[push] $\delta_c \subseteq (Q \times \Sigma_c) \times (Q \times \Gamma - \{\perp\})$

[pop] $\delta_r \subseteq (Q \times \Sigma_r \times \Gamma) \times Q$

Az átlátható veremautomaták definíciója II

Az átlátható veremautomaták (visibly pushdown automata, VPA) definíciója

Egy $\hat{\Sigma}$ feletti átlátható veremautomata a következő:

$M = (Q, Q_{in}, \Gamma, \delta, Q_F)$, ahol

- Q az állapotok véges halmaza,
- Γ a veremábécé, $\perp \in \Gamma$ a veremkezdő szimbólum,
- $Q_{in} \subseteq Q$ a kezdőállapotok halmaza,
- $Q_F \subseteq Q$ a végállapotok halmaza
- $\delta = \langle \delta_c, \delta_r, \delta_\ell \rangle$, ahol

[push] $\delta_c \subseteq (Q \times \Sigma_c) \times (Q \times \Gamma - \{\perp\})$

[pop] $\delta_r \subseteq (Q \times \Sigma_r \times \Gamma) \times Q$

[local] $\delta_\ell \subseteq (Q \times \Sigma_\ell) \times Q$

Veremtartalom, konfiguráció, futás

- lehetséges veremtartalmak: $St := (\Gamma \setminus \{\perp\})^* \perp$
- konfigurációk: $Q \times St$
- futás a $w = a_1 \dots a_k \in \Sigma^*$ szón:
 $(q_0, \sigma_0) \dots (q_k, \sigma_k)$, ahol $q_0 \in Q_{in}$, $\sigma_0 = \perp$, és $\forall 1 \leq i \leq k$ -ra

Veremtartalom, konfiguráció, futás

- lehetséges veremtartalmak: $St := (\Gamma \setminus \{\perp\})^* \perp$
- konfigurációk: $Q \times St$
- futás a $w = a_1 \dots a_k \in \Sigma^*$ szón:
 $(q_0, \sigma_0) \dots (q_k, \sigma_k)$, ahol $q_0 \in Q_{in}$, $\sigma_0 = \perp$, és $\forall 1 \leq i \leq k$ -ra
[push] Ha $a_i \in \Sigma_c$, akkor $\exists \gamma \in \Gamma \setminus \{\perp\} : (q_i, a_i, q_{i+1}, \gamma) \in \delta_c$ és
 $\sigma_{i+1} = \gamma \sigma_i$.

Veremtartalom, konfiguráció, futás

- lehetséges veremtartalmak: $St := (\Gamma \setminus \{\perp\})^* \perp$
- konfigurációk: $Q \times St$
- futás a $w = a_1 \dots a_k \in \Sigma^*$ szón:
 $(q_0, \sigma_0) \dots (q_k, \sigma_k)$, ahol $q_0 \in Q_{in}$, $\sigma_0 = \perp$, és $\forall 1 \leq i \leq k$ -ra

[push] Ha $a_i \in \Sigma_c$, akkor $\exists \gamma \in \Gamma \setminus \{\perp\} : (q_i, a_i, q_{i+1}, \gamma) \in \delta_c$ és $\sigma_{i+1} = \gamma \sigma_i$.

[pop] Ha $a_i \in \Sigma_r$, akkor $\exists \gamma \in \Gamma : (q_i, a_i, q_{i+1}, \gamma) \in \delta_r$ és $\gamma \neq \perp$ és $\sigma_i = \gamma \sigma_{i+1}$ vagy $\gamma = \perp$ és $\sigma_i = \sigma_{i+1} = \perp$.

Veremtartalom, konfiguráció, futás

- lehetséges veremtartalmak: $St := (\Gamma \setminus \{\perp\})^* \perp$
- konfigurációk: $Q \times St$
- futás a $w = a_1 \dots a_k \in \Sigma^*$ szón:
 $(q_0, \sigma_0) \dots (q_k, \sigma_k)$, ahol $q_0 \in Q_{in}$, $\sigma_0 = \perp$, és $\forall 1 \leq i \leq k$ -ra

[push] Ha $a_i \in \Sigma_c$, akkor $\exists \gamma \in \Gamma \setminus \{\perp\} : (q_i, a_i, q_{i+1}, \gamma) \in \delta_c$ és $\sigma_{i+1} = \gamma \sigma_i$.

[pop] Ha $a_i \in \Sigma_r$, akkor $\exists \gamma \in \Gamma : (q_i, a_i, q_{i+1}, \gamma) \in \delta_r$ és $\gamma \neq \perp$ és $\sigma_i = \gamma \sigma_{i+1}$ vagy $\gamma = \perp$ és $\sigma_i = \sigma_{i+1} = \perp$.

[local] Ha $a_i \in \Sigma_\ell$, akkor $(q_i, a_i, q_{i+1}) \in \delta_\ell$ és $\sigma_{i+1} = \sigma_i$.

Veremtartalom, konfiguráció, futás

- lehetséges veremtartalmak: $St := (\Gamma \setminus \{\perp\})^* \perp$
- konfigurációk: $Q \times St$
- futás a $w = a_1 \dots a_k \in \Sigma^*$ szón:
 $(q_0, \sigma_0) \dots (q_k, \sigma_k)$, ahol $q_0 \in Q_{in}$, $\sigma_0 = \perp$, és $\forall 1 \leq i \leq k$ -ra

[push] Ha $a_i \in \Sigma_c$, akkor $\exists \gamma \in \Gamma \setminus \{\perp\} : (q_i, a_i, q_{i+1}, \gamma) \in \delta_c$ és $\sigma_{i+1} = \gamma \sigma_i$.

[pop] Ha $a_i \in \Sigma_r$, akkor $\exists \gamma \in \Gamma : (q_i, a_i, \gamma, q_{i+1}) \in \delta_r$ és $\gamma \neq \perp$ és $\sigma_i = \gamma \sigma_{i+1}$ vagy $\gamma = \perp$ és $\sigma_i = \sigma_{i+1} = \perp$.

[local] Ha $a_i \in \Sigma_\ell$, akkor $(q_i, a_i, q_{i+1}) \in \delta_\ell$ és $\sigma_{i+1} = \sigma_i$.

- A futás sikeres, ha $q_k \in Q_F$
- Felismert nyelv: $L(M)$ azon szavak, melyeken van sikeres futás.

Néhány megjegyzés

Néhány megjegyzés

- A modell nemdeterminisztikus, de az input szó minden pozíción meghatározza az elvégzendő veremművelet típusát (push/pop/–).

Néhány megjegyzés

- A modell nemdeterminisztikus, de az input szó minden pozíción meghatározza az elvégzendő veremművelet típusát (push/pop/–).
- Lokális átmenetek nem olvashatják a verem tetején levő szimbólumot (de az tárolható lenne a belső állapotokban a veremábécé $\Gamma \times \Gamma$ -ra bővítésével).

Néhány megjegyzés

- A modell nemdeterminisztikus, de az input szó minden pozíción meghatározza az elvégzendő veremművelet típusát (push/pop/–).
- Lokális átmenetek nem olvashatják a verem tetején levő szimbólumot (de az tárolható lenne a belső állapotokban a veremábécé $\Gamma \times \Gamma$ -ra bővítésével).
- Az elfogadás végállapotokkal definiált, az üres veremmel való elfogadás túl szigorú lenne, pl. minden Σ_ℓ feletti szót elfogadna.

Néhány megjegyzés

- A modell nemdeterminisztikus, de az input szó minden pozíción meghatározza az elvégzendő veremművelet típusát (push/pop/–).
- Lokális átmenetek nem olvashatják a verem tetején levő szimbólumot (de az tárolható lenne a belső állapotokban a veremábécé $\Gamma \times \Gamma$ -ra bővítésével).
- Az elfogadás végállapotokkal definiált, az üres veremmel való elfogadás túl szigorú lenne, pl. minden Σ_ℓ feletti szót elfogadna.
- Nincsenek ε -átmenetek, de a veremhez nem nyúló ε -átmenetek nem növelnék a kifejező erőt.

Átlátható verem nyelvek (Visibly pushdown languages)

$\hat{\Sigma}$ -VPL definíciója

Egy $L \subseteq \Sigma^*$ nyelv **átlátható verem nyelv** a $\hat{\Sigma}$ felbontásra nézve, ha létezik olyan $\hat{\Sigma}$ -feletti M átlátható veremautomata, melyre $L = L(M)$. Röviden az L nyelv $\hat{\Sigma}$ -VPL.

Átlátható verem nyelvek (Visibly pushdown languages)

$\hat{\Sigma}$ -VPL definíciója

Egy $L \subseteq \Sigma^*$ nyelv **átlátható verem nyelv** a $\hat{\Sigma}$ felbontásra nézve, ha létezik olyan $\hat{\Sigma}$ -feletti M átlátható veremautomata, melyre $L = L(M)$. Röviden az L nyelv $\hat{\Sigma}$ -VPL.

Példák

Átlátható verem nyelvek (Visibly pushdown languages)

$\hat{\Sigma}$ -VPL definíciója

Egy $L \subseteq \Sigma^*$ nyelv **átlátható verem nyelv** a $\hat{\Sigma}$ felbontásra nézve, ha létezik olyan $\hat{\Sigma}$ -feletti M átlátható veremautomata, melyre $L = L(M)$. Röviden az L nyelv $\hat{\Sigma}$ -VPL.

Példák

- $\{a^n b^n | n \geq 0\}$ $\hat{\Sigma}$ -VPL, a $\Sigma_c = \{a\}$, $\Sigma_r = \{b\}$, $\Sigma_\ell = \emptyset$ felbontásra.

Átlátható verem nyelvek (Visibly pushdown languages)

$\hat{\Sigma}$ -VPL definíciója

Egy $L \subseteq \Sigma^*$ nyelv **átlátható verem nyelv** a $\hat{\Sigma}$ felbontásra nézve, ha létezik olyan $\hat{\Sigma}$ -feletti M átlátható veremautomata, melyre $L = L(M)$. Röviden az L nyelv $\hat{\Sigma}$ -VPL.

Példák

- $\{a^n b^n | n \geq 0\}$ $\hat{\Sigma}$ -VPL, a $\Sigma_c = \{a\}$, $\Sigma_r = \{b\}$, $\Sigma_\ell = \emptyset$ felbontásra.
- $\{b^n a^n | n \geq 0\}$ **nem** $\hat{\Sigma}$ -VPL, a $\Sigma_c = \{a\}$, $\Sigma_r = \{b\}$, $\Sigma_\ell = \emptyset$ felbontásra.

Átlátható verem nyelvek (Visibly pushdown languages)

$\hat{\Sigma}$ -VPL definíciója

Egy $L \subseteq \Sigma^*$ nyelv **átlátható verem nyelv** a $\hat{\Sigma}$ felbontásra nézve, ha létezik olyan $\hat{\Sigma}$ -feletti M átlátható veremautomata, melyre $L = L(M)$. Röviden az L nyelv $\hat{\Sigma}$ -VPL.

Példák

- $\{a^n b^n | n \geq 0\}$ $\hat{\Sigma}$ -VPL, a $\Sigma_c = \{a\}$, $\Sigma_r = \{b\}$, $\Sigma_\ell = \emptyset$ felbontásra.
- $\{b^n a^n | n \geq 0\}$ **nem** $\hat{\Sigma}$ -VPL, a $\Sigma_c = \{a\}$, $\Sigma_r = \{b\}$, $\Sigma_\ell = \emptyset$ felbontásra.
- $\{w \in \{a, b\}^* | |w|_a = |w|_b\}$ **nem** $\hat{\Sigma}$ -VPL semmilyen felbontásra sem.

Zárójeleket használó nyelvek

- Dyck_k - k különböző típusú zárójelpárt tartalmazó szabályosan zárójelezett szavak

$$\Sigma_c = \{[1, [2, \dots, [k\}, \Sigma_r = \{]1,]2, \dots,]k\}$$

Zárójeleket használó nyelvek

- Dyck_k - k különböző típusú zárójelpárt tartalmazó szabályosan zárójelezett szavak
 $\Sigma_c = \{[1, [2, \dots, [k\}, \Sigma_r = \{]1,]2, \dots,]k\}$
- Helyesen zárójelezett aritmetikai kifejezések. Pl:

$$(((a + b) * c) + (b * a))$$

Zárójeleket használó nyelvek

- Dyck_k - k különböző típusú zárójelpárt tartalmazó szabályosan zárójelezett szavak

$$\Sigma_c = \{[1, [2, \dots, [k\}, \Sigma_r = \{]1,]2, \dots,]k\}$$

- Helyesen zárójelezett aritmetikai kifejezések. Pl:

$$(((a + b) * c) + (b * a))$$

- C (CALL) – R (RETURN) sorozatok, melyben minden CALL-nak van RETURN párja, de előfordulhatnak pár nélküli RETURN-ök is. Pl:

C C R C R R C R R R C R

Tétel

Adott $\hat{\Sigma}$ felbontásra nézve, a $\hat{\Sigma}$ -VPL nyelvek osztálya zárt a következő műveletekre:

Tétel

Adott $\hat{\Sigma}$ felbontásra nézve, a $\hat{\Sigma}$ -VPL nyelvek osztálya zárt a következő műveletekre:

- unió (az automaták egyesítésével diszjunkt állapothalamazokkal)

Tétel

Adott $\hat{\Sigma}$ felbontásra nézve, a $\hat{\Sigma}$ -VPL nyelvek osztálya zárt a következő műveletekre:

- unió (az automaták egyesítésével diszjunkt állapothalamazokkal)
- metszet (a verem műveletek szinkronizáltsága miatt készíthető szorzat automata, veremábécé: $\Gamma_1 \times \Gamma_2$)

Tétel

Adott $\hat{\Sigma}$ felbontásra nézve, a $\hat{\Sigma}$ -VPL nyelvek osztálya zárt a következő műveletekre:

- unió (az automaták egyesítésével diszjunkt állapothalmazokkal)
- metszet (a verem műveletek szinkronizáltsága miatt készíthető szorzat automata, veremábécé: $\Gamma_1 \times \Gamma_2$)
- átnevezés $f : \hat{\Sigma} \rightarrow \hat{\Sigma}'$, melyre $f(\Sigma_c) \subseteq \Sigma'_c$, $f(\Sigma_r) \subseteq \Sigma'_r$, $f(\Sigma_\ell) \subseteq \Sigma'_\ell$.

Tétel

Adott $\hat{\Sigma}$ felbontásra nézve, a $\hat{\Sigma}$ -VPL nyelvek osztálya zárt a következő műveletekre:

- unió (az automaták egyesítésével diszjunkt állapothalmazokkal)
- metszet (a verem műveletek szinkronizáltsága miatt készíthető szorzat automata, veremábécé: $\Gamma_1 \times \Gamma_2$)
- átnevezés $f : \hat{\Sigma} \rightarrow \hat{\Sigma}'$, melyre $f(\Sigma_c) \subseteq \Sigma'_c$, $f(\Sigma_r) \subseteq \Sigma'_r$, $f(\Sigma_\ell) \subseteq \Sigma'_\ell$.
- konkatenáció (nem determinisztikusan megsejti a felbontást)

Tétel

Adott $\hat{\Sigma}$ felbontásra nézve, a $\hat{\Sigma}$ -VPL nyelvek osztálya zárt a következő műveletekre:

- unió (az automaták egyesítésével diszjunkt állapothalmazokkal)
- metszet (a verem műveletek szinkronizáltsága miatt készíthető szorzat automata, veremábécé: $\Gamma_1 \times \Gamma_2$)
- átnevezés $f : \hat{\Sigma} \rightarrow \hat{\Sigma}'$, melyre $f(\Sigma_c) \subseteq \Sigma'_c$, $f(\Sigma_r) \subseteq \Sigma'_r$, $f(\Sigma_\ell) \subseteq \Sigma'_\ell$.
- konkatenáció (nem determinisztikusan megsejti a felbontást)
- Kleene-iteráció (kicsit bonyolultabb mert elfogadáskor maradhatnak a veremben szimbólumok.)
Ld. [Alur], STOC'04 paper extended version.

Definíció

Egy átlátható $M = (Q, Q_{in}, \Gamma, \delta, Q_F)$ veremautomata **determinisztikus**, ha $|Q_{in}| = 1$ és minden

- $a \in \Sigma_\ell$ -re $(q, a, q') \in \delta_\ell$ legfeljebb egy q' állapotra teljesül,
- $a \in \Sigma_c$ -re $(q, a, q', \gamma) \in \delta_c$ legfeljebb egy q' állapotra és γ veremszimbólumra teljesül,
- $a \in \Sigma_r$ -re és $\gamma \in \Gamma$ -ra $(q, a, \gamma, q') \in \delta_r$ legfeljebb egy q' állapotra teljesül.

Tétel

Minden $\widehat{\Sigma}$ feletti M átlátható veremautomatához megadható olyan M' determinisztikus átlátható veremautomata, melyre $L(M) = L(M')$. Továbbá, ha M -nek n állapota van, akkor konstruálhatunk hozzá olyan M' automatát, melynek állapotszáma $\mathcal{O}(2^{n^2})$ veremábécéje pedig $\mathcal{O}(2^{n^2} \cdot |\Sigma_c|)$ elemű.

Tétel

Minden $\widehat{\Sigma}$ feletti M átlátható veremautomatához megadható olyan M' determinisztikus átlátható veremautomata, melyre $L(M) = L(M')$. Továbbá, ha M -nek n állapota van, akkor konstruálhatunk hozzá olyan M' automatát, melynek állapotszáma $\mathcal{O}(2^{n^2})$ veremábécéje pedig $\mathcal{O}(2^{n^2} \cdot |\Sigma_c|)$ elemű.

Bizonyítás

Ld. Kövesi Nagy Dániel előadását.

Tétel

Minden $\hat{\Sigma}$ feletti M átlátható veremautomatához megadható olyan M' determinisztikus átlátható veremautomata, melyre $L(M) = L(M')$. Továbbá, ha M -nek n állapota van, akkor konstruálhatunk hozzá olyan M' automatát, melynek állapotszáma $\mathcal{O}(2^{n^2})$ veremábécéje pedig $\mathcal{O}(2^{n^2} \cdot |\Sigma_c|)$ elemű.

Bizonyítás

Ld. Kövesi Nagy Dániel előadását.

Következmények

- 1 Az átlátható verem nyelvek osztálya zárt a komplementerképzésre nézve.
- 2 Az átlátható verem nyelvek determinisztikus környezetfüggetlen nyelvek (DCFL).

Általánosan gépekre, szavakra és nyelvekre

- Membership (elem-e): Adott automata felismer-e egy adott szót?

Általánosan gépekre, szavakra és nyelvekre

- Membership (elem-e): Adott automata felismer-e egy adott szót?
- Emptiness (üresség): Adott automata nyelve üres-e?

Általánosan gépekre, szavakra és nyelvekre

- Membership (elem-e): Adott automata felismer-e egy adott szót?
- Emptiness (üresség): Adott automata nyelve üres-e?
- Universality (univerzalitás): Adott automata nyelve minden szót felismer-e?

Általánosan gépekre, szavakra és nyelvekre

- Membership (elem-e): Adott automata felismer-e egy adott szót?
- Emptiness (üresség): Adott automata nyelve üres-e?
- Universality (univerzalitás): Adott automata nyelve minden szót felismer-e?
- Equivalence (ekvivalencia): Két adott automata nyelve megegyezik-e?

Általánosan gépekre, szavakra és nyelvekre

- Membership (elem-e): Adott automata felismer-e egy adott szót?
- Emptiness (üresség): Adott automata nyelve üres-e?
- Universality (univerzalitás): Adott automata nyelve minden szót felismer-e?
- Equivalence (ekvivalencia): Két adott automata nyelve megegyezik-e?
- Inclusion (tartalmazás) : Egy automata nyelve tartalmazza-e egy másikat?

Összehasonlítás zártsági tulajdonságok és eldöntési kérdések alapján

A környezetfüggetlen nyelvek részosztályai közt

Összehasonlítás zártági tulajdonságok és eldöntési kérdések alapján

A környezetfüggetlen nyelvek részosztályai közt

	Unió	Metszet	Kompl.	Konkat.	Kleene-*
Reg	✓	✓	✓	✓	✓
VPL	✓	✓	✓	✓	✓
DCFL	X	X	✓	X	X
CFL	✓	X	X	✓	✓

Összehasonlítás zártági tulajdonságok és eldöntési kérdések alapján

A környezetfüggetlen nyelvek részosztályai közt

	Unió	Metszet	Kompl.	Konkat.	Kleene-*
Reg	✓	✓	✓	✓	✓
VPL	✓	✓	✓	✓	✓
DCFL	X	X	✓	X	X
CFL	✓	X	X	✓	✓

	Emptiness	Univ./Equiv.	Inclusion
NFA	NLOGSPACE	PSPACE	PSPACE
VPA	PTIME	EXPTIME	EXPTIME
DPDA	PTIME	Eldönthető	Eldönthetetlen
PDA	PTIME	Eldönthetetlen	Eldönthetetlen

Monadikus másodrendű logika MSO_μ

- Elsőrendű változók: $x, y, z, \dots$ (pozíció értékűek)
- Másodrendű változók: $X, Y, \dots$ (pozícióhalmaz értékűek)
- Atomi formulák:
 - $Q_a(x)$ (x -en a betű van)
 - $X(x)$ (x benne van X -ben)
 - $x \leq y$ (y nem előzi meg x -et)
 - $\mu(x, y)$ (x -en call betű van, melynek a return párja y -on van)
- Formulák: atomi formulák, $\neg\varphi$, $\varphi \vee \psi$, $\exists x\varphi$, $\exists X\varphi$

Például $\hat{\Sigma} = \langle \{a\}, \{b\}, \{d\} \rangle$ esetén

$$\forall x (Q_a(x) \rightarrow \exists y \exists z (Q_b(y) \wedge Q_d(z) \wedge x \leq z \wedge z \leq y \wedge \mu(x, y)))$$

Tétel

$L \hat{\Sigma}\text{-VPL} \Leftrightarrow$ létezik olyan $\varphi \text{ MSO}_\mu$ mondat, ami L -et definiálja.

Verem fák (stack trees)

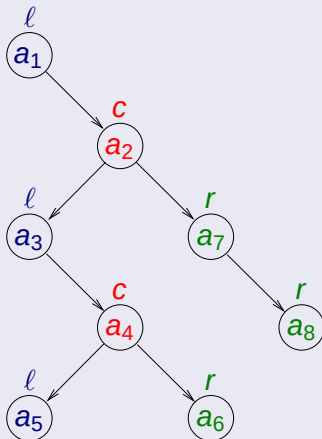
- Rögzítsük Σ egy $\hat{\Sigma}$ felbontását.
- Minden $\hat{\Sigma}$ feletti w szónak meg tudunk feleltetni kölcsönösen egyértelmű módon egy Σ betűivel címkézett véges bináris $\eta(w)$ fát.
- $w := cw_1rw_2$ esetén, ha c és r összetartozó call-return pozíciók, akkor a fában a c betűt reprezentáló csúcs bal részfája lesz w_1 , jobb részfájába pedig rw_2 reprezentációja.
- A return nélküli call betűknek megfelelő csúcsoknak a fában nem lesz jobb részfájuk.
- A return és local csúcsok megfelelőinek nem lesz bal részfájuk.
- Az $\eta(w)$ alakú fákat verem fáknak (stack trees) nevezzük.

Példa

Legyen $\Sigma_c = \{a_2, a_4\}$, $\Sigma_r = \{a_6, a_7, a_8\}$, $\Sigma_\ell = \{a_1, a_3, a_5\}$.
Ekkor a $w = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8$ szóhoz rendelt $\eta(w)$ fa a következő:

Példa

Legyen $\Sigma_c = \{a_2, a_4\}$, $\Sigma_r = \{a_6, a_7, a_8\}$, $\Sigma_\ell = \{a_1, a_3, a_5\}$.
Ekkor a $w = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8$ szóhoz rendelt $\eta(w)$ fa a következő:



Stack fa nyelvek

- Mivel az $\eta(w)$ fa preorder bejárásával visszacapjuk w -t, η bijektív.
- η természetes módon kiterjeszthető átlátható verem szavakról nyelvekre.
- Legyen $St := \eta(\Sigma^+)$.
- Belátható, hogy St reguláris fanyelv.

Stack fa nyelvek

- Mivel az $\eta(w)$ fa preorder bejárásával visszkapjuk w -t, η bijektív.
- η természetes módon kiterjeszthető átlátható verem szavakról nyelvekre.
- Legyen $St := \eta(\Sigma^+)$.
- Belátható, hogy St reguláris fanyelv.

Tétel

$L \in \widehat{\Sigma}\text{-VPL} \Leftrightarrow \eta(L)$ reguláris fanyelv. Továbbá minden átlátható veremnyelv előáll reguláris fanyelv η általi inverzképeként.

Átlátható verem nyelvtanok

Egy $G = (V, \Sigma, S, P)$ környezetfüggetlen nyelvtan **átlátható verem nyelvtan (visibly pushdown grammar)** a

$\hat{\Sigma} = \langle \Sigma_c, \Sigma_r, \Sigma_\ell \rangle$ felbontásra nézve, ha V két diszjunkt V^0 és V^1 halmazra osztható, úgy hogy a P -beli szabályok az alábbi alakúak:

Átlátható verem nyelvtanok

Egy $G = (V, \Sigma, S, P)$ környezetfüggetlen nyelvtan **átlátható verem nyelvtan (visibly pushdown grammar)** a

$\hat{\Sigma} = \langle \Sigma_c, \Sigma_r, \Sigma_\ell \rangle$ felbontásra nézve, ha V két diszjunkt V^0 és V^1 halmazra osztható, úgy hogy a P -beli szabályok az alábbi alakúak:

- $X \rightarrow \varepsilon$;

Átlátható verem nyelvannak

Egy $G = (V, \Sigma, S, P)$ környezetfüggetlen nyelvann **átlátható verem nyelvann (visibly pushdown grammar)** a

$\hat{\Sigma} = \langle \Sigma_c, \Sigma_r, \Sigma_\ell \rangle$ felbontásra nézve, ha V két diszjunkt V^0 és V^1 halmazra osztható, úgy hogy a P -beli szabályok az alábbi alakúak:

- $X \rightarrow \varepsilon$;
- $X \rightarrow aY$, ahol
ha $X \in V^0$, akkor $a \in \Sigma_\ell$ és $Y \in V^0$;

Átlátható verem nyelvannak

Egy $G = (V, \Sigma, S, P)$ környezetfüggetlen nyelvann **átlátható verem nyelvann (visibly pushdown grammar)** a $\hat{\Sigma} = \langle \Sigma_c, \Sigma_r, \Sigma_\ell \rangle$ felbontásra nézve, ha V két diszjunkt V^0 és V^1 halmazra osztható, úgy hogy a P -beli szabályok az alábbi alakúak:

- $X \rightarrow \varepsilon$;
- $X \rightarrow aY$, ahol
 ha $X \in V^0$, **akkor** $a \in \Sigma_\ell$ és $Y \in V^0$;
- $X \rightarrow aYbZ$, ahol $a \in \Sigma_c$, $b \in \Sigma_r$ és $Y \in V^0$, továbbá
 ha $X \in V^0$, **akkor** $Z \in V^0$.

Átlátható verem nyelvtanok

Egy $G = (V, \Sigma, S, P)$ környezetfüggetlen nyelvtan **átlátható verem nyelvtan (visibly pushdown grammar)** a $\hat{\Sigma} = \langle \Sigma_c, \Sigma_r, \Sigma_\ell \rangle$ felbontásra nézve, ha V két diszjunkt V^0 és V^1 halmazra osztható, úgy hogy a P -beli szabályok az alábbi alakúak:

- $X \rightarrow \varepsilon$;
- $X \rightarrow aY$, ahol
 ha $X \in V^0$, **akkor** $a \in \Sigma_\ell$ és $Y \in V^0$;
- $X \rightarrow aYbZ$, ahol $a \in \Sigma_c$, $b \in \Sigma_r$ és $Y \in V^0$, továbbá
 ha $X \in V^0$, **akkor** $Z \in V^0$.

Tétel

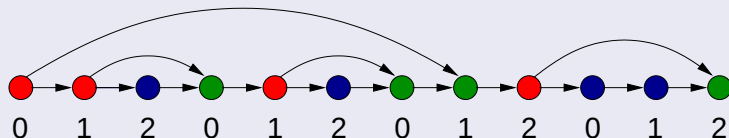
$L \hat{\Sigma}\text{-VPL} \Leftrightarrow L$ generálható átlátható verem nyelvtannal.

Egymásbaágyazott szavak (nested words) I.

- Az átlátható veremautomaták és nyelvek elmélete átfogalmazható segítségükkel.
- A call és return pozíciók egyeztetéséből adódó hierarchikus viszonyokat tegyük explicitté.
- Legyen Σ egy ábécé (most nem osztjuk fel.)
- Beágyazott szó =
 - szokásos lineáris rendezés + helyesen egymásbaágyazott élek
 - a pozíciók Σ betűivel címkézettek

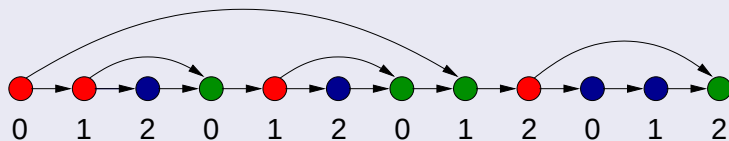
Egymásbaágyazott szavak (nested words) I.

- Az átlátható veremautomaták és nyelvek elmélete átfogalmazható segítségükkel.
- A call és return pozíciók egyeztetéséből adódó hierarchikus viszonyokat tegyük explicitté.
- Legyen Σ egy ábécé (most nem osztjuk fel.)
- Beágyazott szó =
 - szokásos lineáris rendezés + helyesen egymásbaágyazott élek
 - a pozíciók Σ betűivel címkézettek
- Például:



- A pozíciók osztályozása: **call**, **return**, **local**

A pozíciók osztályozása



- A call pozíciók, azok melyeknek mind lineáris mind hierarchikus rákövetkezőjük van.
- A return pozíciók, azok melyeknek mind lineáris mind hierarchikus megelőzőjük van.
- Lokális pozíciók: a többi.

Automaták egymásbaágyazott szavakon: Nested word automata

Nested word automata

- NWA (Σ felett): $\mathcal{A} = (Q, Q_{in}, \delta, Q_F)$, ahol
- Q : állapothalmaz,
- Q_{in}, Q_F : kezdő és végállapotok,
- $\delta = \langle \delta_c, \delta_r, \delta_\ell \rangle$, ahol
 - $\delta_c : Q \times \Sigma \rightarrow 2^Q$;
 - $\delta_r : Q \times \Sigma \times \mathbf{Q} \rightarrow 2^Q$;
 - $\delta_\ell : Q \times \Sigma \rightarrow 2^Q$.
- Return pozíció olvasásakor a NWA, az előző állapoton kívül a hierarchikus megelőző pozíció olvasásakor érvényes állapotot is megkapja.

Egymásbaágyazott szavak $\equiv$ átlátható verem szavak

Egymásbaágyazott szavak kódolása

Minden Σ feletti egymásbaágyazott szó kódolható átlátható verem szóval a $\hat{\Sigma}' = \langle \Sigma'_c, \Sigma'_r, \Sigma'_\ell \rangle$ ábécé felett, ahol

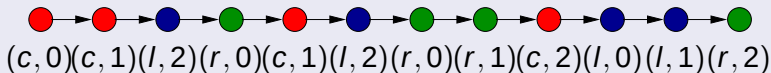
- $\Sigma'_c = \{(c, \sigma) | \sigma \in \Sigma\}$
- $\Sigma'_r = \{(r, \sigma) | \sigma \in \Sigma\}$
- $\Sigma'_\ell = \{(l, \sigma) | \sigma \in \Sigma\}$

Egymásbaágyazott szavak $\equiv$ átlátható verem szavak

Egymásbaágyazott szavak kódolása

Minden Σ feletti egymásbaágyazott szó kódolható átlátható verem szóval a $\hat{\Sigma}' = \langle \Sigma'_c, \Sigma'_r, \Sigma'_\ell \rangle$ ábécé felett, ahol

- $\Sigma'_c = \{(c, \sigma) | \sigma \in \Sigma\}$
- $\Sigma'_r = \{(r, \sigma) | \sigma \in \Sigma\}$
- $\Sigma'_\ell = \{(l, \sigma) | \sigma \in \Sigma\}$

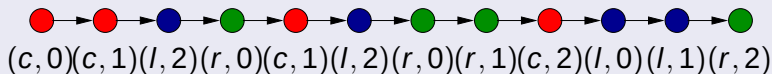


Egymásbaágyazott szavak $\equiv$ átlátható verem szavak

Egymásbaágyazott szavak kódolása

Minden Σ feletti egymásbaágyazott szó kódolható átlátható verem szóval a $\hat{\Sigma}' = \langle \Sigma'_c, \Sigma'_r, \Sigma'_\ell \rangle$ ábécé felett, ahol

- $\Sigma'_c = \{(c, \sigma) | \sigma \in \Sigma\}$
- $\Sigma'_r = \{(r, \sigma) | \sigma \in \Sigma\}$
- $\Sigma'_\ell = \{(l, \sigma) | \sigma \in \Sigma\}$



Átlátható verem szavak kódolása

Minden $\hat{\Sigma}'$ feletti átlátható verem szónak, *melyben nincsenek pár nélküli call és return betűk*, megfeleltethető egy $\Sigma'_c \cup \Sigma'_r \cup \Sigma'_\ell$ feletti egymásbaágyazott szó.

NWA-k és VPA-k ekvivalenciája

Tétel

L reguláris egymásbaágyazott szó nyelv $\Leftrightarrow \eta(L)$ átlátható verem nyelv.

NWA-k és VPA-k ekvivalenciája

Tétel

L reguláris egymásbaágyazott szó nyelv $\Leftrightarrow \eta(L)$ átlátható verem nyelv.

Megjegyzések

NWA-k és VPA-k ekvivalenciája

Tétel

L reguláris egymásbaágyazott szó nyelv $\Leftrightarrow \eta(L)$ átlátható verem nyelv.

Megjegyzések

- Így minden átlátható veremautomatára és nyelvre bizonyított eredmény igaz lesz egymásbaágyazott szó automatákra és nyelvekre is.

Tétel

L reguláris egymásbaágyazott szó nyelv $\Leftrightarrow \eta(L)$ átlátható verem nyelv.

Megjegyzések

- Így minden átlátható veremautomatára és nyelvre bizonyított eredmény igaz lesz egymásbaágyazott szó automatákra és nyelvekre is.
- Visszafele: csak azok az átlátható verem szavak felelnek meg egymásbaágyazott szavaknak, melyekben nincsenek pár nélküli call és return betűk.

Tétel

L reguláris egymásbaágyazott szó nyelv $\Leftrightarrow \eta(L)$ átlátható verem nyelv.

Megjegyzések

- Így minden átlátható veremautomatára és nyelvre bizonyított eredmény igaz lesz egymásbaágyazott szó automatákra és nyelvekre is.
- Visszafele: csak azok az átlátható verem szavak felelnek meg egymásbaágyazott szavaknak, melyekben nincsenek pár nélküli call és return betűk.
- De ez kiküszöbölhető lenne, a magányos pozícióknak egy a szó utolsó betűje utáni, illetve első betűje előtti „elképzelt pozíció”-hoz való társítással.

Tétel

L reguláris egymásbaágyazott szó nyelv $\Leftrightarrow \eta(L)$ átlátható verem nyelv.

Megjegyzések

- Így minden átlátható veremautomatára és nyelvre bizonyított eredmény igaz lesz egymásbaágyazott szó automatákra és nyelvekre is.
- Visszafele: csak azok az átlátható verem szavak felelnek meg egymásbaágyazott szavaknak, melyekben nincsenek pár nélküli call és return betűk.
- De ez kiküszöbölhető lenne, a magányos pozícióknak egy a szó utolsó betűje utáni, illetve első betűje előtti „elképzelt pozíció”-hoz való társítással.
- Enélkül azonban az elmélet elegánsabb és egyszerűbb.

- Általánosítás végtelen szavakra: ω -VPL [AM STOCS04]

További eredmények, kutatási területek

- Általánosítás végtelen szavakra: ω -VPL [AM STOCS04]
- Kongruenciák, Myhill-Nerode típusú tételek, minimalizáció [AKMV ICALP05]

További eredmények, kutatási területek

- Általánosítás végtelen szavakra: ω -VPL [AM STOCS04]
- Kongruenciák, Myhill-Nerode típusú tételek, minimalizáció [AKMV ICALP05]
- Nested Word Temporal Logic: NWTL $\equiv$ FO [Imm LATA07], LICS 2007

További eredmények, kutatási területek

- Általánosítás végtelen szavakra: ω -VPL [AM STOCS04]
- Kongruenciák, Myhill-Nerode típusú tételek, minimalizáció [AKMV ICALP05]
- Nested Word Temporal Logic: NWTL $\equiv$ FO [Imm LATA07], LICS 2007
- Nested Trees [ACM CAV06]

További eredmények, kutatási területek

- Általánosítás végtelen szavakra: ω -VPL [AM STOCS04]
- Kongruenciák, Myhill-Nerode típusú tételek, minimalizáció [AKMV ICALP05]
- Nested Word Temporal Logic: NWTL $\equiv$ FO [Imm LATA07], LICS 2007
- Nested Trees [ACM CAV06]
- Más modellek: Visibly Stack Automata [TVOwww], Visibly Basic Process Algebra, Visibly One-Counter Automata [Srba CSL06]

További eredmények, kutatási területek

- Általánosítás végtelen szavakra: ω -VPL [AM STOCS04]
- Kongruenciák, Myhill-Nerode típusú tételek, minimalizáció [AKMV ICALP05]
- Nested Word Temporal Logic: NWTL $\equiv$ FO [Imm LATA07], LICS 2007
- Nested Trees [ACM CAV06]
- Más modellek: Visibly Stack Automata [TVOWww], Visibly Basic Process Algebra, Visibly One-Counter Automata [Srba CSL06]
- Visibly Pushdown Games [LMS FSTTCS04]

További eredmények, kutatási területek

- Általánosítás végtelen szavakra: ω -VPL [AM STOCS04]
- Kongruenciák, Myhill-Nerode típusú tételek, minimalizáció [AKMV ICALP05]
- Nested Word Temporal Logic: NWTL $\equiv$ FO [Imm LATA07], LICS 2007
- Nested Trees [ACM CAV06]
- Más modellek: Visibly Stack Automata [TVOWww], Visibly Basic Process Algebra, Visibly One-Counter Automata [Srba CSL06]
- Visibly Pushdown Games [LMS FSTTCS04]
- Alkalmazások
 - streaming XML [KMV WWW07]
 - modellellenőrzés, pl. CARET
 - (természetes) nyelvészet (?)

Hivatkozások I.



[Alur] Rajeev Alur publikációi és előadásai.

<http://www.cis.upenn.edu/~alur/onlinepub.html>



[AM STOCS04] R. Alur and P. Madhusudan. Visibly pushdown languages. *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, STOC 2004, Chicago, IL, USA, June 13-16, 2004*, 202–211.



[AKMV ICALP05] Rajeev Alur, Viraj Kumar, P. Madhusudan, Mahesh Viswanathan. Congruences for Visibly Pushdown Languages. in proc.: *ICALP 2005, LNCS 3580*, 1102-1114



[AM DLT06] Rajeev Alur, P. Madhusudan. Adding Nesting Structure to Words. in proc: *DLT2006, LNCS 4036*, 1-13



[ACM CAV06] Rajeev Alur, Swarat Chaudhuri, P. Madhusudan. Languages of Nested Trees. in proc: *CAV 2006 LNCS 4144*, 329-342

Hivatkozások II.



[KMV WWW07] V. Kumar, P. Madhusudan, M. Viswanathan. Visibly Pushdown Automata for Streaming XML, in: *Proceedings of the Sixteenth International World Wide Web Conference (WWW2007) May 8-12, 2007, Banff, Alberta, CANADA*, 1053-1062



[Imm LATA07] Niel Immerman. Descriptive Complexity and Nested Words, invited survey talk given at LATA 2007, Tarragona, Spain, March, 2007.



[LMS FSTTCS04] Christof Löding, P. Madhusudan, Olivier Serre: Visibly Pushdown Games. in proc: *FSTTCS 2004, LNCS 3328*, 408-420



[Srba CSL06] Jirí Srba. Visibly Pushdown Automata: From Language Equivalence to Simulation and Bisimulation. in proc: *CSL 2006, LNCS 4206*: 89-103



[TVOWww] Nguyen Van Tang -Mizuhito Ogawa, Visibly Stack Automata
www.jaist.ac.jp/joint-workshop/verite/VSA-VERITE.pdf