Síla pozičních ekogramatických systémů

Miroslav Langer

IT4Innovation a Slezská univerzita v Opavě
Bezručovo nám. 13, 746 01 Opava
Email: miroslav.langer@fpf.slu.cz

Abstrakt

Poziční ekogramatické systémy (zkráceně PEG systémy) byly zavedeny v Langer (2005a). V tomto příspěvku shrnujeme pohled na generativní sílu PEG systémů. Zabýváme se generativní silou PEG systémů jak v kontextu jiných gramatik a gramatických systémů, tak s ohledem na vlastnosti PEG systémů samotných, čili na generativní sílu s ohledem na počet agentů v prostředí a počet typů agentů v definici PEG systému.

1 Úvod

V tomto příspěvku se budeme věnovat pozičním ekogramatickým systémům (zkráceně PEG systémy) (viz Langer (2005a)). Jedná se o model multiagentového systému zavedeného v roce 2005. Agenti jsou vsazeni do vyvíjejícího se prostředí, přičemž klademe důraz na fyzickou přítomnost agenta v prostředí. Agentova přítomnost v prostředí je dána speciálním identifikátorem. Motivace k zavedení našeho modelu byla inspirována dvěma již zavedenými modely, ekogramatickými systémy a PM-koloniemi.

Ekogramatické systémy byly zavedeny v Csuhaj-Varjú a spol. (1994) ve snaze popsat vzájemné působení mezi komunitou agentů a vyvíjejícím se prostředím, ve kterém žijí. Prostředí, 0L schéma, je dvojice $E = (V_E, P)$, kde V_E je abeceda a P je konečná množina pravidel typu $a \to \alpha, a \in V_E, \alpha \in V_E^*$. Každý agent ekogramatického systému je dán svým 0L schématem, konečnou množinou akčních pravidel a konečnou množinou evolučních pravidel. Prostředí i agenti se tedy mohou vyvíjet na základě vlastních evolučních pravidel. Vzájemná interakce mezi prostředím a agenty je zachycena pomocí funkcí φ a ψ. Funkce φ vybírá evoluční pravidla agentů na základě stavu prostředí, podle kterých dochází k vývoji agenta. Funkce ψ vybírá akční pravidla agenta na základě stavu agenta, pomocí kterých agent působí v prostředí. Vývoj agenta je tudíž ovlivněn stavem prostředí a zásahy agenta v prostředí jsou ovlivněny jeho stavem.

PM-kolonie na rozdíl od ekogramatických systémů nemají vyvíjející se prostředí. Prostředí je tvořeno řetězcem nad abecedou V a je ohraničeno speciálními hraničními symboly. Společenství agentů žijících v prostředí jej může ovlivňovat svými akčními pravidly. Akční pravidla mají formu $(a, A, b) \to \alpha$, kde A je identifikátor agenta, symboly $a, b \in V$ symbolizují kontext agenta A. Agenti působí v prostředí pomocí svých akčních pravidel. Agent se může v prostředí pohybovat, může nahradit, vložit nebo vymazat znak prostředí nebo může zaniknout. Interakce mezi prostředím a agenty je zachycena v pravidlech agentů. Na základě kontextu, čili stavu prostředí v jeho bezprostředním okolí, agent může buď působit na prostředí, nebo může zaniknout.

Na rozdíl od PM-kolonií (Martin-Vide a Păun (1998)) mají PEG systémy vyvíjející se prostředí, 0L schéma, a agenti PEG systémů mají pouze jednostranný kontext. Agenti PEG systémů mohou do prostředí vkládat víc než jeden symbol. Na rozdíl od agentů ekogramatických systémů jsou přítomni přímo v prostředí a jejich přítomnost je dána speciálním symbolem. Sami se nevyvíjejí a prostředí modifikují pomocí svých pravidel na základě svého kontextu. Taktéž nemohou, na rozdíl od agentů ekogramatických systémů, ve dvou po sobě jdoucích derivačních krocích působit na symboly vzdálené od sebe více jak jeden symbol. To nám dovoluje studovat lokální změny prostředí, které agenti způsobují. Přesné určení pozice agenta speciálním symbolem nám do jisté míry dovoluje predikovat jejich chování a taktéž řídit evoluci prostředí.

Motivací pro zavedení PEG systémů je snaha zachytit interakci mezi vyvíjejícím se prostředím a komunitou agentů žijících v tomto prostředí, přičemž klademe důraz na fyzickou přítomnost agenta v prostředí. Inspiraci můžeme hledat kdekoliv, kde komunita agentů působí na vyvíjející se prostředí. Primárním zájmem je studium lokálních změn prostředí v okolí agenta. Vývoj agenta je potlačen, je nedůležitý nebo jej lze popsat pomocí typů agentů. Takovým systémem může být kupříkladu živá tkáň napadená virovým onemocněním.

Inspiraci můžeme hledat například i na listech broskvoní napadených kadeřavostí. Působením parazitických organizmů dochází k degeneraci části rostliny. Změny jsou pouze lokálního charakteru, v místech napadení. Růst rostlin lze navíc popsat Lindenmayerovými systémy, které tvoří prostředí PEG systémů. Působení agentů pak může zapříčinit lokální deformace, změnu vývoje prostředí.

2 Poziční ekogramatické systémy

V této kapitole uvedeme definice pozičních ekogramatických systémů.

Prostředím PEG systémů je 0L schéma. Agenti jsou přítomni v prostředí a jejich umístění je dáno identifikátorem. Pravidla agentů jsou typu $a[1] \to \alpha$ nebo $[1]a \to \alpha$, kde $[1]$ je identifikátor agenta, a je symbol prostředí a α je řetězec složený z libovolného počtu znaků prostředí a identifikátorů agentů v libovolném pořadí. Z toho plyne, že agent může modifikovat svůj kontext, vkládat libovolný počet znaků prostředí, vytvořit libovolný počet agentů, transformovat se na jiný typ agenta, případně se vymazat. Každý agent v každém derivačním kroku musí aplikovat jedno ze svých pravidel, jinak je derivace zastavena, stejně tak, pokud se dva agenti pokusí modifikovat týž symbol. Tyto dvě vlastnosti se v průběhu výzkumu ukázaly být velmi užitečné. Jsou velmi silným nástrojem pro simulaci jiných systémů a gramatik.

Definice 1 *Poziční ekogramatický systém (zkráceně PEG systém) stupně m, $m \geq 1$, je $(m+3)$ - tice $\Sigma = (V_E, N_B, E, B_1, \ldots, B_m)$, kde*

- *V_E je konečná neprázdná abeceda prostředí,*
- *$N_B = \{[j] : 1 \leq j \leq m\}$ je množina identifikátorů agentů, $[j]$ určuje pozici j. typu agenta v prostředí,*
- *$E = (V_E, P_E)$ je $0L$ schéma s abecedou V_E a vývojových pravidel P_E – prostředí,*
- *$B_j = ([j], Q_j)$, je j. typ agenta pro $1 \leq j \leq m$ a Q_j množina pravidel typu $a[j]b \to u$, kde $ab \in V_E$ je symbol po pravé nebo levé straně agenta $[j]$ a $u \in (V_E \cup N_B)^*$.*

V definici pozičního ekogramatického systému vyžadujeme přítomnost alespoň jednoho typu agenta.

Vnitřní agent může vzniknout nebo zaniknout prostřednictvím pravidel a jeho přítomnost v prostředí je dána symbolem z množiny N_B. V prostředí se může vyskytovat několik kopií jednoho typu agenta. Agenti pracují paralelně. Zbylé symboly, které nebyly přepsány agenty, jsou paralelně přepsány pravidly prostředí.

Definice 2 *Konfigurace PEG systému $\Sigma = (V_E, N_B, E, B_1, \ldots, B_m)$ je řetězec v nad $(V_E \cup N_B)$. Počáteční konfiguraci budeme nazývat axiom.*

Definice 3 *Průmět slova $w \in (V \cup N_B)^*$ do množiny V_E je morfismus $\gamma(a) = a$ pro $a \in V_E$ a $\gamma(b) = \varepsilon$ pro $b \in N_B$.*

Definice 4 *Průmět slova $w \in (V \cup N_B)^*$ do množiny N_B je morfismus $\overline{\gamma}(a) = \varepsilon$ pro $a \in V_E$ a $\overline{\gamma}(b) = b$ pro $b \in N_B$.*

Definice 5 *Derivační krok PEG systému $\Sigma = (V_E, N_B, E, B_1, \ldots, B_m)$ je binární relace $\Longrightarrow_\Sigma$ nad $(V_E \cup N_B)^*$ taková, že $w \Longrightarrow_\Sigma w'$ tehdy a jen tehdy, když*

- *$w = \alpha_0 a_1 [j_1] b_1 \alpha_1 \ldots \alpha_{m-1} a_m [j_m] b_m \alpha_m$, kde $\alpha_k \in V_E^*$ pro $0 \leq k \leq m$ a $a_k b_k \in V_E, [j_k] \in N_B, 1 \leq k \leq m$,*
- *$w' = \alpha'_0 \beta_1 \alpha'_1 \ldots \alpha'_{m-1} \beta_m \alpha'_m$, kde $a_k[j_k]b_k \to \beta_k \in Q_k$ pro $1 \leq k \leq m$ a $\alpha_k \Rightarrow_E \alpha'_k$ pro $0 \leq k \leq m$.*

Zápisem $\Longrightarrow^*$ označujeme reflexivní a tranzitivní uzávěr relace $\Longrightarrow$.

V každém kroku derivace agent přepíše jeden symbol po své levé či pravé straně. Ostatní symboly se přepíší podle pravidel 0L prostředí. Všichni agenti pracují paralelně. V případě konfliktu nebo neaktivity kteréhokoliv agenta je odvození zablokováno. Konfliktem rozumíme případ, kdy se dva agenti snaží aplikovat své pravidlo na týž symbol. Neaktivitou potom stav, kdy agent nemůže aplikovat žádné ze svých pravidel.

Jazyk definovaný pozičním ekogramatickým systémem je množina všech slov, která mohou být odvozena z axiomu a ze kterých vynecháme identifikátory vnitřních agentů.

Definice 6 *Jazyk definovaný PEG systémem $\Sigma = (V_E, N_B, E, B_1, \ldots, B_m)$ a axiomem $w \in (V_E \cup N_B)^*$ je množina slov*

$$L(\Sigma, w) = \{\gamma(u) \in V_E^* : w \Rightarrow_\Sigma^* u, u \in (V_E \cup N_B)^*\}.$$

Třídu všech jazyků definovanou PEG systémy (PEG jazyky) budeme označovat $\mathscr{L}(PEG)$.

3 Síla PEG systémů

V průběhu výzkumu PEG systémů jsme se zabývali stanovením generativní síly, jednak v kontextu již zavedených gramatik a systémů, tak v kontextu vlastností PEG systémů, jako je počet agentů v prostředí a počet typů agentů v definici PEG systému. V této kapitole předkládáme nejdůležitější výsledky. Ukážeme, že PEG systémy nejsou výpočetně kompletní a stanovíme hierarchii PEG systémů v závislosti na počtu agentů v prostředí a v závislosti na počtu typů agentů v prostředí.

3.1 Horní hranice generativní síly PEG systémů

V Langer a Kelemenová (2013) jsme ukázali, že třída generovaná čistými gramatikami typu 0 je vlastní podtřídou PEG jazyků. Agent pohybující se v prostředí může připomínat hlavu Turingova stroje a vyvolat tak pocit, že jsme schopni generovat i všechny rekurzivně vyčíslitelné jazyky. V následující větě ukážeme, že tomu tak není, čímž ukážeme horní hranici generativní síly PEG systémů.

Věta 1 *PEG systémy nejsou výpočetně kompletní*
$\mathscr{L}(PEG) \subsetneq \mathscr{L}(RE)$.

Důkaz: Mějme jazyk $L = L(\Sigma, \alpha)$ generovaný PEG systémem Σ s axiomem α. Nejprve ukážeme, že pro nějaké $k \geq 1$ takové, že pro každé $w \in L \setminus \{\alpha\}$ existuje $u \in L$ takové, že $|w| \leq k|u|$.

Mějme v PEG systému Σ odvození: $\alpha \Rightarrow^* \bar{u} \Rightarrow \bar{w}$ a $w = \gamma(\bar{w}),\ u = \gamma(\bar{u})$.
Poslední krok odvození zajistí, že pro každý symbol z N_B existuje v řetězci $\bar{u}$ alespoň jeden symbol z V_E, čili $|u| \leq |\bar{u}| \leq 2|u|$, kde $u = \gamma(\bar{u})$. Označme p_E délku nejdelší pravé strany pravidla prostředí, $p_E = \max\{|\beta| : a \to \beta \in P_E,\ a \in V_E\}$ a q označme délku nejdelší pravé strany pravidla agenta bez identifikátorů agentů, $q = \max\{|\gamma(\beta')| : a[i]b \to \beta' \in Q_i, ab \in V_E, 1 \leq i \leq m\}$. Pak $|w| \leq |\bar{w}| \leq \max\{p_E, p_A\}|\bar{u}| \leq 2\max\{p_E, p_A\}|u|$, tudíž $k = 2\max\{p_E, p_A\}$.

Z poslední nerovnosti plyne, že například jazyk $L = \{a^{n^n} : n \geq 0\}$ nemůže být generovaný žádným PEG systémem, a PEG systémy tudíž nejsou výpočetně kompletní.

$\square$

3.2 Generativní síla podle vlastností PEG systémů

V pozičních ekogramatických systémech je pozice agentů v prostředí dána speciálními symboly. Tyto komponenty způsobují v paralelním prostředí lokální změny v konfiguraci systému. Umístění agenta v prostředí a jeho kontext přímo ovlivňuje generativní sílu PEG systému.

V této podkapitole rozdělíme PEG systémy s ohledem na maximální počet typů agentů v definici PEG systému a s ohledem na maximální počet agentů v prostředí. Ukážeme, že v obou případech toto rozdělení určuje nekonečnou hierarchii PEG jazyků.

3.2.1 Hierarchie na základě počtu agentů v prostředí

V této podkapitole se budeme věnovat rozdělení PEG systémů na základě počtu agentů přítomných v prostředí. Budeme uvažovat PEG systémy s maximálně n agenty libovolného typu v prostředí v každém derivačním kroku. Jinými slovy, zajímá nás celkový počet identifikátorů agentů v prostředí.

Definice 7 *PEG systém $\Sigma = (V_E, N_B, E, B_1, \ldots, B_m)$ s axiomem $w,\ w \in (V_E \cup N_B)^+$ je PEG systém s indexem n, značíme $(\Sigma, w) \in PEG_n$, jestliže $|u|_{N_B} \leq n$ pro všechna $u \in (V_E \cup N_B)^*$ taková, že $w \Rightarrow^*_\Sigma u$.*

Jazyk L je PEG jazyk s indexem n, $L \in \mathscr{L}(PEG_n)$, jestliže existuje PEG_n systém Σ s axiomem w takový, že $L = L(\Sigma, w)$ a pro žádný PEG_m systém Σ' a w' $m < n$ neplatí $L = L(\Sigma', w')$.

Příklad 1 Mějme jazyk $L_1 = \{a^i b^i c^i : i \geq 1\}$ a PEG systém $\Sigma_1 = (\{a, b, c\}, \{[1]\}, E, B_1)$, kde:

- $E = (\{a, b, c\}, \{a \to a, b \to b, c \to c\})$,
- $B_1 = ([1], \{a[1] \to aa[1]b, [1]c \to [1]cc\})$.

Nechť $w_1 = a[1]b[1]c$ je axiom systému Σ_1. Pak
$a[1]b[1]c \Rightarrow aa[1]bb[1]cc \Rightarrow^{n-2} a^n[1]b^n[1]c^n \Rightarrow a^n a[1]bb^n[1]cc^n \Rightarrow \ldots$

Protože $L(\Sigma_1, w_1) = L_1$ a (Σ_1, w_1) je s indexem 2, pak $(\Sigma_1, w_1) \in PEG_2$. Tudíž $L_1 \in \mathscr{L}(PEG_t)$ pro $t \geq 2$.

Abychom dokázali, že $L_1 \in \mathscr{L}(PEG_2)$, ověříme, že L_1 nenáleží do $\mathscr{L}(PEG_1)$, tudíž neexistuje PEG systém Σ s axiomem w s indexem 1 takový, že $L(\Sigma, w) = L_1$.

Předpokládejme naopak, že L_1 náleží do $\mathscr{L}(PEG_1)$ a PEG_1 systém $\Sigma_0 = (\{a, b, c\}, \{[1]\}, E, B_1, \ldots, B_m)$ s axiomem $w_0 = u_0[1]v_0$ generuje L_1. Čtenář si může ověřit, že $E = (\{a, b, c\}, P)$ v Σ_0 je deterministický a $P = \{a \to a, b \to b, c \to c\}$. Všechny změny v prostředí tedy zajišťují agenti.

Nechť je $u[j]v, 1 \leq j \leq m$ slovo odvozené ze Σ_0 z axiomu a $uv = a^t b^t c^t$ pro nějaké $t \geq 1$. Předpokládejme, že v dalším derivačním kroku je použito pravidlo $x[j]y \to z$, kde $x, y \in \{a, b, c, \varepsilon\}, x = \mathrm{suf}(u), y = \mathrm{pref}(v)$, jeden ze symbolů x, y je prázdný a v z je přítomen libovolný agent [i]. Agent $[j]$ se může nacházet v následujících pozicích, pro které platí:

- $y = a$. Pak musí být v následujícím kroku použito pravidlo typu $a[j] \to a^r[i]$ nebo $[j]a \to a^r[i]$ pro nějaké $r \neq 1$. (Pozice a typ agenta $[i]$ na pravé straně použitého pravidla jsou libovolné.) Tudíž pro $u[j]v \Rightarrow w$ platí $|w|_a \neq |w|_b$ a $\gamma(w)$ nenáleží do L_1.
- $y = b$. Pak aplikací $u[j]v \Rightarrow w$ získáme $|w|_c < |w|_a$ a $|w|_c < |w|_b$. Čili $\gamma(w)$ nenáleží do L_1.
- $y = c$. Pak $u[j]v \Rightarrow w$, kde $|w|_a < |w|_b$, nebo $|w|_a < |w|_c$. Čili $\gamma(w)$ není slovo z jazyka L_1.

Ve všech možných případe dostaneme spor s předpokladem, že L_1 patří do $\mathscr{L}(PEG_1)$, tudíž $L_1 \in \mathscr{L}(PEG_2)$.

Na základě této úvahy můžeme vyslovit následující teorém.

Věta 2 *Třída PEG jazyků tvoří, s ohledem na index PEG systému, nekonečnou hierarchii: $\mathscr{L}(PEG_n) \subset \mathscr{L}(PEG_{n+1})$ pro $n \geq 1$ (tj. vlastní podmnožina).*

Důkaz: Zjevně $\mathscr{L}(PEG_n) \subseteq \mathscr{L}(PEG_{n+1})$ pro $n \geq 1$.

Uvažujme jazyk

$$L_{n+1} = \{a_1^i a_2^i \ldots a_{2n}^i a_{2n+1}^i : i \geq 1\}.$$

Ukážeme, že $L_{n+1} \in \mathscr{L}(PEG_{n+1}) - \mathscr{L}(PEG_n)$.

Nejprve sestrojíme PEG systém Σ_{n+1}, který z axiomu w generuje jazyk L_{n+1} a v jehož prostředí se vyskytuje $n+1$ kopií agenta [1]: $\Sigma_{n+1} = (V_E, \{[1]\}, E, B)$, kde

- $V_E = \{a_1, a_2, \ldots, a_{2n+1}\}$,
- $E = (V_E, \{a_1 \to a_1, a_2 \to a_2, \ldots, a_{2n+1} \to a_{2n+1}\})$,
- $B = (\{[1]\}, \{a_{2k-1}[1] \to a_{2k-1}^2[1]a_{2k} : 1 \leq k \leq n\} \cup \{[1]a_{2n+1} \to [1]a_{2n+1}^2\})$.

Axiom systému PEG_{n+1} je $w = a_1[1]a_2a_3[1]a_4 \ldots a_{2n-1}[1]a_{2n}[1]a_{2n+1}$.

Odvození probíhá následovně:
$a_1[1]a_2 \ldots a_{2n-1}[1]a_{2n}[1]a_{2n+1} \Rightarrow$
$a_1^2[1]a_2^2 \ldots a_{2n-1}^2[1]a_{2n}^2[1]a_{2n+1}^2 \Rightarrow$
$a_1^3[1]a_3^3 \ldots a_{2n-1}^3[1]a_{2n}^3[1]a_{2n+1}^3 \Rightarrow$
$a_1^4[1]a_4^4 \ldots a_{2n-1}^4[1]a_{2n}^4[1]a_{2n+1}^4 \Rightarrow \ldots$

Tudíž $L(\Sigma_{n+1}, w) = \{a_1^i a_2^i \ldots a_{2n}^i a_{2n+1}^i : i \geq 1\}$.

Protože $L(\Sigma_{n+1}, w) = L_{n+1}$ a Σ_{n+1} je indexu $n+1$, tj. $L(\Sigma_{n+1}, w) \in PEG_{n+1}$. Pak $L_{n+1} \in \mathscr{L}(PEG_t)$, pro nějaké $t,\ t \leq n+1$.

Abychom dokázali, že $L_{n+1} \in \mathscr{L}(PEG_{n+1})$, prověříme, že $L_{n+1} \notin \mathscr{L}(PEG_n)$, tj. neexistuje PEG systém Σ s indexem n a axiom w takový, že $L(\Sigma, w) = L_{n+1}$.

Předpokládejme naopak, že L_{n+1} patří do $\mathscr{L}(PEG_n)$ a PEG_n systém $\Sigma_n = (V_E, N_B, E, B_1, \ldots, B_m)$ s axiomem w_n generuje L_{n+1}. Ověříme, že $E = (V_E, P)$ ze Σ_n je deterministický a $P = \{a_i \to a_i : 1 \leq i \leq 2n+1\}$.

Ze struktury jazyka L_{n+1} vyplývá, že pro $a_i \to \alpha \in P$ je $\alpha = a_i^s$ pro nějaké s. Předpokládejme, že P není deterministické a $a_i \to a_i^j, a_i \to a_i^k \in P$, pro $j < k$. Nechť je v odvození $u = a_1^r a_2^r \ldots a_{2n}^r a_{2n+1}^r \in L_{n+1}$, pro nějaké r použito pravidlo $a_i \to a_i^j$. Pak ale i slovo $v = a_1^r a_2^r \ldots a_i^{r+k-j} \ldots a_{2n}^r a_{2n+1}^r \in L(\Sigma_n, w)$, ale $v \notin L_{n+1}$.

Zjevně tedy $a_i \to a_i^j, 1 \leq i \leq 2n+1$ v deterministickém P platí $j = 1$.

Tudíž všechny změny prostředí vykonávají agenti. Nechť $u[j]v \to w$ náleží B_j. pak ze struktury jazyka L_{n+1} vyplývá, že $|\gamma(w) \cap V_E| \leq 2$. Tím pro odvození $x \Rightarrow y$, kde x obsahuje nejvýše n symbolů z N_B platí, $|x|_{a_i} = |y|_{a_i}$, pro alespoň jedno i. Tudíž buď $\gamma(x)$, nebo $\gamma(y)$ nepatří do L_{n+1} a $L(\Sigma_n, w_n) \neq L_{n+1}$.

3.2.2 Hierarchie na základě počtu typů agentů

V této podkapitole se budeme zabývat rozdělením pozičních ekogramatických systémů na základě počtu typů agentů v definici PEG systému. Ukážeme, že počet typů agentů v definici PEG systému (čili kardinalita množiny N_B) ovlivňuje generativní sílu PEG systému. Abychom ukázali, že takováto hierarchie existuje, musíme najít jazyk L, který není $0L$ jazykem a pro který platí, že L nelze generovat PEG systémem s n typy agentů, ale lze jej generovat PEG systémem s $n+1$ typy agentů. Zcela nepřekvapivě je struktura takového jazyka podobná struktuře jazyka použitého v předešlém důkazu.

Definice 8 *PEG systém s nejvýše n typy agentů je takový PEG systém $\Sigma = (V_E, N_B, E, B_1, \ldots, B_m)$, kde $n \geq |N_B|$, a značíme jej PEG^n.*

Třídu jazyků generovanou PEG^n systémem označíme $\mathscr{L}(PEG^n)$.

Ukažme si nejprve jednoduchý příklad.

Příklad 2 Uvažujme jazyk $L^2 = \{a^i b^i a^j b^j a^k : i \geq 1, j = 2i, k = 3i\}$ a PEG systém $\Sigma^2 = (\{a, b\}, \{[1], [2]\}, E, B_1, B_2)$, kde:

- $E = (\{a, b\}, \{a \to a, b \to b\})$,
- $B_1 = ([1], \{[1]a \to a[1]ab, [1]b \to aa[1]bbb\})$,
- $B_2 = ([2], \{a[2] \to aaaa[2]\})$.

Nechť axiom systému je $w^2 = [1]abaaa[1]bbbaaa[2]$, pak
$[1]abaaa[1]bbbaaa[2] \Rightarrow$
$a[1]abbaaaa[1]bbbbaaaaaaa[2] \Rightarrow$
$aa[1]ab^3a^6[1]b^6a^9[2] \Rightarrow$
$a^3[1]ab^4a^8[1]b^8a^{12}[2] \Rightarrow a^4[1]ab^5a^{10}[1]b^{10}a^{15}[2] \ldots$,
tudíž $L(\Sigma^2, w^2) = L^2$, Σ má dva typy agentů, a proto $L^2 \in \mathscr{L}(PEG^t)$ pro $t \geq 2$.

Abychom ukázali, že $L^2 \in \mathscr{L}(PEG^2)$, ověříme, že L^2 nepatří do $\mathscr{L}(PEG^1)$, tj. neexistuje PEG systém Σ s $|N_B| = 1$ a axiom w, takové, že $L(\Sigma, w) = L^2$. Předpokládejme naopak, že L^2 patří do $\mathscr{L}(PEG^1)$ a PEG^1 systém $\Sigma^1 = (\{a, b\}, \{[1]\}, E, B_1)$ s axiomem $w_0 = u_0[1]v_0[1]x_0[1]y_0$ generuje L^2. Laskavý čtenář si snadno ověří, že $E = (\{a, b\}, P)$ systému Σ^1 je deterministické a $P = \{a \to a, b \to b\}$. Tudíž že všechny změny prostředí zajišťují agenti.

Nechť jsou $u[1]v[1]x[1]y$ slova odvozená z axiomu v Σ^1 a $uvxy = a^t b^t a^{2t} b^{2t} a^{3t}$ pro nějaké $t \geq 1$. Podle definice PEG systému může každý agent reagovat pouze na symbol po své levé nebo pravé straně, nikoli s oběma současně. Toto omezení nám dovolí uvažovat následující případy:

- Nechť $u = a^{t-1}$, $v = ab^t a^{2t}$, $x = b^{2t} a^{3t}$, $y = \varepsilon$, pak Q_1 musí obsahovat pravidla $[1]a \to a[1]ab, [1]b \to aa[1]bbb, a[1] \to aaaa[1]$. Tato sada pravidel umožnuje v následujícím derivačním kroku odvodit slovo w, pro které platí, že $\gamma(w) = a^{t+3}b^{t+1}a^{t+2}b^{t+2}a^{t+3}$, které však nenáleží do L^2.
- Nechť $u = a^{t-1}$, $v = ab^t a^{2t}$, $x = b^{2t}$, $y = a^{3t}$, pak Q_1 musí obsahovat pravidla $[1]a \to a[1]ab, [1]b \to aa[1]bbb, [1]a \to [1]aaaa$. Tato sada pravidel umožnuje v následujícím derivačním kroku odvodit slovo w, pro které platí, že $\gamma(w) = a^{t+3}b^{t+1}a^{2t+2}b^{2t+2}a^{3t+3}$, které však nenáleží do L^2.
- Nechť $u = a^{t-1}$, $v = ab^t a^{2t}$, $x = b^{2t}$, $y = a^{3t}$, pak Q_1 musí obsahovat pravidla $[1]a \to a[1]ab, [1]b \to aa[1]bbb, b[1] \to b[1]aaa$. Tato

sada pravidel umožnuje v následujícím derivačním kroku odvodit slovo w, pro které platí, že $\gamma(w) = a^{t+1}b^{t+1}a^{2t+2}b^{2t+2}aba^{3t}$, které však nenáleží do L^2.

Ve všech případech dostáváme spor s předpokladem, že L^2 patří do $\mathscr{L}(PEG^1)$, tudíž $L^2 \in \mathscr{L}(PEG^2)$.

Obecně můžeme říct, že fakt, že agent může použít pouze jednostranný kontext, nám dovolí inkrementovat dva shodné podřetězce typu a^ib^i o pouze dvě odlišná čísla.

Věta 3 *Třída PEG jazyků s ohledem na počet typů agentů tvoří nekonečnou hierarchii: $\mathscr{L}(PEG^n) \subset \mathscr{L}(PEG^{n+1})$ pro $n \geq 1$ (tj. vlastní podmnožinu).*

Důkaz: Zjevně $\mathscr{L}(PEG^n) \subseteq \mathscr{L}(PEG^{n+1})$ pro $n \geq 1$.

Uvažujme jazyk

$$L^{n+1} = \{a^ib^ia^{2i}b^{2i}\dots$$

$$\dots a^{(2n-1)i}b^{(2n-1)i}a^{2ni}b^{2ni}a^{(2n+1)i} : i \geq 1\}.$$

Ukážeme, že $L^{n+1} \in \mathscr{L}(PEG^{n+1}) - \mathscr{L}(PEG^n)$.

Tudíž ukážeme, že potřebujeme alespoň $n + 1$ typů agentů ke generování jazyka L^{n+1}, n typů agentů nepostačuje.

Nejprve sestavíme PEG systém Σ^{n+1} s $n + 1$ typy agentů, který generuje jazyk L^{n+1}: $\Sigma^{n+1} = (V_E, \{[1]\}, E, B_1, B_2, \dots, B_n, B_{n+1})$, kde

- $V_E = \{a, b\}$,
- $E = (V_E, \{a \to a, b \to b\})$,
- $B_1 = ([1], \{[1]a \to a[1]ab, [1]b \to aa[1]bbb\})$,
- $B_2 = ([2], \{[2]a \to aaa[2]abbb, [2]b \to aaaa[2]bbbbb\})$,
- $\dots$
- $B_n = ([n], \{[n]a \to a^{2n-1}[n]ab^{2n-1}, [n]b \to a^{2n}[n]bb^{2n}\})$,
- $B_{n+1} = ([n+1], \{a[n+1] \to aa^{2n+1}[n+1]\})$.

Axiom systému PEG^{n+1} je

$$w = [1]abaa[1]bbaa[2]abbbaaaa[2]bbbb\dots$$

$$\dots a^{2n-2}[n]ab^{2n-1}a^{2n}[n]b^{2n}a^{2n+1}[n+1].$$

Ukážeme několik prvních kroků odvození:
$[1]abaa[1]bbaa[2]abbbaaaa[2]bbbb\dots$
$\dots a^{2n-2}[n]ab^{2n-1}a^{2n}[n]b^{2n}a^{2n+1}[n+1] \Rightarrow$
$a[1]abba^4[1]b^4a^5[2]ab^6a^8[2]b^8\dots$
$\dots a^{4n-3}[n]ab^{4n-2}a^{4n}[n]b^{4n}a^{4n+2}[n+1] \Rightarrow$
$aa[1]abbba^6[1]b^6a^8[2]ab^9a^{12}[2]b^{12}\dots$
$\dots a^{6n-4}[n]ab^{6n-3}a^{6n}[n]b^{6n}a^{6n+3}[n+1] \Rightarrow \dots$

Tudíž $L(\Sigma^{n+1}, w) = \{a^ib^ia^{2i}b^{2i}\dots a^{(2n-1)i}b^{(2n-1)i}a^{2ni}b^{2ni}a^{(2n+1)i} : i \geq 1\}$.

Abychom ukázali, že $L^{n+1} \in \mathscr{L}(PEG^{n+1})$, ověříme, že $L^{n+1} \notin \mathscr{L}(PEG^n)$, čili neexistuje PEG systém Σ s $|N_B| = n$ a axiom w takový, že $L(\Sigma, w) = L^{n+1}$.

Předpokládejme naopak, že L^{n+1} patří do $\mathscr{L}(PEG^n)$ a PEG^n systém $\Sigma^n = (V_E, N_B, E, B_1, \dots, B_n)$ s axiomem w^n generuje L^{n+1}. Ověříme, že $E = (V_E, P)$ v Σ^n je deterministické a $P = \{z \to z : z \in \{a, b\}\}$.

Ze struktury jazyka L^{n+1} plyne, že pro $z \to \alpha \in P$ je $\alpha = z^s$ pro nějaké s. Předpokládejme, že P není deterministická a $z \to z^j \in P$ pro $j \geq 0$. Nechť je v odvození $u = a^rb^ra^{2r}b^{2r}\dots a^{(2n-1)r}b^{(2n-1)r}a^{2nr}b^{2nr}a^{(2n+1)r} \in L_{n+1}$ pro nějaké r použito pravidlo $z \to z^j$. Pak ale i slovo $v = a^{j(r-2)+1}b^ra^{2(r)}b^{2(r)}\dots a^{r(2n-1)}b^{r(2n-1)}a^{2nr}b^{2nr}a^{r(2n+1)} \in L(\Sigma, w)$, ale $v \notin L^{n+1}$. Zjevně pro pravidla $z \to z^j$ v deterministické P platí $j = 1$. V opačném případě ne všechna slova z L^{n+1} jsou v $L(\Sigma^n, w^n)$

Tudíž všechny změny v prostředí způsobují svou aktivitou agenti.

Nechť $u_0[1]u_1[1]u_2[2]u_3[2]\dots u_{2n-1}[n]u_{2n}[n]u_{2n+2}[o]u_{2n+3}$, kde $u_0 = a^{t-1}$, $u_1 = ab^ta^{2t}, u_2 = b^{2t}a^{3t-1}, u_3 = ab^{3t}a^{4t}, \dots, 1 \leq o \leq n$, je slovo odvozené z axiomu v Σ^n a $u_0\dots u_{2n+3} = a^tb^ta^{2t}b^{2t}\dots a^{(2n-1)t}b^{(2n-1)t}a^{2nt}b^{2nt}a^{(2n+1)t}$ pro nějaké $t \geq 1$. Bez újmy na obecnosti můžeme předpokládat, že $o = 1$. V souladu s definicí PEG systému může každý agent přepsat symbol po své pravé nebo levé straně, ale ne oba současně. Díky tomuto omezení můžeme brát v potaz pouze podřetězec $u_{2n+2}[o]u_{2n+3}$, a to v následujících případech:

- Nechť $u_{2n+2} = b^{2nt-1}, u_{2n+3} = ba^{t(2n+1)}$, pak Q_1 musí obsahovat pravidla $[1]a \to a[1]ab$, $[1]b \to aa[1]bbb, [1]b \to [1]ba^{2n+1}$. Taková sada pravidel nám však dovolí odvodit v následujícím kroku odvození slovo w takové, že $\gamma(w) = a^{t+1}b^{t+1}a^tba^{2n+1}b^t\dots a^{2n(t+1)}\ b^{2n(t+1)}\ a^{(2n+1)(t+1)}$, které ovšem nepatří do L^{n+1}.
- Nechť $u_{2n+2} = b^{2nt}, u_{2n+3} = a^{t(2n+l)}$, pak Q_1 musí obsahovat pravidla $[1]a \to a[1]ab$, $[1]b \to aa[1]bbb, b[1] \to b[1]a^{2n+1}$. Taková sada pravidel nám však dovolí odvodit v následujícím kroku odvození slovo w takové, že $\gamma(w) = a^{t+1}b^{t+1}a^{t+2}b^{t+2}\dots a^{2n(t+1)}b^{2n(t+1)}aaba^{t(2n+1)-1}$, které ovšem nepatří do L^{n+1}.
- Nechť $u_{2n+2} = b^{2nt}, u_{2n+3} = a^{t(2n+l)}$, pak Q_1 musí obsahovat pravidla $[1]a \to a[1]ab$, $[1]b \to aa[1]bbb, [1]a \to [1]aa^{2n+1}$. Taková sada pravidel nám však dovolí odvodit v následujícím kroku odvození slovo w takové, že $\gamma(w) = a^{t+2n+1}b^ta^{t+2}b^{t+2}\dots a^{2n(t+1)}b^{2n(t+1)}a^{(2n+1)(t+1)}$, které ovšem nepatří do L^{n+1}.

- Nechť $u_{2n+2} = b^{2nt}a^k, u_{2n+3} = a^l$, kde $k + l = t(2n + 1)$, pak Q_1 musí obsahovat pravidla $[1]a \rightarrow a[1]ab, [1]b \rightarrow aa[1]bbb, [1]a \rightarrow [1]aa^{2n+1}$. Taková sada pravidel nám však dovolí odvodit v následujícím kroku odvození slovo w takové, že $\gamma(w) = a^{t+2n+1}b^t a^{t+2} b^{t+2} \dots a^{2n(t+1)} b^{2n(t+1)} a^{(2n+1)(t+1)}$, které ovšem nepatří do L^{n+1}.

- Nechť $u_{2n+2} = b^{2nt}a^k, u_{2n+3} = a^l$, kde $k + l = t(2n + 1)$, pak Q_1 musí obsahovat pravidla $[1]a \rightarrow a[1]ab, [1]b \rightarrow aa[1]bbb, a[1] \rightarrow a^{2n+1}a[1]$. Taková sada pravidel nám však dovolí odvodit v následujícím kroku odvození slovo w takové, že $\gamma(w) = a^{t+2n+1}b^t a^{t+2} b^{t+2} \dots a^{2n(t+1)} b^{2n(t+1)} a^{(2n+1)(t+1)}$, které ovšem nepatří do L^{n+1}.

- Nechť $u_{2n+2} = b^{2nt}, u_{2n+3} = a^{t(2n+1)}$, pak Q_1 musí obsahovat pravidla $[1]a \rightarrow a[1]ab$, $[1]b \rightarrow aa[1]bbb, a[1] \rightarrow a^{2n+1}a[1]$. Taková sada pravidel nám však dovolí odvodit v následujícím kroku odvození slovo w takové, že $\gamma(w) = a^{t+2n+1}b^{t+1}a^t ba^{2n+1}b^t \dots a^{2n(t+1)}b^{2n(t+1)}\ a^{(2n+1)(t+1)}$, které ovšem nepatří do L^{n+1}.

Ve všech případech dostaneme spor s předpokladem, že L^{n+1} patří do $\mathscr{L}(PEG^n)$ a platí $L^{n+1} \in \mathscr{L}(PEG^{n+1})$. Tudíž $\mathscr{L}(PEG^n) \subsetneq \mathscr{L}(PEG^{n+1})$.

4 Závěr

V tomto článku jsme předložili nejdůležitější výsledky týkající se generativní síly PEG systémů. Stanovili jsme horní hranici a ukázali jsme, že generativní síla je závislá jak na počtu agentů působících v prostředí, tak na počtu typů agentů v definici PEG systému. Byť jsme schopni pomocí PEG systémů generovat víc než čisté jazyky typu 0, PEG systémy nejsou výpočetně kompletní. Další výsledky týkající se generativní síly PEG systémů, zejména jejich vztah k čistým řízeným gramatikám, lze najít v Langer (2007), Langer a Kelemenová (2012), Langer a Kelemenová (2013). Nadále zůstává nezodpovězena otázka, jaký je vztah PEG systémů a čistých řízených gramatik s kontrolou výskytu. Další výsledky, které se týkají především uzávěrových vlastností PEG systémů, nalezneme např. v Langer (2005a), Langer (2005b).

Poděkování

Článek vznikl ve spojitosti s projektem IT4Innovations Centre of Excellence, reg. no. CZ.1.05/1.1.00/02.0070. Výzkum je dále podpořený projektem SGS/24/2013.

Literatura

Csuhaj-Varjú, E., Kelemen, J., Kelemenová, A. a Păun, G. (1994). Eco(grammar) systems – a preview. *Cybernetics and Systems '94*, str. 941–948.

Langer, M. (2005a). Agents placed in the environment of eco-grammar systems – positioned eco-grammar systems. V *Pre-Procs. of the 1st Doctoral Workshop on Mathematical a Engineering Methods in Computer Science*, str. 31–37.

Langer, M. (2005b). Agenty umístěné v prostředí ekogramatických systému – poziční ekogramatické systémy. V *Kognice a umělý život V*, str. 339–350.

Langer, M. (2007). Poziční ekogramatické systémy a (řízené) pure gramatiky. V *Kognice a umělý život VII*, str. 215–220.

Langer, M. a Kelemenová, A. (2012). Positioned agents in eco-grammar systems with border markers and pure regulated grammars. *Kybernetika*, 48:502–517.

Langer, M. a Kelemenová, A. (2013). On positioned eco-grammar systems and pure grammars of type 0. *Neural Network World*, 2(13):81–91.

Martin-Vide, C. a Păun, G. (1998). Pm-colonies. *Computers and Artificial Intelligence*, 17:553–582.