Pattern Languages

Diplomová Práca

Peter Bartek

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA INFORMATIKY

Študijný odbor: Informatika

Vedúci diplomovej práce: RNDr. Dana Pardubská, PhD.

BRATISLAVA 2007

Abstrakt

Táto práca sumarizuje mnohé výsledky dosiahnuté v posledných rokoch v oblasti jazykov definovaných pomocou vzorov, tzv. Pattern Languages. Vzor je reťazec pozostávajúci z terminálov a premenných, pričom premenné nahrádzame slovami tak, že výskyt rovnakej premennej nahradíme vždy tým istým slovom. Predstavíme si rôzne spôsoby definovania formálnych jazykov pomocou vzorov a generatívnych systémov založených na vzoroch. Zameriame sa hlavne na základné vlastnosti takto definovaných tried jazykov, porovnáme ich navzájom medzi sebou, ako aj s inými všeobecne známymi triedami jazykov Chomského hierarchie a 0L-systémov, predstavíme ich uzáverové vlastnosti, niekde naznačíme aj dopad a súvis získaných výsledkov pre niektoré iné oblasti. V niektorých prípadoch rozoberieme aj niektoré iné aspekty týkajúce sa napr. rozhodnuteľnosti niektorých problémov. Hoci je práca zameraná hlavne na prehľad už získaných výsledkov, na viacerých miestach je doplnená vlastnými jednoduchými dôkazmi, príkladmi, prípadne aj komentármi, ktoré majú napomôcť tomu, aby čitateľ ľahšie pochopil niektoré preberané skutočnosti, či dôkazy.

Kľúčové slová:
vzor (pattern), pattern language, multivzor, dvojúrovňové štruktúry

Obsah

Kapitola 1

Úvod

Myšlienka vzorov sa objavila už v práci autora A. Thue [o1] v roku 1906, avšak jazyky definované vzorom tak, ako sú prezentované v tejto práci boli predstavené D. Angluinom až oveľa neskôr v [1].

Formálny jazyk sa obvykle zvykne definovať pomocou nejakých generatívnych systémov alebo rozpoznávacích zariadení, ako sú napríklad gramatiky alebo automaty. V niektorých prípadoch je však z teoretických aj praktických dôvodov výhodnejšie použiť voľnejšiu definíciu - napríklad zadaním vzoru, ktorý by mali rešpektovať všetky slová v danom jazyku. Užitočné je to napríklad vtedy, keď chceme nájsť spoločný vzor pre všetky slová k danej množine vzoriek. Toto je typický problém induktívnej inferencie, čo je proces odvodzovania všeobecných pravidiel zo špecifických príkladov. Pokiaľ sa daná množina vzoriek pomocou nejakého procesu zväčšuje, potom daný problém patrí aj do teórie strojového učenia. Často sa pri tom snažíme nájsť vzor, ktorý čo najlepšie charakterizuje danú množinu príkladov (vzoriek). Niekedy môžeme nájsť aj viacero "najlepších" vzorov, ktoré popisujú danú množinu vzoriek (jazyky definované týmito vzormi sú navzájom neporovnateľné a akýkoľvek vzor, ktorý definuje menšiu množinu ako ktorýkoľvek z "najlepších" vzorov už neobsahuje všetky slová z množiny vzoriek). Práve aplikácie takéhoto druhu boli pôvodnou motiváciou pre skúmanie "pattern languages" (jazykov definovaných vzorom). Môžeme mať popri tom aj druhú množinu slov, ktorej prvky nechceme zahrnúť do jazyka (tzv. negatívne príklady). V takom prípade hľadáme vzor, ktorý má popisovať množinu pozitívnych príkladov a zároveň jazyk, ktorý je ním definovaný, nesmie obsahovať dané negatívne príklady.

Vzory môžeme hľadať aj v programovacích jazykoch, napr. medzi hlavné nebezkontextové prvky programovacích jazykov patrí nutnosť definovať návestia a deklarovať identifikátory. Napríklad: správny program obsahujúci návestia by mal rešpektovať vzor

$$X_1\ Y{:}Z\ X_2\ \text{goto}\ Y\ X_3$$

kde X_1, X_2, X_3 sú premenné reprezentujúce nejaké časti programu, Z je premenná pre nejaký príkaz a Y je premenná pre návestie. Bodkočiarka a "goto" sú konštanty (vyjadrené ako reťazce). Pri interpretovaní tohoto vzoru musíme nahradiť premennú Y vždy rovnakým reťazcom.

My sa budeme v tejto práci zaoberať práve takto definovanými jazykmi. Vzor (pattern) teda nie je nič iné ako reťazec pozostávajúci z terminálnych symbolov a premenných. Nahrádzaním premenných terminálnymi reťazcami potom získavame slová patriace do jazyka, pričom v prípade, že sa niektorá premenná objaví v danom vzore viackrát, treba všetky jej výskyty nahradiť tým istým reťazcom. Na slová, ktorými nahrádzame premenné, možno samozrejme klásť rôzne obmedzenia. Hneď v ďalšej kapitole (číslo 2) si predstavíme tzv. *vymazávajúce* (resp. *nevymazávajúce*) *vzory*, kde za premenné možno dosadzovať ľubovoľné (prípadne ľubovoľné neprázdne) reťazce. Budeme skúmať hlavne základné vlastnosti takto definovaných tried jazykov z hľadiska teórie formálnych jazykov, ako sú uzáverové vlastnosti a porovnanie veľkosti tried s inými dobre známymi triedami Chomského hierarchie, pozrieme sa aj (vo všeobecnosti menej podrobne) na problém ekvivalencie a inklúzie s neobvyklým výsledkom v rámci teórie formálnych jazykov. Stručne popíšeme aj prístup vychádzajúci z pôvodnej

motivácie pre skúmanie pattern languages (teda že k danej vzorke hľadáme popisný vzor) a taktiež sa na chvíľu zastavíme pri probléme viacznačnosti vo vzoroch, ktorý možno označiť za istý druh nedeterminizmu. V tretej kapitole pristúpime k prirodzenému zovšeobecneniu, keď namiesto jedného vzoru budeme uvažovať nejakú konečnú množinu vzorov. Jazyk definovaný týmito vzormi bude zjednotením jazykov definovaných jednotlivými vzormi. V kapitole číslo 4 potom daný model opäť zovšeobecníme a to tak, že ku každej premennej priradíme nejakú doménu (jazyk, z ktorého budeme vyberať slová pri jej nahrádzaní) a okrem toho budeme uvažovať aj nekonečné množiny vzorov. V tomto prípade pôjde už o akési dvojúrovňové štruktúry, kde na prvej úrovni bude musieť byť nejaké zariadenie (napr. gramatika) na definovanie nekonečnej množiny vzorov, ktoré potom na druhej úrovni interpretujeme. Okrem už tradičných základných vlastností takto definovaných tried jazykov aspoň stručne spomenieme niektoré výsledky ich porovnania s paralelnými komunikujúcimi systémami gramatík (v skratke: PCGS). V piatej kapitole do istej miery opustíme dovtedajší prístup a s využitím obvyklej stratégie teórie formálnych jazykov zadefinujeme isté generatívne zariadenia založené na vzoroch, konkrétne pôjde o tzv. *gramatiky založené na vzoroch*, kde na začiatku budeme mať množinu vzorov a terminálnych slov (tzv. axiómy). Nahradením premenných vo vzoroch axiómami získame ďalšiu množinu slov a tieto novovzniknuté slová môžeme v ďalších krokoch použiť na získanie ešte väčšej množiny slov, atď.. V šiestej (záverečnej) kapitole si predstavíme zovšeobecňujúci model, tzv. *Pattern-systémy*, ktorý pokrýva mnohé dovtedy definované triedy pattern languages.

Pokúsime sa na konci každej kapitoly spomenúť otvorené problémy týkajúce sa preberanej oblasti, prípadne aj možnosti ďalšieho výskumu s odkazom na literatúru, z ktorej možno čerpať ďalšie informácie o danej oblasti.

Na viacerých miestach práce porovnávame triedy definované vzormi s triedami L-systémov. Keďže L-systémy nemusia byť súčasťou základného kurzu formálnych jazykov, pridali sme na záver práce dodatok, v ktorom definujeme základné triedy OL-systémov.

Základné definície a označenia

Hoci tento text je určený tým, ktorí už prešli základným kurzom teórie formálnych jazykov, uvedieme aj niektoré základné definície, a to hlavne kôli označeniam používaným v tejto práci, keďže v rôznych učebniciach a článkoch sa tieto niekedy navzájom líšia. Vopred chceme najmä upozorniť na označovanie vlastnej inklúzie.

Abeceda je konečná a neprázdna množina symbolov (písmen). Budeme ju označovať Σ. Jej prvky budeme zapisovať malými písmenami zo začiatku latinskej abecedy, niekedy aj s indexami (a, b, c, ..., a_1, a_2, a_3, ...).

Slovo nad abecedou Σ je konečná postupnosť symbolov zo Σ. Slová budeme označovať písmenami z konca latinskej abecedy (u, v, w, x, ...), niekedy aj s indexami.
Prázdne slovo budeme označovať symbolom ε.
Dĺžku slova w označíme ako $|w|$.
Počet symbolov a v slove *w* označíme ako $\#_a w$.

Jazyk nad abecedou Σ je ľubovoľná množina slov nad abecedou Σ. Jazyky budeme označovať symbolom L, často s indexami (L_1, L_2, ...).

Označovanie niektorých operácii nad slovami a jazykmi:
Nech *w* je slovo a L jazyk.
Reverz slova *w* budeme označovať w^R, *reverz* jazyka L zase L^R .
Komplement jazyka L označíme ako L^c.

Pre triedy jazykov (nielen) Chomského hierarchie budeme používať nasledujúce označenia:
SNG – jednoslovné jazyky
FIN – konečné jazyky
REG – regulárne jazyky
LIN – lineárne jazyky
CF – bezkontextové jazyky
CS resp. ECS – kontextové resp. rozšírené kontextové jazyky
RE – rekurzívne vyčísliteľné jazyky

OL, EOL, DOL, TOL, EDTOL, atď. – triedy jazykov definované L-systémam – ich definície možno nájsť v dodatku

AFL (abstraktná trieda jazykov) je trieda jazykov uzavretá na zjednotenie, prienik s regulárnymi jazykmi, kladnú iteráciu (Kleeneho +), zreťazenie, nevymazávajúci homomorfizmus a inverzný homomorfizmus.
anti-AFL je trieda jazykov, ktorá nie je uzavretá ani na jednu z uvedených operácii.

Chceme upozorniť na nasledujúce označovanie inklúzie:
$\subset$ - vlastná inklúzia
$\subseteq$ - inklúzia

Iné označenia:
^ - umocňovanie (hlavne v indexoch)

Kapitola 2

Vzory a jazyky

2.1 Základné definície

V nasledujúcich niekoľkých definíciách sformalizujeme neformálny popis pattern languages z úvodu.

Definícia 2.1.1: Nech Σ je abeceda terminálnych symbolov a V spočítateľná množina premenných. *Vzor* (*pattern*) je konečný a neprázdny reťazec nad $(\Sigma \cup V)$. Vzory budeme označovať malými gréckymi písmenami zo začiatku gréckej abecedy (α, β, γ, ...) alebo ako π, prípadne aj s indexami (π_1, π_2, ...), prvky Σ zase malými písmenami latinskej abecedy (a, b, c, ..., prípadne aj s indexami) a premenné veľkými písmenami z konca abecedy (X, Y, X_1, X_2, ...).

Definícia 2.1.2: Nech Σ je množina terminálnych symbolov a V množina premenných. Označme symbolom $H_{\Sigma,V}$ množinu homomorizmov s nasledujúcimi vlastnosťami

$$H_{\Sigma,V} = \{h:(\Sigma \cup V)^* \rightarrow \Sigma^* \mid h(a)=a, \text{ ak } a \in \Sigma;\ h(X_i) \in \Sigma^*,\ X_i \in V,\ i \in \{1,...,n\}\}$$

Uvedomme si, že takýto homomorfizmus v danom vzore ponechá terminálne symboly bez zmeny a premenné nahradí nejakým slovom zo Σ^*, taktiež je zabezpečená jedna zo základných požiadaviek pri nahrádzaní premenných vo vzore slovami zo Σ a síce, že rovnakú premennú musíme nahradiť vždy tým istým slovom. Podľa toho, či budeme premenné v danom vzore nahrádzať ľubovoľnými alebo neprázdnymi slovami, budeme príslušný vzor nazývať *vymazávajúci* resp. *nevymazávajúci* (*E-pattern, E-vzor* resp. *NE-pattern, NE-vzor*; E = erasing, NE = non-erasing). Teda rozdiel medzi vymazávajúcimi a nevymazávajúcimi vzormi sa týka jazyka, ktorý je nimi definovaný, nie vzorov samotných.

Definícia 2.1.3:
Jazyk generovaný E-vzorom $\pi \in (\Sigma \cup V)^+$ nad abecedou Σ je $L_{E,\Sigma}(\pi) = \{h(\pi) \mid h \in H_{\Sigma,V}\}$.
Jazyk generovaný NE-vzorom $\pi \in (\Sigma \cup V)^+$ nad abecedou Σ je $L_{NE,\Sigma}(\pi) = \{h(\pi) \mid h \in H_{\Sigma,V}$ je nevymazávajúci, teda $h(Xi) \in \Sigma^+\}$

Pre istotu zdôrazňujeme, že množinu slov takto definovaných jazykov tvoria obrazy daného vzoru získané pomocou *všetkých* homomorfizmov z množiny $H_{\Sigma,V}$.

Definícia 2.1.4: Homomorfizmus, ktorý ponecháva terminálne symboly bez zmeny nazveme *stabilným pre terminály*.

Všetky homomorfizmy z množiny $H_{\Sigma,V}$ majú práve definovanú vlastnosť a pri definovaní rôznych typov pattern languages (ako uvidíme neskôr) sa budú využívať práve homomorfizmy s touto vlastnosťou.

Označenie: Triedy jazykov generované vymazávajúcimi resp. nevymazávajúcimi vzormi budeme označovať ako PL_E resp. PL_{NE}. Budeme ich nazývať *vymazávajúce* resp. *nevymazávajúce* pattern languages (E-pattern languages resp. NE-pattern languages). Obzvlášť z hľadiska rozhodnuteľnosti určitých vlastností je významný rozdiel medzi triedami PL_E a PL_{NE}. Keď budeme hovoriť naraz o oboch triedach jazykov, budeme používať označenie PL_Z, $Z \in \{E,NE\}$ (napríklad ak bude uvedené, že $w \in PL_Z$, bude to znamenať, že $w \in PL_E$ a $w \in PL_{NE}$).

Dohoda: Ak budú množiny Σ, V známe, budeme namiesto $H_{\Sigma,V}$ písať len H a namiesto $L_{E,\Sigma}(\pi)$ resp. $L_{NE,\Sigma}(\pi)$ len $L_E(\pi)$ resp. $L_{NE}(\pi)$.

Definícia 2.1.5: Hovoríme, že Z-vzory π_1 a π_2 sú ekvivalentné, ak $L_Z(\pi_1) = L_Z(\pi_2)$, $Z \in \{E,NE\}$.

2.2 Základné vlastnosti tried PL_E a PL_{NE}

Kvôli získaniu základnej predstavy o pattern languages (pattern-jazykoch) začneme niekoľkými príkladmi. Niektoré z týchto príkladov neskôr využijeme pri dokazovaní určitých vlastností pattern languages. Písmeno Z bude znamenať buď E alebo NE.

Príklad 2.2.1:
Majme vzor $\pi_1 = XX$
Je zrejmé, že $L_E(\pi_1) = \{ww \mid w \in \Sigma^*\}$ a $L_{NE}(\pi_1) = \{ww \mid w \in \Sigma^+\}$

Príklad 2.2.2:
Nech $\pi_2 = XYX$ je E-pattern.
Tento vzor definuje jazyk, v ktorom prefix = sufix, pričom sa neprekrývajú

Príklad 2.2.3:
Jazyk $\{a\}^+ \in PL_Z$.
Majme vzory $\pi_1 = X$ a $\pi_2 = aX$, $\Sigma = \{a\}$.
Je zrejmé, že $L_{NE}(\pi_1) = L_E(\pi_2) = \{a\}^+$.

Lema 2.2.4: Jednoslovné jazyky patria do PL_Z (okrem jazyka $L = \{\varepsilon\}$).

Dôkaz: Jazyk $L = \{w \in \Sigma^+\}$ je generovaný vzorom $\alpha = w$ (vzor podľa definície nemusí obsahovať premenné, musí to však byť neprázdny reťazec, preto tam nepatrí jazyk $\{\varepsilon\}$).

□

Lema 2.2.5: $\Sigma^* \in PL_E$ (resp. $\Sigma^+ \in PL_{NE}$).

Dôkaz: Uvedené jazyky sú generované vzorom X, ostatné vyplýva priamo z definície.

□

Hoci z uvedených príkladov je zrejmé, že takýmto spôsobom možno generovať aj nebezkontextové jazyky, teraz si ukážeme príklady regulárnych jazykov, ktoré do PL_Z nepatria.

Lema 2.2.6: $a^+ \cup b^+ \notin PL_Z$

Dôkaz: Keďže jazyk je nekonečný, tak vzor, ktorým by sa mal dať generovať, musí obsahovať aspoň jednu premennú. Lenže premenné možno nahrádzať ľubovoľnými slovami nad danou abecedou (resp. ľubovoľnými neprázdnymi slovami v prípade nevymazávajúcich vzorov). Keďže chceme generovať jazyk nad abecedou $\Sigma = \{a, b\}$, tak v tomto prípade sa nedá vyhnúť slovám obsahujúcim oba symboly naraz.

□

Lema 2.2.7: $a^+b^+ \notin PL_{NE}$, $a^*b^* \notin PL_E$

Dôkaz: Opäť máme nekonečné jazyky, takže vzory, ktorými be sme ich chceli generovať by museli obsahovať aspoň jednu premennú, ktorú možno nahrádzať ľubovoľnými slovami nad abecedou $\{a, b\}^+$ resp. $\{a, b\}^*$, teda napríklad aj slovom *abbbab* a je úplne zrejmé, že takéto podslovo sa nikde v slovách uvedených jazykov nevyskytuje.

□

Veta 2.2.8: Triedy jazykov PL_E a PL_{NE} sú anti-AFL, nie sú uzavreté ani na prienik a komplement.

Dôkaz: Postupne dokážeme neuzavretosť na všetky operácie, $Z \in \{E, NE\}$.

Zjednotenie:
Jazyky a^+ a $b^+ \in PL_Z$, $a^+ \cup b^+ \notin PL_Z$

Zreťazenie:
Jazyky a^* a $b^* \in PL_E$, $a^*b^* \notin PL_E$
Jazyky a^+ a $b^+ \in PL_{NE}$, $a^+b^+ \notin PL_{NE}$

Kleeneho +:
$\{ab\} \in PL_Z$, $\{ab\}^+ \notin PL_Z$ (opäť – je to nekonečný jazyk, vo vzore pre tento jazyk musí byť aspoň 1 premenná, tým pádom sa určite v mnohých slovách naruší požadovaná štruktúra, keď za premennú dosádzame ľubovoľné reťazce nad $\{a, b\}$).

Prienik s regulárnymi jazykmi:
$\Sigma^* \in PL_E$ ($\Sigma^+ \in PL_{NE}$) pre všetky Σ. Pre $|\Sigma| \geq 2$ existujú regulárne jazyky, ktoré nepatria do PL_Z, nech je to napr. a^+b^+. Veľmi ľahko vidno, že $a^+b^+ \cap \Sigma^* = a^+b^+$ (tak isto aj pre Σ^+).

Homomorfizmus (nevymazávajúci):
Nech $h(a) = ab$. Jazyk $a^+ \in PL_Z$, no $h(a^+) = \{ab\}^+ \notin PL_Z$.

Inverzný homomorfizmus:

Nech $h(a) = \varepsilon$, $h(b) = b$. Jazyk $\{b\} \in PL_Z$, ale $h^{-1}(b) = a^*ba^* \notin PL_Z$ (opäť platia podobné argumenty ako v predchádzajúcich príkladoch).

Prienik:
Dôkaz urobíme pre PL_E.
Nech $\Sigma = \{a, b\}$ a $\alpha = XXab$, $\beta = XbaX$ nech sú vzory. Prienik jazykov definovaných týmito vzormi je:

$$L_E(\alpha) \cap L_E(\beta) = \{w \in \Sigma^* \mid w = uuab = vbav \text{ pre nejaké slová } u, v \in \Sigma^*\}$$

Skúsme vyriešiť rovnicu *uuab = vbav*. Jedno triviálne riešenie je $u = v = b$, teda slovo bbab patrí do prieniku. V prípade, že $v \neq b$, musí byť jeho suffix *ab*, teda $v = v_1ab$ pre nejaké v_1. Odtiaľ dostávame, že $vbav = v_1abbav_1ab$ a keďže celé sa to musí rovnať slovu *uuab*, tak ľahko vidno, že musí platiť: $u = v_1ab = bav_1$. Z ďalšej analýzy rovnosti $v_1ab = bav_1$ (prefix v_1 je *ba*, teda $bav_2ab = babav_2$ pre nejaké v_2 a pre v_2 dostávame rovnakú rovnosť ako pre v_1, atď...) vyplýva, že $v_1 = (ba)^kb$, $k \geq 0$ a následne pre slová u aj v platí: $u = v = (ba)^kbab = (ba)^{k+1}b$.

Slová z prieniku budú mať teda tvar (pre $k \geq 0$):
$L_E(\alpha) \cap L_E(\beta) = \{(ba)^{k+1}b(ba)^{k+1}bab)\} \cup \{bbab\} = \{(ba)^kb(ba)^{k+1}b, k \geq 0\}$.

Jazyk obsahujúci takéto slová však nemôže patriť do PL_E, čo sa opäť dokáže podobne ako v predchádzajúcich prípadoch. Keďže je nekonečný, tak vo vzore, ktorým by sme ho chceli generovať by musela byť aspoň 1 premenná, ktorú možno nahradiť napríklad aj slovom *aa*, lenže *aa* nie je podslovom žiadneho slova z prieniku.

Pozornému čitateľovi je iste zrejmé, že to isté platí aj pre PL_{NE} (maximálne len s nepatrnými zmenami v dôkaze zohľadňujúcimi fakt, že za premenné dosádzame len neprázdne slová)

Komplement:
Jazyk $L_1 = \{a, b\}^*ab\{a, b\}^* \in PL_E$ (je generovaný vzorom XabY), $L_1^c = b^*a^* \notin PL_E$
Jazyk $L_2 = \{a, b\}^+ab\{a, b\}^+ \in PL_{NE}$ (je generovaný vzorom XabY), komplement jazyka L_2 obsahuje ε a trieda PL_{NE} obsahuje len bez-ε jazyky.

□

Veta 2.2.9: Triedy PL_E, PL_{NE} sú uzavreté na zrkadlový obraz.

Dôkaz: $L_Z(\alpha)^R = L_Z(\alpha^R)$, $Z \in \{E, NE\}$.

□

Situácia nemusí byť až taká zlá, pokiaľ sa obmedzíme len na jazyky nad rovnakou abecedou, aj keď oveľa lepšie to nebude. *Triedy jazykov nad nejakou fixnou abecedou Σ si označme ako* $PL_{Z,\Sigma}$, $Z \in \{E, NE\}$. Začnime najprv tým horším prípadom, ktorý nie je pokrytý v predchádzajúcej vete:

Veta 2.2.10: Triedy jazykov $PL_{E,\Sigma}$ a $PL_{NE,\Sigma}$ nie sú uzavreté na zjednotenie, ak $|\Sigma| \geq 2$.

Dôkaz: Uvažujme jazyky nad abecedou $\Sigma = \{a, b\}$ generované vzormi *aaX* a *bbX*. Slová z každého jazyka majú vždy iný (ale fixný) prefix. Ľahko vidno, že pomocou jedného vzoru nevieme definovať jazyk obsahujúci oba typy slov.

□

Veta 2.2.11: Triedy jazykov $PL_{E,\Sigma}$ a $PL_{NE,\Sigma}$ sú uzavreté na zreťazenie.

Dôkaz: Nech $L_{Z,\Sigma}(\alpha)$ a $L_{Z,\Sigma}(\beta)$ sú jazyky definované vzormi α a β, $Z \in \{E, NE\}$. Bez ujmy na všeobecnosti môžeme predpokladať, že množiny premenných, ktoré sa vyskytujú v α resp. v β sú disjunktné (keby neboli, tak premenné vhodne premenujeme). Je zrejmé, že potom vzor $\alpha\beta$ generuje jazyk $L(\alpha)L(\beta) = L(\alpha\beta)$ (pre istotu ešte raz zdôraznime, že abecedy oboch jazykov musia byť rovnaké).

□

Teraz porovnáme triedy PL_E a PL_{NE} s triedami jazykov Chomského hierarchie.

Veta 2.2.12: PL_E a PL_{NE} sú neporovnateľné s REG, CF, sú vlastnou podmnožinou ECS.

Dôkaz: *Neporovnateľnosť s REG a CF*: Nech Σ je abeceda obsahujúca aspoň dva symboly. Už vieme, že $\{ww \mid w \in \Sigma^*\} \in PL_E$ a podobne aj $\{ww \mid w \in \Sigma^+\} \in PL_{NE}$, ale oba jazyky nie sú ani bezkontextové. Naopak, jazyk $a^+ \cup b^+ \in$ REG, ale nepatrí ani do jednej z tried PL_E resp. PL_{NE}.

$PL_Z \subset$ ECS – uvedieme len myšlienku dôkazu: Lineárne ohraničený automat dostane na vstup slovo a má overiť, či to slovo rešpektuje daný vzor. Automat si tipne, ktoré časti slova prislúchajú k jednotlivým častiam vzoru a potom overí, či je to naozaj tak. Napríklad musí skontrolovať, či časti slova vzniknuté nahradením viacnásobného vyskytu jednej premennej sú zhodné. To sa dá tak, že si automat pri úvodnom tipovaní označí písmená takto vzniknutých podslov a potom pri viacnásobnom (ale určite konečnom) počte prechodov cez vstup overí, či sa písmená na zhodných pozíciách vytipovaných podslov zhodujú (značí si, čo už skontroloval).

□

Veta 2.2.13: Nech $|\Sigma| = 1$. Potom jazyky tried $PL_{E,\Sigma}$ a $PL_{NE,\Sigma}$ sú regulárne.

Teraz zadefinujeme relácie, ktoré budú užitočné pri dokazovaní niektorých vlastností jazykov definovaných vzormi, hlavne pri dokazovaní istých rohododovacích problémov, tak isto sa pomocou nich dajú iným spôsobom definovať jazyky generované vzormi.

Definícia 2.2.14 (relácie $\leq$ a $\leq_+$):
Nech π_1 a π_2 sú vzory nad $(\Sigma \cup V)^+$ a nech $h:(\Sigma \cup V) \to (\Sigma \cup V)^*$ je homomorfizmus stabilný pre terminály

$\pi_1 \leq \pi_2$	práve vtedy, keď	$\pi_1 = h(\pi_2)$,	pre ľubovoľné h s uvedenou vlastnosťou
$\pi_1 \leq_+ \pi_2$	práve vtedy, keď	$\pi_1 = h(\pi_2)$,	h je nevymazávajúce
$\pi_1 \equiv \pi_2$	práve vtedy, keď	$\pi_1 = h(\pi_2)$,	h je premenovanie premenných

Zdôrazňujeme, že na rozdiel od homomorfizmov z množiny $H_{\Sigma,V}$, ktoré každú premennú zobrazia na terminálne slovo, v tomto prípade sa premenná môže zobraziť aj na reťazec z $(\Sigma \cup V)^*$ (terminály ponechá nezmenené).

Je zrejmé, že uvedené relácie sú tranzitívne.

Nie je ťažké si uvedomiť, že pre každý vzor α a β platí: Ak $\alpha \leq_+ \beta$, potom $L_{NE}(\alpha) \subseteq L_{NE}(\beta)$ a podobne, ak $\alpha \leq \beta$, tak potom $L_E(\alpha) \subseteq L_E(\beta)$.
Pre lepšiu predstavu si zoberme nejaké vzory α, β také, že $\alpha \leq_+ \beta$, teda $\alpha = h(\beta)$. Terminály vo vzore β sa zobrazia sami na seba (h je ako z definície uvedených relácií). Každá premenná z β sa vo všeobecnosti zobrazí na nejaký neprázdny reťazec zo $(\Sigma \cup V)^+$ (môže to byť iná premenná, reťazec premenných, terminálne slovo, ...) čiže vzor α je v istom zmysle *špecifickejší* (resp. *menej všeobecný*) ako β, čím sa zúži množina potenciálne vygenerovateľných slov, pričom je zrejmé, že to, čo sa dá vygenerovať pomocou α sa dá určite aj pomocou β.

Príklad 2.2.15: Nech $\beta = X$, $h(X) = YY$, Σ nech je abeceda terminálnych symbolov.
Nech $\alpha = h(\beta) = YY$. $L_{NE}(\beta) = \Sigma^+$, $L_{NE}(\alpha) = \{ww \mid w \in \Sigma^+\}$, ľahko vidno, že $L_{NE}(\alpha) \subseteq L_{NE}(\beta)$.

Ak vzor $\alpha = h(\beta)$, tak hovoríme, že α je inštanciou β. Ak navyše platí, že $\alpha \in \Sigma^*$, tak potom sa nazýva terminálnou inštanciou β.

Pomocou takto definovaných relácii sa teraz dajú ľahko definovať jazyky generované E- resp. NE-vzorom π:

$$L_{E,\Sigma}(\pi) = \{w \in \Sigma^* \mid w \leq \pi\}$$
$$L_{NE,\Sigma}(\pi) = \{w \in \Sigma^* \mid w \leq_+ \pi\}$$

Pre $\leq_+$ navyše platí: $\pi_1 \equiv \pi_2$ práve vtedy, keď $\pi_1 \leq_+ \pi_2$ a súčasne $\pi_2 \leq_+ \pi_1$.
Pre π_1 a π_2 spĺňajúce $\pi_1 \leq \pi_2$ a $\pi_2 \leq \pi_1$ nie je známa žiadna jednoduchá charakterizácia a táto skutočnosť spôsobuje ťažkosti pri rozhodovaní ekvivalencie nevymazávajúcich vzorov.

2.3 Problém ekvivalencie a inklúzie

Triedy jazykov PL_{NE} a PL_E sú jedny z mála prirodzene definovaných tried, pre ktoré je problém ekvivalencie rozhodnuteľný (pre PL_{NE} dokonca triviálne) a problém inklúzie nerozhodnuteľný (pre PL_{NE} aj PL_E). Ekvivalencia pre PL_E je až na najeké špeciálne prípady zatiaľ otvorený problém, ale predpokladá sa, že by tento problém mal byť rozhodnuteľný (aj keď zatiaľ je teoreticky možný akýkoľvek záver).

Problém inklúzie úzko súvisí aj s niektorými inými oblasťami, napríklad veľa problémov v kombinatorike na slovách sa dá vyjadriť v termínoch problému inklúzie pre pattern languages, tak isto aj isté problémy z oblasti prepisovania termov. Z tohoto hľadiska ani nie je veľkým prekvapením, že sa daný problém ukázal byť nerozhodnuteľný. Avšak vzhľadom na to, že dokazovanie ekvivalencie pre PL_{NE} je pomerne jednoduché, sa zdá ,že ľudia, ktorí s týmto problémom pracovali, si zrejme mysleli opak, teda že problém inklúzie by mal byť rozhodnuteľný. Poďme sa teraz bližšie pozrieť na problém ekvivalencie pre NE-pattern languages.

Definícia 2.3.1: Nech Σ je abeceda terminálnych symbolov a nech $|\Sigma| \geq 2$. Nech $V=\{X_1, X_2, ...\}$ je množina premenných. Uvažujme ďalej nasledovnú možinu homomorfizmov stabilných pre terminály, pre ktoré platí:

$$f_a(X_i) = a$$
$$f_b(X_i) = b$$

$$g_j(X_i) = a \qquad \text{ak } i = j$$
$$g_j(X_i) = b \qquad \text{ak } i \neq j$$

a g_j je definované pre všetky $j \geq 1$ a tak isto aj $i \geq 1$.
Pre daný vzor α definujme $F(\alpha) = \{ f_a(\alpha), f_b(\alpha) \} \cup \{ g_i(\alpha) \mid i \geq 1\}$. $F(\alpha)$ je konečný jazyk a platí: $F(\alpha) \subseteq L_{NE}(\alpha)$.

Poznámka 2.3.2: Keby $|\Sigma| = 1$, tak potom by pattern languages nad Σ boli regulárne a v takom prípade by sa ekvivalencia dala určiť ľahko.

Možno stojí za povšimnutie, že ak α neobsahuje premenné, tak $F(\alpha) = L_{NE}(\alpha) = \{\alpha\}$, teda $|F(\alpha)| = 1$, ak α obsahuje 1 premennú, tak $|F(\alpha)| = 2$ a ak α obsahuje $n \geq 2$ premenné, tak $|F(\alpha)| = n+2$.

Lema 2.3.3: Nech α a β sú vzory rovnakej dĺžky, teda $|\alpha| = |\beta|$, a nech $F(\alpha) \subseteq L_{NE}(\beta)$. Potom platí: $\alpha \leq_+ \beta$.

Lema 2.3.4: Nech α a β sú vzory také, že $|\alpha| = |\beta|$. Potom $L_{NE}(\alpha) \subseteq L_{NE}(\beta)$ práve vtedy, keď $\alpha \leq_+ \beta$.

Dôkaz: Nech $L_{NE}(\alpha) \subseteq L_{NE}(\beta)$. Keďže $F(\alpha) \subseteq L_{NE}(\alpha)$, tak potom $F(\alpha) \subseteq L_{NE}(\beta)$ a podľa lemy 2.3.3 platí: $\alpha \leq_+ \beta$. Opačná implikácia je triviálna.

Veta 2.3.5: Pre PL_{NE} platí: $L_{NE}(\pi_1) = L_{NE}(\pi_2)$ práve vtedy, keď $\pi_1 \equiv \pi_2$.

Dôkaz: Ak $L_{NE}(\pi_1) = L_{NE}(\pi_2)$, tak potom nutne $|\pi_1| = |\pi_2|$, keďže používame nevymazávajúci homomorfizmus. Z lemy 2.3.4 dostávame $\pi_1 \leq_+ \pi_2$ a $\pi_2 \leq_+ \pi_1$, teda $\pi_1 \equiv \pi_2$. Opačná implikácia je triviálna.

□

Z dôkazu vyplýva algoritmus s lineárnou časovou zložitosťou na testovanie ekvivalencie pre PL_{NE}.

Veta 2.3.6: Je nerozhodnuteľné určiť, či $L_Z(\pi_1) \subseteq L_Z(\pi_2)$ pre dané vzory π_1 a π_2 $(Z \in \{E, NE\})$.

Pre Z = E, teda pre vymazávajúce vzory, sa veta dokazuje redukciou na problém existencie akceptačného výpočtu nedeterministického 2-počítadlového automatu bez vstupu, ktorý je nerozhodnuteľný (pozri napr. [o2]), kokrétnejšie: skonštruujú sa vzory $\alpha, \beta \in (\Sigma \cup V)^*$ také, že

$L_{E,\Sigma}(\alpha) \not\subseteq L_{E,\Sigma}(\beta)$ práve vtedy, keď 2-počítadlový automat má akceptačný výpočet. Pre nevymazávajúce vzory sa problém inklúzie dokazuje zase redukciou na ten istý problém pre vymazávajúce vzory. Tieto dôkazy sú pomerne dlhé a technické, celé ich možno nájsť v [4].

Už vieme, že platí nasledovné:
Ak $\pi_1 \leq_+ \pi_2 \Rightarrow L_{NE}(\pi_1) \subseteq L_{NE}(\pi_2)$ (platí aj pre PL_E a $\leq$)

Ak $|\pi_1| = |\pi_2|$, z lemy 2.3.4 vyplýva pre PL_{NE} aj opačná inklúzia. Pre rôzne dĺžky vzorov však táto inklúzia už vo všeobecnosti neplatí, dôkazom čoho je nasledujúci kontrapríklad:

Nech $\Sigma = \{a, b\}$, $\pi_1 = aXbaXXb$, $\pi_2 = XXY$.
Ak $w \in L_{NE}(\pi_1)$, potom w možno písať v tvare $w = aubauub$, $u \in \Sigma^+$.
Vhodným rozdelením slova w ľahko vidno, že v slove w možno zakaždým nájsť vzor $\pi_2 = XXY$, teda $w \in L_{NE}(\pi_2)$, treba si samozrejme daný prípad aspoň trochu rozobrať:

1) ak u = av, potom w=a|a|vbaavavb
2) ak u = bv, potom w=abvb|abvb|vb

teda $L_{NE}(\pi_1) \subseteq L_{NE}(\pi_2)$ a je zrejmé, že $\pi_1 \neq h(\pi_2)$, teda $\pi_1 \not\leq_+ \pi_2$.

Problém inklúzie je však rozhodnuteľný v špeciálnych prípadoch - napríklad pre E-patterns neobsahujúce terminály, avšak nie je známe, či niečo podobné platí pre NE-patterns bez terminálnych symbolov.

Zastavme sa ešte na chvíľu pri *probléme ekvivalencie pre E-patterns*. Videli sme, že dva NE-vzory (nevymazávajúce vzory) sú ekvivalentné práve vtedy, keď sú, až na názvy premenných, zhodné. Pre vymazávajúce vzory (E-patterns) nie je známy žiaden vzťah, podľa ktorého by sa dalo určiť, či sú ekvivalentné alebo nie, dokonca zostáva otvorenou otázka, či je problém evivalencie vôbec rozhodnuteľný. Ako uvidíme ďalej, známe sú len niektoré čiastkové výsledky týkajúce sa tohoto problému.

Kvôli zjednodušeniu práce s vymazávajúcimi vzormi definujeme pojem štandardnej reprezentácie.

Definícia 2.3.7: Nech $\alpha \in (\Sigma \cup V)^+$ je vzor (pattern). Pod štandardnou reprezentáciou α rozumieme rozklad $\alpha = \alpha_0 u_1 \alpha_1 u_2 \alpha_2 .....\alpha_{m-1} u_m \alpha_m$, kde $\alpha_0, \alpha_m \in V^*$, $\alpha_i \in V^+$ pre i = 1, ..., m-1, $u_j \in \Sigma^+$, j = 1, ..., m-1. M-ticu terminálnych symbolov $(u_1,....,u_m)$ budeme nazývať terminálový segment štandardnej reprezentácie.
Hovoríme, že vzory α a β sú podobné, ak ich štandardné reprezentácie majú rovnaký terminálový segment.

Veta 2.3.8: Nech $\alpha, \beta \in (\Sigma \cup V)^+$ Predpokladajme, že $L_{E,\Sigma}(\alpha) = L_{E,\Sigma}(\beta)$.

1. Ak $|\Sigma| \geq 3$, potom α a β sú podobné.
2. Nech $|\Sigma| \geq 4$ a nech $\alpha = \alpha_0 u_1 \alpha_1 u_2 \alpha_2 .....\alpha_{m-1} u_m \alpha_m$ a $\beta = \beta_0 u_1 \beta_1 u_2 \beta_2 .....\beta_{m-1} u_m \beta_m$
   (z 1. bodu vieme, že musia byť podobné).
   Potom $L_{E,\Sigma}(\alpha_i) = L_{E,\Sigma}(\beta_i)$ pre i = 0,....,m.

Opačné tvrdenie neplatí: teda z rovnosti $L_{E,\Sigma}(\alpha_i) = L_{E,\Sigma}(\beta_i)$ vo všeobecnosti rovnosť $L_{E,\Sigma}(\alpha) = L_{E,\Sigma}(\beta)$ nevyplýva.

Predchádzajúca veta neplatí, ak abeceda terminálnych symbolov má iba 2 prvky, dokazuje nám to nasledujúci príklad.

Príklad 2.3.9: Nech $\Sigma = \{0, 1\}$ a α = X01Y0Z a β = X0Y10Z nech sú vzory. Oba jazyky definované týmito vzormi sú rovnaké, teda $L_{E,\Sigma}(\alpha) = L_{E,\Sigma}(\beta)$ (dokonca sú regulárne), pričom vzory α a β nie sú podobné.

Dá sa ukázať, že pre podobné vymazávajúce vzory (similar E-patterns) je problém ekvivalencie aj inklúzie rozhodnuteľný, ak terminálna abeceda obsahuje 2 písmená, ktoré sa nevyskytujú vo vzoroch.

Veta 2.3.10: Nech $\alpha, \beta \in (\Sigma \cup V)^+$ sú podobné vzory. Ak Σ obsahuje dve rôzne písmená a,b nevyskytujúce sa v α a β, potom sú nasledujúce tvrdenia ekvivalentné:

1. $L_{E,\Sigma}(\alpha) \subseteq L_{E,\Sigma}(\beta)$
2. $\tau_{|\beta|,a,b}(\alpha) \in L_{E,\Sigma}(\beta)$
3. Existuje homomorfizmus h: var(β) $\rightarrow$ var(α)* taký, že h(β) = α.

Dôsledok 2.3.11: Pre dva podobné vzory α, β a abecedu Σ obsahujúcu dva rôzne terminály nevyskytujúce sa v α a β je rozhodnuteľné, či $L_{E,\Sigma}(\alpha) \subseteq L_{E,\Sigma}(\beta)$.

A z predchádzajúcich tvrdení vyplýva ďalej:
Dôsledok 2.3.12: Ak Σ obsahuje aspoň 3 písmená a zároveň aj dva rôzne terminály nevyskytujúce sa vo vzoroch, tak je rozhodnuteľný aj problém ekvivalencie pre E-patterns (vymazávajúce vzory).

Tieto výsledky naznačujú, že riešenie problému ekvivalencie pre E-patterns sa stáva jednoduchším, ak do terminálnej abecedy zavedieme nové písmená. Zatiaľ však zostáva *otvorenou* otázka, či sa ekvivalencia zachová po rozšírení abecedy, ak originálna abeceda má aspoň 3 písmená (pre 2-písmenovú abecedu sa nezachováva – pozri príklad vyššie).

Zdá sa, že hlavná príčina problémov v rozhodovaní ekvivalencie pre vymazávajúce vzory (E-patterns) je v tom, že E-patterns môžu obsahovať veľa nadbytočných premenných, ktoré nie sú potrebné pre definovanie jazyka. Napríklad každý bezterminálový E-pattern, ktorý obsahuje premennú vyskytujúcu sa v ňom práve raz, je ekvivalentný vzoru X. Z toho vyplýva, že aj veľmi rozdielne vyzerajúce E-vzory môžu byť ekvivalentné. Existuje istá technika na elimináciu nadbytočných premenných, ktorá sa dá použiť na riešenie ekvivalencie v istých špeciálnych prípadoch. Pre hlbšie štúdium naznačenej problematiky odporúčame [4] a [o8].

2.4 Pohľad z „druhej strany”

Teraz trochu bližšie predstavíme trochu iný pohľad na pattern languages, ktorý vychádza z pôvodnej motivácie pre ich skúmanie, uvedieme aj niekoľko otvorených problémov v tejto oblasti. Na upresnenie toho, čo napovedá nadpis tejto podkapitoly, povieme toľko, že teraz budeme mať najskôr k dispozícii nejaký jazyk (množinu vzoriek) a budeme hľadať čo možno *najlepší popisný vzor* (descriptive pattern), ktorý rešpektujú všetky slová v danom jazyku. Požiadavka na čo najlepší popisný vzor má zmysel, keby sme hľadali len nejaký popisný vzor, tak napríklad vzor X by vyhovoval pre každú neprázdnu množinu vzoriek, ten nám však nedáva žiadnu predstavu o štruktúre slov, ak tam nejaká existuje. Jazyk definovaný týmto hľadaným vzorom bude samozrejme často väčšia množina, ako jazyk, ku ktorému ten vzor hľadáme, musí však obsahovať všetky jeho slová. Najlepším vzorom sa myslí vzor, ktorý definuje čo možno najmenší jazyk (v zmysle množinovej inklúzie), ktorý je nadmnožinou pôvodného jazyka. Niekedy môžeme nájsť aj viacero “najlepších” vzorov, medzi ktorými nemožno rozhodnúť, ktorý z nich je lepší (jazyky definované týmito vzormi sú navzájom neporovnateľné a akýkoľvek vzor, ktorý definuje menšiu množinu ako ktorýkoľvek z “najlepších” vzorov už neobsahuje všetky slová z množiny vzoriek). Takýto prístup je obzvlášť vhodný, keď sa množina vzoriek (vzorových príkladov) zväčšuje, napríklad pri nejakom procese učenia. Tu sa pattern languages ukazujú ako veľmi prirodzený model na reprezentáciu nejakej množiny vzoriek.

V teórii strojového učenia sa neraz stretávame aj s dvoma množinami - prvky jednej z nich chceme zhrnúť do cieľového jazyka (môžeme ju nazvať množinou pozitívnych príkladov) a prvky druhej nie (množina negatívnych príkladov). Prvá z nich – množina pozitívnych príkladov. V takom prípade hľadáme vzor, ktorý má poposivať množinu pozitívnych príkladov a zároveň jazyk, ktorý je ním definovaný, nesmie obsahovať dané negatívne príklady.

Príklad 2.4.1:
F = {010100, 00100100, 01101100, 0001000100, 0111011100, 010110101100, 001010010100}
Ak sa bližšie pozrieme na slová, ktoré tvoria množinu F, určite si všimneme, že napríklad všetky slová začínajú znakom 0 a končia 00, túto skutočnosť možno ľahko vyjadriť vzorom 0X00. Skúsme nájsť niekoľko lepších nevymazávajúcich vzorov, ktoré popisujú množinu F: pri bližšom pohľade ľahko vidno, že dané slová rešpoktujú vzor 0X0X00, $X \in \{0, 1\}^+$. Ďalšie nevymazávajúce vzory popisujúce F sú napríklad: XX00, 0XX0, XXYY, 0X00, 0X100. Ak pripustíme aj vymazávajúce vzory, tak popisným vzorom môže byť napr. aj 0X10X100. V niektorých prípaduch sú prípustné aj iné typy vzorov okrem tých základných, ktoré sme definovali na začiatku druhej kapitoly. Premenných vo vzore môže byť viacero typov, napríklad X^R (premennú X^R nahrádzame reverzom slova, ktorým sme nahradili premennú X). V takom prípade je popisným vzorom pre množinu F aj $0X10X^R100$. Pri trochu bližšom pohľade musí byť hneď zrejmé, že nie všetky vzory popisujú F rovnako dobre. Najlepším popisom pre množnu F sú samozrejme tie vzory, ktoré generujú najmenšiu množinu (vzhľadom na množinovú inklúziu). V tomto prípade sú to 0X10X100 a $0X10X^R100$ (sú vymazávajúce, no samozrejme, niekedy môžeme chcieť nájsť čo možno najlepší nevymazávajúci vzor, v tomto prípade by to boli vzory 0X100 a 0X0X00). Takýmto prístupom sa pokúšať odhadnúť štruktúru slov napr. keď nepoznáme celú množinu (resp. výsledkov, ktoré možno popísať ako slová z nejakého jazyka). Pokiaľ získame nové výsledky alebo sa naša neúplná množina nejakým spôsobom rozrastie (napr. pri procese učenia), je dosť možné, že postupne budeme môcť vylúčiť niektoré z najlepších alternatívnych vzorov a priblížiť sa tak očakávanému tvaru slov, ktoré majú do množiny patriť.

Teraz stručne sformalizujme pojem popisný vzor (descriptive pattern):

Definícia 2.4.2: Nech $Z \in \{E, NE\}$ a Σ abeceda terminálnych symbolov. Hľadaný *popisný vzor* π pre množinu $F \subseteq \Sigma^+$ musí spĺňať:

1) $F \subseteq L_{Z,\Sigma}(\pi)$
2) Neexistuje π_0 taký, že $F \subseteq L_{Z,\Sigma}(\pi_0) \subset L_{Z,\Sigma}(\pi)$

Inak povedané, $L_{Z,\Sigma}(\pi)$ je najmenšia množina, ktorá obsahuje všetky slová z F. Ako už bolo spomenué v úvode, nedá sa vylúčiť, že nájdeme viacero vzorov π s danou vlastnosťou, pričom príslušné množiny jazykov $L_{Z,\Sigma}(\pi)$ budú navzájom neporovnateľné.

Ak F je konečná, vieme zostrojiť popisný vzor algoritmicky (pre NE- aj E- patterns), pričom pre NE-patterns existuje za splnenia určitých podmienok aj efektívny algoritmus (nie horší ako polynomiálny), pre E-patterns nie je známe, či existuje efektívny algoritmus. Taktiež nie známe, či nekonečná množina (napr. nejaký bezkontextový jazyk) nutne musí mať popisný vzor.

Pokiaľ uvažujeme konečné množiny pozitívnych aj negatívnych príkladov, označme si ich ako F_1 a F_2, tak potom hľadáme taký vzor π, ktorý spĺňa nasledujúcu podmienku:

$$F_1 \subseteq L_{NE}(\pi) \text{ a } F_2 \cap L_{NE}(\pi) = \varnothing$$

Problém hľadania takéhoto vzoru sa nazýva *pattern consistency problem*.

Nebudeme sa hlbšie púšťať do teórie strojového učenia, tu sme aspoň naznačili, aký význam môže mať nami študovaná problematika aj pre iné oblasti. Pre čitateľa s hlbším záujmom o naznačené problémy môžeme odporučiť napr. [3], [o3] (hľadanie popisného vzoru k danej množine), [o4] (efektívnejší algoritmus pre nájdenie popisnéhom vzoru v rámci triedy PL_{NE}), či [o5] (k pattern consistency problem).

2.5 Viacznačnosť

Na tomoto mieste si niečo povieme o viacznačnosti vo vzoroch alebo presnejšie, ak máme daný nejaký vzor α a slovo $w \in L_Z(\alpha)$ ($Z \in \{E, NE\}$), tak môže existovať viacero možností nahradenia premenných terminálnymi slovami takých, ktoré vedú k slovu w. Rôznym nahradeniam premenných budú samozrejme zodpovedať rôzne homomorfizmy, ktoré vzor α zobrazia na to isté slovo. Ide teda o istý druh nedeterminizmu vo vzoroch. Túto tému nebudeme rozoberať príliš podrobne, naznačíme však niektoré otvorené problémy. Podrobnejšie túto problematiku rozoberajú články [7] a [o6].

Príklad 2.5.1: Majme vzor $\pi = XYX$ a slovo z $L_{NE}(\pi)$, $w = a^3ba^3$.
Existujú 3 rôzne nevymazávajúce homomorfizmy h_i, pre ktoré $w = h_i(\pi)$, $i = 1, 2, 3$, konkrétne:

$h_1(X) = a$	$h_2(X) = a^2$	$h_3(X) = a^3$
$h_1(Y) = a^2ba^2$	$h_2(Y) = aba$	$h_3(Y) = b$

V prípade E-patterns existujú okrem troch uvedených ešte ďalšie dva homomorfizmy, ktoré zobrazia vzor π na slovo w, a to

$h_4(X) = \varepsilon$	$h_5(X) = a^3ba^3$
$h_4(Y) = a^3ba^3$	$h_5(Y) = \varepsilon$

Hovoríme, že stupeň neurčitosti slova w je 3 v prípade NE-patterns a 5 v prípade E-patterns.

Označenie: Stupeň neurčitosti slova w (degree of ambiguity) budeme označovať $deg_a(w)$.

Označenie: Nech π je vzor a w terminálny reťazec. Označme $H_Z(\pi, w, \Sigma)$, $Z \in \{E, NE\}$, množinu homomorfizmov (resp. NE-homomorfizmov) zo $(\Sigma \cup var(\pi))^* \to \Sigma^*$ stabilných pre terminály takých, že $h(\pi)=w$. Pod $var(\pi)$ rozumieme množinu premenných vyskytujúcich sa vo vzore π.

Definícia 2.5.2: Nech $Z \in \{E, NE\}$. *Stupeň neurčitosti* (degree of ambiguity) pre daný Z-pattern (Z-vzor) π je také číslo $k \geq 1$, pre ktoré platí:

1) $|H_Z(\pi, w, \Sigma)| \leq k$ pre všetky $w \in \Sigma^*$
2) $|H_Z(\pi, w, \Sigma)| = k$ pre nejaké $w \in \Sigma^*$

Ak také k neexistuje, potom stupeň neurčitosti je ∞.

Drobná poznámka k danej definícii: Nie je zlé si uvedomiť, že ak niektoré $w \in \Sigma^*$ nepatrí do jazyka definovaného daným vzorom, tak mohutnosť príslušnej množiny homomorfizmov je 0.

Ak stupeň neurčitosti vzoru π bude k, budeme to v ďalšom značiť ako $deg_a(\pi) = k$.

Prejdime ďalej, k jazykom:

Definícia 2.5.3: **N**ech Z ∈ {E, NE}. Z-pattern language L má stupeň neurčitosti k ≥ 1, ak $L = L_Z(\pi)$ pre nejaký vzor π, pre ktorý platí: $deg_a(\pi) = k$ a neexistuje π_1 taký, že $L = L_Z(\pi_1)$ a $deg_a(\pi_1) < k$. Ak také k neexistuje, potom $deg_a(L) = \infty$. Ak $deg_a(L) = 1$, hovoríme, že L je jednoznačný, v opačnom prípade hovoríme, že L je vnútorne viacznačný.

Veta 2.5.4: Nech π je nejaký NE-pattern. Potom $deg_a(\pi) = deg_a(L_{NE}(\pi))$.

Dôkaz: Uvedené tvrdenie triviálne vyplýva z faktu, že dva jazyky z PL_{NE} sú ekvivalentné práve vtedy, keď ich definujú dva vzory, ktoré sú až na názvy premenných, zhodné (pozri vetu 2.3.5).

Pre E-patterns analogická veta neplatí. Napríklad Σ^* sa dá definovať E-patternom X, ktorého stupeň neurčitosti je 1, preto aj stupeň neurčitosti jazyka Σ^* je 1 (teda je to jednoznačný jazyk). Ten istý jazyk je ale možné definovať aj E-patternom XY, ktorého stupeň neurčitosti je ∞.

Určovanie stupňa neurčitosti

Veta 2.5.5: Nech Z ∈ {E, NE}. Pre daný Z-pattern π a prirodzené číslo k vieme efektívne rozhodnúť, či π je viacznačný stupňa aspoň k.

Z predchádzajúcej vety vyplýva, že vieme efektívne rozhodnúť, či $deg_a(\pi)=k$, $deg_a(\pi)<k$ alebo či π je jednoznačný.

Otvoreným problémom zostáva rozhodovanie otázky, či $deg_a(\pi) = \infty$.

Z viet 2.5.4 a 2.5.5 vyplýva algoritmická rozhodnuteľnosť otázky, či $deg_a(L) = k$ pre nejaký NE-pattern language. Pre E-pattern languages (PL_E) zostáva táto istá otázka zatiaľ *otvoreným problémom* a situácia je v tomto prípade možno o to komplikovanejšia, že zatiaľ aj ekvivalencia pre PL_E je otvorený problém.

Je jednoduché konštruovať vzory s $deg_a = 1$ alebo ∞.
Vo všeobecnosti je ťažké dokázať, že daný vzor má konečný $deg_a > 1$.
Pre zaujímavosť možno uviesť, že napríklad taký vzor XabXbcaYabcY má $deg_a = 2$ (ako NE- aj E-vzor). Najlepší známy príklad vzoru s $deg_a = 3$ má dĺžku 324 a najkratšie slovo, ktoré má 3 rôzne rozklady podľa tohoto vzoru má dĺžku 1018.

Veta 2.5.6: Nech Z ∈ {E, NE}. Pre všetky m,n ≥ 0 sa dá efektívne skonštruovať Z-pattern so stupňom neurčitosti (deg_a) 2^m3^n.
Otvorená otázka je, či pre všetky k ≥ 1 existuje vzor so stupňom neurčitosti presne k.

Pre čitateľa, ktorého by táto problematika zaujala môžeme odporučiť napríklad [7] a [o6].

Kapitola 3

Multi-pattern languages

Prejdime teraz k prirodzenému zovšeobecneniu definície pattern languages z predchádzajúcich kapitol. Teraz sa nebudeme pri definícii jazyka obmedzovať len na jeden vzor, ale k dispozícii budeme mať konečnú množinu vzorov. V takom prípade bude výsledný jazyk zjednotením jazykov definovaných jednotlivými vzormi - budeme hovoriť o multi-pattern languages. Priebežne budeme daný model komplikovať aj tým, že premenné v danom vzore nebudeme nahrádzať ľubovoľným, prípadne ľubovoľným neprázdnym slovom, ako tomu bolo doteraz, ale ku každej premennej bude priradený konkrétny jazyk, ktorého slovami budeme nahrádzť príslušnú premennú.

Ešte krátka poznámka k praktickým aplikáciám: multi-patterns alebo multivzory možno v mnohých prípadoch použiť aj na oveľa lepší popis množiny vzoriek, ako keď hľadáme len jeden popisný vzor pre všetky slová. Môže ísť napríklad o množinu slov s dvoma rôznymi prefixami, v takom prípade sa na popis hodia oveľa lepšie dva vzory ako len jeden.

3.1 Základné definície

Definícia 3.1.1: Multivzor Π je konečná množina vzorov, teda $\Pi = \{\pi_1, ..., \pi_n\}$, $\pi_i \in (\Sigma \cup V)^+$, kde Σ je terminálna abeceda, $V = \{X_1, ..., X_n\}$ je množina premenných, $i = 1, ..., n$. Jazyk definovaný Z-multivzorom Π je $L_Z(\Pi) = L_Z(\pi_1) \cup \ldots \cup L_Z(\pi_n)$, $Z \in \{E, NE\}$.

Označme $MPL_E(n)$, $MPL_{NE}(n)$ – triedy vymazávajúcich resp. nevymazávajúcich multi-pattern languages stupňa n, tieto triedy dostaneme pomocou multivzoru s práve n prvkami, formálne:
$MPL_Z(n) = \{L \mid L = L_Z(\Pi)$ pre nejaký multivzor $\Pi = \{\pi_1, ..., \pi_n\}\}$, $Z \in \{E, NE\}$.

Ďalej označme MPL_E, MPL_{NE} – triedy vymazávajúcich resp. nevymazávajúcich multi-pattern languages: $MPL_Z = \cup_{n\geq 0} MPL_Z(n)$, $Z \in \{E, NE\}$.

Veta 3.1.2: Triedy jazykov MPL_E a MPL_{NE} sú až na existenciu jazykov obsahujúcich ε rovnaké, teda $MPL_E = MPL_{NE} \cup \{L \cup \{\varepsilon\}, L \in MPL_E\}$.

Na základe tejto vety budeme často písať miesto MPL_E a MPL_{NE} len MPL.

Napriek rovnosti tried MPL_E a MPL_{NE}, existujú rozdiely medzi vymazávajúcim a nevymazávajúcim prípadom, ak je stanovený pevný počet vzorov, teda triedy $MPL_E(n)$ a $MPL_{NE}(n)$ už vo všeobecnosti nie sú rovnaké pre pevné n.

Teraz dokážeme nekonečnú hierarchiu tried $MPL_Z(n)$, $Z \in \{E, NE\}$.

Veta 3.1.3: Triedy jazykov $MPL_E(n)$ a $MPL_{NE}(n)$ tvoria nekonečnú hierarchiu pre $n \geq 1$:

1) $MPL_E(n) \subset MPL_E(n+1)$
2) $MPL_{NE}(n) \subset MPL_{NE}(n+1)$

Dôkaz: Uvažujme postupnosť prvočísel: $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, ...
Majme ďalej jednoprvkovú abecedu $\Sigma = \{a\}$ a multivzor $\Pi_k = \{X^{p1}, X^{p2}, ..., X^{pk}\}$, $k \geq 1$ a nech $Z \in \{E, NE\}$. Slovo $a^{pk+1} \in L_{Z,\Sigma}(\Pi_{k+1})$, stačí nahradiť premennú X vo vzore X^{pk} slovom a. Pozrime sa teraz na to, či sa dané slovo a^{pk+1} nachádza v jazyku $L_{Z,\Sigma}(\Pi_k)$. Štruktúra slov v tomto jazyku vyzerá nasledovne: premenné nahrádzame slovami zo $\Sigma^* = a^*$ (v prípade E-vzorov) resp. zo $\Sigma^+ = a^+$ (v prípade NE-vzorov), teda na základe definície budú v $L_{Z,\Sigma}(\Pi_k)$ slová $a^{m.p1}$, $a^{m.p2}$, ..., $a^{m.pk}$, $m \geq 0$ (pre E-vzory) resp. $m \geq 1$ (pre NE-vzory). Pre $m \geq 2$ budú teda exponenty neprvočíselné, najdlhšie slovo s prvočíselným exponentom je a^{pk}, teda a^{pk+1} určite nepatrí do $L_{Z,\Sigma}(\Pi_k)$, teda tvrdenie vety platí.

□

Dôsledok 3.1.4: $PL_Z \subset MPL$, $Z \in \{E, NE\}$.

Dôkaz: Je zrejmé, že $PL_Z = MPL_Z(1)$, $Z \in \{E, NE\}$. Ostatné vyplýva z definície 3.1.1 a vety 3.1.3.

□

3.2 Uzáverové vlastnosti MPL

Podobne ako triedy PL_E a PL_{NE}, aj trieda MPL má veľmi zlé uzáverové vlastnosti, ktoré sú zhrnuté v nasledujúcej vete

Veta 3.2.1: MPL je anti-AFL a nie je uzavretá ani na prienik a komplement.

Dôkaz: je takmer identický s dôkazom uzáverových vlastností pre triedy PL_E a PL_{NE}. V prípade jazykov nepatriacich do MPL možno použiť podobné argumenty ako pre PL_E, PL_{NE} (pozri vetu 2.2.8).

Zjednotenie, zreťazenie:
Jazyky a^+ a $b \in MPL$, $a^+ \cup b \notin MPL$ ani $a^+b \notin MPL$

Iterácia:
$\{ab\} \in MPL$, $\{ab\}^+ \notin MPL$

Prienik s regulárnymi jazykmi:
$\Sigma^* \in MPL$ pre všetky Σ. Pre $|\Sigma| \geq 2$ existujú regulárne jazyky, ktoré nepatria do MPL. Nech je to opäť napr. a^+b. Veľmi ľahko vidno, že $a^+b \cap \Sigma^* = a^+b$.

Homomorfizmus (nevymazávajúci):
Nech $h(a) = ab$. Jazyk $a^+ \in$ MPL, no $h(a^+) = \{ab\}^+ \notin$ MPL.

Inverzný homomorfizmus:
Nech $h(a) = \varepsilon$, $h(b) = b$. Jazyk $\{b\} \in$ MPL, ale $h^{-1}(b) = a^*ba^* \notin$ MPL.

Prienik:
Dokážeme rovnako ako pre PL_Z, $Z \in \{E, NE\}$, teda:
Nech $\Sigma = \{a, b\}$ a $\alpha = XXab$, $\beta = XbaX$ nech sú vzory. Prienik jazykov definovaných týmito vzormi je: $L(\alpha) \cap L(\beta) = \{w \in \Sigma^* \mid w = uuab = vbav$ pre nejaké slová $u, v \in \Sigma^*\} =$
$= \{(ba)^k b(ba)^{k+1} b, k \geq 0\}$.
Jazyk obsahujúci takéto slová však nemôže patriť do MPL, pretože napríklad aa nie je podslovom žiadneho slova.

Komplement:
Jazyk $L = \{a, b\}^*ab\{a, b\}^* \in$ MPL, $L^c = b^*a^* \notin$ MPL

V nasledujúcej vete uvidíme aspoň jeden malý rozdiel v porovnaní s triedami PL_Z.

Veta 3.2.2: MPL je uzavretá na zjednotenie a zreťazenie pre dva jazyky nad rovnakou abecedou.

Dôkaz:
Zjednotenie:
Nech $L_1, L_2 \in$ MPL sú jazyky nad rovnakými abecedami generované multivzormi Π_1 resp. Π_2. Na základe definície 3.1.1 ľahko vidno, že jazyk $L = L_1 \cup L_2$ je generovaný multivzorom $\Pi = \Pi_1 \cup \Pi_2$.

Zreťazenie:
Ak zabezpečíme, že množiny premenných vyskytujúcich sa v multivzoroch Π_1 resp. Π_2 budú disjunktné (napríklad vhodným premenovaním premenných v jednom z multivzorov, napr. v Π_2, označme si Π_2 po premenovaní premenných ako Π_2'), tak potom multivzor Π generujúci jazyk L_1L_2 obsahuje vzory $\alpha\beta$, kde $\alpha \in \Pi_1$ a $\beta \in \Pi_2$'.

Pozrime sa ešte nakrátko na problém rozhodovania ekvivalencie a inklúzie pre MPL.

Veta 3.2.3: Problémy ekvivalencia aj inklúzie sú pre triedu MPL nerozhodnuteľné.

Dôkaz: Keďže $L_\Sigma(\alpha) \subseteq L_\Sigma(\beta)$ práve vtedy, keď $L_\Sigma(\beta) = L_\Sigma(\alpha, \beta)$, tak uvedené tvrdenie vyplýva z nerozhodnuteľnosti problému inklúzie pre PL_Z (veta 2.3.6), $Z \in \{E, NE\}$.

Uvedomme si, že veta platí bez ohľadu na to, či je zatiaľ otvorený problém ekvivalencie pre PL_E rozhodnuteľný.

Teraz pristúpime k ďalšiemu zovšeobecneniu triedy MPL a síce, že ku každej premennej priradíme nejaký jazyk, ktorého slovami ju budeme môcť nahrádzať. Už nebudeme rozlišovať medzi vymazávajúcimi a nevymazávajúcimi vzormi, v ďalších úvahách to nebude dôležité a to aj preto, že na veľkosť definovaných tried jazykov to až na prázdne slovo nebude mať žiaden vplyv, čo dokážeme neskôr pomocou všeobecnejšieho tvrdenia v ďalšej kapitole (lema 4.1.2).

Definícia 3.2.4: Nech Σ je terminálna abeceda a $V = \{ X_1, ..., X_n \}$ nech je množina premenných. Nech je ku každej premennej X_i je priradený nejaký jazyk $D_i \subseteq \Sigma^*$, $i = 1, ..., n$. Postupnosť jazykov $D_1, ..., D_n$ označme D. D_i budeme nazývať doménou X_i (zn.: $D_i = dom(X_i)$). Majme ďalej množinu homomorfizmov H_D, pre ktorú platí:

$$H_D = \{h:(\Sigma \cup V) \rightarrow \Sigma^* \mid h(a)=a \text{ pre všetky } a \in \Sigma,\ h(X_i) \in D_i\ ,\ X_i \in V,\ i \in \{1,...,n\}\}.$$

Jazyk definovaný vzorom $\pi \in (\Sigma \cup V)^+$ s postupnosťou domén premenných D je $L_D(\pi)=\{ h(\pi) \mid h \in H_D\}$.

Poznámka: Pre vymazávajúce (E-patterns) resp. nevymazávajúce vzory (NE-patterns), teda pre triedy PL_E a PL_{NE} a tak isto aj pre MPL_E a MPL_{NE} platí: $D_i = \Sigma^*$ resp. $D_i = \Sigma^+$ pre všetky $i = 1, ..., n$.

Definícia 3.2.5: Nech $\Pi = \{\pi_1, ..., \pi_n\}$ je multivzor, $V = \{X_1, ..., X_n\}$ množina premenných a $D = D_1, ..., D_n$ nech je postupnosť domén, ktoré sú priradené k premenným. Jazyk definovaný multivzorom Π s postupnosťou domén D je $L_D(\Pi) = L_D(\pi_1) \cup \ldots \cup L_D(\pi_n)$.

Označenie: Nech F je nejaká trieda jazykov. Potom triedu jazykov $\{L_D(\Pi) \mid \Pi$ je multivzor, $D = D_1, ...,D_n$ pre nejaké $D_1, ...,D_n \in F\}$ budeme označovať ako MPL_F. Teda napríklad pre F = FIN (REG, CF, ...) máme MPL_{FIN} (MPL_{REG}, MPL_{CF}, ...).

Príklad 3.2.6:
Nech $V = \{ X_1 \}$, $\Sigma = \{a, b\}$, $\pi_1 = X_1X_1$, $D_1 = \Sigma^*$

$$L_D(\pi_1) = \{ww \mid w \in \{a, b\}^*\}$$

V tomto prípade sa samozrejme jedná len o „obyčajný“ E-pattern, všimnime si však, že $L_D(\pi_1) \in MPL_{REG}$, hoci $L_D(\pi_1) \notin CF$.

Príklad 3.2.7:
Nech $V = \{ X_1, X_2 \}$, $\Sigma = \{a, b, c\}$, $\pi_2 = X_1X_2cX_2X_1$, $D_1 = \{a\}^+$, $D_2 = \{b\}^+$

$$L_D(\pi_2) = \{a^nb^mcb^ma^n \mid m,n \geq 1\}$$
$L_D(\pi_2) \in MPL_{REG}$, $L_D(\pi_2) \notin REG$

Príklad 3.2.8:
Nech $V = \{ X, Y \}$, $\Sigma = \{a, b, c\}$, $\pi_3 = baXaYX$, $dom(X) = D_1 = \{b\}^*$, $dom(Y) = D_2 = \{c\}^*$

$$L_D(\pi_3) = \{bab^mac^nb^m \mid m,n \geq 0\}.$$

Lema 3.2.9: Ak $L \subseteq \Sigma^*$ a súčasne $L \in F$, kde F je nejaká trieda jazykov, potom $L \in MPL_F$.

Dôkaz: Nech $V = \{ X_1 \}$, $\pi = X_1$, $D_1 = L$, odtiaľ ihneď dostávame: $L_D(\pi) = L$.

Nasledujúce lemy uvedieme bez dôkazu, ich platnosť vyplýva zo všeobecnejších tvrdení v ďalšej kapitole.

Lema 3.2.10: $\{ a^n b^n \mid n \geq 1 \} \notin MPL_{REG}$

Lema 3.2.11: $\{ a^n b^n c^n \mid n \geq 1 \} \notin MPL_{CF}$

Lema 3.2.12: Pre každý multivzor Π (nad abecedou Σ a množinou vzorov $V = \{ X_1, ..., X_n \}$) a $D = D_1, \ ... \ , D_n$, $D_i \in F$ $(i = 1, ..., n)$ existuje multivzor Π' (nad Σ a V) taký, že $L_D(\Pi) = L_{D'}(\Pi')$ pre $D' = (D_1 - \{\varepsilon\}, ... , D_n - \{\varepsilon\})$.

Kapitola 4

Nekonečné multivzory

V tejto kapitole pristúpime k ďalšiemu zovšeobecneniu, konkrétne, multivzor bude teraz môcť byť aj nekonečná množina, teda Π bude pozostávať z nekonečného počtu vzorov nad danou abecedou a množinou premenných. Samozrejme, že na množinu Π bude treba klásť nejaké rozumné obmedzenia. Budeme sa zaoberať prípadmi, keď Π bude regulárny alebo bezkontextový jazyk. Všimnime si, že v tomto prípade sa implicitne jedná o akúsi dvojúrovňovú štruktúru, kde na prvej úrovni bude musieť byť nejaké zariadenie (napr. gramatika) na definovanie nekonečnej množiny vzorov, ktoré potom na druhej úrovni interpretujeme.

4.1 Základné definície a vlastnosti

Definícia 4.1.1: Nech Π je nekonečný multivzor, $V = \{X_1, ..., X_n\}$ množina premenných a $D = D_1, ..., D_n$ nech je postupnosť domén, ktoré sú priradené k premenným. Jazyk definovaný nekonečným multivzorom Π s postupnosťou domén D je $L_D(\Pi) = \cup_{\pi \in \Pi} L_D(\Pi)$.

Označenie: Triedy jazykov definované pomocou nekonečných multivzorov bodeme označovať ako $PL(F_1, F_2)$, kde $\Pi \in F_1$ a domény premenných budú z F_2.

Poznámka: Pre už známe triedy typu MPL_F v zmysle práve definovaného označenia dostávame: $MPL_F = PL(FIN, F)$, kde F je nejaká trieda jazykov, z ktorej sú domény premenných.

Nasledujúce dve lemy nám v istom zmysle definujú akési normálne tvary domén premenných, presnejšie, že ak zakážeme konečné a bez-epsilonové domény premenných, veľkosť triedy $PL(F_1, F_2)$ s uvedenými obmedzeniami sa nezmení.

Lema 4.1.2: Nech F_1 je trieda jazykov uzavretá na zjednotenie a ľubovoľný homomorfizmus a F_2 nech je uzavretá na rozdiel s $\{\varepsilon\}$. Potom pre každý jazyk $L \in PL(F_1, F_2)$ existuje $\Pi' \in F_1$ a $D_1', ... , D_n' \in F_2$ také, že $L = L_{D'}(\Pi')$ a žiadne D_i', i = 1, ... , n, neobsahuje ε. (Samozrejme, že $L = L_D(\Pi)$ pre nejaké $\Pi \in F_1$ a domény pre premenné v postupnosti D sú z triedy F_2.)

Dôkaz: Predpokladajme, že $L = L_D(\Pi)$, pričom niektoré jazyky z D môžu obsahovať ε. Skonštruujeme Π' také, že $L = L_{D'}(\Pi')$, kde v D' budú tie isté jazyky ako v D, ale bez ε. Nech E_D je množina takých homomorfizmov, že pre každé $h \in E_D$ platí nasledovné:

$h(a) = a$ pre $a \in \Sigma$
$h(X_i) = X_i$ ak $\varepsilon \notin D_i$
$h(X_i) \in \{ \varepsilon, X_i \}$ ak $\varepsilon \in D_i$
(samozrejme, že i = 1, ..., n)

Množina E_D je konečná. Naša nová množina vzorov $\Pi = \cup_{(h \in E_D)} h(\Pi)$ a $D' = (D_1-\{\varepsilon\}, ..., D_n-\{\varepsilon\})$. Je zrejmé, že $\Pi' \in F_1$ a $L_D(\Pi) = L_{D'}(\Pi')$. Π' obsahuje všetky tie vzory, čo Π, okrem toho je obohatená o všetky také vzory, ktoré možno získať nahrádzaním premenných vo vzoroch z Π za ε (samozrejme len tam, kde príslušná doména danej premennej to ε obsahuje).

□

Lema 4.1.3: Nech F_1 je trieda jazykov uzavretá na zjednotenie a ľubovoľný homomorfizmus a F_2 nech je ľubovoľná trieda jazykov. Potom pre každý jazyk $L \in PL(F_1, F_2)$ existuje $\Pi \in F_1$ a $D_1, ..., D_n \in F_2$ také, že $L = L_D(\Pi)$ a každý z jazykov D_i (i = 1, ..., n) je nekonečný.

Dôkaz: Základná myšlienka je, že ak je k niektorej premennej (premenných je konečný počet) priradený konečný jazyk, tak príslušnú premennú nahrádzame postupne všetkými slovami z daného konečného jazyka vo všetkých vzoroch (vzorov môže byť samozrejme nekonečne veľa; zavedie sa vhodná množina homomrfizmov). Tým získame novú množinu vzorov, v ktorých sa premenná, ku ktorej bol priradený konečný jazyk, už nevyskytuje a daný konečný jazyk môžeme vyškrtnúť z postupnosti jazykov priradených k premenným. Tento postup opakujeme, až kým sa nezbavíme všetkých premenných s konečnou doménou. Malo by byť zrejmé, že jazyk získaný z novej množiny vzorov bude ten istý.

Formálne: Nech $\Pi \in (V \cup \Sigma)^*$ a $D_1, ..., D_n \subseteq \Sigma^*$. Predpokladajme, že niektoré D_i $(1 \le i \le n)$ je konečné. Uvažujme E_i – množinu homomorfizmov $h: (V \cup \Sigma)^* \to (V \cup \Sigma)^*$ takú, že

$h(a) = a \quad a \in \Sigma$
$h(X_j) = X_j \quad$ pre $j \ne i$, $(1 \le j \le n)$
$h(X_i) \in D_i$

Množina E_i je konečná, takže aj naša nová množina vzorov $\Pi' = \cup_{h \in Ei} h(\Pi)$ patrí do triedy F_1. Vzory v Π' už neobsahujú premennú X_i. Pre $D' = D \setminus Di$ platí: $L_D(\Pi) = L_{D'}(\Pi')$. Opakovaním tohoto postupu pre každú premennú s konečnou doménou sa nakoniec dopracujeme k nejakej množine vzorov, povedzme Π'', a množine domén D", pre ktoré platí: $L_D(\Pi) = L_{D''}(\Pi'')$ a každý jazyk v postupnosti D" je nekonečný (môže sa stať aj to, že D" bude prázdna).

□

Na základe predchádzajúcich 2 liem budeme môcť pri práci s $PL(F_1, F_2)$ predpokladať, že domény premenných sú nekonečné a bez ε (okrem prípadu, keď $F_1 = SNG$).

Lema 4.1.4 (pumpovacia pre PL(REG, REG)): Pre každý jazyk $L \in PL(REG, REG)$ existujú konštanty p a s také, že každé $w \in L$, $|w| > p$, sa dá napísať v tvare $w = u_1 x u_2 x ... u_k x u_{k+1}$, $1 \le k \le s$, $x \ne \varepsilon$ a $u_1 x^i u_2 x^i ... u_k x^i u_{k+1} \in L$ pre všetky $i \ge 1$.

Dôsledok 4.1.5: Ak $L = L_D(\Pi)$ pre nekonečné $\Pi \in REG$, tak potom existuje $w \in L$ také, že $w = uxy$, $x \neq \varepsilon$ a $ux^iy \in L$ pre všetky $i \geq 1$. Na type jazykov z D nezáleží.

Lema 4.1.6 (pumpovacia pre PL(CF, CF)): Pre každý jazyk $L \in PL(CF, CF)$ existujú konštanty p a s také, že každé $w \in L$, $|w| > p$, sa dá napísať v tvare $w = u_1xvyu_2xvy...u_kxvyu_{k+1}$, $1 \leq k \leq s$, $xy \neq \varepsilon$ a $u_1x^ivy^iu_2x^ivy^i ...u_kx^ivy^iu_{k+1} \in L$ pre všetky $i \geq 1$.

Dôkaz: Nech $L = L_D(\Pi)$, $\Pi \in CF$, $D = (D_1, ..., D_n)$, $D_i \in CF$ ($i = 1, ..., n$). Keďže $\Pi \in CF$, platí pre neho pumpovacia lema pre bezkontextové jazyky, teda existuje číslo $q \geq 1$ také, že každé $\pi \in \Pi$, $|\pi| > q$, sa dá napísať v tvare $\pi = \alpha\beta\delta\gamma\rho$, $|\beta\delta\gamma| \leq q$, $\beta\gamma \neq \varepsilon$ a $\alpha\beta^i\delta\gamma^i\rho \in \Pi$ pre všetky $i \geq 1$. Tak isto aj pre každé D_j existuje konštanta $q_j \geq 1$ taká, že každé $z \in D_j$, $|z| > q_j$, sa dá napísať v tvare $z = uvwxy$, $|vwx| \leq q_j$, $vx \neq \varepsilon$ a $uv^iwx^iy \in D_j$ pre všetky $i \geq 1$.

Uvažujme teraz nasledovné hodnoty p a s:

$p = q.\max\{ q_j \mid j = 1, ..., n \}$

$s = q$

Uvažujme ďalej nejaké $w \in L$, $|w| > p$. Slovo w vzniklo z nejakého vzoru $\pi \in \Pi$ pomocou nejakého homomorfizmu $h \in H_D$, teda $w = h(\pi)$. Môžu nastať 2 prípady:

1) $|\pi| > q$. Takže $\pi = \alpha\beta\delta\gamma\rho$, $|\beta\delta\gamma| \leq q$, $\beta\gamma \neq \varepsilon$ a $\alpha\beta^i\delta\gamma^i\rho \in \Pi$ pre všetky $i \geq 1$.
   Je jasné, že $h(\alpha\beta^i\delta\gamma^i\rho) = h(\alpha)h(\beta)^ih(\delta)h(\gamma)^ih(\rho) \in L$ pre všetky $i \geq 1$ a $h(\beta)h(\gamma) \neq \varepsilon$ (na základe lemy 4.1.2 predpokladáme bez-ε domény). Tvrdenie lemy platí.

2) $|\pi| \leq q$. V takomto prípade existuje nejaké j, $1 \leq j \leq n$, také, že X_j sa aspoň raz vyskytne v π a $|h(X_j)| > q_j$. Potom možno π napísať v tvare $\pi = \alpha_1X_j\alpha_2X_j ...\alpha_kX_j\alpha_{k+1}$, pričom X_j sa nevyskytuje v žiadnom z $\alpha_1, ..., \alpha_{k+1}$.
   Je jasné, že $k \leq q$. Podľa pumpovacej lemy pre CF-jazyky $h(X_j) = uxvyz$, $vy \neq \varepsilon$ a $ux^ivy^iz \in D_j$ pre všetky $i \geq 1$. Potom
   $w = h(\pi) = h(\alpha_1X_j\alpha_2X_j ...\alpha_kX_j\alpha_{k+1}) = h(\alpha_1)uxvyzh(\alpha_2)uxvyz ...h(\alpha_k)uxvyzh(\alpha_{k+1})$.
   Teraz stačí len vhodne rozdeliť vzniknuté slovo a lema bude dokázaná.

   Konkrétne: $h(\alpha_1)u \mid xvy \mid zh(\alpha_2)u \mid xvy \mid zh(\alpha_3)u \mid ... \mid zh(\alpha_k)u \mid xvy \mid zh(\alpha_{k+1})$

u_1	xvy	u_2	xvy	u_3	u_k	xvy	u_{k+1}

   Ak teraz uvažujeme iný homomorfizmus z H_D, ktorý premennú X_j nahradí slovom ux^ivy^iz pre nejaké $i \geq 1$ a všetky ostatné premenné tým istým slovom ako pôvodný homomorfizmus, tak dostaneme, že slovo $u_1x^ivy^iu_2x^ivy^i ...u_kx^ivy^iu_{k+1}$ je tiež z L. To sa dá samozrejme urobiť pre všetky $i \geq 1$, čbtd.

□

Dôsledok 4.1.7: Ak $L = L_D(\Pi)$ pre nekonečné $\Pi \in CF$, tak potom existuje $w \in L$ také, že $w = uxvyz$, $xy \neq \varepsilon$ a $ux^ivy^iz \in L$ pre všetky $i \geq 1$. Na type jazykov z D nezáleží.

Dôkaz: Ako v predchádzajúcej leme, využijú sa pumpovacie vlastnosti Π a prípad 1).

Dôsledok 4.1.8: Ak $L = L_D(\Pi)$, $D = (D_1, ..., D_n)$, $D_i \in REG$ ($i = 1, ..., n$), každé D_i je nekonečné a ak Π obsahuje aspoň 1 vzor, ktorý obsahuje aspoň 1 premennú (Π môže byť ľubovoľného typu), potom existuje $w \in L$ také, že $w = u_1xu_2x ...u_kxu_{k+1}$, $x \neq \varepsilon$ a $u_1x^iu_2x^i ...u_kx^iu_{k+1} \in L$ pre všetky $i \geq 1$.

Dôkaz: Uvažujme vzor $\pi \in \Pi$, ktorý obsahuje aspoň 1 premennú a jednu jeho premennú, napr. X_j, nahraďme dostatočne dlhým reťazcom $uxv \in D_j$, na ktorý možno použiť pumpovaciu lemu pre regulárne jazyky.

□

Lema 4.1.9: Nech $F \in \{REG, CF\}$. Potom platí: $PL(FIN,F) \subset PL(REG,F) \subset PL(CF,F)$.

Lema 4.1.10: $\{a^nb^n \mid n \geq 1\} \notin PL(REG, REG)$

Dôkaz: vyplýva z lemy 4.1.4

Lema 4.1.11: $\{a^nb^nc^n \mid n \geq 1\} \notin PL(CF, CF)$

Dôkaz: vyplýva z lemy 4.1.6

Na záver ako inšpiráciu pre ďalšie rozšírenie si obzoru spomeňme napríklad porovnanie Multi pattern languages s paralelnými komunikujúcimi systémami gramatík, kde bolo dokázané, že pattern languages sa dajú simulovať pomocou PCGS. Takýmto porovnaním možno získať presnejšiu predstavu o sile PCGS, navyše, už známe výsledky pre pattern languages sa môžu dať použiť aj pre PCGS, viac informácii spolu so známymi výsledkami možno nájsť v[8].

Kapitola 5

Gramatiky založené na vzoroch

Gramatiky založené na vzoroch

V tejto kapitole ukážeme trochu iný spôsob nahrádzania premenných reťazcami. Využijeme pritom obvyklú stratégiu známu z teórie formálnych jazykov. Budeme vychádzať z konečnej množiny terminálnych slov, ktorými budeme nahrádzať premenné v danej konečnej množine vzorov. Tým získame ďalšie slová, ktoré budeme môcť ďalej používať na nahrádzanie premenných v našej množine vzorov. V princípe možno uvažovať dva základné spôsoby získavania nových slov: takzvaný *nesynchronizovaný* - za premenné možno dosadzovať hicijaké doteraz odvodené slová alebo *synchronizovaný* – za premenné sa budú dosadzvať len slová získané v predchádzajúcom kroku odvodenia. Z tejto oblasti sú zatiaľ známe len viaceré základné výsledky, prípadné hlbšie prepojenia napr. s Chomského gramatikami, L-systémami, ako aj možné aplikácie tohoto prístupu zatiaľ zostávajú otvorené.

5.1 Základné definície a vlastnosti

Definícia 5.1.1: Gramatika založená na vzoroch je štvorica $G = (\Sigma, V, A, P)$, kde Σ je abeceda terminálnych symbolov, V je množina premenných, $A \subseteq \Sigma^*$ je konečná množina axióm, $P \subseteq (\Sigma \cup V)^* V(\Sigma \cup V)^*$ je konečná množina vzorov (patterns), pričom, ako je zrejmé z definície, v tomto prípade musí každý vzor obsahovať aspoň 1 premennú.

Podľa spôsobu odvodzovania slov budeme príslušnú gramatiku nazývať buď synchronizovanou alebo nesynchronizovanou.

Označenie: Majme množinu vzorov P a jazyk $L \subseteq \Sigma^*$. Množinu slov, ktorú získame nahradením všetkých premenných vo vzoroch z P reťazcami z L, označíme P(L).

Formálne:
$P(L) = \{u_1x_{i1}u_2x_{i2} \ldots u_kx_{ik}u_{k+1} \mid u_1X_{i1}u_2X_{i2} \ldots u_kX_{ik}u_{k+1} \in P;\ u_i \in \Sigma^*;\ X_{ij} \in V;\ x_{ij} \in L;\ i = 1, \ldots, k+1;\ j = 1, \ldots, k\}$.

Alternatívny zápis pomocou množiny homomorfizmov stabilných pre terminály používanej v predchádzajúcich kapitolách: $P(L) = \{h(\alpha), \alpha \in P\}$
Homomorfizmus $h \in \{h:(\Sigma \cup V) \rightarrow \Sigma^*$, $h(a) = a$ pre $a \in \Sigma$, $h(X) \in L$ pre všetky $X \in V\}$

Pre istotu ešte dodajme, že tak, ako doteraz, rôzne výskyty tej istej premennej sa nahrádzajú rovnakým reťazcom.

Definícia 5.1.2: Jazyk generovaný nesynchronizovanou gramatikou G (zn: NL(G)) je najmenšia množina $L \subseteq \Sigma^*$, pre ktorú platí:

1) $A \subseteq L$
2) $P(L) \subseteq L$

NL(G) teda obsahuje všetky slová, ktoré možno získať tak, že premenné vo vzoroch nahrádzame doteraz získanými slovami, vychádzame pri tom z axióm. Každé slovo z NL(G) sa dá získať konečným počtom použití množiny vzorov.

Dá sa to vyjadriť aj takto:
$NL(G) = A \cup P(A) \cup P(A \cup P(A)) \cup P(A \cup P(A) \cup P(A \cup P(A)))...$

Definícia 5.1.3: Jazyk generovaný synchronizovanou gramatikou G (zn: SL(G)) je:
$SL(G) = A \cup P(A) \cup P(P(A)) \cup P(P(P(A))) \cup ...$

Triedy takto získaných jazykov označíme PL_{NG} pre nesynchronizované gramatiky resp. PL_{SG} pre synchronizované gramatiky založené na vzoroch. V prípade synchronizovaných gramatík sa budeme bližšie zaoberať jazykmi generovanými takýmito gramatikami s jednoprvkovou množinou vzorov, v tomto špeciálnom prípade sa príslušnú triedu jazykov označíme ako iterované (iterated) pattern languages, skrátene IPL. Avšak najprv sa budeme podrobnejšie zaoberať triedou PL_{NG}.

Najskôr na niekoľkých príkladoch ilustrujeme, ako tieto gramatiky pracujú. Uvedieme aj príklady jazykov, ktoré nepatria do triedy definovanej nesynchronizovanými gramatikami a ako už býva zvykom, takéto jazyky nám neskôr poslúžia na dokazovanie určitých vlastností danej triedy jazykov.

Príklady jazykov, ktoré sú v PL_{NG}:

Príklad 5.1.4:
$G_1 = (\{a\}, \{X\}, \{a\}, \{aX\})$
Máme jedinú axiómu – „a“, takže za premennú X v jedinom vzore „aX“ môžeme dosadiť len „a“. Tým získame slovo „aa“, ktoré potom „v druhom kole“ opäť môžeme dosadiť za premennú X do vzoru „aX“ a získame slovo „aaa“ atď.....
Teda $NL(G_1) = \{a\} \cup \{aa\} \cup \{aa, aaa\} \cup \{aa, aaa, aaaa\} \cup .... = \{a^n; n \geq 1\}$.

Príklad 5.1.5:
$G_2 = (\{a\}, \{X\}, \{a\}, \{XX\})$
$NL(G_2) = \{a\} \cup \{aa\} \cup \{aa, aaaa\} \cup \{aa, aaaa, aaaaaaaa\} \cup ... = \{a\text{^}2^n; n \geq 0\}$

Príklad 5.1.6:

$G_3 = (\{a, b\}, \{X\}, \{a, b\}, \{XX\})$

$NL(G_3) = \{a, b\} \cup \{aa, bb\} \cup \{aa, bb, aaaa, bbbb\} \cup ... = \{a\text{^}2^n;\ n \geq 0\} \cup \{b\text{^}2^n;\ n \geq 0\}$

Príklad 5.1.7:

$G_4 = (\Sigma, \{X_1, X_2\}, \Sigma \cup \{\varepsilon\}, \{X_1X_2\})$

$NL(G_4) = \Sigma^*$

Príklad 5.1.8:

$G_5 = (\{a, b\}, \{X\}, \{ab\}, \{aXb\})$

$NL(G_5) = \{ab\} \cup \{aabb\} \cup \{aabb, aaabbb\} \cup ... = \{a^nb^n;\ n \geq 1\}$

Príklad 5.1.9:

$G_6 = (\{a, b\}, \{X_1, X_2\}, \{\varepsilon\}, \{X_1X_2, aX_1b\})$

$NL(G_6) = \{\varepsilon\} \cup \{\varepsilon, ab\} \cup \{\varepsilon, ab, abab, aabb\} \cup \{\varepsilon, ab, abab, aabb, abababab, aababb,$
$aabbaabb, aaabbb\} \cup ...$

$L(G_6)$ je Dyckov jazyk nad abecedou $\{a, b\}$.

A ešte jeden jednoduchý príklad, ktorý jasnejšie poukáže na rozdiely medzi PL_{NG} a PL_{SG} (uvidíme neskôr):

Príklad 5.1.10:

$G_7 = (\{a, b\}, \{X_1, X_2\}, \{a, b\}, \{X_1X_2\})$

$NL(G_7) = \{a, b\} \cup \{aa, ab, ba, bb\} \cup \{aa, ab, aaa, aab, aba, abb, ba, bb, baa, bab, bba,$
$bbb, aaaa, aaab, ...(\text{všetky ostatné slová dĺžky } 4)\} \cup ... = \{a, b\}^+$

Veta 5.1.11: Nech $L \subseteq \Sigma^*$ je konečný jazyk. Potom L, L^+, L^* patria do PL_{NG}.

Dôkaz: $L = L(G_1)$, kde $G_1 = (\Sigma, \{X\}, L, \{X\})$

$L^+ = L(G_2)$, kde $G_2 = (\Sigma, \{X_1, X_2\}, L, \{X_1X_2\})$ (porovnaj s príkladom 7)

$L^* = L(G_3)$, kde $G_3 = (\Sigma, \{X_1, X_2\}, L \cup \{\varepsilon\}, \{X_1X_2\})$ (p. príklad 4)

□

Príklady jazykov, ktoré nie sú v PL_{NG}:

Lema 5.1.12: $L_1 = a^* \cup b^*$ a $L_2 = a^* \cup b$ nepatria do PL_{NG}.

Dôkaz: Sporom: Nech $L_1 = NL(G)$ pre nejakú gramatiku $G = (\{a, b\}, V, A, P)$. Žiadny vzor v P nemôže obsahovať terminál alebo 2 rôzne premenné, inak by sa dali generovať reťazce obsahujúce oba terminálne symboly („a“ aj „b“). Keďže L_1 je nekonečný, tak v P musia existovať vzory s dĺžkou väčšou ako 1. Berúc do úvahy všetky obmedzenia vyplývajúce z doterajších úvah, tak jedinou možnosťou zostáva, že všetky vzory v P majú tvar X^i, $i \geq 1$. Nech teda množina vzorov P vyzerá takto:

$$P = \{V^{i1}, V^{i2}, ..., V^{ik}\}\ (i_j \geq 1, k \geq 1, j = 1, ..., k)$$

Z takejto množiny vzorov sa dajú generovať len reťazce tvaru a^m alebo b^m, pričom m môže byť len tvaru $m = i_1^{s1} i_2^{s2} i_3^{s3} ... i_k^{sk}$ ($s_j \geq 0$, j = 1, ..., k). To znamená, že m je buď 1 z čísel i_1, i_2, ..., i_k alebo súčinom ľubovoľných (nezáporných, celočíselných) mocnín týchto čísel. Z toho vyplýva, že sa nedá vygenerovať žiaden reťazec a^m alebo b^m s prvočíselným m väčším ako max $\{i_1, i_2, ..., i_k\}$, teda $L_1 \neq NL(G)$. Keďže neexistuje gramatika, ktorá by generovala L_1, tak samozrejme $L_1 \notin PL_{NG}$.

V prípade L_2 platia zčasti podobné argumenty, konkrétne: Každá gramatika generujúca L_2 musí v P obsahovať vzory tvaru X^i ($i \geq 1$) a aspoň v jednom takom vzore je $i > 1$, keďže L_2 je nekonečný. Tým pádom sa ale dajú generovať aj reťazce s väčším počtom b-čok ako 1, čo nechceme (a máme spor).

□

Lema 5.1.13: $L = \{a^{2\hat{}n} b^{2\hat{}n} ; n \geq 0\} \notin PL_{NG}$

Dôkaz: (opäť, ako inak, sporom)
Predpokladajme, že L = NL(G) pre nejakú gramatiku G = ({a, b}, V, A, P). Ak je v P vzor tvaru $u_1X_1u_2X_2u_3$, $u_1, u_2, u_3 \in (\{a, b\} \cup V)^*$, tak bez ohľadu na to, či $X_1 = X_2$ alebo $X_1 \neq X_2$, dajú sa vygenerovať reťazce tvaru $u1'a^{2\hat{}n} b^{2\hat{}n} u_2'a^{2\hat{}n} b^{2\hat{}n} u_3'$, ktoré do L nemôžu patriť. Podobne, ani vzory tvaru ubvXw alebo uXvaw nemôžu patriť do P, inak by do jazyka L museli patriť slová, ktoré by obsahovali podslovo ba. Takže každý vzor musí mať tvar a^jXb^k. Keďže každé slovo v jazyku obsahuje rovnaký počet a-čok a b-čok, musí platiť, že i = k. Keby sme teda mali v P vzor a^jXb^j, tak by sme mohli X nahradiť slovami ab aj a^2b^2, tým by sme získali $a^{j+1}b^{j+1}$ resp. $a^{j+2}b^{j+2}$. Tieto slová patria do L len vtedy, keď j = 0, teda P = {X}, z čoho vyplýva, že L(G) = A a to je samozrejme spor.

□

Nasledujúce dve lemy (5.1.14 a 5.1.16) nám do istej miery priblížia štruktúru slov triedy jazykov PL_{NG}.

Lema 5.1.14: Ak $L \in PL_{NG}$, tak potom existuje konštanta k taká, že pre všetky $w \in L$, $|w| > k$, existujú slová x, y, z, pre ktoré platí:

a) w = xyz
b) $xz \neq \varepsilon$
c) $y \in L$

Dôkaz: Majme nesynchronizovanú gramatiku G = (Σ, V, A, P), ktorá generuje L.. Nech $k = \max \{ |x|, x \in A \}$. Ak nejaké slovo $w \in L$ má väčšiu dĺžku ako k, tak potom to samozrejme nemôže byť axióma, čiže vzniklo niekoľkonásobným použitím vzorov z P. Aspoň jeden z týchto vzorov má dĺžku väčšiu ako 1. Zoberme posledný takýto vzor (s dĺžkou väčšou ako 1), ktorý bol použitý pri odvodení slova w. Predpokladajme, že je to vzor $u_1X_1u_2X_2...u_sX_su_{s+1}$, $u_i \in \Sigma^*$, $X_i \in V$ pre všetky zmysluplné i; $s \geq 1$. Potom $z = u_1w_1u_2w_2...u_sw_su_{s+1}$, $w_i \in L$, i = 1, ..., s. Potrebné rozdelenie slova w dostaneme takto: $x = u_1$, $y = w_1$, $z = u_2w_2...u_sw_su_{s+1}$.

□

Dôsledok 5.1.15: Do PL_{NG} nepatria nasledujúce jazyky:

- $L_1 = ab^+a$

- $L_2 = a^+b^+c^+$
- $L_3 = \{a^nb^nc^n; n \geq 1\}$

Lema 5.1.16: Ak $L \in PL_{NG}$ je nekonečný jazyk ($L \subseteq \Sigma^*$), potom existuje $u \in \Sigma^+$ také, že pre všetky $n \geq 1$ platí: $u^nv \in L$ alebo $vu^n \in L$.

Dôkaz: Nech nesynchronizovaná gramatika $G = (\Sigma, V, A, P)$ generuje L. Keďže L je nekonečný, tak v P sa musí nachádzať aspoň jeden vzor s dĺžkou viac ako 1, nech je to π. Rozlíšime 4 prípady:

a) $\pi = uXx$ ($u \in \Sigma^+, X \in V, x \in (\Sigma \cup V)^*$).
Ľahko vidno, že L v tomto prípade musí obsahovať slová požadovaného tvaru, konkrétne u^nzy_n, $n \geq 1$, $z \in A$, y_n sa získa z x nahradením prípadných premenných v x za slová z A.
Pre presnejšiu predstavu uvedieme podrobnejší postup:
Na začiatku máme len axiómy. Rozhodneme sa použiť vzor uXx. Premennú X nahradíme slovom $z \in A$ a všetky prípadné premenné v x za slová z A.
Vznkne nám nejaké slovo uzt_1. Opäť použijeme vzor uXx, pričom tentoraz dosadíme za x slovo uzt_1, vznikne slovo $uuzt_1t_2$, t_2 opäť vzniko z x rovnakým spôsobom ako v predošlom kroku t_1, teda nahradením prípadných premenných v x za slová z A. Označme $y_2 = t_1t_2$. Takto postupujeme ďalej, pričom v každom ďalšom kroku nahradíme X posledným novovzniknutým slovo a tak postupne vznikajú slová $uuuzt_1t_2t_3$ ($y_3 = t_1t_2t_3$), u^4zy_4, u^5zy_5, atď....., teda našli sme slovo u také, že $u^nv \in NL(G)$ ($v = zy_n$).

b) $\pi = xXu$ ($u \in \Sigma^+, X \in V, x \in (\Sigma \cup V)^*$)
Toto je v istom zmysle opačná situácia ako predošlý prípad, tu sa rovnakou metódou ako v predchádzajúcom prípade dopracujeme zase k záveru, že $vu^n \in NL(G)$.

c) $\pi = X_1xX_2$ ($X_1, X_2 \in V$, nezáleží na tom, či $X_1 = X_2$, $x \in (\Sigma \cup V)^+$)
Nech $z_1, z_2 \in A$ a y nech sa získa nahradením všetkých premenných v x za slová z A. Ak v n krokoch nahrádzame premennú X_1 slovom z_1, premenné v x stále tými istými premennými (teda zakaždým získame rovnaké y) a premennú X_2 vždy novovzniknutým slovom (na začiatku nahradíme X_2 slovom z_2), tak postupne získavame slová z_1yz_2, $z_1yz_1yz_2$, $z_1yz_1yz_1yz_2$, $z_1yz_1yz_1yz_1yz_2$, ...Teraz už ľahko vidno, že všeky slová tvaru $(z_1y)^nz_2 \in NL(G)$. Označme $z_1y = u$ a $z_2 = v$, teda opäť sme našli vhodné slová, ktoré spĺňajú tvrdenie lemy. Ešte treba podotknúť, že v prípade $A = \{\varepsilon\}$ musí byť počet terminálov v slove x väčší ako 1, inak by bol NL(G) konečný.

d) $\pi = X_1X_2...X_k$ ($k \geq 2$; $X_i \in V$ pre i = 1, ..., k; nezáleží na tom, či sa niektoré $X_i = X_j$)
Ak všetky vzory v P majú takýto tvar, potom musí v A existovať neprázdne slovo, inak by bol jazyk L konečný, nech je to z. Potom sa v NL(G) nachádzajú všetky slová tvaru $z^{k\wedge s}$, $s \geq 1$. Nech u = z, potom je zrejmé, že $u^nv \in NL(G)$ pre ľubovoľné $n \geq 1$ ($v = z^m$, pre vhodné $m \leq k^s - 1$).

□

Dôsledok 5.1.17: PL_{NG} neobsahuje jazyky s nasl. vlastnosťou: Ak $uz^kv \in L$, potom $z = \varepsilon$, $k \geq 2$.

5.2 Porovnanie PL_{NG} s inými triedami jazykov

Veta 5.2.1: trieda PL_{NG} je neporovnateľná s týmito triedami jazykov: REG, CF, DOL, OL, EOL, DTOL, TOL.

Dôkaz: Neporovnateľnosť PL_{NG} s REG a CF vyplýva z nasledujúcich faktov:

a) Jazyk $(a^* \cup b) \notin PL_{NG}$ a súčasne je regulárny (a teda samozrejme aj bezkontextový).

b) Jazyk $\{a\text{^}2^n; n \geq 0\} \in PL_{NG}$ a nie je bezkontextový (a samozrejme ani regulárny).

Neporovnateľnosť s DOL, OL:
$\{a^{2\text{^}n}b^{2\text{^}n}; n \geq 0\} \notin PL_{NG}$, ale patrí do DOL (dokonca do PDOL - pozri dodatok)
.... (teda aj OL, TOL, DTOL, EOL)

Z faktu, že TOL neobsahuje všetky konečné jazyky a zo vzťahov medzi triedami tu spomínaných OL-systémov (pozri dodatok) vyplýva neporovnateľnosť s DOL, OL, DTOL, TOL. Ešte potrebujeme nájsť jazyk z PL_{NG}, ktorý nepatrí do EOL: taký jazyk nám generuje napríklad gramatika $G = (\{a, b\}, \{X\}, \{a\}, \{Xb, bX, XX\})$.

□

Veta 5.2.2: $PL_{NG} \subset EDTOL$.

Dôsledok: $PL_{NG} \subset ECS$.

5.3 Uzáverové vlastnosti PL_{NG}

Veta 5.3.1: PL_{NG} nie je uzavretá na:

a) zjednotenie a zreťazenie s jednoslovnými jazykmi (z toho vyplýva aj neuzavretosť pre celú triedu PL_{NG}, keďže každý konečný jazyk je z PL_{NG})
b) prienik
c) prienik s regulárnymi jazykmi
d) komplement
e) Kleeneho $^+$
f) inverzný homomorfizmus

Dôkaz:

a) $a^* \in PL_{NG}$, $(a^* \cup b) \notin PL_{NG}$

Uvažujme ďalej jazyk $ab^+ = NL(G)$, kde $G = (\{a, b\}, \{X\}, \{ab\}, \{Xb\})$
$ab^+ \in PL_{NG}$, $ab^+a \notin PL_{NG}$

b) Nech $G_1 = (\{a, b\}, \{X\}, \{aa, ab\}, \{Xa, Xb\})$

$G_2 = (\{a, b\}, \{X\}, \{aa, ba\}, \{aX, bX\})$
Označme $L_1 = NL(G_1) = a\{a, b\}^+$ a $L_2 = L_1^R = \{a, b\}^+a$
$L = L_1 \cap L_2 = a\{a, b\}^+a$ - nepatrí do PL_{NG}, keby patril, tak podľa lemy 5.1.14 existuje konštanta k taká, že pre všetky $w \in L$, $|w| > k$, existujú slová x, y, z, pre ktoré platí: $w = xyz$, $xz \neq \varepsilon$, $y \in L$. Do prieniku však patria aj slová tvaru ab^ka, z čoho je hneď zrejmé, že stred resp. prefix alebo sufix slova nemôže mať požadovaný tvar, teda $L \notin PL_{NG}$.

c) Neuzavretosť na prienik s regulárnymi jazykmi vyplýva z prípadu b), keďže jazyky L_1 a L_2 sú regulárne.

d) Nech $G = (\{a, b\}, \{X\}, \{ab, ba\}, \{Xa, Xb, aX, bX\})$
Táto gramatika generuje jazyk $NL(G) = \{w \in \{a, b\}^* \mid \#_a w \geq 1, \#_b w \geq 1\}$. Komplement tohoto jazyka - jazyk $a^* \cup b^*$ však do PL_{NG} nepatrí, čo sme už dokázali.

e) Už vieme, že $L = \{a^nb^n, n \geq 1\} \in PL_{NG}$. Aby sme dokázali, že $L^+ \notin PL_{NG}$, opäť využijeme lemu 5.1.14. Slovo $a^nb^n \in L^+$ pre všetky $n \geq 1$. Nech teda $L^+ \in PL_{NG}$, potom pre dosť veľké n platí: $a^nb^n = xyz$, $xz \neq \varepsilon$, $y \in L^+$. Toto zodpovedá nejakému vzoru xXz z množiny vzorov nejakej gramatiky pre L^+. Vzor z dvoma premennými nemožno použiť, pretože za premenné môžeme dosadiť len slová z L^+, týmto spôsobom by sme určite nevygenerovali a^nb^n. Navyše, slovo z L^+, ktorým môžeme nahradiť premennú X tak, aby sme dostali a^nb^n, musí mať tvar a^jb^j. Z toho vyplýva, že $x = a^i$ a $y = b^i$ pre $i = n-j$. Lenže v L^+ je aj slovo $a^{2i}b^{2i}a^{2i}b^{2i}$, takže ak je v množine vzorov vzor xXz, tak potom nahradením premennej X slovom $a^{2i}b^{2i}a^{2i}b^{2i}$ by sme získali slovo $a^{3i}b^{2i}a^{2i}b^{3i} \notin L^+$, čo je spor.

f) Nech $G = (\{a, b, c\}, \{X\}, \{abc\}, \{Xbc\})$
Nech $L = NL(G) = a(bc)^+$
Definujme teraz homomorfizmus $h:\{a, b, c\}^* \rightarrow \{a, b, c\}^*$

$$h(a) = ab, h(b) = cb, h(c) = c$$

Jazyk $h^{-1}(L) = ab^*c$ nie je z PL_{NG}, opäť sa triviálne využije lema 5.1.14

□

Veta 5.3.2: PL_{NG} je uzavretá na (ľubovoľný) homomorfizmus a zrkadlový obraz.

Dôkaz: Nech $L = NL(G)$ pre nejakú gramatiku $G = (\Sigma_1, V, A, P)$ a nech $h: \Sigma_1^* \rightarrow \Sigma_2^*$ je nejaký homomorfizmus. Jazyk h(L) možno generovať gramatikou $G' = (\Sigma_2, V, A', P')$, kde $A' = h(A)$ a P' vznikne z P nahradením každého terminálu v každom vzore jeho homomorfným obrazom, premenné zostanú bez zmeny (teda pre premenné použijeme nejaký homomorfizmus $h_1: (\Sigma_1 \cup V)^* \rightarrow (\Sigma_2 \cup V)^*$, pre ktorý platí: $h_1(a) = h(a)$ pre $a \in \Sigma_1$ a $h_1(X) = X$ pre $X \in V$ (h_1 je stabilný pre premenné)).

Zrkadlový obraz: Nech $L = NL(G)$ pre nejakú gramatiku $G = (\Sigma, V, A, P)$. Gramatika G_1 pre L^R bude vyzerať nasledovne: $G_1 = (\Sigma, V, A^R, P^R)$.

□

5.4 Iterované pattern languages

Ako už bolo spomenuté na začiatku tejto kapitoly, v tomto prípade sa jedná iba o miernu modifkáciu spôsobu odvodzovania slov v jazyku pomocou gramatík založených na vzoroch. Len pre zopakovanie uveďme, že tu budeme uvažovať len jednoprvkovú množinu vzorov a že v každom kroku sa za premenné môžu dosadzovať len terminálne slová, ktoré vznikli v predchádzajúcom kroku odvodenia, pôjde teda o *synchronizovaný* spôsob odvodzovania slov z danej gramatiky založenej na vzoroch.

Definícia 5.4.1: Iterovaný pattern language je generovaný synchronizovanou gramatikou založenou na vzoroch s jednoprvkovou množinou vzorov $G = (\Sigma, V, A, \{\pi\})$, kde Σ je abeceda terminálnych symbolov, V je množina premenných, A je konečná množina axióm, $\pi \in (\Sigma \cup V)^* V(\Sigma \cup V)^*$ je vzor.

Jazyk generovaný takouto gramatikou budeme označovať ako SL(G), keďže ide len o špeciálny prípad synchronizovaných gramatík založených na vzoroch. Kôli zjednodušeniu budeme v definícii gramatiky niekedy vynechávať zložené zátvorky na mieste množiny vzorov, keďže v tomto prípade je iba jeden, takže v tomto prípade budeme občas písať: $G = (\Sigma, V, A, \pi)$.

Triedu všetkých iterovaných pattern-jazykov označme IPL.

Príklady jazykov, ktoré sú v IPL:

Príklad 5.4.2:

$G_1 = (\{a\}, \{X\}, \{a\}, \{aX\})$

$SL(G_1) = \{a\} \cup \{aa\} \cup \{aaa\} \cup \{aaaa\} \cup .... = \{a^n; n \geq 1\}$.

Ide o tú istú gramatiku ako v príklade 1 pre PL_{NG}. Je dobré si všimnúť rozdiel v postupnom generovaní jazyka v porovnaní s NL(G), hoci výsledok je nakoniec rovnaký; to isté platí aj pre ďalšie 2 príklady, ktoré sú len synchronizovanou obdobou príkladov, ktoré boli použité pre PL_{NG}.

Príklad 5.4.3:

$G_2 = (\{a\}, \{X\}, \{a\}, \{XX\})$

$SL(G_2) = \{a\} \cup \{aa\} \cup \{aaaa\} \cup \{ aaaaaaaa\} \cup ... = \{a\text{^}2^n; n \geq 0\}$

Príklad 5.4.4:

$G_3 = (\{a, b\}, \{X\}, \{ab\}, \{aXb\})$

$NL(G_3) = \{ab\} \cup \{aabb\} \cup \{aaabbb\} \cup ... = \{a^n b^n; n \geq 1\}$

A konečne aj príklad, kde budú rozdielne aj jazyky (tá istá gramatika ako v príklade 5.1.10 pre PL_{NG}):

Príklad 5.4.5:

$G_4 = (\{a, b\}, \{X_1, X_2\}, \{a, b\}, \{X_1X_2\})$

$SL(G_4) = \{a, b\} \cup \{aa, ab, ba, bb\} \cup \{a, b\}^4 \cup \{a, b\}^8 \cup ... = \{w \in \{a, b\}^*, |w| = 2^n, n \geq 0\}$

(Pod $\{a, b\}^n$ sa myslia všetky slová dĺžky n nad abecedou $\{a, b\}$.)

Veta 5.4.6: Nech $L \subseteq \Sigma^*$ je konečný jazyk. Potom $L, L^* \in$ IPL.

Dôkaz: (rovnako ako pre PL_{NG})

$L = L(G_1)$, kde $G_1 = (\Sigma, \{X\}, L, \{X\})$

$L^* = L(G_2)$, kde $G_2 = (\Sigma, \{X_1, X_2\}, L \cup \{\varepsilon\}, \{X_1X_2\})$

□

Nasledujú príklady jazykov, ktoré do IPL nepatria a využijeme ich pri dokazovaní ďalších vlastností IPL.

Lema 5.4.7: Jazyk $L = a^* \cup b^*$ nepatrí do IPL.

Dôkaz: : Sporom - podobne ako pre PL_{NG}: Nech $L = SL(G)$ pre nejakú gramatiku $G = (\{a, b\}, V, A, \pi)$. Vzor π nemôže obsahovať žiaden terminál alebo 2 rôzne premenné, inak by sa dali generovať reťazce obsahujúce oba terminálne symboly *a* aj *b*. Keďže L_1 je nekonečný, tak π musí mať dĺžku väčšou ako 1, teda $\pi = X^i$, $i \geq 1$. Ale to sa potom nedá vygenerovať žiaden reťazec tvaru a^m resp. b^m, kde m je prvočíslo väčšie ako i. Takže neexistuje synchronizovaná gramatika, ktorá by generovala L, teda $L \notin PL_{SG}$ a následne $L \notin$ IPL, keďže IPL je podmnožinou PL_{SG}.

□

Lema 5.4.8: Jazyk $L_2 = \{a^{2\hat{}n}\, b^{2\hat{}m}\, ; m, n \geq 0\} \notin$ IPL.

Dôkaz: Predpokladajme, že $L = SL(G)$ pre nejakú gramatiku $G = (\{a, b\}, V, A, \pi)$. vzor π nemôže obsahovať dve premenné (nezáleží na tom, či sú rôzne alebo nie), inak by museli byť v L aj slová tvaru $xa^{2\hat{}n}\, b^{2\hat{}m}\, ya^{2\hat{}n}\, b^{2\hat{}m}\, z$, ktoré tam nemajú čo hľadať. Neostáva teda iná možnosť, π musí byť tvaru uXv, $u, v \in \{a, b\}^*$. Slovo u nemôže obsahovať symbol b a v nemôže obsahovať symbol a, inak by sa dali generovať slová obsahujúce podslovo ba. Takže vzor π musí mať tvar a^jXb^k. Reťazce $a^{j+1}b^{k+1}$ a $a^{j+2}b^{k+2}$ získané z π nahradením premennej X slovami ab a aabb sú z L len vtedy, keď $j = k = 0$, teda $\pi = X$. Z toho vyplýva, že L je konečný (keďže množina axióm je konečná) a máme spor.

□

Lema 5.4.9: Jazyk $L = \{ww \mid w \in \{a, b\}^*\} \notin$ IPL.

Dôkaz: Nech $L = SL(G)$ pre nejakú gramatiku $G = (\{a, b\}, V, A, \pi)$. Ak má takáto gramatika generovať jazyk L, potom zrejme $\pi = \alpha\alpha$, kde α je nejaký vzor (inak by sa dali generovať slová aj iného tvaru ako ww). Môžu nastať 2 prípady:

1) $|\alpha|_a + |\alpha|_b$ je párne číslo (počet terminálnych symbolov vo vzore α). P(A) obsahuje len slová párnej dĺžky (bez ohľadu na počet terminálov v α). Keď za premenné v α dosadíme slová z P(A), dostaneme slovo párnej dĺžky. Slová vzniknuté zo vzoru $\pi = \alpha\alpha$ budú mať vždy párnu dĺžku (a len také budeme za premenné vo vzoroch dosadzovať v ďalších krokoch), tým pádom pre všetky $k \geq 2$ platí: $P^k(A) = \{ww \mid w \in \{a, b\}^*$, kde $|w|$ je párne číslo$\}$. Jazyk L obsahuje však aj ľubovoľné slová tvaru ww, kde dĺžka w je nepárne číslo, tie sa však v tomoto prípade nedajú vygenerovať všetky.

2) $|\alpha|_a + |\alpha|_b$ je nepárne číslo. Tu zase podobnou úvahou ako v predchádzajúcom prípade prídeme na to, že pre všetky $k \geq 2$ platí: $P^k(A) = \{ww \mid w \in \{a, b\}^*$, kde $|w|$ je nepárne číslo$\}$. V tomto prípade je teda zase problém vygenerovať všetky slová tvaru ww s párnou dĺžkou w.

Keďže neexistuje spôsob, ako by synchronizovaná gramatika s jednoprvkovou množinou vzorov generovala ľubovoľné slová tvaru ww, tak jazyk L nepatrí do IPL (vzor π je daný na začiatku jednoznačne, takže vždy nastane práve jeden z rozoberaných prípadov, nikdy nie oba naraz).

□

Lema 5.4.10: Dyckov jazyk nad {a, b} nie je iterovaný pattern language.

Dôkaz: Predpokladajme, že Dyckov jazyk nad {a, b} sa dá generovať nejakou synchronizovanou gramatikou $G = (\{a, b\}, V, A, \pi)$. Rozlišujeme 2 prípady:

1) prvý symbol π je terminál a keďže sa jedná o Dyckov jazyk, musí to byť a, teda $\pi = au\alpha$, $u \in \{a, b\}^*$, $\alpha \in V(V \cup \{a, b\})^*$ (podľa definície každý vzor musí v tomto prípade obsahovať aspoň 1 premennú). Keďže máme k dispozícii len 1 vzor, tak všetky slová z P(A) budú mať prefix au. V ďalšom kroku dosadzujeme len takéto slová, takže v $P^2(A)$ budú mať všetky slová prefix $(au)^2$. Vo všeobecnosti teda reťazce v $P^k(A)$, $k \geq 1$ majú tvar $(au)^k x$ pre nejaké $x \in \{a, b\}^*$. Opäť môžu nastať 2 prípady, konkrétne: slovo u obsahuje symbol b – potom sa ale nedajú vygenerovať reťazce tvaru $a^j b^j$, $j = k(|u| + 1)$. V opačnom prípade $u = a^t$, $t \geq 0$, potom ale pre dostatočne veľké m existujú reťazce tvaru $(ab)^m$, ktoré nie je možné vygenerovať.

2) prvý symbol vo vzore π je premenná $\pi = X\alpha$ $\alpha \in (V \cup \{a, b\})^*$. Keby bolo $\varepsilon \in P(A)$, tak $\pi \in V^+$. Potom by v P(A) neboli reťazce tvaru $a^k b^k$, $k \geq \max\{|w|, w \in A\}$. Aby sme mohli vygenerovať takéto reťazce, museli by sme každú premennú nahradiť nejakým slovom z Dyckovho jazyka (množina axióm A samozrejme môže obsahovať len také slová), avšak jediné slová, s ktorými by sa to dalo dosiahnuť sú ε a $a^k b^k$.
Takže vzor π nemôže pozostávať len z premenných. V takom prípade si zoberme slovo $a^k b^k \notin A \cup P(A)$ pre nejaké $k \geq \max\{|w|, w \in A\}$. Ak by sme ho chceli vygenerovať, museli by sme nahradiť premennú X jeho vlastným prefixom, ktorý určite do Dyckovho jazyka patriť nemôže a máme spor (ako je z definície zrejmé, na nahrádzanie premenných možno vždy použiť len slová, ktoré do daného jazyka patria).

□

Veta 5.4.11: Trieda IPL je neporovnateľná s triedami REG, CF, EDTOL, PL_{NG}, PL_E, PL_{NE}.

Dôkaz: Neporovnateľnosť s REG a CF vyplýva z nasledujúcich faktov: jazyk $a^* \cup b^*$ je regulárny a teda aj bezkontextový, ale nepatrí do IPL. Na druhej strane jazyk $\{a^{2^{\wedge}n} \mid n \geq 0\} \cup \{b^{2^{\wedge}n} \mid n \geq 0\} \in$ IPL, ale nie je ani bezkontextový a teda ani regulárny a jeho

príslušnosť do triedy IPL dokazuje fakt, že ho generuje synchronizovaná gramatika G = ({a, b}, {X}, {a, b}, XX}.

Neporovnateľnosť s EDTOL a PL_{NG}: jazyk $\{w \in \{a, b\}^+ , \text{ kde } |w| = 2^k, k \geq 0\} \notin$ EDTOL a teda nepatrí ani do PL_{NG}, ale patrí do IPL (pozri príklad 5.4.5). Na druhej strane, Dyckov jazyk patrí do triedy PL_{NG} a teda aj do EDTOL a nepatrí do IPL (pozri príklad 5.1.9 a lemu 5.4.10).

Neporovnateľnosť s PL_Z, $Z \in \{E, NE\}$: jazyk $\{a^{2\wedge n} \mid n \geq 0\} \cup \{b^{2\wedge n} \mid n \geq 0\} \notin PL_Z$ a patrí do IPL a jazyky $\{ww \mid w \in \{a, b\}^*\} \in PL_E$ resp. $\{ww \mid w \in \{a, b\}^+\} \in PL_{NE}$ nie sú z IPL.

□

Veta 5.4.12: Trieda IPL nie je uzavretá na zjednotenie, zreťazenie, prienik s regulárnymi jazykmi, Kleeneho* a inverzný homomorfizmus.

Dôkaz: Využijeme pri tom jazyky, o ktorých na základe príkladov a tvrdení už vieme, že patria resp. nepatria do triedy IPL.

Zjednotenie: $a^*, b^* \in$ IPL, $a^* \cup b^* \notin$ IPL.

Zreťazenie: $L_1 = \{a^{2\wedge n}, n \geq 0\}$ a $L_2 = \{b^{2\wedge n}, n \geq 0\}$ patria do IPL
$L_1L_2 = \{a^{2\wedge n} b^{2\wedge m} ; m, n \geq 0\} \notin$ IPL

Prienik s regulárnymi jazykmi:
Nech L = SL(G) = $\{a, b\}^*$, kde G = ({a, b}, {X_1, X_2}, {ε, a, b} , X_1X_2) a R = $a^* \cup b^*$.
$L \cap R = a^* \cup b^* \notin$ IPL.

Kleeneho*: $L = \{a^n b^n \mid n \geq 0\} \in$ IPL (L = SL(G), kde G = ({a, b}, {X}, {ε, ab}, aXb) – malá modifikácia príkladu 5.4.4). Ukážeme, že $L^* \notin$ IPL. Keby L^* patril do IPL, tak gramatika, ktorá by ho generovala by musela vyzerať takto: G_1 = ({a, b}, {X}, A, uXv). Vzor musí mať tvar uXv, nemôže obsahovať 2 premenné, inak by sme dostali aj slová iného ako požadovaného tvaru, (napr. $a^n b^n$ má v L^* iba jeden vlastný podreťazec). Potom zrejme $u = a^i$ a $v = b^i$, $i > 0$. Ale nahradením premennej X napr. slovom $a^{2i}b^{2i}$ $a^{2i}b^{2i}$, ktoré musí patriť do L^*, by sme dostali slovo $a^{3i}b^{2i}$ $a^{2i}b^{3i}$, ktoré do L^* nepatrí, máme spor.

□

Inverzný homomorfizmus:
Nech G = ({a, b, c}, {X}, {abc}, Xbc}), potom SL(G) = $\{a(bc)^n \mid n \geq 1\}$ samozejme patrí do IPL.
Nech h je homomorfizmus definovaný takto:

h(a) = ab
h(b) = cb
h(c) = c

Potom $h^{-1}(SL(G)) = \{ab^n c \mid n \geq 0\} \notin$ IPL. Zvyšok dôkazu je ponechaný na čitateľa.

□

Veta 5.4.13: IPL je uzavretá na homomorfizmus.

Dôkaz: Nech L = SL(G) pre nejakú gramatiku G = (Σ_1, V, A, π) a nech h: $\Sigma_1^* \rightarrow \Sigma_2^*$ je nejaký homomorfizmus. Jazyk h(L) možno generovať gramatikou G' = (Σ_2, V, A', π'), kde A' = h(A) a π' vznikne z π nahradením každého terminálu jeho homomorfným obrazom, premenné zostanú bez zmeny.

5.5 Možnosti na ďalšie rozpracovanie a pattern gramatiky

Niektoré ďalšie výsledky týkajúce sa tried PL_{NG} a IPL možno nájsť v [10] a [11]. Na záver spomeňme ešte niekoľko otvorených problémov týkajúcich sa tried PL_{NG} a IPL:
Vie sa, že problém príslušnosti daného bezkontextového jazyka do triedy PL_{NG} je nerozhodnuteľný. *Otvorené otázky* sú napríklad, či je rozhodnuteľné, či daný regulárny jazyk patrí do PL_{NG}? Je rozhodnuteľné, či daná nesynchronizovaná gramatika generuje regulárny resp. bezkontextový jazyk? Platí IPL $\subseteq$ ETOL?

Aké sú vlastnosti triedy jazykov generovaných synchronizovanými gramatikami založených na vzoroch s ľubovoľným počtom vzorov? Pokiaľ vieme, táto problematika ešte nebola v dostupných článkoch skúmaná.

Úplne na záver tejto kapitoly ešte stručne spomenieme aj ďalší, trochu odlišný spôsob od toho predchádzajúceho, ako využiť vzory - tzv. *pattern gramatiky* – tu sa jedná o gramatiky s konečnou množinou pravidiel tvaru $\alpha \rightarrow \beta$, kde α a β sú vzory. Na pravidlá možno klásť rôzne obmedzenia podobne, ako v prípade frázových gramatík. Takto definované gramatiky majú značnú generatívnu silu, dokonca pomocou kontextových pravidiel (ľavá strana pravidla nie je dlhšia ako pravá) sa dajú vygenerovať aj jazyky, ktoré nie sú kontextové. Pre bližšie zoznámenie sa s týmito gramatikami môže čitateľ nahliadnuť do [o7].

Kapitola 6

Pattern systémy

Predstavíme model, ktorý pokrýva mnohé z doteraz uvedených spôsobov definovania pattern languages a poskytuje v istom zmysle pre ne akýsi jednotný rámec. Základná myšlienka je, že na začiatku máme ku každej premennej priradené dve množiny: množinu vzorov a množinu terminálnych slov. Nové slová generujeme tak, že v nasledujúcom kroku nahrádzame premenné vo vzoroch slovami, ktoré sú priradené k jednotlivým premenným, pričom tieto slová musia rešpektovať niektorý zo vzorov, ktorý je priradený k príslušnej premennej, tým získame ďalšie slová. Môžeme pri tom postupovať synchronizovane alebo nesynchronizovane (podobne ako v prípade gramatík založených na vzoroch), pričom v tomto prípade existujú dva druhy synchronizácie. Keďže v tomto prípade nie je až také jednoduché hneď na prvýkrát pochopiť, ako systém pracuje, uvedieme aj veľa príkladov.

6.1 Základné definície

Definícia 6.1.1: Pattern systémom nazývame štvoricu $\Gamma = (\Sigma, V, p, t)$, kde Σ resp. V sú ako zvyčajne množiny terminálnych symbolov resp. premenných, p a t sú zobrazenia z množiny premenných do množiny podmnožín $(\Sigma \cup V)^* V (\Sigma \cup V)^*$ resp. Σ^*, formálne:

$$p: V \rightarrow 2^{(\Sigma \cup V)^* V (\Sigma \cup V)^*}$$
$$t: V \rightarrow 2^{\Sigma^*}$$

V našom prípade uvažujeme len *konečné* a *neprázdne* podmnožiny $(\Sigma \cup V)^* V (\Sigma \cup V)^*$ resp. Σ^*.

Význam tejto definície je nasledovný: ku každej premennej je na začiatku priradená konečná a neprázdna množina slov, ktorými môže byť na začiatku odvodzovacieho procesu nahradená, to nám určuje zobrazenie t. Zobrazenie p zase ku každej premennej priradí nejakú konečnú a neprázdnu množinu vzorov a slová, ktorými v ďalších krokoch odvodzovania nahrádzame príslušnú premennú musia tieto vzory rešpektovať. Teraz si zadefinujeme rôzne spôsoby procesu odvodzovania, v zásade sú dva a to opäť synchronizovaný a nesynchronizovaný, avšak v prvom prípade rozlišujeme dva typy synchronizácie.

Definícia 6.1.2: Silno synchronizovaný pattern systém (v skratke SSPS) $\Gamma = (\Sigma, V, p, t)$ rekurzívne generuje pre každé $i \geq 0$ postupnosť n-tíc terminálnych slov, označme si ju $D^{(i)}(\Gamma)$, nasledovným spôsobom:

$(w_1^{(0)}, ..., w_n^{(0)}) \in D^{(0)}(\Gamma)$, ak $w_j^{(0)} \in t(X_j)$, $j = 1, ..., n$

$(w_1^{(i+1)}, ..., w_n^{(i+1)}) \in D^{(i+1)}(\Gamma)$, ak existuje $(w_1^{(i)}, ..., w_n^{(i)}) \in D^{(i)}(\Gamma)$ a $\alpha_j \in p(X_j)$, $j = 1, ..., n$ také, že $w_j^{(i+1)}$ sa získa z α_j nahradením všetkých výskytov každej premennej X_k slovom $w_k^{(i)}$, platí to pre všetky $k \in \{1, ..., n\}$, pre ktoré sa premenná X_k vo vzore α_j vyskytuje.

Jazyk generovaný silno synchronizovaným pattern systémom Γ začínajúc od X_j je:

$$L_s(\Gamma, X_j) = \{w_j^{(i)} \mid i \geq 0\}.$$

Nové n-tice získavame teda tak, že pre každú premennú vyberieme nejaký vzor z množiny k nej priradených vzorov a premenné v týchto vzoroch nahradíme potom príslušnými slovami z n-tice slov z predchádzajúceho kroku výpočtu (premennú X_k slovom $w_k^{(\text{predch. krok})}$).

Definícia 6.1.3: Slabo synchronizovaný pattern systém (v skratke WSPS; W = weakly) $\Gamma = (\Sigma, V, p, t)$ rekurzívne generuje pre každé $i \geq 0$ postupnosť n-tíc terminálnych slov, zase si ju označme $D^{(i)}(\Gamma)$, nasledovným spôsobom:

$(w_1^{(0)}, ..., w_n^{(0)}) \in D^{(0)}(\Gamma)$, ak $w_j^{(0)} \in t(X_j)$, $j = 1, ..., n$

$(w_1^{(i+1)}, ..., w_n^{(i+1)}) \in D^{(i+1)}(\Gamma)$, ak existuje $\alpha_j \in p(X_j)$, $j = 1, ..., n$, a pre každé takéto α_j existuje $(w_1^{(i)}, ..., w_n^{(i)}) \in D^{(i)}(\Gamma)$ také, že $w_j^{(i+1)}$ sa získa z α_j nahradením všetkých výskytov každej premennej X_k slovom $w_k^{(i)}$, a to pre všetky $k \in \{1, ..., n\}$, pre ktoré sa premenná X_k vo vzore α_j vyskytuje.

Jazyk generovaný slabo synchronizovaným pattern systémom Γ začínajúc od X_j je:

$$L_w(\Gamma, X_j) = \{w_j^{(i)} \mid i \geq 0\}.$$

Na rozdiel od silno synchronizovaného pattern systému možno v tomto prípade v každom kroku použiť pre rôzne vzory rôzne n-tice.

Definícia 6.1.4: Nesynchronizovaný pattern systém (v skratke NSPS) $\Gamma = (\Sigma, V, p, t)$ generuje nasledovnú postupnosť $D_j^{(i)}(\Gamma)$ pre každé $j = 1, ..., n$ a $i \geq 0$ nasledovným spôsobom:

$D_j^{(0)}(\Gamma) = t(X_j)$, $j = 1, ..., n$

$D_j^{(i+1)}(\Gamma)$ je zjednotením medzi $D_j^{(i)}(\Gamma)$ a množinou všetkých terminálnych slov získaných z každého vzoru $\alpha \in p(X_j)$ tým, že sa v ňom nahradí každý výskyt každej premennej X_k nejakým slovom z $D_k^{(i)}(\Gamma)$, $k = 1, ..., n$.

Jazyk generovaný nesynchronizovaným pattern systémom Γ začínajúc od X_j je:

$$L_n(\Gamma, X_j) = \{w \in \Sigma^* \mid \text{existuje } i \geq 0 \text{ také, že } w \in D_j^{(i)}(\Gamma)\}.$$

Rozdiel medzi synchronizovaným a nesynchronizovaným pattern systémom je v podstate ten istý, s akým sme sa stretli aj v prípade gramatík založených na vzoroch, teda v synchronizovanom prípade sa premenné môžu nahrádzať len slovami vygenerovanými v predchádzajúcom kroku odvodenia, kým v nesynchronizovanom prípade možno na nahradenie premenných použiť akýkoľvek terminálny reťazec vyprodukovaný v hociktorom z predchádzajúcich krokov výpočtu.

Triedy jazykov definované SSPS resp. WSPS resp. NSPS označme ako SSPL resp. WSPL resp. NSPL.

Definícia 6.1.5: Pattern systém $\Gamma = (\Sigma, V, p, t)$ sa nazýva deterministický, ak pre všetky $X_i \in V$ sú množiny $p(X_i)$ a $t(X_i)$ jednoprvkové. Príslušné triedy jazykov pre tri vyššie definované spôsoby odvodzovania si označme ako DSSPL, DWSPL, DNSPL.

V ďalšom môžeme bez ujmy na všeobecnosti predpokladať, že v $L_f(\Gamma, X_j)$ je $j = 1$ (vždy môžeme vhodne premenovať premenné), preto, ak nebude z nejakých dôvodov nutné uvádzať aj premennú, budeme písať len $L_f(\Gamma)$ ($f \in \{s,w,n\}$).

Teraz uvedieme pár príkladov, ktoré nám lepšie ukážu ako taký pattern systém pracuje a z ktorých viaceré použijeme aj neskôr pri rôznych dôkazoch.

Príklad 6.1.6: Majme pattern systém $\Gamma_1 = (\{a, b\}, \{X_1, X_2, X_3\}, p, t)$

$p(X_1) = \{X_2X_3\}$ $\quad t(X_1) = \{ab\}$
$p(X_2) = \{X_2X_2\}$ $\quad t(X_2) = \{a\}$
$p(X_3) = \{X_3X_3\}$ $\quad t(X_3) = \{b\}$

Keďže množiny vzorov resp. axióm prislúchajúcich k jednotlivým premenným sú len jednoprvkové, ide o deterministický pattern systém. Nie je ťažké vidieť, že v každom kroku odvodenia je v synchronizovanom prípade k dispozícii len jedna n-tica, preto vygenerovaný jazyk bude rovnaký pre slabú aj silnú synchronizáciu. Pre deterministické pattern systémy teda dostávame aj prvý triviálny výsledok: DSSPL = DWSPL

Lema 6.1.7: DSSPL = DWSPL.

Vráťme sa ale k príkladu 6.1.6 a poďme sa teraz bližšie pozrieť na to, ako systém pracuje, najskôr synchronizovaný prípad: na začiatku máme

$D^{(0)}(\Gamma_1) = (ab, a, b)$ je to jediná n-tica slov prislúchajúcich k premenným v poradí X_1, X_2, X_3

Jediná n-tica vzorov, ktoré prislúchajú k premenným je: (X_2X_3, X_2X_2, X_3X_3)
Premennú X_i môžeme nahradiť len slovom, ktoré je na i-tej pozícii v n-tici slov $D^{(0)}(\Gamma_1)$ ($i = 1, 2, 3$), teda $D^{(1)}(\Gamma_1) = (ab, a^2, b^2)$. Rovnakým spôsobom získame $D^{(2)}(\Gamma_1)$ s použitím n-tice slov z $D^{(1)}(\Gamma_1)$, $D^{(3)}(\Gamma_1)$ získame zo slov z $D^{(2)}(\Gamma_1)$, atď....
Niekoľko prvých iterácii vyzerá takto:

$D^{(0)}(\Gamma_1) = (ab, a, b)$
$D^{(1)}(\Gamma_1) = (ab, a^2, b^2)$
$D^{(2)}(\Gamma_1) = (a^2b^2, a^4, b^4)$
$D^{(3)}(\Gamma_1) = (a^4b^4, a^8, b^8)$
$D^{(4)}(\Gamma_1) = (a^8b^8, a^{16}, b^{16})$

Jasne vidno, že $L_s(\Gamma_1) = L_w(\Gamma_1) = \{a^{2\wedge n}\, b^{2\wedge n} \mid n \geq 0\}$.

Pre nesynchronizovaný režim použijúc nasledovný zápis:
$D^{(i)}(\Gamma_1) = (D_1^{(i)}(\Gamma_1), D_2^{(i)}(\Gamma_1), D_3^{(i)}(\Gamma_1))$, i = 0, 1, 2, ..., dostávame:

$D^{(0)}(\Gamma_1) = (\{ab\}, \{a\}, \{b\})$
$D^{(1)}(\Gamma_1) = (\{ab\}, \{a, a^2\}, \{b, b^2\})$
$D^{(2)}(\Gamma_1) = (\{ab, ab^2, a^2b, a^2b^2\}, \{a, a^2, a^4\}, \{b, b^2, b^4\})$
$D^{(3)}(\Gamma_1) = (\{ab, ab^2, ab^4, a^2b, a^2b^2, a^2b^4, a^4b, a^4b^2, a^4b^4\}, \{a, a^2, a^4, a^8\}, \{b, b^2, b^4, b^8\})$

$L_n(\Gamma_1) = \{a^{2\wedge m}\, b^{2\wedge n} \mid m, n \geq 0\}$.

Nech Γ_1' je ten istý systém ako Γ_1, s nasl. zmenami: $p(X_1) = \{X_2, X_3\}$, $t(X_1) = \{a, b\}$.
Potom $L_n(\Gamma_1') = \{a^{2\wedge n} \mid n \geq 0\} \cup \{b^{2\wedge n} \mid n \geq 0\}$.

Príklad 6.1.8: Majme pattern systém $\Gamma_2 = (\{a, b\}, \{X_1, X_2\}, p, t)$
$p(X_1) = \{X_2X_2\}$ $t(X_1) = \{\varepsilon\}$
$p(X_2) = \{aX_2b\}$ $t(X_2) = \{\varepsilon\}$

V tomto prípade obsahuje každý vzor výskyty len jednej premennej, preto sa budú zhodovať aj jazyky pre všetky tri režimy odvodzovania. Nie je ťažké vidieť, že:
$L_s(\Gamma_2) = L_w(\Gamma_2) = L_n(\Gamma_2) = \{a^nb^na^nb^n \mid n \geq 0\}$.

Príklad 6.1.9: $\Gamma_3 = (\{a, b\}, \{X_1, X_2, X_3, X_4\}, p, t)$
$p(X_1) = \{X_2X_3\}$ $t(X_1) = \{a\}$
$p(X_2) = \{X_1\}$ $t(X_2) = \{\varepsilon\}$
$p(X_3) = \{X_4\}$ $t(X_3) = \{b\}$
$p(X_4) = \{X_3\}$ $t(X_4) = \{a\}$

Na prehľadnejšie odvodzovanie slov použijeme nasledujúcu tabuľku:

V	$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{X_3}$	$\mathbf{X_4}$
$\mathbf{p(X_j)}$	X_2X_3	X_1	X_4	X_3
$\mathbf{D^{(0)}(\Gamma_3) = t(X_j)}$	a	ε	b	a
$\mathbf{D^{(1)}(\Gamma_3)}$	b	a	a	b
$\mathbf{D^{(2)}(\Gamma_3)}$	a^2	b	b	a
$\mathbf{D^{(3)}(\Gamma_3)}$	b^2	a^2	a	b
$\mathbf{D^{(4)}(\Gamma_3)}$	a^3	b^2	b	a
$\mathbf{D^{(5)}(\Gamma_3)}$	b^3	a^3	a	b
$\mathbf{D^{(6)}(\Gamma_3)}$	a^4	b^3	b	a
$\mathbf{D^{(7)}(\Gamma_3)}$	b^4	a^4	a	b

Zaujímajú nás slová, ktoré sú v stĺpci pre X_1. Dá sa ľahko tušiť, že $L_s(\Gamma_3) = a^+ \cup b^+$.

Zaujímavé je, že tento jazyk vznikne z postupnosti slov (stĺpec pod X_1), v prípade bežných zariadení, ako sú automaty, „obyčajné" gramatiky alebo L-systémy, sa takýto jazyk nedá definovať ako postupnosť slov. Podobne možno definovať aj jazyk $L_s(\Gamma_3') = a^+ \cup b^+ \cup c^+$, kde $\Gamma_3' = (\{a, b, c\}, \{X_1, X_2, X_3, X_4, X_5, X_6\}, p, t)$, kde zobrazenia p a t sú definované nasledovne:

$p(X_1) = \{X_3X_4\}$ $t(X_1) = \{a\}$
$p(X_2) = \{X_1\}$ $t(X_2) = \{\varepsilon\}$
$p(X_3) = \{X_2\}$ $t(X_3) = \{\varepsilon\}$
$p(X_4) = \{X_5\}$ $t(X_4) = \{c\}$
$p(X_5) = \{X_6\}$ $t(X_5) = \{b\}$
$p(X_6) = \{X_4\}$ $t(X_6) = \{a\}$

Pozorný čitateľ iste sám dokáže nahliadnuť, že takýto systém naozaj generuje horeuvedený jazyk.

Príklad 6.1.10: $\Gamma_4 = (\{X_1, X_2, X_3\}, \{a, b, c\}, p, t)$
$p(X_1) = \{X_2X_3\}$ $t(X_1) = \{abc\}$
$p(X_2) = \{aX_2b\}$ $t(X_2) = \{ab\}$
$p(X_3) = \{X_3c\}$ $t(X_3) = \{c\}$

V	$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{X_3}$
$\mathbf{p(X_j)}$	X_2X_3	aX_2b	X_3c
$\mathbf{D^{(0)}(\Gamma_4)}$	abc	ab	c
$\mathbf{D^{(1)}(\Gamma_4)}$	abc	aabb	cc
$\mathbf{D^{(2)}(\Gamma_4)}$	aabbcc	aaabbb	ccc
$\mathbf{D^{(3)}(\Gamma_4)}$	$a^3b^3c^3$	a^4b^4	c^4
$\mathbf{D^{(4)}(\Gamma_4)}$	$a^4b^4c^4$	a^5b^5	c^5
$\mathbf{D^{(5)}(\Gamma_4)}$	$a^5b^5c^5$	a^6b^6	c^6

Tento systém v synchronizovanom režime generuje dobre známy jazyk nepatriaci do CF:
$L_s(\Gamma_4) = \{a^nb^nc^n \mid n \geq 1\}$.

Príklad 6.1.11: $\Gamma_5 = (\{X_1, X_2, X_3\}, \{a, b, c\}, p, t)$
$p(X_1) = \{X_2cX_2\}$ $t(X_1) = \{aca, bcb\}$
$p(X_2) = \{X_2aX_2b\}$ $t(X_2) = \{a, b\}$

V	$\mathbf{X_1}$	$\mathbf{X_2}$
$\mathbf{p(X_j)}$	X_2cX_2	X_2aX_2b
$\mathbf{D^{(0)}(\Gamma_4)}$	{aca, bcb}	{a, b}
$\mathbf{D^{(1)}(\Gamma_4)}$	{aca, bcb}	{aa, ab, ba, bb}
$\mathbf{D^{(2)}(\Gamma_4)}$	{aacaa, abcab, bacba, bbcbb}	{aaa, aab, aba, abb, baa, bab, bba, bbb}

$L_s(\Gamma_5) = L_n(\Gamma_5) = \{xcx \mid x \in \{a, b\}^+\}$, čo je ďalší dobre známy nebezkontextový jazyk.

Keďže v ďalšom sa budeme odvolávať na niektoré jazyky, tu je ich malé zhrnutie spolu s triedami, do ktorých patria:

$L_1 = \{a^{2\wedge n} \mid n \geq 0\} \cup \{b^{2\wedge n} \mid n \geq 0\} \in$ NSPL
$L_2 = \{a^nb^na^nb^n \mid n \geq 0\} \in$ DNSPL $\cap$ DSSPL
$L_3 = a^+ \cup b^+ \in$ DSSPL

$L_4 = \{a^n b^n c^n \mid n \geq 1\} \in$ DSSPL

Veta 6.1.12: Každý konečný jazyk patrí do DNSPL a aj do DWSPL = DSSPL.

Dôkaz: Nech $F = \{w_1, w_2, ..., w_n\}$ je nejaký konečný jazyk nad abecedou Σ. Systém generujúci F vyzerá nasledovne:

$$\Gamma_F = (\Sigma, \{X_1, X_2, ..., X_n\}, p, t\}$$
$$p(X_i) = \{X_{i+1}\},\ i = 1, ..., n-1$$
$$p(X_n) = \{X_n\}$$
$$t(X_i) = \{w_i\},\quad i = 1, ..., n.$$

Jasne vidno, že $L_n(\Gamma_F, X_1) = L_s(\Gamma_F, X_1) = F$.

□

Teraz uvedieme aj niekoľko príkladov jazykov, ktoré nepatria do niektorých tried definovaných pattern systémami. Tieto jazyky nám, ako to už býva zvykom, neskôr poslúžia na dokazovanie istých tvrdení týkajúcich sa vzťahov medzi triedami.

Lema 6.1.13: $L_4 = \{a^n b^n c^n \mid n \geq 1\} \notin$ NSPL

Dôkaz: Predpokladajme, že $L_4 = L_n(\Gamma)$, kde $\Gamma = (\Sigma, V, p, t)$, $\Sigma = \{a, b, c\}$, $V = \{X_1, ..., X_n\}$, $n \geq 1$. Pre množiny $D_j^{(i)}$ (pozri definíciu nesynchronizovaného pattern systému), $j = 1, ..., n$, $i \geq 0$ platí: $D_j^{(0)}(\Gamma) = t(X_j)$ a $D_j^{(i)}(\Gamma) \subseteq D_j^{(i+1)}(\Gamma)$ (tento vzťah jasne vyplýva z definície), z čoho nám ďalej vyplýva, že $L_n(\Gamma) = \lim_{i\to\infty} D_1^{(i)}(\Gamma)$.

Nech $w = a^m b^m c^m \in L_4$, pričom $m > \max\{|\alpha|; \alpha \in p(X_j)\}.\max\{|x|; x \in t(X_j)\}$ pre $j = 1, ..., n$. Pre takéto m nám ihneď vyplýva prvý dôležitý fakt: $w \notin D_1^{(0)}(\Gamma) \cup ... \cup D_n^{(0)}(\Gamma)$. Avšak existuje $i \geq 1$ také, že $w \in D_1^{(i)}(\Gamma)$. Zoberme najmenšie i s touto vlastnosťou. Slovo $w = h_1(\alpha_1)$ pre nejaké $\alpha_1 \in p(X_1)$ a $h_1: (V \cup \Sigma)^* \to \Sigma^*$ je homomorfizmus stabilný pre terminály taký, že premenné zobrazuje na slová z $D_j^{(i-1)}(\Gamma)$, $j = 1, ..., n$ (len simulujeme prácu nesynchronizovaného pattern systému).

Poďme sa teraz pozrieť na to, akú štruktúru môže mať vzor α_1. Najskôr vylúčme možnosť, že α_1 obsahuje terminály, inak by sa dali generovať slová, ktoré do L_4 nepatria, takže α_1 obsahuje len premenné. Ak sa v ňom vyskytuje premenná X_1, tak potom $\alpha_1 = X_1$ alebo okrem X_1 obsahuje premenné, ktoré možno nahradiť len prázdnymi slovami. Iná možnosť vedie k sporu, pretože X_1 sa nahrádza slovami z $D_1^{(i-1)}$ a táto množina obsahuje len slová z jazyka L_4, teda slová tvaru $a^k b^k c^k$. Pridaním prípadných ďalších symbolov po nahradení X_1 by sa odvodili slová, ktoré by nemali požadovaný tvar. Lenže ak $\alpha_1 = X_1$, potom $w \in D_1^{(i-1)}(\Gamma)$, čo je spor s tým, že i je najmenšie také, že $w \in D_1^{(i)}(\Gamma)$ (rovnaký argument platí pochopiteľne aj pre druhý prípad, kde okrem X_1 sú aj premenné, ktoré sa nahrádzajú prázdnym slovom) , preto sa v α_1 nemôže X_1 vyskytovať.

Predpokladajme teda, že v α_1 sa vyskytuje nejaká premenná $X_j \neq X_1$, ktorá sa pri odvodzovaní nahradí nejakým slovom $y_j \in D_j^{(i-1)}(\Gamma)$. Ak $\alpha_1 \neq X_j$, tak množina $D_j^{(i-1)}(\Gamma)$ musí byť iba jednoprvková, inak by sme pri nahrádzaní X_j mohli použiť aj iný reťazec ako y_j (pri zachovaní nahradenia prípadných iných premenných rovnakými slovami ako predtým) a tým by sme dostali slovo nepatriace do L_4. Lenže ak množina $D_j^{(i-1)}(\Gamma)$ je jednoprvková, potom $D_j^{(i-1)}(\Gamma) = t(X_j)$, z čoho zase vyplýva, že w je výsledkom nahradenia premenných v α_1 slovami z $t(X_j)$, $j = 1, ..., n$ a to je spor s výberom m. Ešte zostala možnosť, že $\alpha_1 = X_j$, $X_j \neq X_1$. Z toho vyplýva, že $w \in D_j^{(i-1)}(\Gamma)$ a zákonite musí existovať vzor $\alpha_j \in p(X_j)$ taký, že $w = h_j(\alpha_j)$ pre $h_j: (V \cup \Sigma)^* \to \Sigma^*$,

h_j je stabilný pre terminály a $h_j(X_r) \in D_r^{(i-2)}(\Gamma)$, r = 1, ..., n. Takto by sme mohli stále pokračovať smerom k $D_j^{(0)}(\Gamma)$, no keďže slovo $w \notin D_1^{(0)}(\Gamma) \cup ... \cup D_n^{(0)}(\Gamma)$, opäť by sme prišli k záveru, že w sa získa z nejakého vzoru α_j, ktoré pozostáva z viac ako jednej premennej, o čom už vieme, že to vedie k sporu, teda $L_4 = \{a^n b^n c^n \mid n \geq 1\} \notin$ NSPL.

□

Nasledujúcej lema do istej miery charakterizuje niektoré jazyky z DNSPL, dá sa pomocou nej dokazovať aj to, že niektoré jazyky do DNSPL nepatria.

Lema 6.1.14: Ak $L \in$ DNSPL, pričom $L = L_1 \cup L_2$, kde $L_1 \subseteq \Sigma_1^*$, $L_2 \subseteq \Sigma_2^*$ sú nekonečné jazyky a abecedy Σ_1 a Σ_2 sú disjunkné, potom existuje konečný jazyk $F \subseteq \Sigma_1^* \cup \Sigma_2^*$ a konštanta $t \geq 2$ taká, že

$$L \subseteq F \cup \{x^t \mid x \in \Sigma_1^* \cup \Sigma_2^*\}$$

Dôkaz: Nech $L = L_n(\Gamma, X_1)$ pre nejaký deterministický pattern systém $\Gamma = (\Sigma_1 \cup \Sigma_2, V, p, t)$, $V = \{X_1, ..., X_n\}$. Keďže Γ je deterministický, tak $|p(X_j)| = |t(X_j)| = 1$ pre j = 1, ..., n. Bez ujmy na všeobecnosti môžeme predpokladať, že Γ neobsahuje žiadne zbytočné premenné, teda také, ktoré sa nezúčastňujú na generovaní jazyka L. Keby tam také premenné boli, vieme ich jednoducho algoritmicky odstrániť pomocou grafu, kde by premenné X_i a X_j boli spojené hranou práve vtedy, keď sa premenná X_j vyskytuje vo vzore z $p(X_i)$. Zbytočná premenná je taká, ktorú s X_1 nespája žiadna cesta.

Keďže abecedy jazykov L_1 a L_2 sú disjunktné, tak žiadny vzor použitý na generovanie slov z L_1 nemôže obsahovať symboly zo Σ_2 a podobne aj naopak, žiadny vzor použitý na generovanie slov z L_2 nemôže obsahovať symboly zo Σ_1. Navyše Γ je deterministický a neobsahuje žiadne zbytočné premenné. Z uvedených faktov vyplýva, že všetky vzory pozostávajú len z premenných.

Keby všetky vzory mali dĺžku jeden, potom by bol jazyk $L_n(\Gamma)$ konečný, čo nie je, takže aspoň jeden vzor musí mať dĺžku väčšiu ako jeden, nech je to $\beta = X_{i1} ... X_{ik}$, $k \geq 2$. Uvažujme ďalej najkratšiu možnú postupnosť premenných $X_{j1}, ..., X_{jr}$, $r \geq 0$, takú, že $p(X_1) = X_{j1}$, $p(X_{j1}) = X_{j2}$, ..., $p(X_{j(r-1)}) = X_{jr}$, $p(X_{jr}) = \alpha$, kde α je dĺžky aspoň dva (môže nastať aj prípad, že $p(X_1) = \alpha$). Bez použitia vzoru α sa dá získať len konečný počet slov, ktoré sú zahrnieme do F. Všetky ostatné slová sa získajú nahradením premenných vo vzore α terminálnymi reťazcami. To platí pre slová z L_1 aj z L_2, teda $D_{is}^{(j)}(\Gamma)$ obsahuje reťazce zo Σ_1^* aj zo Σ_2^* pre všetky $j \geq 1$ a s = 1, ..., k. Keby niektoré $i_u \neq i_v$, u, v ∈ {1, ..., k}, tak pre dosť veľké j, j', $D_{iu}^{(j)}(\Gamma)$ by obsahovalo reťazec zo Σ_1^+ a $D_{iv}^{(j')}(\Gamma)$ zase reťazec zo Σ_2^+ (alebo naopak), potom by však bolo možné vygenerovať slovo obsahujúce symboly súčasne z oboch abecied, čo je nežiadúce. Takže buď $D_{is}^{(j)}(\Gamma) = \{\varepsilon\}$ alebo $X_{iu} = X_{iv}$, presnejšie: $\alpha = \alpha_1 X_{iu} \alpha_2 X_{iu} ... \alpha_t X_{iu} \alpha_{t+1}$, pričom všetky α_q, q = 1, ..., t+1 sa skladajú z premenných X_{is}, pričom $D_{is}^{(j)}(\Gamma) = \{\varepsilon\}$. Pre prípad, že t = 1, môžeme všetky vzory tvaru $\alpha = \alpha_1 X_{iu} \alpha_2$ nahradiť novým $\alpha = X_{iu}$ bez toho, aby sa zmenil jazyk definovaný daným pattern systémom (keďže α_1 a α_2 je možné nahradiť len prázdnym slovom). Po tejto úprave získame trochu iný pattern systém Γ', pre ktorý budeme opäť uvažovať najkratšiu možnú postupnosť premenných $X_{j1}, ..., X_{jr}$, $r \geq 0$, takú, že $p(X_1) = X_{j1}$, $p(X_{j1}) = X_{j2}$, ..., $p(X_{j(r-1)}) = X_{jr}$, $p(X_{jr}) = \alpha$. Týmto postupom nakoniec nájdeme taký vzor $\alpha = \alpha_1 X_{iu} \alpha_2 X_{iu} ... \alpha_t X_{iu} \alpha_{t+1}$, kde $t \geq 2$, z čoho jasne vyplýva, že po nahradení premennej X_{iu} sa získa požadovaný tvar generovaných reťazcov x^t.

□

Dôsledok 6.1.15: Pre $L_1 = \{a^{2^{\wedge}n} \mid n \geq 0\} \cup \{b^{2^{\wedge}n} \mid n \geq 0\}$ a $L_3 = a^+ \cup b^+$ platí: $L_1, L_3 \notin$ DNSPL.

6.2 Simulácia doteraz známych tried pattern languages pomocou pattern systémov a vzťahy s inými triedami jazykov

Teraz si ukážeme, ako sa pomocou pattern systémov dajú simulovať mnohé doteraz spomínané triedy pattern languages.

Lema 6.2.1: $PL_E \subset NSPL$ a $PL_{NE} \subset NSPL$.

Dôkaz: Nech $\Sigma = \{a_1, a_2, ..., a_k\}$, $V = \{X_1, X_2, ..., X_n\}$ sú množiny terminálov a premenných a nech $\pi \in (\Sigma \cup V)^+$ je vzor. Ako už vieme, triedy PL_E resp. PL_{NE} sú tvorené jazykmi pozostávajúcimi zo slov, ktoré vzniknú nahradením premenných v danom vzore ľubovoľnými resp. ľubovoľnými neprázdnymi reťazcami. Pattern systém, ktorý nám vygeneruje ľubovoľný jazyk z PL_E resp. z PL_{NE} vyzerá nasledovne:

$\Gamma = (\Sigma, V \cup \{X_0\}, p, t)$
$p(X_0) = \{\pi\}$, $t(X_0) = \{w\}$, w je nejaké slovo z $L_E(\pi)$
$p(X_i) = \{a_j X_i \mid i = 1, ..., k\}$, $t(X_i) = \{\varepsilon\}$, $i = 1, ..., n$

Premenné $X_1, ..., X_n$ produkujú reťazce zo Σ^*, ktoré sa potom použijú na nahrádzanie premenných vo vzore π, čím sa dokonale simuluje spôsob získavania slov pre vymazávajúce vzory, teda $L_E(\pi) = L_n(\Gamma, X_0)$.

Pre nevymazávajúce vzory stačí uvažovať pattern systém Γ', ktorý bude zhodný s Γ s jediným rozdielom: $t(X_i) = \Sigma$, $i = 1, ..., n$. Potom $L_{NE}(\pi) = L_n(\Gamma', X_0)$.

Stačí už len dokázať, že uvažované inklúzie sú vlastné. Na to nám poslúži napríklad jazyk $L_1 = \{a^{2^{\wedge}n} \mid n \geq 0\} \cup \{b^{2^{\wedge}n} \mid n \geq 0\}$, ktorý patrí do NSPL, ale nepatrí ani do jednej z tried PL_E resp. PL_{NE}, pretože, ako je zrejmé z definície týchto dvoch tried, ak uvažujeme jazyky nad abecedou $\{a, b\}$, tak každý nekonečný jazyk z $PL_E \cup PL_{NE}$ obsahuje slová, v ktorých sa vyskytujú súčasne oba symboly naraz (neprázdne slová dosadzované za premenné sú ľubovoľné, môžu teda obsahovať akékoľvek symboly z danej abecedy).

□

Lema 6.2.2: $MPL_E \subset NSPL$ a $MPL_{NE} \subset NSPL$.

Dôkaz: Hlavná myšlienka dôkazu je rovnaká ako pre predchádzajúci prípad, opäť uvažujme abecedy terminálnych symbolov $\Sigma = \{a_1, a_2, ..., a_k\}$ a premenných $V = \{X_1, X_2, ..., X_n\}$. Nech Π je multivzor (teda konečná množina vzorov z $(\Sigma \cup V)^+$). Majme pattern systém:

$\Gamma = (\Sigma, V \cup \{X_0\}, p, t)$
$p(X_0) = \Pi$, $t(X_0) = \{w\}$, w je nejaké slovo z $L_E(\pi)$
$p(X_i) = \{a_j X_i \mid i = 1, ..., k\}$, $t(X_i) = \{\varepsilon\}$, $i = 1, ..., n$

Jediný drobný rozdiel oproti predchádzajúcemu prípadu je v množine $p(X_0)$. Následne dostávame: $L_E(\Pi) = L_n(\Gamma, X_0)$.

Nevymazávajúci prípad pre MPL_{NE} opäť ošetríme tak, že položíme $t(X_i) = \Sigma$, i = 1, ..., n.

Inklúzie sú vlastné, na dôkaz možno opäť použiť jazyk $L_1 = \{a^{2^{\wedge}n} \mid n \geq 0\} \cup \{b^{2^{\wedge}n} \mid n \geq 0\}$, ktorý nepatrí do MPL_E resp. do MPL_{NE} z rovnakého dôvodu ako pre PL_E a PL_{NE}.

□

Lema 6.2.3: $PL_{NG} \subset NSPL$.

Dôkaz: Nech Σ, V, A, P sú (v uvedenom poradí) množiny terminálov, premenných, axióm a vzorov. Pre danú gramatiku založenú na vzoroch G = (Σ, V, A, P) skonštruujeme nasledovný pattern systém:

$$\Gamma = (\Sigma, V, p, t)$$
$$p(X_i) = P \quad i = 1, ..., n$$
$$t(X_i) = A \quad i = 1, ..., n$$

V tomto prípade naozaj ľahko vidno, že $NL(G) = L_n(\Gamma, X_i)$ pre každé $i \in \{1, ..., n\}$. Stačilo zabezpečiť, aby si jednotlivé premenné boli navzájom rovné tak, ako je tomu v prípade uvažovaných gramatík.

Vlastná inklúzia sa dokáže pomocou jazyka $a^* \cup b^*$, o ktorom už vieme, že nepatrí do PL_{NG}. Na druhej strane je to bezkontextový jazyk a keďže platí, že $CF \subset NSPL$ (dokážeme neskôr), tak musí platiť aj $a^* \cup b^* \in NSPL$.

□

Lema 6.2.4: $IPL \subset WSPL$.

Dôkaz: Nech Σ, V, A sú (v uvedenom poradí) množiny terminálov, premenných, axióm a nech π je vzor. Pre danú gramatiku založenú na vzoroch G = (Σ, V, A, π) skonštruujeme nasledovný pattern systém:

$$\Gamma = (\Sigma, V, p, t)$$
$$p(X_i) = \pi \quad i = 1, ..., n$$
$$t(X_i) = A \quad i = 1, ..., n$$

Nie je ťažké vidieť, že synchronizovaný spôsob odvodzovania slov v uvažovanej gramatike zodpovedá slabej synchronizácii pattern systému Γ, teda $SL(G) = L_w(\Gamma, X_i)$, i = 1, ..., n, tým je dokázaná inklúzia.

Vlastnú inklúziu dokážeme opäť pomocou vyššie uvedeného jazyka $L_3 = a^+ \cup b^+ \in DSSPL$. Ako už vieme, DSSPL = DWSPL a triviálne platí: $DWSPL \subseteq WSPL$, teda $L_3 \in WSPL$. Podobným spôsobom ako pre jazyk $a^* \cup b^*$ sa dokáže, že ani $L_3 \notin IPL$ (pozri dôkaz XXX). Z uvedených skutočností vyplýva vlastná inklúzia.

□

Teraz ešte uvedieme niekoľko výsledkov týkajúcich sa tried definovaných priamo pattern systémami a niektoré z nich porovnáme s niektorými triedami jazykov Chomského hierarchie.

Veta 6.2.5: NSPL $\subset$ WSPL

Dôkaz: Nech $\Gamma = (\Sigma, V, p, t)$, $V = \{X_1, ..., X_n\}$ je nesynchronizovaný pattern systém (NSPS). Skonštruujeme slabo synchronizovaný pattern systém (WSPS) $\Gamma' = (\Sigma, V, p', t)$, kde $p'(X_i) = p(X_i) \cup \{X_i\}$ pre všetky $i = 1, ..., n$. Týmto sa zabezpečí, že $D^{(k)}(\Gamma') \subseteq D^{(k+1)}(\Gamma')$ pre všetky $k \geq 0$, presnejšie: všetky slová, ktorými danú premennú budeme môcť nahrádzať sa týmto prenesú do každého nasledujúceho kroku (teda v predchádzajúcom kroku odvodenia budú k dispozícii aj všetky od začiatku odvodené slová, čím sa zabezpečí ich dostupnosť pri nahrádzaní premenných tak, ako je to typické pre nesynchronizovaný spôsob odvodzovania).

S využitím inklúzie $D^{(k)}(\Gamma') \subseteq D^{(k+1)}(\Gamma')$ sa dá indukciou ľahko ukázať, že $D^{(k)}(\Gamma') = D_1^{(k)}(\Gamma) \times D_2^{(k)}(\Gamma) \times ... \times D_n^{(k)}(\Gamma)$ pre všetky $k \geq 0$, z čoho vyplýva, že $L_n(\Gamma, X_i) = L_w(\Gamma', X_i)$ pre všetky $X_i \in V$. To, že slabo synchronizovaný pattern systém má väčšiu generatívnu silu, dokazuje jazyk $L_4 = \{a^n b^n c^n \mid n \geq 1\}$, o ktorom už vieme, že nepatrí do NSPL, ale patrí do WSPL (patrí dokonca do DWSPL = DSSPL).

□

Veta 6.2.6: CF $\subset$ NSPL

Dôkaz: Nech $G = (N, T, P, S)$ je bezkontextová gramatika, kde N je množina neterminálov, T množina terminálov, P sú pravidlá a S je počiatočný neterminál. Uvažujme nasledovný normálny tvar pre G: pre každý neterminál $A \in N$ existuje pravidlo $A \rightarrow x$, kde $x \in T^*$ (ak také neexistuje, tak ho pridáme, pričom vyberieme nejaké $x \in L(G_A)$, kde $G_A = (N, T, P, A)$) a na pravej strane každého pravidla sa každý neterminál vyskytuje najviac jedenkrát (samozrejme, môže tam byť viac neterminálov, ale všetky musia byť navzájom rôzne, to sa dá zabezpečiť tak, že prípadný viacnásobný výskyt nejakého neterminálu A nahradíme novými neterminálmi $A_1, ..., A_k$ a pridáme nové pravidlá $A_i \rightarrow A$, $A_i \rightarrow x$, $i = 1, ..., k$ a x je pravá strana pravidla $A \rightarrow x$).

Pattern systém $\Gamma = (T, N, p, t)$, kde

$$p(A) = \{\alpha \mid A \rightarrow \alpha \in P, \alpha \notin T^*\}$$
$$t(A) = \{x \mid A \rightarrow x \in P, x \in T^*\}$$

Celkom priamočiaro sa dá vidieť, že $L(G) = L_n(\Gamma, S)$, teda CF $\subseteq$ NSPL.
Vlastná inklúzia vyplýva z toho, že napríklad jazyk $L_1 = \{a^{2\wedge n} \mid n \geq 0\} \cup \{b^{2\wedge n} \mid n \geq 0\} \in$ NSPL nie je bezkontextový.

□

Veta 6.2.7: REG $\subset$ SSPL

Dôkaz: Podobne ako predchádzajúca veta. Každý vzor obsahuje iba jeden výskyt premennej, čo zaručuje bezproblémovú synchronizáciu.
Vlastnosť inklúzie je zrejmá , napr. $L_4 = \{a^n b^n c^n \mid n \geq 1\} \in$ DSSPL (a teda triviálne patrí aj do SSPL), pritom nie je ani bezkontextový.

Získali sme základnú predstavu fungovania pattern-systémov. Tému sme iste nevyčerpali, existujú ešte aj ďalšie výsledky v tejto oblasti vrátane zaujímavých vzťahov s triedami jazykov L-systémov, pre čitateľa s hlbším záujmom o problematiku odporúčame [12]. V tejto oblasti existuje tak isto viacero otvorených problémov.

Záver

Zoznámili sme sa s rôznymi spôsobmi definovania jazykov pomocou vzorov. Spoznali sme niektoré základné vlastnosti takto definovaných tried jazykov. Zdá sa, že u nás takýto spôsob definovania formálnych jazykov nie je veľmi známy. Aj preto sme sa pokúsili pokryť mnohé oblasti, ktorým sa venujú rôzne odborné články. Tému sme rozhodne nevyčerpali, táto práca by mala slúžiť ako úvod do problematiky. Viaceré skúmané oblasti sú pomerne mladé s množstvom otvorených otázok a ešte len čakajú na svoje prípadné uplatnenie (či už praktické alebo teoretické – napríklad môžu pomôcť aj pri rozvíjaní iných súvisiacich teórii). Naznačili sme aj praktické využitie vzorov napríklad v teórii strojového učenia. Pokúsili sme sa zaviesť jednotnú terminológiu, na viacerých miestach sme sa pokúsili zúplniť text jednoduchými príkladmi, komentármi uľahčujúcimi pochopenie danej témy, či komentár k niektorým dôkazom, aby čitateľ nebol zahltený len formálnymi detailami, niekde sme aj doplnili niektoré jednoduché tvrdenia.

Keďže počas spracovávania témy sme narazili na dosť veľa otvorených problémov prakticky v každej preberanej oblasti, otvára sa tým veľký priestor na ďalší výskum. Mnohé z tém môžu byť dobrou inšpiráciou pre ďalšie diplomové práce, ktoré môžu na rozdiel od tohoto prehľadu so širokým záberom podrobnejšie rozpracovať niektorú z uvedených oblastí, pokúsiť sa do nej vniknúť hlbšie a možno aj vyriešiť niektoré z otvorených otázok. Niektoré možnosti ďalšieho výskumu sme už priamo naznačili v hlavnom texte alebo z neho pripamo vyplývajú. Ako príklad na záver uveďme problém ekvivalencie pre triedu PL_{NE}, kde zatiaľ nie je známe, či je rozhodnuteľný. Témou, ktorá zatiaľ podľa našich vedomostí nebola rozpracovaná v článkoch sú synchronizované gramatiky založené na vzoroch (zovšeobecnenie iterovaných pattern languages). Aké sú základné vlastnosti triedy jazykov definovaných takýmito gramatikami? Aký je ich vzťah k pattern-systémom? Existuje tam nejaká hierarchia tried jazykov podľa počtu vzorov?

Dodatok

L-systémy

Definícia: 0L-systém je trojica $G = (\Sigma, P, w)$, kde Σ je abeceda, $P \subseteq \Sigma \times \Sigma^*$ sú prepisovacie pravidlá, pričom pre každé $a \in \Sigma$ existuje aspoň 1 pravidlo (teda $proj_1(P) = \Sigma$) a $w \in \Sigma^+$ je axióma.

Krok odvodenia je relácia $\Rightarrow$ na slovách zo Σ^*, pre ktorú platí: $u \Rightarrow w$, ak $u = a_1 \ldots a_n$, $v = b_1 \ldots b_n$, pričom $a_i \rightarrow b_i \in P$ pre $i = 1, \ldots, n$. Prepisuje sa teda každý symbol.

Jazyk definovaný 0L-sysémom G je $L(G) = \{ v \in \Sigma^* \mid w \Rightarrow^* v \}$.

Triedu jazykov generovanú 0L-systémami označme jednoducho OL.

Túto základnú definíciu 0L-systému možno samozrejme rôznymi spôsobmi modifikovať a tým získame ďalšie triedy jazykov. Definujeme si tie spôsoby, ktoré budú užitočné z hľadiska tejto práce, nepôjde teda o vyčerpávajúci prehľad.

Definícia:
D0L-systém je 0L-systém, kde pre každé $a \in \Sigma$ existuje práve jedno prepisovacie pravidlo.
P0L-systém je 0L-systém, v ktorom $P \subseteq \Sigma \times \Sigma^+$, teda žiadne písmeno sa nemôže prepísať na ε.

T0L-systém trojica $G = (V, \{P_1, \ldots, P_k\}, w)$, teda miesto jednej sady pravidiel máme nejakú konečnú množinu. V danom kroku odvodenia použijeme vždy práve jednu sadu pravidiel.

E0L-systém je štvorica $G = (N, T, P, w)$, kde N resp. T sú disjunktné abecedy neterminálnych resp. terminálnych symbolov, $P \subseteq (N \cup T) \times (N \cup T)^*$ a podobne ako pre 0L-systémy aj tu existuje aspoň 1 pravidlo pre všetky $a \in (N \cup T)$, $w \in (N \cup T)^+$. Krok odvodenia je relácia $\Rightarrow$ na slovách z $(N \cup T)$ definovaná ďalej rovnako ako pre 0L-systémy. Jazyk generovaný takýmto systémom je $L(G) = \{v \in T^* \mid w \Rightarrow^* v\}$.

Triedy jazykov definované týmito systémami označme: DOL, POL, TOL, EOL, HOL. Význam jednotlivých písmen je nasledovný: D-deterministic, P-propagating, T-table, E-extended.
Práve definované rozšírenia 0L-systémov možno ľubovoľne kombinovať, môžeme mať teda napríklad aj PD0L-systém, ET0L-systém, PDT0L-systém atď. Napríklad pre posledne menovaný systém platí, že máme viac sád pravidiel (**T**able), pre každý symbol abecedy existuje aspoň 1 pravidlo (**D**eterministic), pravé strany pravidiel majú dĺžku aspoň 1 (**P**ropagating).
Pri rôznom kombinovaní nám teda môžu vzniknúť rôzne triedy jazykov, ktoré budeme pri zachovaní uvedeného poradia písmen všeobecne označovať ako [E][P][D][T]OL. Písmená v hranatých zátvorkách sa použijú podľa potreby, pri označení konkrétnej triedy vynecháme samozrejme aj zátvorky, teda napríklad PDOL, ETOL, PDTOL, EDTOL, EPOL, EOL, atď.

Nech $G = (\{a, b\}, \{a \rightarrow aa, b \rightarrow bb\}, ab)$. Potom $L(G) = \{a^{2^n} b^{2^n} ; n \geq 0\}$ $(\in$ PDOL$)$

Niektoré základné vzťahy: $OL \subset TOL \subset EOL \subset ETOL \subset ECS$
$CF \subset EOL$
OL, TOL sú neporovnateľné s FIN, REG, CF

Použitá literatúra

[1] D. Angluin, Finding patterns common to a set of strings, STOC 17 (1979) 130-141

[2] K. Salomaa, Patterns (2003)

[3] T. Jiang, E. Kinber, A. Salomaa, K. Salomaa, S. Yu, Pattern languages with and without erasing, Intern J. Computer Math 50 (1994) 147-163

[4] T. Jiang, A. Salomaa, K. Salomaa, S. Yu, Decision problems for patterns, Journal of Computer and System Sciences 50 (1995) 53-63

[5] L. Kari, A. Mateescu, G. Păun, A. Salomaa, Multi-pattern languages, Theoretical Computer Science 141 (1995) 253-268

[6] S. Dumitrescu, G. Păun, A. Salomaa, Languages associated to finite and infinite sets of patterns

[7] A. Mateescu, A. Salomaa Finite degrees of ambiguity in pattern languages, Theoretical Informatics and Applications (1994) 233-253

[8] S. Dumitrescu, G.Păun, A. Salomaa, Pattern languages versus parallel communicating grammar systems (1996)

[9] V. Mitrana, Patterns and Languages: An Overview

[10] J. Dassow, G. Păun, A. Salomaa, Grammars based on patterns

[11] V. Mitrana, Iterated pattern languages

[12] V. Mitrana, G. Păun, G. Rozenberg, A. Salomaa, Pattern Systems

[13] A. Mateescu, A. Salomaa Hanbook of Formal Languages Vol. 1-3

Ďalej boli použité poznámky z prednášok:
Prof. RNDr. Branislav Rovan, PhD.:
1. Formálne jazyky a automaty
2. Teória paralelných výpočtov
3. Vybrané partie z teórie jazykov

RNDr. Mária Pastorová
Biologicky motivovaná teória jazykov

Ďalšia odporúčaná literatúra

[o1] A.Thue: Über unendliche Zeichenreihen, Norske Vid. Selsk. Skr., I Mat. Nat. Kl. Krisitiania 7 (1906) 1-22

[o2] O. Ibarra, Reversal-bounded multicounter machines and their decision problems, Journal of the ACM 25 (1978) 116-133

[o3] D. Angluin, Finding patterns common to a set of strings, Journal of Computer and System Sciences 21 (1980) 46-62

[o4] K. Jantke, Polynomial time inference of general pattern languages. Proc. STACS'84, Lect. Notes Comput. Sci. 166, 1984, pp. 314-325

[o5] K.-I Ko and W.-G. Tzeng, Three Σ_2^P-complete problems in computational learning theory. Computational Complexity 1 (1991) 269-310

[o6] A. Mateescu, A. Salomaa, Nondeterminism in patterns, Proc. STACS'94, Lect. Notes Comput. Sci. 775, Springer, 1994, pp. 661-668

[o7] G. Phaun, G. Rozenberg, A. Salomaa, Pattern grammars, Journal of Automata, Languages and Combinatorics 1: 219-235 (1996)

[o8] E. Ohlebusch and E. Ukkonen, On the equivalence problem for E-pattern languages, Theoretical Computer Science 186 (1997) 231-248