UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

E0L Formy

Diplomová práca

2014
Bc. Jozef Fekiač

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

E0L Formy

Diplomová práca

Študijný program: Informatika
Študijný odbor: 9.2.1 Informatika
Číslo študijného odboru: 2508
Školiace pracovisko: Katedra Informatiky FMFI
Školitel: RNDr. Mária Pastorová

Bratislava, 2014
Bc. Jozef Fekiač

64598018

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Jozef Fekiač
Študijný program: informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.2.1. informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský

Názov: E0L formy

Cieľ: Skúmanie Lindenmayerových systémov s vyčlenením terminálnych symbolov v abecede. Rozšírené 0L formy, zhodnotenie vzhľadom na Chomského hierarchiu. Vizualizácia.

Vedúci: RNDr. Mária Pastorová
Katedra: FMFI.KI - Katedra informatiky
Vedúci katedry: doc. RNDr. Daniel Olejár, PhD.

Spôsob sprístupnenia elektronickej verzie práce:
archív

Dátum zadania: 28.11.2012

Dátum schválenia: 28.11.2012

prof. RNDr. Branislav Rovan, PhD.
garant študijného programu

....
študent

....
vedúci práce

Abstrakt

Práca sa zaoberá E0L systémami, E0L formami a reláciou interpretácie, ich vlastnosťami, normálnymi E0L formami, ďalej vzťahom E0L foriem vzhľadom na Chomského hierarchiu, vizualizáciou operácií nad E0L formami.
Kľúčové slová: E0L systém, E0L forma, dfl-substitúcia, vizualizácia E0L foriem

Abstrakt

The thesis deals with E0L systems, E0L forms and relation of interpretations, their properties, normal forms of E0L. Furthermore with relationship of E0L forms and Chomsky hierarchy, visualization of E0L forms and operations on them.
Keywords: E0L system, E0L form, dfl-substitution, visualization of E0L forms.

Čestné prehlásenie

Čestne prehlasujem, že som túto prácu vypracoval sám s použitím uvedených zdrojov.

Poďakovanie

Ďakujem RNDr. Márii Pastorovej za poskytnutie potrebnej literatúry,

materiálov a užitočných rád, rovnako za morálnu podporu pri písaní práce.

Ďakujem svojim rodičom a starým rodičom, priateľke a všetkým blízkym, ktorí sa podieľali na rozhodnutí nevzdať sa, dotiahnuť úsilie do konca, a zvíťaziť.

Obsah

Úvod

L-systémy Prvý záznam o vytvorení modelu pre najjednoduchšie L-systémy sa datuje na rok 1968, kde A. Lindemayer popísal matematický model bunkovej interakcie vláknitých rias.
L-systémy sa zakladajú na paralelnom prepisovaní symbolov, čím popisovali správanie bunkových organizmov z hľadiska ich vývoja. Neskôr sa podarilo model obohatiť o rozšírenia ktoré umožnili modelova zložitejšie organizmy, rastliny a fraktály.
O značnú časť teoretického výskumu o ktoré sa zaslúžil P. Prusinkiewicz, G. Rozenberg, A.Salomaa ale aj iní.
Podarilo sa pochopiť a vysvetliť fungovanie rastu stromov a bylín, na základe ktorých zostrojili formálny model stromových 0L systémov, čo bolo neskôr rozšírené na uzátvorkované 0L systémy, ktoré sa už ľahšie čítali, a boli lepšie pochopiteľné.

V ďalších pasážach sa môžeme dočítať o stochastických 0L systémoch, ktoré mali pri pravidlách uvedenú pravdepodobnosť použitia, čo neznamenalo väčšiu rozmanitosť vygenerovaných slov-modelov rastlín, ale mohla sa skúmať pravdepodobnosť vzniku.

Podobným spôsobom pokrývajú aj systémy s interakciou, kde to bolo zložitejšie, nakoľko prirodzene sú generované slová lineárne, v jednom rade, bez vetvení, ktoré je umelo vsunuté zátvorkami alebo iným, podobným spôsobom. Zaviedli možnosť definície množiny symbolov, ktoré treba vynechať pri hľadaní kontextu pri aplikovaní pravidiel na prepis.
Inšpirácia z klasickej teórie jazykov v spojení s paralelným prepisovaním neskôr našla využitie v počítačovej grafike na vykresľovanie realistických objektov, čím vzniklo množstvo podnetov pre ďalší výskum.
Jedno z prvých zásadných zmien, ktoré zasiahli 0L systémy bolo rozšírenie o terminálnu abecedu, čím sa umožnilo vytvoriť systém neobsahujúci všetky slová pre ktoré existuje odvodenie pomocou pravidiel. Slová z jazyka začali byť obmedzené faktom, že museli obsahovať iba terminálne symboly.
Týmto rozšírením sa dosiahlo pokrytie triedy bezkontextových jazykov.

Práve kvôli porovnaniu s gramatikami, ktoré mali sekvenčný prepis, začalo byť zaujímavé skúmanie doterajších tried jazykov v porovnaní s rôznymi špeciálnymi tvarmi E0L systémov.

Tento fakt motivoval zavedenie pojmu E0L forma, spolu s definovaním mechanizmu interpretácie.
E0L forma je nič iné, ako E0L systém. Rozdiel je uhol pohľadu, z ktorého skúmame konštrukcie.
Práve interpretácia E0L systému, ktorá sa zakladá na pridávaní a vynechávaní pravidiel, ktoré sa podobajú pôvodným pravidlám v skúmanej E0L forme, nám dáva nový, abstraktnejší pohľad na problematiku E0L systémov.

Zo záverečných prác v slovenčine, ktoré sú na FMFI UK pomerne dobre dostupné, sú z hľadiska ozrejmenia fungovania L-systémov sú to nasledovné:

T0L systémy nad dospelými jazykmi, J. Hudáček, 2004

Lindenmayerove systémy s interakciou, J. Minárik, 2006

Rozšírenia Lindenmayerových systémov s interakciou, M.Vaško, 2010

Dospelé jazyky k Lindenmayerovym systémom, J. Fekiač, 2011

Najväčšou inšpiráciou pri E0L systémoch bola práca

Rozšírenia Lindenmayerovych systémov s interakciou od M. Vaška sa zaoberá najmä tabuľkovými a rozšírenými L-systémemi s interakciou.

Zaujíma sa o zaradenie rozšírených systémov do Chomského hierarchie, čo bolo dobrým podkladom pre skúmanie pohľadu cez E0L formy v tejto práci. Spracúva základy o E0L systémoch, ich uzáverové vlastnosti a uvádza ilustrujúce príklady, čím ukázala možné smerovania výskumu.

Výsledky sa vyskutujú aj pre ET0L systémy, niektoré z nich môžu byť použiteľné aj pre E0L systémy, nakoľko E0L systém je špeciálnym prípadom ET0L.

Pri nich sa spomína normálny tvar so zamietnutím, synchronizovaný ET0L systém a jeho konštrukcia, vyjadrené je aj zaradenie triedy jazykov generovaných E0L a ET0L medzi triedy jazykov generovaných CF a CS gramatikami.

Práca Dospelé jazyky k Lindenmayerovym systémom pokrýva základné vedomosti o 0L systémoch, dospelých jazykoch k 0L systémom. Obsahuje ďalšie vlastnosti E0L systémov, ktoré sú neskôr využité pri ozrejmovaní do-teraz nedokázaných o T0L systémoch, AT0L a ET0L systémoch.

Dokázaná je rovnosť množín jazykov generovaných rozšírenými tabuľkovými L systémami a dospelými jazykmi k tabuľkovým L systémom, čím sa vybudovalo prepojenie medzi E0L systémami a bezkontextovými gramatikami.

Táto práca má 4 kapitoly. V prvej kapitole sa budeme zaoberať zadefinovaním E0L systémov, normálnymi tvarmi, zisťovaním dôležitých vlastností a ich ilustrovaním na príkladoch,čím zaručíme podrobné vysvetlenie.

V druhej kapitole sa zaoberáme zadefinovaním E0L foriem, čo zahŕňa definovanie dfl-substitúcie, definovanie relácie interpretácie,
ďalej ukazujeme interpretáciu E0L formy na príklade, popisujeme dôležité vlastnosti relácie interpretácie, zisťujeme charakteristické vlastnosti dfl-substitúcie, charakterizujeme vznik tried jazykov z E0L foriem a definujeme špeciálne tvary E0L foriem.

Tretia kapitola sa zaoberá vzťahmi E0L foriem vzhľadom na Chomského hierarchiu. Tu ukazujeme príklady E0L foriem ktoré generujú jednotlivé triedy jazykov z Chomského hierarchie.

Štvrtá kapitola sa zaoberá vizualizáciou E0L foriem a operácií na nich, čím sa dosahuje ľahšie porozumenie a skúmanie mechanizmov.

1 L-systémy s vymedzením terminálnej abecedy

Túto triedu rozšírených L-systémov označujeme E0L.

1.1 Základné pojmy a tvrdenia

Definícia 1.1.1.
E0L systém je definovaný ako štvorica $\mathcal{S}$=(V,P,x,Σ),
kde
V≠{}, V je konečná množina symbolov(nazývame ju kompletná abeceda), obsahuje terminálne aj neterminálne symboly.

Σ je terminálna abeceda.
P je množina pravidiel, P $\subseteq V \times V^*$, ktorá je tiež konečná, zároveň musí byť úplná (t.j. $\forall a \in V \exists (a, \alpha) \in P$)
Pozn.: Intuitívnejší zápis dvojice (a, α)bude $a \to \alpha$.
Pre pravidlá s rovnakou ľavou stranou môžeme použiť skrátený zápis na základe predlohy v nasledujúcom príklade:

$$a \to \alpha, a \to \beta \text{ skrátime na } a \to \alpha \mid \beta.$$

Množinu P nazývame tabuľka pravidiel.
$x \in V^+$ je axióma, počiatočné slovo.

Aj pri E0L systémoch môžeme vyčleniť podtriedu deterministických ED0L systémov, kde pre každý symbol abecedy existuje práve jedno pravidlo, čím je zaručené deterministicky jeho použitie v odvodení, ak je to možné.

Definícia 1.1.2.
Krok odvodenia v E0L systéme $\mathcal{S}$=(V,P,x,Σ) je definovaný nasledovne:

Nech
$$v = a_1a_2a_3...a_k \in V^*, a_i \in V, i = 1...k \in \mathbb{N}\ , \qquad w \in V^*$$
potom
$$\text{v} \implies \text{w (v systéme } \mathcal{S}\text{) vtedy a len vtedy,}$$
ak
$$\exists w = \alpha_1\alpha_2\alpha_3...\alpha_k, k \in \mathbb{N}, a_i \to \alpha_i (i = 1...k)$$

Viac krokov znázorňujeme v $\overset{k}{\implies}$ w pre daný počet krokov k resp. v $\overset{+}{\implies}$ w pre ľubovoľný kladný počet krokov.

Definícia 1.1.3.
Jazyk generovaný systémom $\mathcal{S}$=(V,P,x,Σ)
je množina

$$\mathrm{L}(\mathcal{S})' = \{v \in V^*| \ x \ \stackrel{*}{\Longrightarrow} \ w\} \cap \Sigma^*$$

kde $\stackrel{*}{\Longrightarrow}$ je reflexívno-tranzitívny uzáver.

V skratke povedané, do jazyka generovaného E0L systémom patria všetky slová, ktoré sú odvodené konečným počtom platných krokov odvodenia a zároveň obsahujú výhradne terminálne symboly.

Príklad 1.1.1. Pre ilustráciu odvodenia terminálneho slova v E0L systéme uvedieme jednoduchý príklad:

Nech $\mathcal{S} = \{V, p, x, \Sigma\}$,
$V = \{A, B, C, a, b, c\}$
$x = ABC$,
$P = \{A \to Aa \mid a$
$B \to Bb \mid b$
$C \to Cc \mid c$
$r \to r, r \in \{a, b, c\}\}$
$\Sigma = \{a, b, c\}$

Na tomto systéme môžeme vidieť, že máme 3 neterminály, ktoré sa prepisujú na samého seba + terminálny symbol. Nedeterminizmus výberu pravidla zaručí, že v istom okamihu sa prvý(a jediný krát, pravidlá ale nemusia nutne byť tohoto tvaru. Pripomeňme si, E0L systémy majú pravidlá tvaru VxV^*, pričom V obsahuje aj terminálne aj neterminálne symboly.) prepíše neterminálny symbol na terminálny. Je zjavné, že jazyk generovaný týmto E0L systémom je $L = \{a^x b^y c^z | x, y, z > 0\}$

Definícia 1.1.4. Počiatočný jazyk z neterminálov v S
Nech $\mathcal{S} = \{V, P, x, \Sigma\}$,
Potom jazyk $L_{innt}(S)$ je definovaný nasledovne:
$L_{innt}(S) = \{x \in \Sigma^*|$ existuje odvodenie R slova x také, že každé slovo y z odvodenia R obsahuje iba neterminálne symboly$\}$

Je jasné, že vo všeobecnosti platí, že $L(S) \neq L_{innt}(S)$

1.2 Synchronizovaný tvar E0L systému

Veta 1.
Ku každému E0L systému $\mathcal{S}$ existuje ekvivalentný synchronizovaný $\mathcal{S}'$, kde $v \in L(\mathcal{S})$ sa odvodia synchronizovane
(t.j. $\underset{a\in\Sigma}{\forall} a \overset{+}{\Longrightarrow} v, v \notin \Sigma^+$)

Dôkaz:
Zaveďme nasledovný homomorfizmus:
$h(a) = \{$
$\bar{a} \mid a \in \Sigma$
$a \mid a \in V - \Sigma$
$\}$
$\bar{\Sigma} = \{\bar{a} \mid a \in \Sigma\}$, tieto symboly sa nevyskytujú vo V
$\mathcal{S}' = (V', P', x', \Sigma)$
$V' = V \cup \bar{\Sigma} \cup \{F\}$, kde $F \notin V$
x'=h(x)
$P' = \{a \to F \mid a \in \Sigma \cup \{F\}\} \cup$
$\cup\{\bar{a} \to h(\alpha) \mid a \in V, a \to \alpha \in P\}\cup$
$\cup\{\bar{a} \to a \mid a \in \Sigma\}$

Pre slová $v \in L(\mathcal{S})$ vždy v $\mathcal{S}'$ pred vznikom terminálneho slova existuje neterminálne, ktoré má symboly s pruhom. Pri nesynchronizovanom odvodení sa terminálny symbol, ktorý sa prepísal skôr, pri nasledovnom kroku odvodenia prepíše na zamietací symbol, ktorý je neterminálny. Tým zaručíme, že nesynchronizovaným odvodením nevzniknú slová jazyka $L(\mathcal{S}')$.

Dôsledok 1. $\mathcal{L}(E0L) = \mathcal{L}_{innt}(E0L)$

Na základe predchádzajúcej vety je zjavné, že každý E0L systém S vieme vyjadriť v synchronizovanom tvare,
teda každý jazyk L(S) vieme vyjadriť ako $L_{innt}(\bar{S})$, kde $\bar{S}$ je synchronizovaný E0L systém skonštruovaný z S podľa vety 1.

Na základe tohoto príkladu bude jednochšie pochopiteľná predošlá veta.

Príklad 1.2.1. .
Nech $L = \{a^n b^n c^n \mid n > 0\}$.

Tento jazyk vieme pomerne jednoducho vygenerovať E0L systémom, pokiaľ zavedieme správnu axiómu a odvodzovacie pravidlá.

Nech $\mathcal{S} = \{V, P, x, \Sigma\}$,
pričom $V = \{A, B, C, a, b, c, \bar{a}, \bar{b}, \bar{c}, F\}$
$x = ABC$,
$P = \{A \to A\bar{a} \mid a$
$B \to B\bar{b} \mid b$
$C \to C\bar{c} \mid c$
$\bar{a} \to \bar{a} \mid a$
$\bar{b} \to \bar{b} \mid b$
$\bar{c} \to \bar{c} \mid c$
$r \to F, r \in \{a, b, c, F\}\}$
$\Sigma = \{a, b, c\}$

Týmito pravidlami máme zaručené, že pokiaľ symboly A,B,C, ktoré sú zodpovedné za rast slova, sa premenia na terminálne, tak už to musia urobiť všetky naraz, spolu s neterminálnymi symbolmi s pruhom(z ktorých sa stanú terminály bez pruhu).

V ďalšom kroku odvodenia všetky terminálne znaky sa zmenia na neterminálny zamietací symbol F. Z definície E0L do jazyka patria len slová zložené z terminálov, teda zaručene sa nám budú odvodzovať všetky slová synchronizovane.

Synchronizáciou vieme kontrolovať tvorenie symbolov na viacerých miestach bez toho, aby sme potrebovali využit pravidlá interakciou, ktoré sú dostupné až v systémoch iného druhu ako je E0L.

Synchronizovaný tvar bude témou, ktorou sa budeme zaoberať rovnako aj v ďalšej kapitole, nakoľko je jednou zo silných vlastností E0L, bez ktorej by sme nedokázali generovať jazyky význačné pre E0L, ktoré nie sú z triedy bezkontextových jazykov.
Porovnanie s Chomského hierarchiou bude témou tejto práce až neskôr.

2 E0L formy

2.1 Základné definície a tvrdenia

E0L formy sú druhom abstrakcie nad E0L systémami, a dajú sa pomocou nich zaviesť triedy podobných systémov, rovnako ako jazykov generovaných týmito triedami systémov.

E0L forma sa tvorí model alebo hlavný E0L systém, schopný definovať triedu podobných E0L systémov pomocou interpretačného mechanizmu.

Štúdium E0L foriem je podobná ako štúdium gramatických foriem pre frázové gramatiky. Týmto štúdiom bolo možné pozerať na niektoré aspekty okolo teórie E0L systémov z iného uhlu.

Dôležité je zdôrazniť, že teória o E0L systémoch ako aj o všeobecných L systémch je rozsiahla a rozmanitá, preto sa budeme zaoberať len vybranými časťami, ktoré sú pre nás najpodstatnejšie.

V skratke povedané, E0L forma nie je nič všeobecnejšie ako E0L systém. Pokiaľ sa pozeráme na E0L systém S ako na E0L formu, potom môžeme využiť mechanizmus pre výrobu interpretácii S' vytvorené z S, symbolicky zapísané S' $\triangleleft$ S. Každá interpretácia pôvodného E0L systému je tiež E0L systém. Všetky interpretácie S' vytvorené z S tvoria triedu označenú $\mathcal{G}$(S), a všetky jazyky generované týmito interpretáciami tvoria triedu $\mathcal{L}$(S).

Potrebné je teraz ale definovať konštrukciu interpretácie. Pre zdôraznenie základnej idei začneme všeobecnejšie.

Definícia 2.1.1. Dfl-substitúcia
Substitúcia σ definovaná na abecede Σ sa nazýva disjunktná konečnopísmenná substitúcia, v skratke nazývaná dfl-substitúcia, ak pre každé písmeno a z abecedy Σ je $\sigma(a)$ konečná neprázdna množina písmen, a zároveň pre ľubovoľné dve písmená a,b z abecedy Σ platí

$$\sigma(a) \cap \sigma(b) = \emptyset \text{ ak a} \neq \text{b}.$$

Písmená v substitúcii môžu byť aj z inej abecedy ako je Σ.

Keďže už vieme čo je to dfl-substitúcia, môžeme si ukázať jej fungovanie na jednoduchom príklade.

Príklad 2.1.1. Zoberme ľubovoľný systém S=(Σ,P,s), kde Σ je abeceda a P je konečná množina pravidiel tvaru x → y, s je axióma.
S v tomto prípade označujeme prepisovacia forma.

Nech μ je dfl-substitúcia. Definujme ju jednoducho: pre pravidlo x' → y' existuje v prepisovacej forme pravidlo x → y také, že $x' \in \mu(x), y' \in \mu(y)$

Potom prepisovacia forma S'=(Σ',P',s') sa vola interpretácia formy S modulo μ , symbolicky $S' \triangleleft S(\mu)$ alebo skrátene $S' \triangleleft S$, pokiaľ

$$\Sigma' \subseteq \bigcup_{a\in\Sigma} \mu(a) = \mu(\Sigma) \text{ a P' } \subseteq \mu(P).$$

Teda interpretácia S vzniká priradením každému písmenu z abecedy Σ istý počet interpretácií nasledovnou substitúciou

$$\mu(a) = \{a_1, a_2, \dots, a_k\}$$

tak, aby sme zachovali disjunktnosť dfl-substitúcie, teda žiadne písmeno nemôže byť interpretáciou dvoch rôznych písmen.

Substituovaním všetkých písmen v pravidlách z P dostaneme novú množinu pravidiel $\mu(P)$. Dôležité je poznamenať, že môžeme vybrať ľubovoľnú podmnožinu $\mu(P)$, teda nie nutne všetky interpretácie. V opačnom prípade by sme boli príliš obmedzovaní.

Voľnosť výberu interpretácii má ale aj nevýhody. Niektoré odvodenia podľa pôvodného S sa v T môžu stratiť.

Môže byť zaujímavé pozorovať, že naše definície môžu byť priamo použité na prípad, kde S je konečný orientovaný graf, ktorý označíme hlavný graf, s množinou vrcholov Σ a hrán P. Naše definície potom tvoria triedu interpretácii hlavného grafu.

Pokiaľ medzi vrcholmi x a y nebola hrana, potom ani medzi x_i, y_j nebude hrana, pokiaľ x_i resp. y_j je interpretácia x resp. y. Na druhej strane existencia hrany medzi x a y nezaručuje existenciu hrany medzi x_i, y_j.

Dokázali by sme rozšíriť tie to tvrdenia aj na nekonečné množiny relácii, ale týmto sa nebudeme zaoberať.

Grafmi sa zaoberá kapitola 4: Vizualizácia.

Veta 2. Pokiaľ S, S' a S" sú prepisovacie systémy také, že platí $S' \lhd S$ a $S'' \lhd S'$, potom platí aj $S'' \lhd S$. Relácia interpretácie je tranzitívna. Pokiaľ dostaneme dva prepisovacie systémy S, S', vieme rozhodnúť o ich vzťahu vzhľadom na interpretáciu.

Veta 3. Prepisovacie systémy S a S' sú vo vzťahu S' $\lhd S$ iff $\mathcal{G}(S') \subseteq \mathcal{G}(S)$. Je rozhodnuteľné, či dva dané prepisovacie systémy sú striktne formovo ekvivalentné alebo nie.

Definícia 2.1.2. E0L forma a relácia interpretácie

Nech E0L forma je E0L systém S=(V, P, x,Σ), $x \in V - \Sigma$

Potom E0L systém S'=(V',Σ', P', x') je interpretáciou S modulo μ, symbolicky zapísané $S' \lhd S(\mu)$, skrátene $S' \lhd S$, ak je μ dfl substitúcia definovaná na Σ, a sú splnené nasledujúce podmienky:

(i) $\mu(\Sigma) \supseteq \Sigma'$ a $\mu(V) \supseteq V'$
(ii) $x' \in \mu(x)$;
(iii) $P' \subseteq \mu(P) = \bigcup_{\alpha \to x \in P} \mu(\alpha) \to \mu(x)$

Zhrnuté, interpretácia zachováva terminálnu abecedu, axiómu, a množina pravidiel je podmnožinou všetkých interpretácií pôvodných pravidiel.

Pre jednoduchosť budeme množine substituovaných pravidiel $\mu(P)$ hovoriť aj množina potenciálnych pravidiel.

Tu je dôležité poznamenať, že pravidlo (iii) je navyše obmedzené faktom, že interpretácia E0L systému musí byť E0L systém. Toto znamená, že pre každý symbol musí existovať aspoň jedno pravidlo na prepis. Výsledky predchádzajúcich viet 2 a 3 sú ale zachované.

2.2 Triedy generované interpretáciou E0L formy

Definícia 2.2.1. Triedy generované interpretáciou E0L schémy

Triedy E0L systémov a jazykov generovaných S sú definované nasledovne:

$$\mathcal{G}(S) = \{S' | S' \lhd S(\mu) \text{ pre nejaké } \mu\}$$

$$\mathcal{L}(S) = \{L(S') | S' \lhd S(\mu) \text{ pre nejaké } \mu\}$$

Definícia 2.2.2. (Striktná) Formová ekvivalencia

Dve E0L formy F a G sú formovo ekvivalentné(resp striktne formovo ekvivalentné), ak

$$\mathcal{L}(F) = \mathcal{L}(G) \text{ (resp } \mathcal{G}(F) = \mathcal{G}(G))$$

Príklad 2.2.1. Ukážeme príklad dvoch E0L systémov, kde platí, že druhý je interpretáciou prvého.
$S = (V, P, x, \Sigma)$
$V = \{S, A, B, C, R, a, b, c\}$
$\Sigma = \{a, b, c\}$
$P = \{S \to ABC, A \to a, B \to b, C \to c, C \to R,$
$a \to a, b \to b, c \to c, R \to a|b|c\}$

Vidno, že na 2 kroky odvodenia odvodíme jediné terminálne slovo, a to abc. Každý ďalší krok odvodenia nám pridá terminálne slová aba,abb.

Ukážme nejakú interpretáciu tejto E0L formy. Musí platiť, že množiny substitúcie rôznych symbolov musia byť disjunktné a axióma musí byť zachovaná.

Keď sa nad týmto zamyslíme, z našej E0L formy vieme spraviť len nasledovné:

1. Substitúcia S na rôzne symboly nám žiadnu silu nepridá, nakoľko S prepisujeme len jediný krát.
2. Substitúcia ľubovoľného iného symbolu nám spôsobí viacero možností na odvodenie.
3. Substitúcia terminálov môže spôsobiť, že vzniknú pravidlá na prepis terminálov na iné terminály (dfl-substitúcia môže terminál nahradiť len terminálom). Tomuto ale môžeme zabrániť správnym výberom pravidiel z množiny

potenciálnych pravidiel.

Zoberme substitúciu μ, definovanú nasledovne:
$\mu(X) = \{X\}, X1 \notin \{C, c\}$
$\mu(C) = \{C, D\}$
$\mu(c) = \{c, d\}$

Množina potenciálnych pravidiel bude nasledovná:
$P_{pot} = P \cup \{S \to ABD, C \to c, C \to d, D \to c, D \to d, D \to R, R \to a|b|c|d, c \to d, d \to c, d \to d\}$

Pretože máme možnosť výberu pravidiel z $\mu(P)$, môžeme napríklad rozhodnúť, že d sa bude odvodzovať výhradne na 2 kroky (vynecháme pravidlo $R \to d$), a teda že sa bude odvodzovať výhradne z neterminálneho symbolu D(vynecháme pravidlo $D \to c$).

Čiže naša nová E0L forma T bude na 2 kroky generovať slová abc, abd a na 3 kroky ešte navyše slová aba,abb.
Po zavedení substitúcie μ na systém S a vybraním z množiny potenciálnych pravidiel vznikne E0L forma nasledovného tvaru:
$T = (V', P', x, \Sigma')$
$V' = \{S, A, B, C, R, a, b, c, d\}$
$\Sigma' = \{a, b, c, d\}$
$P' = \{S \to ABC|ABD, A \to a, B \to b, C \to c, C \to R,$
$a \to a, b \to b, c \to c, R \to a|b|c, D \to d, d \to d\}$

Teraz je zaujímavé ukázať, či S a T sú formovo alebo striktne formovo ekvivalentné.

Príklad 2.2.2. Vezmime E0L systémy S a T z predchádzajúceho príkladu a skúsme zistiť, či sú v relácii $\triangleleft$

Postupovať budeme v dvoch krokoch.

Prvý krok, dokázať, že T $\triangleleft$ S, je zjavný.Systém T vznikol substitúciou zo systému S, teda táto relácia platí.

Druhý krok je S $\triangleleft$ T.

Jediná závažná zmena nastala na množine pravidiel.
$P = \{S \to ABC, A \to a, B \to b, C \to c, C \to R,$
$a \to a, b \to b, c \to c, R \to a|b|c\}$
$P' = \{S \to ABC|ABD, A \to a, B \to b, C \to c, C \to R,$

$a \to a, b \to b, c \to c, R \to a|b|c, D \to d, d \to d\}$

Zavedením substitúcie, tvaru $mu(x) = x$, odstránením pravidiel obsahujúcich symboly D alebo d, upravením V a Σ vznikne presne S.
Odstránili sme pravidlá $S \to ABD, D \to d, d \to d$ z P', čím vznikla presne množina P.

Tieto dva systémy sú preto striktne formovo ekvivalentné, teda aj formovo ekvivalentné, nakoľko rovnaké triedy systémov generujú nutne rovnaké jazyky.

Príklad 2.2.3. Uveďme príklad, kde platí relácia $\triangleleft$ iba v jednom smere.
$S = (V, P, x, \Sigma)$
$V = \{S, A, B, C, R, a, b, c\}$
$\Sigma = \{a, b, c\}$
$P = \{S \to ABC, A \to a, B \to b, C \to c, C \to R,$
$a \to a, b \to b, c \to c, R \to a|b|c\}$

Interpretáciou, pri ktorej vynecháme pravidlo $C \to R$, a nedosiahnuteľné pravidlo pre R, sa zjavne stráca možnosť spätného interpretovania na pôvodnú formu.

Pre nový systém $T = (V, P', x, \Sigma)$,
$P' = \{S \to ABC, A \to a, B \to b, C \to c, a \to a, b \to b, c \to c\}$
neplatí relácia S $\triangleleft$ T, pretože dfl-substitúciou nevieme vytvoriť interpretáciu pôvodného systému, ktorej pravidlá by umožnili dlhšie odvodenie obsahujúce len neterminálne slová. (v pôvodnom systéme S sa vyskytuje odvodenie $S \to ABC \to abR \to abc$, v novom systéme T toto odvodenie neexistuje a nie je možné ho vytvoriť ani interpretáciou T).

Vieme prehlásiť, že S a T nie sú striktne formovo ekvivalentné. Ostáva nám otázka, či sú formovo ekvivalentné.

V tomto prípade je dôležité nezabudnúť na disjunktnosť dfl-substitúcie. To znamená, že odvodenie slova aba v T ani jeho interpretáciách nemôže existovať, pretože v pôvodnej forme T neexistovali 2 pravidlá, ktoré sa líšia len na ľavej stranem, pričom na pravej strane je ten istý terminál.

Teda S a T nie sú ani formovo ekvivalentné.

Na záver uvedieme príklad, kde dve formy sú formovo ekvivalentné, ale nie sú striktne formovo ekvivalentné.

Príklad 2.2.4. Skrátenie odvodenia
Nech
$S = (V, P, S, \Sigma)$
$V = \{S, A, B, C, R, a, b, c\}$
$\Sigma = \{a, b, c\}$
$P = \{S \to ABC, A \to a, B \to b, C \to c,$
$a \to a, b \to b, c \to c\}$

V druhej forme, označenej T, budú neterminály vždy mať jednu medzifázu.
$T = (V, P', S, \Sigma)$
$V = \{S, A, B, C, R, a, b, c\}$
$\Sigma = \{a, b, c\}$
$P = \{S \to \bar{A}\bar{B}\bar{C}, \bar{A} \to A, \bar{B} \to B, \bar{C} \to C, A \to a, B \to b, C \to c,$
$a \to a, b \to b, c \to c\}$

Je zjavné, že v T budú vždy odovodenia dĺžky 3 kroky, na rozdiel od S, kde odvodenia sú dĺžky 2 kroky.
Túto vlastnosť zjavne nevieme dfl substitúciou nijak porušiť, najmä kvôli disjunktnosti, a vlastnosti pruhovaných neterminálov, pre ktoré neexistuje pravidlo, kde sa prepisujú na terminálny symbol.

Každej interpretácii v S, kde sa substituuje neterminálny symbol, vieme nájsť zodpovedajúcu substitúciu v T, na rovnakom neterminálnom symbole.

Substitúciou pruhovaných neterminálov nezískame žiadnu silu navyše, pretože ku každému pruhovanému neterminálu zodpovedá jeden nepruhovaný neterminál. Teda pokiaľ interpretujeme pruhovaný neterminál, pravé strany jeho pravidiel(v tomto prípade zatiaľ jediného) sa vždy na niektoré symboly z interpretácie symbolu jemu zodpovedajúcemu nepruhovanému neterminálu.

Ako príklad uveďme nové pravidlo $S \to \bar{A}'\bar{B}\bar{C}, \bar{A}' \to A'$. Odvodeniu v T ktoré obsahuje tieto dve pravidlá zodpovedá odvodenie podľa pravidla, ktoré bude v S nasledovné: $S \to A'BC$.

Pretože interpretáciou S alebo T vieme dostať cyklus v pravidlách jedine na terminálnych symboloch, do ktorých sa dostaneme vždy v rovnakom kroku odvodenia (vo forme S v druhom, vo forme T v treťom).To znamená, že substitúcie terminálnych symbolov a výbery pravidiel môžeme robiť rovnako aj v S aj v T.

Dôsledok 2. Operácia skrátenia Existuje operácia skrátanie odvodenia, ktorou narušíme striktnú formovú ekvivalenciu dvoch foriem, pričom ostanú formovo ekvivalentné, pretože pre každé odvodenie v jednom existuje ekvivalentné odvodenie v druhom systéme.
Skrátený E0L systém musí ale zachovať všetky možné vetvy odvodenia z pôvodného systému.

T.j. pre odvodenie zo symbolu x: Ak v systéme S na k krokov odvodíme zo symbolu X slovo α, tak v skrátenom systéme T odvodíme zo symbolu X slovo α na k-1 krokov.

Nutné je skrátiť všetky ďalšie odvodenia z axiómy, obsahujúce symbol X.

Pre prehľadnosť môžeme pri úprave používať pomocné značenie pre určenie počtu krokov odvodenia v pôvodnom systéme.

Nech v systéme S existujú pravidlá $X \to AB, A \to R_1, B \to R_2$, potom môžeme vytvoriť systém T obsahujúci pravidlá, simulujúce 2 kroky odvodenia naraz (označme si túto množinu P_2) $X \to_2 R_1R_2$ pre všetky trojice pravidiel s fixným symbolom X nasledovného tvaru: $X \to AB, A \to R_1, B \to R_2$. Zároveň bude obsahovať množinu pravidiel simujúcich jeden krok odvodenia (nie nutne neprázdnu), označme ju P_1, kde skracovanie nebolo nutné.

Pretože v niektorých prípadoch môže vzniknúť terminálne slovo v priebehu skráteného odvodenia musíme si dať pozor, v ktorých prípadoch môžeme vykonať skrátenie.

Postup pre skrátenie môžeme vykonať v prípade, že vieme zaručiť, že sa nebudú používať v jednom kroku odvodenia zároveň pravidlá z P_1 aj P_2.
Táto vlastnosť sa zachováva v relácii interpretácie, t.j interpretáciou pravidla z P_2 vzniká pravidlo z P_2, interpretáciou pravidla z P_1 vzniká pravidlo z P_1, teda ak sa v kroku odvodenia použili pravidlá jedného typu, zodpovedajúce pravidlá v interpretácii budú rovnako jedného typu.

Na pravidlách, ktoré menia terminálne slovo menia po svojom odvodení, skrátenie nie je možné vykonať, pretože by sa tým vynechali slová z jazyka.

Tu by bolo výhodné pracovať vždy so synchronizovanými E0L systémami, pretože ku každému E0L systému existuje synchronizovaný E0L systém, ktorý generuje rovnaký jazyk. Pri synchronizovaných E0L systémoch je terminálne slovo jediným terminálnym slovom v odvodení, a neexistuje odvodenie, ktoré by obsahovalo terminálne slová dve.

Vyvstáva tu otázka, či je E0L systém formovo ekvivalentný so zodpovedajúcim synchronizovaným E0L systémom.

Lema 2.2.1. Formová neekvivalencia systému a zodpovedajúceho synchronizovaného systému podľa vety 1
Majme systémy S a T, pričom T je ekvivalentný synchronizovaný systém k S.

Každému odvodeniu terminálneho slova v S zodpovedá odvodenie v T, pričom toto pokrýva aj odvodenia, kde sa terminálne slová prepisujú na iné terminálne slová, resp. terminálne slová na neterminálne a naspäť na terminálne.

Interpretáciou ľubovoľného symbolu v S vzniknú nové pravidlá, pričom je zachovaná terminálnosť symbolov.
Rovnakú interpretáciu vieme aplikovať na pravidlá v T, ktoré neobsahujú terminály (pozn.: pravidlá obsahujúce terminály sú špeciálne synchronizačné pravidlá).

Interpretáciou symbolu v T, ktorý je terminál zo systému S s pruhom, môžu vzniknúť nové javy.

Keďže pravidlá v S pre terminály môžu prepisovať:

- Terminál na neterminály,

- Terminál na terminály.

Uvedieme protipríklad:
Systém S je nesynchronizovaný. $S = (V, P, x, \Sigma)$
$V = \{S, A, B, a, b\}$
$\Sigma = \{a, b\}$
$P = \{S \to AB, A \to A|a, B \to b, b \to b, a \to a\}$

Odvodenie $S \Rightarrow AB \Rightarrow Ab \Rightarrow ab$ zjavne je nesynchronizované. V systéme T podľa konštrukcie existuje pravidlo pre $\bar{b}$, tvaru $\bar{b} \to \bar{b}$ a pravidlo $\bar{b} \to b$.

Vynechaním pravidla $\bar{b} \to b$ a symbolu b pri interpretácii T nenarušíme úplnosť T, ale dôsledkom toho vynecháme všetky slová z jazyka obsahujúce symbol b.

Takúto operáciu na forme S nevieme vykonať bez narušenia úplnosti.

Vo všeobecnosti vieme vynechať pri synchronizačnej konštrukcii z vety 1 interpretáciou všetky slová obsahujúce niektorý symbol, preto môžeme prehlásiť, že systém T skonštruovaný synchronizáciou S nie je formovo ekvivalentný s S.

Príklad 2.2.5. Nech
$S = (V, P, x, \Sigma)$
$V = \{S, A, B, C, R, a, b, c\}$
$\Sigma = \{a, b, c\}$
$P = \{S \to ABC, A \to a, B \to b, C \to c, C \to R,$
$a \to a, b \to b, c \to c, R \to a|b|c\}$

$T = (V, P', x, \Sigma)$
$V = \{S, A, B, C, R, a, b, c\}$
$\Sigma = \{a, b, c\}$
$P' = \{S \to ABC|ABR, A \to a, B \to b, C \to c,$
$a \to a, b \to b, c \to c, R \to a|b|c\}$

sú skúmané E0L formy, ktoré nie sú striktne formovo ekvivalentné.

Striktne formovo ekvivalentné nie sú, lebo existuje odvodenie v S, ($S \Rightarrow ABC \Rightarrow abR$), ktoré obsahuje slovo pozostávajúce z terminálov aj neterminálov. Takéto odvodenie terminálneho slova v T nenájdeme. Nakoľko v T máme odvodenia, ktoré obsahujú neterminály len po druhý krok odvodenia, nie je možné zostrojiť takú interpretáciu T, ktorá obsahuje odvodenie, ktoré obsahuje neterminály v treťom kroku. (Toto vyplýva z vlastnosti dfl-substitúcie: zachováva terminálnosť symbolu) Obidva systémy generujú jazyk L= {abc, aba, abb}.

Pretože nie sú žiadnym spôsobom synchronizované, t.j. nezáleží na dĺžke odvodenia, dá sa dokázať, že pre každú interpretáciu S existuje interpretácia T taká, že generujú rovnaký jazyk(a naopak).

Dôkazom tohoto dokážeme formovú ekvivalenciu S a T.

Dôkaz tvrdou silou:
Všimnime si odvodenia,ktoré neobsahujú neterminál R.
Kvôli dobrému označeniu symbolov vidíme, že tu je mechanizmus odvodenia

totožný. (v opačnom prípade by sme museli použiť operáciu premenovania symbolov tak, aby táto časť E0L formy bola totožná.)

Prejdime k odvodeniam, ktoré obsahujú neterminál R.
Pre formu S: $S \Rightarrow ABC \Rightarrow abR \Rightarrow aba, abb, abc$
Pre formu T: $S \Rightarrow ABR \Rightarrow aba, abb, abc$

Pri formovej ekvivalencii sa zapodievame všetkými interpretáciami jazykov, ktorých je nekonečne veľa, je jednoduchšie porovnať, čo sa môže stať s odvodeniami v S a T po interpretácii.

Všimnime si, že interpretáciou pravidla $a \rightarrow a$ na pravidlo $a \rightarrow \bar{a}$ vo forme S môžu vzniknúť odvodenia, ktoré zo slova abR, odvodia napríklad slovo $\bar{a}ba$, ktoré sa odvádza len v tejto vetve odvodenia. v prípade formy T sa odvodilo synchronizovane. V prípade formy S by sme odvodili slovo aba vôbec. Pretože máme v S a T vetvu, ktorá je identická v oboch formách, bude pravidlo výskyt pravidla $a \rightarrow \bar{a}$ v S podmieňovať jeho výskyt v T a naopak.

V odvodeniach neobsahujúcich neterminál R vieme postupovať pri S aj T rovnako: ak interpretáciou S pridáme pravidlo, vieme rovnako intepretovať T aby vzniklo totožné pravidlo a naopak.

Pri odvodeniach obsahujúcich R je situácia iná.
Odvodenie obsahujúce neterminál R nie je synchronizované.
Pokiaľ interpretujeme symbol A,B alebo C, musíme si klásť zásadnú otázku:

Ak vznikne nové slovo v systéme, ktorý interpretáciou S, musí vzniknúť nové slovo v systéme, ktorý je interpretáciou v T na rovnaký počet krokov(a naopak)?

Pre účely striktnej formovej ekvivalencie S a T je nutné, aby odvodenie v pôvodnom a novom systéme boli rovnaké, t.j. musíme zistiť interpretáciu, ktorou vznikne zo systému S systém T.
Pri formovej ekvivalencii môže byť odvodenie dlhšie aj kratšie.
Aby sme ukázali formovú ekvivalenciu, ukážeme, že je možné, aby sa pri odvodení aba nepoužilo v S pravidlo $a \rightarrow a$, ktoré v interpretácii existovať nebude.

Chceme, aby v odvodení $S \Rightarrow ABC \Rightarrow abR \Rightarrow aba, abb, abc$ nevystupovalo slovo abR ale $\bar{a}bR$. Môžeme interpretovať prepis symbolu A, to nám ale naruší vetvy ktoré sú totožne v S aj v T.

Jediné čo nám ostáva, je interpretovať pravú stranu pravidla o krok skôr v odvodení, $S \to ABC$ na $S \to \bar{A}BC$. Následne pridáme pravidlá $\bar{A} \to a_n, a_n \to a$, pričom a_n je nový terminál, čím dosiahneme, že bude interpretácia korektná, a zároveň sa nebude vyskytovať v žiadnom terminálnom slove.
Tým zastáva funkciu podobnú neterminálu.
Dôsledok postupu je oneskorenie odvodenia terminálneho symbolu o 1 krok.
Podobným postupom by sme vedeli oneskoriť odvodenie o ľubovoľný konečný počet krokov.
Dôležité je si všimnúť kedy sa odvodí posledný terminálny symbol v slove.

Definícia 2.2.3. Pseudoneterminálny symbol
Pseudoneterminálny symbol v E0L systéme $S = (V, P, x, \Sigma)$ je symbol z množiny Σ, ktorý sa nikdy nevyskytuje v terminálnom slove.

Dôsledok 3. Zavedením pseudoneterminálov prostredníctvom relácie interpretácie môžeme oneskoriť, alebo urýchliť odvodenie niektorých terminálnych symbolov, čím vieme zjednodušiť rozhodovanie o formovej ekvivalencii dvoch systémov prostredníctvom porovnávania odvodení.

Definícia 2.2.4. Operácie vykonané pri interpretácii
Interpretovaním podľa substitúcie (označme ju μ) vieme v princípe spraviť tri podinterpretácie, ktoré reprezentujú nasledovné operácie:

- Premenovanie symbolov tak, aby premenovanie neporušovalo disjunktnosť, a zároveň každý symbol sa prepíše na práve jeden
- Pridanie pravidiel a symbolov
- Odobranie pravidiel a/alebo symbolov

Potom každú interpretáciu S podľa $\mu(X)$ vieme rozdeliť na tri interpretácie podľa troch rôznych substitúcií, pričom zložením týchto troch interpretácií dostaneme pôvodnú. Pretože relácia interpretácie je tranzitívna, zložením týchto troch interpretácií dostaneme znovu interpretáciu S.

Nech μ_1 je substitúcia, kde nemusí nutne platiť $X \in \mu_1(X)$. Našim zámerom je, aby sme v tejto prvej (premenovávacej) podinterpretácii premenovali všetky symboly na konečný stav, aby pre platilo $\forall X \in V, X \in \mu_2(X)$

Preto prvá operácia je založená na jednoduchej interpretácii podľa μ_1, ktorá len premenúva symboly, ale nepridáva ani neuberá pravidlá. Zároveň je účelom, aby sa tu určil nový symbol axiómy, ak sme sa pri interpretácii podľa μ rozhodli ju premenovať.

Platí S=(V,P,x,Σ), T=(V,P',x",Σ'),
$T \lhd S(\mu_1)$,
$\forall X \in V \;\; \exists X' : \mu_1(X) = \{X'\}$
$|P| = |P'|$

Druhá podinterpretácia vyžaduje, aby v abecede existoval znak X z abecedy Σ', pre ktorý $\mu_2(X)$ je množina ktorá obsahuje aspoň dva znaky. Iný spôsob zvyšovania počtu pravidiel neexistuje.
Predpokladajme, že v tomto prípade platí $X \in \mu_2(X)$. (V opačnom prípade vieme zmeniť substitúcie μ_1 a μ_2 tak, aby vyhovovali).

Platí T=(V,P',x",Σ'),U=(V",P",x",Σ'')
$U \lhd T(\mu_2)$,
$|P'| \leq |P''|$

Tretia podinterpretácia sa riadi substitúciou ξ, pričom : $\xi(X) = X$ môže odobrať pravidlá v prípade, že sa nenaruší úplnosť E0L systému. T.j. pravidlá môžeme odoberať, ak pre každý symbol ostane aspoň jedno pravidlo na odvodzovanie. Symbol (označme ho Y) sa nám podarí odobrať, pokiaľ vieme odobrať všetky pravidlá, ktoré obsahujú symbol Y, bez narušenia úplnosti E0L systému.

Platí $U = (V', P', x', \Sigma'), \bar{S} = (\bar{V}, \bar{P}, \bar{x}, \bar{\Sigma})$
$\bar{S} \lhd U(\mu_2)$,
$|P''| \geq |\bar{P}|$

Potom $\bar{S} \lhd S(\mu)$.

Dôkaz: vyplýva z definície interpretácie E0L formy.

Zhrnieme si doterajšie zaujímavé výsledky v nasledujúcej vete:

Veta 4. Relácia $\lhd$ pre E0L systémy je rozhodnuteľná a tranzitívna.Systémy S a S' sú v relácii $S' \lhd S$ práve vtedy,
keď

$$\mathcal{G}(S) \subseteq \mathcal{G}(S')$$

Relácia $S' \lhd S$ implikuje inklúziu $\mathcal{L}(S) \subseteq \mathcal{L}(S')$, ale opačným smerom to vo všeobecnosti neplatí. Striktná formová ekvivalencia pre E0L formy je rozhodnuteľná.

Dôsledok 4. Na základe predošlej vety vieme prehlásiť, že ak sú dve formy S a T striktne formovo ekvivalentné, tak platí

$$\mathcal{G}(S) \subseteq \mathcal{G}(T)$$

a tiež

$$\mathcal{G}(T) \subseteq \mathcal{G}(S)$$

To znamená, že platí aj $S \lhd T$ a $T \lhd S$.

Na dôkaz striktnej formovej ekvivalencie nám stačí nájsť tieto dve interpretácie.

Je zaujímavé si všimnúť časté zámeny použitia výrazov E0L systém a E0L forma - ide o jednu a tú istú vec o ktorej hovoria, ale v druhom prípade prízvukujeme náš úmysel brať do úvahy aj interpretácie. Používame výraz formová ekvivalencia na zdôrazdnenie rozdielu od obyčajnej ekvivalencie dvoch E0L systémov.

Preto formová ekvivalencia pre formy F,G znamená $\mathcal{L}(F) = \mathcal{L}(G)$ a ekvivalencia F,G len $L(F) = L(G)$. Nakoľko rodina jazykov $\mathcal{L}(F)$ invariantná vzhľadom na premenovanie terminálnej abecedy F, dve E0L formy môžu byť formovo ekvivalentné bez toho aby boli ekvivalentné.

Nasledujúci príklad ukáže, že dve E0L formy môžu byť ekvivalentné aj keď nie sú formovo ekvivalentné.

Príklad 2.2.6. Nech F_1 a F_2 sú nasledovné E0L formy:

$F_1 = (\{S, a\}, \{S \to aS \mid a, a \to a\}, S, \{a\})$

$F_2 = (\{S, a\}, \{S \to SS \mid S \mid a, a \to a\}, S, \{a\})$

Analýzou pravidiel vieme jednoznačne povedať, že E0L forma F_1 generuje jazyk $L = \{a^n | n > 0\}$, rovnako ako E0L forma F_2, teda sú ekvivalentné. Líšia sa ale mechanizmom odvodzovania.

Interpretáciami E0L formy F_1 vygenerujeme rodinu jazykov generovaných regulárnymi gramatikami, pričom interpretáciami E0L formy F_2 generujeme jazyky triedy $\mathcal{L}(E0L)$.

Preto F_1 a F_2 nie su formovo ekvivalentné.

Všimneme si, že ľubovoľný regulárny jazyk v bezepsilonovom tvare a bez chain rules sa skladá z pravidiel tvaru $S \to aS', S \to a$ pre vybrané a z množiny terminálov, pričom terminálne symboly sa v regulárnych jazykoch neprepisujú, čo môžeme v E0L systéme zapísať práve pravidlom tvaru $a \to a$ pre nejaké a. Pozorovaním môžeme zistiť, že ak aplikujeme dfl substitúciu na takéto pravidlá, môže sa prepísať niektorý neterminál na terminál. Opačne to ale nie je možné.

Môžu vzniknúť pravidlá, ktoré prepisujú terminály na terminály, ale pridaním vhodných pravidiel, ktoré obsahujú na ľavej strane neterminál, spomedzi ktorých sa bude nedeterministicky vyberať pri odvádzaní terminálneho slova, dostaneme E0L systém, ktorý už pravidlá prepisujúce terminály neobsahuje.
Vo všeobecnosti teda nie je zložité si predstaviť konštrukciu, ktorá by z takéhoto E0L systému dokázala vyrobiť regulárnu gramatiku.
Pripomeňme si, že aby sme zachovali reláciu interpretácie na E0L systémoch, musí pre dfl substitúciu platiť, že zachováva axiómu a terminálnosť symbolov, nevymazáva symboly ani nezvyšuje ich počet.

Veta 5.
Ak F je E0L forma, tak trieda jazykov $\mathcal{L}(F)$ je uzavretá vzhľadom na dfl-substitúciu.

Dôkaz: Nech

$$F' = (V', P', x', \Sigma') \text{ je interpretácia F},$$

a

$$\tau : \Sigma' \to \Sigma'' \text{ je ľubovoľná dfl substitúcia.}$$

Musíme ukázať, že $\tau(L(F'))$ je v $\mathcal{L}(F)$.

Bez ujmy na všeobecnosti môžeme predpokladať, že $\Sigma'' \cap (V' - \Sigma') = \emptyset$. Rozšírime τ na V' definovaním $\tau(A) = A$ pre každé $A \in V' - \Sigma')$. Je zjavné, že toto bude tiež dfl-substitúcia.

Nový E0L systém, ktorým sa budeme zaoberať, vyzerá nasledovne:

$$F'' = (\tau(V'), \tau(P'), S', \tau(\Sigma'))$$

Podľa definície interpretácie je F" interpretáciou F' modulo τ.
Preto podľa vety 2 je F" je interpretáciou F, čo znamená, že L(F") je z triedy $\mathcal{L}(F)$. Na druhej strane je jednoducho vidno, že

$$L(F'') = \tau(L(F'))$$

□

V nasledujúcej časti by sme popísali niektoré výsledky o redukcii E0L foriem nasledovnom význame:
Majme E0L formu F. Chceme zostrojiť E0L formu F_1, ktorá je formovo ekvivalentná s F, a navyše spĺňa niektoré špeciálne kritériá, napríklad že je propagujúca.

Príklad 2.2.7. Zachovanie vlastnosti synchronizovanosti

Nech S=(V, P, x,Σ) je synchronizovaný E0L systém.
To znamená, že každé odvodenie terminálneho slova je zložené výhradne z neterminálov.
Z toho môžeme priamo dedukovať, že existuje zamietacie pravidlo. tvaru $x \to F$ kde x je ľubovoľný terminál a F je neterminál z ktorého nevieme odvodiť žiadne terminálne slovo(tzv. zamietací symbol).

V opačnom prípade by existovalo odvodenie z terminálneho slova do iného terminálneho slova, čo by znamenalo, že E0L systém nie je synchronizovaný.

V synchronizovanom E0L systéme by sme mohli rozdeliť pravidlá na 3 skupiny.

1. Pravidlá typu $N \to N^*$, kde N je jeden neterminál a N^* je ľubovoľný počet neterminálov.
2. Pravidlá typu $N \to T^+$, N je neterminál, T^+ je ľubovoľný kladný počet terminálov
3. Pravidlá typu $T \to F$, ktoré zamedzia odvodeniu terminálneho slova zo slova obsahujúceho terminálny symbol. F je zamietací symbol.

Všetky tri skupiny sú uzavreté vzhľadom na interpretáciu, nakoľko dfl substitúcia nemôže meniť terminálnosť symbolov.
Interpretáciou takéhoto systému dostaneme nutne synchronizovaný systém, nakoľko jediné odvodenie terminálneho slova môže byť prostredníctvom pravidiel druhej skupiny.

Inak ako synchronizovane sa terminálne slovo odvodiť nemôže. Ak by sa odvodil niektorý terminál skôr ako ostatné, v nasledujúcom kroku by sa zmenil na zamietací symbol F, čím by sa už nemohlo odvodiť terminálne slovo.

Preto je odvodenie nutne synchronizované. □

Definícia 2.2.5. Propagujúci 0L systém je taký 0L systém, ktorý pre žiadny symbol z abecedy neobsahuje pravidlo tvaru $a \to \epsilon$.

Je zjavné, že pokiaľ neexistuje pravidlo na mazanie symbolov v odvodení slova v pôvodnej forme, neexistuje dfl substitúcia, ktorá by z ľubovoľného pravidla vytvorila pravidlo na mazanie symbolov. Toto vyplýva z definície dfl substitúcie.

Definícia 2.2.6. Redukovaná forma
Rovnako ako pri bezkontextových gramatikách, aj tu sa nám zíde redukovaná forma.

Redukovaná forma E0L systému S (označme ju S_R) musí spĺňať 2 vlastnosti:

-Pre každý symbol existuje odvodenie z axiómy (v opačnom prípade môžeme zmazať všetky pravidlá, ktoré obsahujú symbol, ktorý sa nedá odvodiť z axiómy). Toto kritérium nazývame aj dostiahnuteľnosťou. (Pozn.: Všimnime si, že ide najmä o symboly na ľavej strane pravidla.)

-Pre každý symbol existuje terminálne slovo, ktoré z neho odvodíme. Výnimkou je špeciálny zamietací symbol.

Výskyt zamietacích symbolov je v odvodeniach E0L systémov často kľúčový.
Preto každý výskyt symbolov, z ktorých sa nedá odvodiť terminálne slovo, nahradíme jediným novým neterminálnym symbolom F.
F bude náš zamietací symbol. Nakoľko sa v odvodení terminálneho slova nevyskytuje, postačí nám na tento účel jeden.
Pre tento zamietací symbol je nutné pridať pravidlo $F \to F$.

Preskúmajme striktnú formovú ekvivalenciu:
Takýto systém S_R síce nemusí byť vo všeobecnosti nutne striktne formovo ekvivalentný so systémom S.
Vynechané pravidlá pre nedosiahnuteľné symboly môžu byť unikátneho tvaru.
Ako príklad uveďme nedosiahnuteľné pravidlo $a \to RRRRR$.
Za predpokladu, že v systéme S neexistuje iné pravidlo, ktoré má pravidlo s pravou stranou dĺžky 5, vynechané pravidlo už nevieme dostať pomocou relácie interpretácie.

Formovo ekvivalentný s pôvodným systémom S ale bude, pretože každému odvodeniu terminálneho slova vieme nájsť ekvivalentné v S_R.

Táto vlastnosť je zjavná, nakoľko každý symbol, ktorý sa nedá odvodiť z axiómy v odvodení nevystupuje,
teda ani pravidlá, v ktorých je na ľavej strane.

Príklad 2.2.8. Majme E0L formu S=(V, P, x,Σ)
x=S
V={S,a,b,A,B,R,N}
$P = \{S \to A|B, R \to N, a \to N, b \to b, B \to Bb, N \to N\}$
Máme tu zastúpený nedostupný symbol R, zamietací symbol N, a neterminál B z ktorého nevieme odvodiť terminálne slovo.

Symbol R a jeho pravidlo $R \to N$ môžeme priamo vymazať, pretože v odvodení z axiómy sa k R nedostaneme.
Symboly B a N, ktoré nefigurujú v odvodeniach terminálneho slova môžeme nahradiť symbolom F. K pravidlám pridáme $F \to F$
V pravidlách obsahujúcich N na pravej strane, symbol N nahradíme symbolom F.
Symbol b sa stal nedosiahnuteľným, teda ho odstránime spomedzi symbolov, a odstránime zodpovedajúce pravidlo.

Ekvivalentná redukovaná forma teda bude
S'=(V', P', x,Σ')
x=S
V'={S,a,F}
$\Sigma' = \{a\}$
$P' = \{S \to A|F, F \to F, a \to F, \}$

Veta 6. o konštrukcii formovo ekvivalentnej binárnej E0L formy

Pre každú E0L formu S môžeme skonštruovať formovo ekvivalentnú formy S_1 kde každé pravidlo z S_1 je jedného tvaru z nasledovných:

$$A \to \epsilon, A \to a, A \to B, A \to BC, a \to A$$

kde A,B,C sú neterminály, a a je terminál.
Navyše, ak S bol propagujúci, S_1 je propagujúci, a ak S bol synchronizovaný, S_1 je synchronizovaný.

Dôkaz: Ako prvé zredukujeme S na formu S_2, kde ľavé strany pravidiel budú najviac dĺžky 2. Označme si maxr(S) dĺžku najdlhšej pravej strany spomedzi všetkých pravidiel v S. To získame opakovaním konštrukcie z nasledovnej lemy:

Lema 2.2.2. Pre každú E0L formu S, pre ktorú platí $maxr(S) = m, m \geq 3$, existuje formovo ekvivalentná E0L forma $\bar{S}$, ktorá má $maxr(\bar{S}) = m - 1$.

Dôkaz: nech $\mathcal{S} = (V, P, x, \Sigma)$, $\bar{\mathcal{S}} = (\bar{V}, \bar{P}, x, \Sigma)$ sú E0L formy skonštruované nasledovne:

Nech p je pravidlo z P, potom množina $\bar{P}$ bude definovaná nasledovne:

$\bar{P} = \{\alpha \to [p], [p] \to x | p : \alpha \to x \in P$ a zároveň $|x| \leq 2\}$
$\cup \{\alpha \to [p][p'], [p] \to \beta, [p'] \to y | p : \alpha \to x \in P$
a zároveň $|x| \geq 3, x = \beta y, \beta \in V, y \in V^*\}$

Je dôležité poznamenať, že [p] a [p'] sú nové neterminály.
Takýmto rozdelením pravidla sa zachovajú odvodenia, akurát budú o jeden krok dlhšie.
Pravé strany pravidiel sú ale zaručene o 1 znak kratšie.

Pri dôkaze ich formovej ekvivalencie využijeme fakt, že každému kroku odvodenia v S zodpovedajú 2 kroky odvodenia v $\bar{S}$.

Pre každú interpretáciu S' ktorá vznikla z S vieme indentifikovať, ktoré pravidlá boli interpretované. Pretože pravidlu z S vieme jednoznačne priradiť dvojicu resp. trojicu pravidiel v $\bar{S}$, interpretácii pravidla z S vieme skonštruovať interpretáciu pravidiel v $\bar{S}$.

T.j. intepretácia pôvodného pravidla z S,

$\alpha \to \beta y$ na pravidlo v $\bar{S}$, $\alpha' \to \beta' y'$

zodpovedá interpretovaniu pravidiel z $\bar{S}$

$\alpha \to [p][p'], [p] \to \beta, [p'] \to y$ na pravidlá $\alpha' \to [p][p'], [p] \to \beta', [p'] \to y'$.

Opačným smerom je to jednoduchšie: postačí identifikovať, ktoré neterminály v $\bar{S}'$ sú interpetáciami [p] a [p'], nájdeme zodpovedajúce odvodenia dĺžky 2 ktoré následne spojíme do jedného pravidla z S'.

Každé trojice pravidiel v $\bar{S}'$, $\bar{S}'$ je interpretácia $\bar{S}$,
tvaru $\alpha' \to [p]'[p']', [p]' \to \beta', [p']' \to y'$,
vieme zredukovať na pravidlo tvaru $\alpha' \to \beta' y'$, ktoré je z S'.

Pre pravidlá, ktoré len predlžujú odvodenie, pretože ich pravá strana je dĺžky 2 alebo menej, to funguje obdobne. $\Box$

Opakovaním postupu z tejto lemy získame formu $S_2 = (V_2, P_2, x, \Sigma)$, pre ktorú $maxr(S_2) \leq 2$, ale niektoré pravidlá ešte stále nemusia byť požadovaného tvaru, napr. $a \to \epsilon, a \to BC, A \to B, a \in \Sigma, A, B, C \in V - \Sigma$.

Na odstránenie pravidiel týchto typov použijeme nasledovnú konštrukciu:

Zadefinujeme formu $\mathcal{S}_1 = (V_1, P_1, x, \Sigma)$.
P_1 je definované nasledovne:

$P_1 = \{\alpha \to [p], [p] \to [p'][p''], [p'] \to \beta, [p''] \to \gamma | p : \alpha \to \beta\gamma$, p je pravidlo z množiny $P_2, \alpha, \beta, \gamma$ sú z množiny $V_2\} \cup$
$\cup \, \{\alpha \to [p], [p] \to [p'], [p'] \to x, |p : \alpha \to x$, p je pravidlo z množiny $P_2, |x| \leq 1\}$

Nezabúdajme na to, že [p],[p'] a [p''] sú nové neterminály. (V prípade kolízie treba zaviesť nový symbol).

3 Zaradenie do Chomského hierarchie

Pretože väčšina E0L foriem prepisuje symboly paralelne, je zaujímavé skúmať špeciálne formy, ktoré sa podobajú na gramatiky v Chomského hierarchii, alebo generujú presne tú istú triedu jazykov.

3.1 Jazyky symbolov

Tieto jazyky by sa dali charakterizovať ako jazyky generované E0L systémami, ktoré neobsahujú pravidlo, ktoré má na pravej strane viac ako jeden symbol, a zároveň má jednopísmenovú axiómu.

Veta 7. Nech $\mathcal{S} = (V, P, x, \Sigma)$ je E0L forma
V={S,a} P=$S \to a, a \to a$
x=S

Potom $\mathcal{L}(Symb) = \mathcal{L}(S)$
Keďže $\mathcal{L}(Symb)$ obsahuje iba konečné, jednopísmenové jazyky, stačí slová vymenovať.
To dosiahneme interpretovaním pravidla $S \to a$, pričom substituovať budeme za symbol a všetky symboly, ktoré chceme, aby boli v jazyku.

Veta 8. Interpretovaním E0L formy z predchádajúcej vety $\mathcal{S}$ v substitúcii $\mu, \mu(S) = S, \mu(a) = \{a, b\}$ môže vzniknúť systém
$\mathcal{S}_2$ o ktorom môžeme zaručene povedať, že obsahuje aspoň dve terminálne slová.
Vybraním pravidiel $S \to a, a \to b, b \to b$ máme zaručenú existenciu dvoch slov.
Navyše každá interpretácia tejto formy minimálny počet rôznych slov zachováva, alebo zvyšuje .
Podobným spôsobom ako sme uviedli vyššie, môžeme vytvoriť množiny pravidiel, z ktorých odobratím ľubovoľného narušíme úplnosť E0L formy. Zároveň najdlhšie odvodenie,na ktorom sa slová neopakujú môže mať ľubovoľnú konečný dĺžku.

3.2 Konečné jazyky

Vyjadrenie konečných jazykov má špecifický tvar E0L formy.

Veta 9. Pre každý konečný jazyk je charakteristická maximálna dĺžka slova v jazyku. Problematické je, že každý jazyk môže mať vo všeobecnosti túto maximálnu dĺžku slova inú.

Pre ľubovoľnú E0L formu, ktorá generuje konečný jazyk, je táto maximálna dĺžka slova tiež konštanta.

Relácia interpretácie neumožňuje meniť dĺžku pravej strany pravidla.

Záver z týchto pozorovaní je jednoznačný:

Každá interpretácia E0L formy S generujúcej konečný jazyk, má najdlhšie slovo maximálne tak dlhé, ako pôvodná forma S.

Popísať ale vieme triedu konečných jazykov s dĺžkou slova práve x ako triedu interpretácií nasledovnej E0L formy:
$\mathcal{S} = (V, P, x, \Sigma)$
$V = \{S, a\}$
$P_1 = \{S \to a^x, a \to a\}$
x=S
$\Sigma = \{a\}$

Analogicky môžeme postupovať pre triedu konečných jazykov s dĺžkou slova najviac x: $\mathcal{S} = (V, P, x, \Sigma)$
$V = \{S, a\}$
$P_1 = \{S \to a^x, a \to a|\epsilon\}$
x=S
$\Sigma = \{a\}$

3.3 REG

Pokiaľ pri E0L forme zabezpečíme, aby počet neterminálov nevzrastal, a zároveň budú pravidlá na prepisovanie neterminálu, ktorých pravá strana obsahuje neterminál, tvaru $S \to aS$ (ľavo regulárne) alebo $S \to Sa$ (pravo regulárne).
Je známe, že ľavo regulárne gramatiky aj pravo regulárne gramatiky majú rovnakú popisnú silu, ktorá je ekvivalentná sile konečných automatov.

Príklad 3.3.1. Ľavé a pravé regulárne jazyky
$\mathcal{S}_1 = (V, P_1, x, \Sigma)$
$\mathcal{S}_2 = (V, P_2, x, \Sigma)$
$V = \{S, a\}$
$P_1 = \{S \to a|aS|\epsilon\, a \to a\}$
$P_2 = \{S \to a|Sa|\epsilon\ a \to a\}$
x=S
$\Sigma = \{a\}$

Je zjavné, že každá z interpretácií S_1 resp. S_1 bude regulárna gramatika. Pri regulárnych gramatikách nás zaujímajú nasledovné konštrukcie: Prepis neterminálu na terminál, prepis neterminálu na iný neterminál, vytvorenie cyklu z pravidiel generujúcich terminálne symboly (t.j. umožnenie pumpovania slova).

Prepis neterminálu na iný neterminál vznikne interpretáciou pravidla $S \to aS$ v dfl substitúcii, kde sa môžu za symbol S substituovať aspoň dva rôzne neterminály. Prepis neterminálu na terminál nám zaručujú interpretácie pravidla $S \to a$. Vytvorenie cyklu z pravidiel vznikne znovu substitúciou pre S s aspoň dvoma neterminálmi.

Pravidlá, ktoré sú prepisujú neterminál na neterminál môžeme zanedbať, nakoľko je možné upraviť každú regulárnu gramatiku do bezepsilonového tvaru s výnimkou prepisu axiómy na prázdny symbol (bez tohoto pravidla by neexistovalo prázdne slovo v jazyku).

Interpretácia S_1 resp S_2 na E0L formu, ktorá generuje jazyk ľubovoľnú ľavo resp pravo regulárnu gramatiku G=(N,T,P',x') bude vyzerať nasledovne:

Nech μ je dfl-substitúcia. Potom postupujme podľa nasledovných krokov:

- identifikujeme či ide o pravo alebo ľavo regulárnu gramatiku - bude sa interpretovať forma S_1 (ľavá) alebo S_2 (pravá).
- v gramatike identifikujeme terminálne (T) a neterminálne symboly (N)
- $\mu(S) = N, \mu(a) = T, \mu(x) = x'$
- podľa pravidiel z množiny P' gramatiky vyberieme pravidlá z $\mu(P)$.
- pre každý terminálny symbol bude v interpretácii S_i existovať jediné pravidlo tvaru $a \to a \quad \forall a \in T$.

Keďže potenciálna množina pravidiel podľa interpretácie μ obsahuje všetky možné pravidlá, pokrýva aj zložitejšie cykly, ktoré mohli byť zahrnuté v gramatike G.

3.4 CF

Bezkontextové jazyky, ako už skôr bolo spomenuté, sa dajú vyjadriť ako E0L jazyky bez prepisovania terminálnych symbolov.

Pre jednoduchšie porozumenie najbližej vety zopakujeme, čo je to dospelá abeceda 0L systému a čo je dospelý jazyk k 0L systému.

Definícia 3.4.1. Dospelý jazyk k 0L systému
Nech $\mathcal{S}$ = (V,P,x) je 0L systém. Potom dospelým jazykom k $\mathcal{S}$ nazývame $L_A(\mathcal{S}) = \{v \in L(\mathcal{S}) \mid ak(v \implies w)tak(v = w)\}$
Slovo $v \in L_A(\mathcal{S})$ nazývame stabilné slovo.

Definícia 3.4.2. Relácia $\Rrightarrow$ je definovaná nasledovne:
X $\Rrightarrow$ Y $\overset{def.}{\Leftrightarrow}$ $((x \implies y) \wedge (x \implies y')) \implies (y' = y)$,
Relácia $\overset{n}{\Rrightarrow}$ obdobne.
[Pozn.: Potom $L_A(\mathcal{S}) = \{w \in L(\mathcal{S}) \mid w \Rrightarrow w\}$.
Pre $a \in V$ budeme označovať dané slovo X_a ak platí $a \overset{+}{\implies} X_a \Rrightarrow X_a$]

Definícia 3.4.3. Dospelá abeceda
Nech $\mathcal{S}$ = (V,P,x) je 0L systém, potom dospelou abecedou nazývame množinu
$V_A(\mathcal{S}) = \{a|(a \in V) \wedge (\exists v \in L_A(S) : \#_a(v) > 0\}$
Označme abc(v) = $\{a \in V \mid \#_a(v) > 0)\}$ potom $V_A(S) = \bigcup_{v \in L_A} abc(v)$
Pre $a \in V$ Budeme označovať X_a, ak platí $a \overset{+}{\implies} X_a \Rrightarrow X_a$

Nasledovná veta z [7] popíše, ako vyzerá E0L systém, ktorá generuje bezkontextový jazyk.

Veta 10. Konštrukcia E0L systému generujúceho bezkontextový jazyk

$L \in \mathcal{L}(E0L)$ **je z triedy bezkontextových jazykov ak k nemu existuje E0L systém** $\mathcal{S}$ **taký, že** $L(\mathcal{S}) = L$ **a zároveň pre terminály** $a \in \Sigma$ **existujú iba pravidlá a** $\rightarrow$ **a**
K E0L jazyku $\mathcal{S} = (V, P, x, \Sigma)$ jednoznačne existuje 0L $\mathcal{S}' = (V', P', x')$ taký, že dospelý jazyk k $\mathcal{S}'$, a platí V=V'$\cup\{F\}, x = x'$.

Aby

$$L_A(\mathcal{S}') = L(\mathcal{S})$$

musíme zaručiť, aby

$$V_A(\mathcal{S}') = \Sigma$$

To zabezpečíme pridaním pravidiel do P'.

P'=$P \cup \{a \to F \mid a \in V - \Sigma\} \cup F \to FF$

pričom F je nový symbol.
Týmito pravidlami zaručíme, že neterminálne symboly nebudú v dospelej abecede.

Sporom: Vezmime neterminálny symbol s, ktorý je v dospelej abecede L_A. Teda existuje stabilné slovo ktoré tento symbol obsahuje. Určite ale existuje pravidlo pre s, ktoré vedie do symbolu F, ktorý sa neobmedzene množí. To znamená, že zo stabilného slova odvodíme nové slovo, pretože symbol F nie je v dospelej abecede. Toto je spor.
$\square$

Dokázali sme, že na základe podmienok kladených na E0L systém S generujúci jazyk L, vieme z neho zostrojiť E0L systém S', ktorého dospelý jazyk je presne jazyk L. Z dokázanej vety 2 v [7] vyplýva, že jazyk L musí byť určite bezkontextový.

Napriek tomu, že pre ľubovoľný bezkontextový jazyk vieme vytvoriť E0L, ktorý ho generuje, nie sú triedou E0L jazykov.

Tento fakt bol dôsledkom všeobecnejšieho tvrdenia v [8]. Tvrdenie sa zakladá na fakte, že ak jazyk
$L = \{a^i b^i c^j d^j | i, j \geq 1\}$ je v triede generovanej E0L formou S, potom trieda $\mathcal{L}(S)$ zaručenie obsahuje aj nejaký jazyk mimo triedy bezkontextových jazykov.

3.5 E0L

V tejto časti vznikla zásadná otázka: ako vyzerá E0L forma, ktorá generuje ľubovoľný E0L systém.

Definícia 3.5.1. E0L forma $\mathcal{S}$ je úplná, ak $\mathcal{L}(\mathcal{S}) = \mathcal{L}(E0L)$

Veta 11. E0L forma

$$\mathcal{S} = (\{S, a\}, \{S \to S, S \to SS, S \to a, a \to S\}, S, \{a\})$$

je úplná.

Dôkaz:

Nech L je ľubovoľný jazyk generovaný E0L systémom. Na základe teorémy 2.1 v [4], je zrejmé, že každý takýto jazyk L sa dá generovať propagujúcim E0L systémom T. Transformáciou podľa vety 6 dostaneme formovo ekvivalentný systém T_1, ktorého všetky pravidlá sú niektorého z typov

$$A \to a, A \to B, A \to BC, a \to A$$

Všimnime si, že táto transformácia zachováva ekvivalenciu systémov, teda

$$L = L(T) = L(T_1)$$

Zjavne ale T_1 je interpretáciou $\mathcal{S}$. Keďže L bol ľubovoľný jazyk generovaný E0L systémom, forma $\mathcal{S}$ je úplná.

□

4 Vizualizácia

4.1 Graf E0L formy

Pri skúmaní E0L foriem, ich vlastností, a najmä (striktnej) formovej ekvivalencie vyvstala otázka, ako to porovnávať efektívne.

Zaviedieme preto grafovú reprezentáciu E0L formy s niekoľkými operáciami ktoré zodpovedajú operáciám ktoré vykonáva interpretácia.

Definícia 4.1.1. Graf E0L formy
Nech $\mathcal{S} = (V, P, x, \Sigma)$ je E0L forma.

Potom zostrojíme graf G=(V_1, V_2,H,Z), kde V_1, V_2 sú vrcholy grafu, H sú hrany grafu, podľa nasledovných pravidiel
(Poznámka: Číslo za hranou špecifikuje, či ide o hranu z vrcholu z množiny V_1 do vrcholu množiny V_2 (x,y,0) alebo naopak (x,y,1)):

1. Nech x=Z je základný vrchol a platí $X \in V_1$ (bude znázornený s dvojitým obrysom)

2. Nech každý symbol N $\in$ V - Σ je znázornený kružnicou, označený N a platí N $\in V_1$

3. Nech každý symbol T $\in \Sigma$ je vrchol znázornený štvorcom

4. Pre každé pravidlo p z množiny P pridáme vrchol reprezentujúci jeho pravú stranu do množiny V_2, tieto vrcholy znázorňujeme elipsou, pre rozlíšenie od samotných symbolov v prípade pravidiel, ktorých pravá strana obsahuje len jeden symbol.

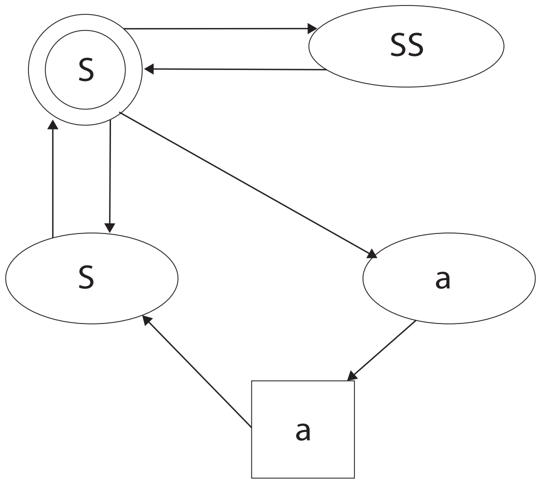
5. Nech platí $\forall p \in P, (p : (x \to y)) \wedge (x, y, 0) \in H$(pravidlo podmieňujúce vznik hrán v grafe reprezentujúcich pravidlá E0L formy)

6. Nech platí $\forall y \in V_2 y = a_1 ... a_n \wedge (y, a_i, 1) \in H$ (pravidlo podmieňujúce vznik hrán z pravej strany pravidla na jednotlivé symboly)

7. Nech každá hrana (x,y,a) z H je zobrazená ako šípka z vrcholu x do vrcholu y.

Príklad 4.1.1. Ako najlepší príklad na jednoduchý graf E0L formy nám poslúži úplná E0L forma

$$\mathcal{S} = (\{S, a\}, \{S \to S, S \to SS, S \to a, a \to S\}, S, \{a\}) \quad .$$

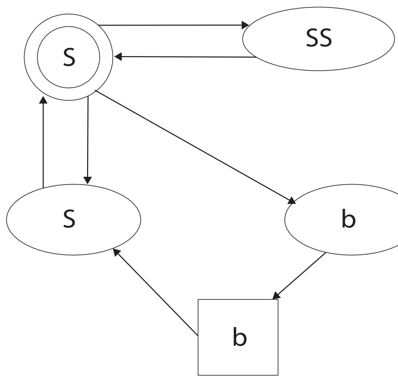


K nemu vezmime interpretáciu μ, definovanú nasledovne:

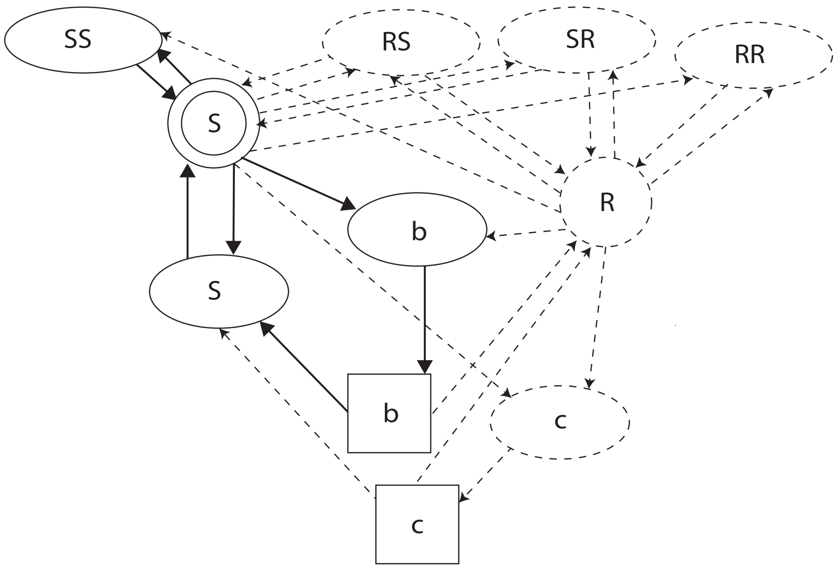
$$\mu(S) = \{S, R\}$$
$$\mu(a) = \{b, c\}$$

Nakoľko sme ukázali, že môžeme rozdeliť interpretáciu E0L systému na tri pod, postupujme podľa týchto krokov.

1. **Premenovanie**. Keďže pre všetky symboly platí, že $x \in \mu(x)$, táto fáza obsahuje identickú substitúciu $\mu_1(S) = \{S\}$, kvôli jednoduchosti, pre a si musíme vybrať: $\mu_1(a) = \{b\}$ alebo $\mu_1(a) = \{c\}$ My volíme prvú možnosť. Rovnako môžeme prehlásiť, že sme S vybrali za axiómu. V opačnom prípade by sme museli práve v tejto operácii určiť novú axiómu prostredníctvom inej substitúcie μ_1' kde $\mu_1'(S) = \{R\}$



2. **Pridanie pravidiel a symbolov**. V tejto operácii sa len ujasňuje, aké pravidlá a symboly môže mať konečná interpretácia. V tomto prípade $\mu_2(S) = \mu(S), \mu_2(b) = \{b, c\}$, nakoľko sme už pri premenovaní premenovali symbol a na b. V tejto fáze vzniká veľa nových pravidiel a hrán, preto znázornenie vyzerá pomerne neprehľadne. Prerušované hrany, elipsy a štvorce sú nové hrany a vrcholy pridané v procese interpretácie.



3. **Vynechávanie pravidiel a symbolov**. Tu je dfl substitúcia identita, t.j. $\mu_3(x) = \{x\}$ pre všetky symboly. Hlavný krok tejto operácie je zvolenie pravidiel a symbolov ktoré budú vynechané, ak je to možné vzhľadom na úplnosť systému. Úplnosť systému dostiahneme tak, že z každého štvorca resp kruhu bude existovať vychádzajúca hrana.

   Vynechávanie prvkov grafu prebieha nasledovne:

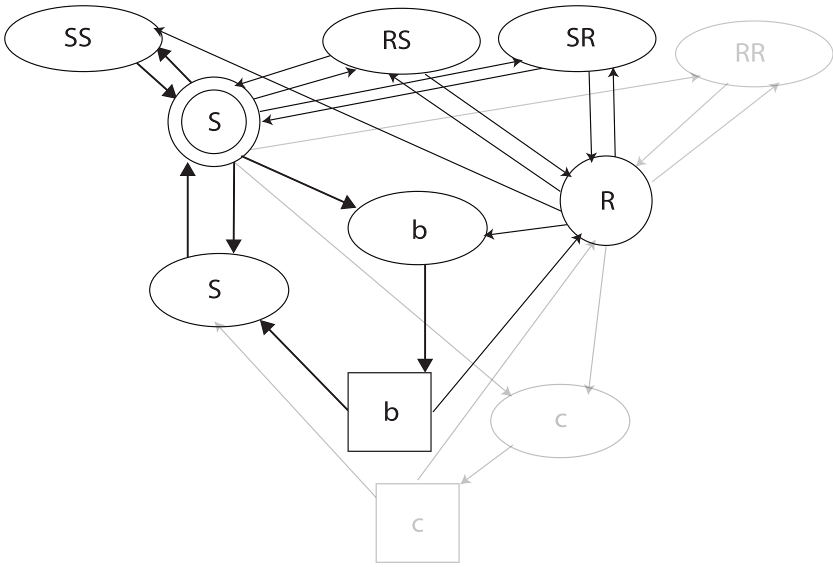
   Vynechaniu pravidla $x \to y$ zodpovedá odstránenie hrany (x,y,0). Pokial do elipsy y nevstupuje žiadna iná hrana, je potrebné odstrániť všetky vystupujúce hrany z y.

   Ilustrujme tento postup na pravidlách $R \to RR$ a $S \to RR$

   Odstránili sme hranu (R,RR,0) a hranu (S,RR,0), čím zostala elipsa RR bez vstupujúcich hrán. Tým nám vzniká povinnosť odstrániť hrany (RR,R,1). Na obrázku znázornené šedou.

   Odstránenie symbolu, bez ujmy na všeobecnosti to bude c, musíme odstrániť štvorec s popisom c a všetky súvisiace elementy- hrany ktoré z neho vystupujú alebo doňho vstupujú, a elipsy z ktorých vychádzajú hrany vstupujúce do c.
   Následne musíme odstrániť všetky hrany ktoré vchádzali do elipsy, ktorú sme odstránili, a skontrolovať, či je E0L forma reprezentovaná grafom úplná, teda z každého vrchola reprezentujúceho symbol vychádza aspoň jedna hrana.

   Pre c: odstránime štvorec c, hrany (c, S,0),(c,R,0) a hranu (c,c,1), čím vzniká nutnosť odstrániť elipsu c.
   Odstránením elipsy c odstránime hrany (R,c,0) a (S,c,0), čím sa dostaneme do konzistentného stavu, čo znamená, že každá elipsa má vstupnú a výstupnú hranu, a každý vrchol reprezentujúci symbol má výstupnú hranu.



Definícia 4.1.2. Krok odvodenia v E0L systéme ku ktorému prislúcha graf G Majme graf G, ktorý je v súlade s našou konštrukciou, potom jeden krok odvodenia prebieha nasledovne, pomocou kameňov s číslami, ktoré vieme posúvať po grafe:

- pred krokom odvodenia sme v slove $x = x_1 x_2 ... x_n$
- nájdeme zodpovedajúce vrcholy z V_1 grafu G, do každého umiestnime kameň s číslom k, kde k je miesto výskytu v slove x.
- pre každý kameň s číslom k zvolíme jednu z hrán vychádzajúcich z vrchola, v ktorom ležia, presunieme kameň do koncového vrcholu danej hrany. Tu nastáva nedeterminizmus.
- teraz, keď každý kameň je vo vrchole znázornenom elipsou, nahradíme každý kameň s číslom k novými, ktoré majú dve číslelné označenia. Prvé je číslo kameňa ktorý sme nahradili, druhé je poradové číslo symbolu v danom vrchole, ktorému kameň zodpovedá.
  T.j. kameň vo vrchole $x_1 x_2$ s číslom k nahradíme kameňmi s označením (k,1) a (k,2).

- Po nahradení novými kameňmi presunieme kamene do vrcholov, ktorým zodpovedajú. Kameň (k,1) putuje do symbolu x_1, kameň (k,2) putuje do symbolu x_2.
- Po týchto operáciách sa každý kameň nachádza vo vrchole, ktorý reprezentuje symbol, ktorý sa odvodil po jednom kroku odvodenia.

Pokiaľ si pre na kameň zapíšeme vrchol, z ktorého sme ho zobrali, usporiadame kamene podľa lexikografického usporiadania ich označenia, a prečítame označenia vrcholov na kameňoch v poradí od najmenšieho po najväčší, vieme presne zistiť aké slovo sa odvodilo na 1 krok odvodenia.

Ak sme ochotní mať kamene s m-rozmerným označením, vieme podobne simulovať m-1 krokov odvodenia.

4.2 Graf odvodenia v E0L forme

Definícia 4.2.1. Graf odvodenia Nech $\mathcal{S} = (V, P, x, \Sigma)$ je E0L forma,
a $W = w_0 p_1 w_1 p_2 ... w_{n-1} p_n w_n$je postupnosť množín použitých pravidiel na prepis.
Potom zostrojíme graf R=(V_1, V_2,H,Z), kde V_1, V_2 sú vrcholy grafu, H sú orientované hrany grafu, podľa nasledovných pravidiel :

1. Nech x=Z je základný vrchol a platí $X \in V_1$ (bude znázornený s dvojitým obrysom)
2. Nech každý neterminálny symbol N z odvodenia W je znázornený kružnicou, označený N a platí N $\in V_1$
3. Nech každý terminálny symbol T z odvodenia W je vrchol znázornený štvorcom
4. Nech pre všetky pravidlá p použité v odvodení W platí: $(p : (x \to y)) \wedge y \in V_2$ (vrcholy pre pravé strany pravidiel)
5. Nech pre všetky pravidlá p použité v odvodení W platí:$(p : (x \to y)) \wedge (x, y, 0) \in H$(hrany zo symbolov do pravých strán)
6. Nech $\forall y \in V_2 y = a_1 ... a_n \wedge (y, a_i, 1) \in H$(Hrany z pravých strán do symbolov)

Príklad 4.2.1. Ukážka grafu odvodenia.

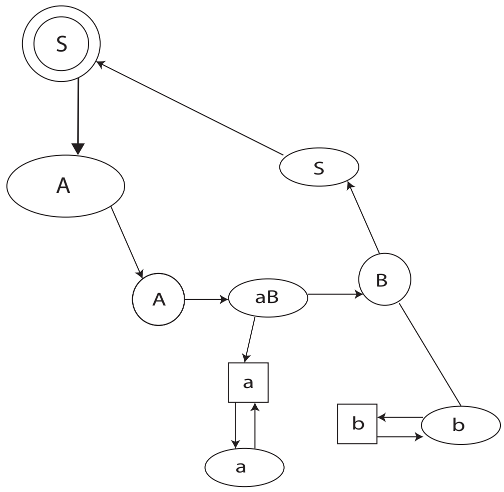
Ukážeme odvodenie, kde nastane cyklus v grafe, čo je jeden z kľúčových javov pri generovaní nekonečných jazykov.

Nech $\mathcal{S} = (V, P, x, \Sigma)$ je E0L forma,
V=$\{S, A, a, B, b, C, C1, C2, c\}$
P=$\{S \to A|abcC, A \to aB, B \to S|b, c \to c, b \to b, a \to a, C \to C1, C1 \to cC2, C2 \to cC|c\}$
x=S
$\Sigma = \{a, b, c\}$

V tejto E0L forme je zaujímavé si všimnúť cyklus z pravidiel.

Je to $S \to A \to B \to S$.
Ako interpretáciu tejto formy môžeme vytvoriť z pravidiel cyklus dvojnásobnej dĺžky.
Postačí zaviesť nové symboly, ktoré si pamätajú, že ešte musí prebehnúť kópia cyklu z pôvodných pravidiel.
$S \to A \to B \to S' \to A' \to B' \to S$,
Tento jav ide vytvoriť pomocou interpretácie $\mu(x) = \{x, x'\}$ pre všetky neterminály.
Výberom vhodných pravidiel, medzi ktorými nie je $B \to b$, zaručíme interpretáciou beh cyklu v požaadovanej dvojnásobnej dĺžke.
Je zjavné, že takouto interpretáciou sme stratili istú generatívnu silu. Podobným spôsobom by sa dal prinútiť beh cyklu opakovať aspoň k-násobne, kde k je konečné číslo.
Spätne interpretovať formu, aby cyklus pôvodnej dĺžky existoval, vo všeobecnosti nevieme, pokiaľ prostredníctvom interpretácie dlhší cyklus nevynecháme, a následne nevygenerujeme kratší z iných pravidiel. Nehovoriac, že odvodenie do prvého neterminálu v cykle môže vygenerovať ďalšie symboly, čo nám naruší ekvivalenciu.
Toto vyplýva priamo z definície DFL substitúcie: nie je možné, aby množiny substitúcie dvoch rôznych symbolov mali neprázdny prienik.
Aby sa graf odvodenia príliš nepodobal pôvodnému grafu E0L formy, pridali sme druhý cyklus do zdrojového E0L systému.

Záznam odvodenia prostredníctvom pôvodného cyklu:



Pre naše účely je odvodenie nezávislé od poradia resp. počtu použitia cyklov, nakoľko pri skúmaní odvodení s cyklami sa s neurčitým počtom opakovaní cyklov stretávame bežne.

Záver

V teórii jazykov zaoberajúcich sa L systémami sa vyskytlo mnoho rôznych variánt a rozšírení, ktoré majú menšie alebo väčšie odlišnosti či ciele, ktoré chcú dosiahnuť. Nezávisle od toho sa dajú skúmať aj spôsobom použitým v tejto práci, skúmaním foriem a ich interpretácií.

V tejto práci sme sa snažili rozpracovať do podrobností skúmanie E0L foriem, ale predpokladáme, že náš prístup by mohol byť prínosný aj v iných oblastiach L-systémov.
V skúmaní formovej ekvivalencie E0L je stále mnoho otvorených otázok, napríklad ako efektívne porovnávať dve rôzne formy vzhľadom na reláciu interpretácie.
Prostredníctvom grafového znázornenia sa ale dá pozerať prehľadnejšie aj na zložité systémy, skoro nezávisle od rozšírení, ale hlavne môže otvárať prístup tradičným metódam porovnávania grafov.

Efektívnejšie zobrazovanie takýchto grafov v priestore môže byť náročnou prekážkou pri nadrozmerných systémoch pre človeka, ale s použitím moderných počítačových zobrazovacích techník je práca značne uľahčená.

Riešenie otázok formovej a striktnej formovej ekvivalencie prostredníctvom grafového zobrazenia môže byť pre človeka tiež značne jednoduchšie, aj napriek mnohým možnostiam výberu pravidiel pri interpretácii.

Rozšíriť graf na systémy s interakciou môže byť zaujímavá prekážka ktorá by inšpiruje vznik nového, lepšieho výpočtového modelu, aplikovateľného v reálnom svete počítačových výpočtov.

Literatúra

[1] Rozenberg G. - Salomaa, A. 1992 Lindenmayer systems: impacts on theoretical computer science computer graphics and developmental biology,Berlin: Springer-Verlag, 1992 ISBN 0387553207

[2] Vaško, M. 2010 Rozšírenia Lindenmayerových systémov s interakciou, Bratislava: Univerzita Komenského, 2010.

[3] Rozenberg G. - Salomaa, A. 1980. The mathematical theory of L systems. New York: Academic Press, 1980. ISBN 0125971400

[4] Hopcroft, J.E -Ullmann, J.D. 1969. Formal languages and their relation to automata. Reading: Addison-Wesley Pub. Co., 1969. ISBN 0201029839

[5] ROZENBERG, G. - SALOMAA, A. 1986. The Book of L. Berlin: Springer-Verlag, 1986. ISBN 0387160221

[6] FEKIAČ, J. 2012. Dospelé jazyky k Lindenmayerovym systémom. Bratislava, Univerzita Komenského 2012

[7] J. Albert, H. Maurer, 1978. The class of context-free languages is not an EOL family. Information Processing Lett., 6 (No. 6) (1978), pp. 190–195