VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMEDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

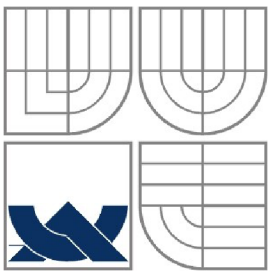
UMĚLECKÉ POUŽITÍ FRAKTÁLOVÝCH FILTRŮ (BEZKONTEXTOVÝCH GRAMATIK)

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

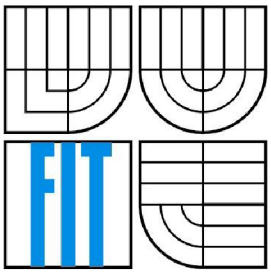
AUTOR PRÁCE
AUTHOR

PETER MEDERLY

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMEDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

UMĚLECKÉ POUŽITÍ FRAKTÁLOVÝCH FILTRŮ (BEZKONTEXTOVÝCH GRAMATIK)

ARTISTIC USE OF FRACTAL FILTERS (CONTEXT-FREE GRAMMARS)

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE
AUTHOR

PETER MEDERLY

VEDOUCÍ PRÁCE
SUPERVISOR

ING. MICHAL FAPŠO

BRNO 2008

Zadání bakalářské práce/6427/2007/xmeder00

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2007/2008

Zadání bakalářské práce

Řešitel: **Mederly Peter**
Obor: Informační technologie
Téma: **Umělecké použití fraktálových filtrů (bezkontextových gramatik)**
Kategorie: Počítačová grafika

Pokyny:

1. Zoznámte sa s grafickými bezkontextovými gramatikami (http://www.contextfreeart.org/) a s jednoduchou kontextovou nadstavbou (http://www.fit.vutbr.cz/~ifapso/cs_cfdg.html)
2. Navrhnite aplikáciu na automatickú konverziu jednoduchého obrázku (napr. jeden ťah štetca, machuľa a pod.) do bezkontextovej gramatiky
3. Implementácia predchádzajúceho bodu
4. Takto vytvorené gramatiky aplikujte na jednoduché obrázky, prípadne fotografie

Literatura:

- http://www.contextfreeart.org/mediawiki/index.php/CFDG_HOWTO
- Podľa pokynov vedúceho

Při obhajobě semestrální části projektu je požadováno:

- 1, 2

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Fapšo Michal, Ing.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2007
Datum odevzdání: 14. května 2008

L.S.

doc. Dr. Ing. Pavel Zemčík
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Peter Mederly**
Id studenta: 78868
Bytem: Panská dolina 66, 949 01 Nitra
Narozen: 04. 07. 1987, Trenčín
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   bakalářská práce

Název VŠKP: Umělecké použití fraktálových filtrů (bezkontextových gramatik)
Vedoucí/školitel VŠKP: Fapšo Michal, Ing.
Ústav: Ústav počítačové grafiky a multimédií
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
   - ☒ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

....

Nabyvatel Autor

Abstrakt

Táto bakalárska práca sa zaoberá využitím bezkontextových gramatík v grafických aplikáciách za účelom generovania obrázkov a hlavne generovaním bezkontextovej návrhovej gramatiky. Obsahuje úvod do formálnych gramatík, popis aplikácií pracujúcich s bezkontextovými návrhovými gramatikami a návrh, implementáciu a použitie generátoru bezkontextovej návrhovej gramatiky gg_cs_cfdg.

Klúčové slova

bezkontextová návrhová gramatika, generátor gramatiky, cfdg, cs_cfdg, gg_cs_cfdg

Abstract

This bachelor's thesis is dealing with the use of context-free grammars in graphic aplications for generating images and mainly with generating context-free design grammars. It contains introduction to formal grammars, description of aplications working with context-free design grammars and concept, implementation and usage of context-free grammar generator gg_cs_cfdg.

Keywords

context-free design grammar, grammar generator, cfdg, cs_cfdg, gg_cs_cfdg

Citace

Mederly Peter: Umělecké použití fraktálových filtrů (bezkontextových gramatik), bakalářská práce, Brno, FIT VUT v Brně, 2008

Umělecké použití fraktálových filtrů (bezkontextových gramatik)

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Michala Fapša, uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Peter Mederly
15. května 2008

Poděkování

Rád by som poďakoval svojmu vedúcemu, Ing. Michalovi Fapšovi, za priateľský prístup a podporu pri tvorení bakalárskej práce.

Obsah

1 Úvod

Počítačová grafika sa v posledných rokoch rýchlo rozvíjala a našla si pevné miesto v umeleckom svete. Či už ide o 2D alebo 3D grafiku niet pochýb, že sa stala neoddeliteľnou súčasťou nášho života. Formálne gramatiky už tiež nie sú v informatike žiadnou novinkou, ich umelecké využitie je však pomerne nová a málo preskúmaná oblasť.

Táto práca podáva komplexný pohľad na generovanie bezkontextovej návrhovej gramatiky. Ako prvé je potrebné spoznať teoretické zázemie formálnych gramatík, vysvetlenie je podporené konkrétnymi príkladmi. Zvládnutie týchto základov je kľúčové pre pochopenie ďalších kapitol. V 50tych rokoch zaviedol Noam Chomsky rozdelenie gramatík do štyroch základných typov. Z nich je tvarom pravidiel zaujímavá bezkontextová gramatika, na ktorej je postavená grafická aplikácia cfdg. Spojili sa v nej jednoduchosť a elegancia bezkontextových gramatík za účelom generovania obrázkov. Je založená na sade nedeterministických pravidiel (bezkontextových návrhových gramatikách), ktoré však musíme sami napísať. Zvláda vykresliť milióny tvarov a vytvárať až 100megapixelov veľké obrázky. Vznikla kontextová nádstavba cs_cfdg, ktorá umožňuje tieto ručne napísané bezkontextové návrhové gramatiky „inteligentne“ aplikovať.

Toto všetko viedlo k návrhu a implementácii generátoru bezkontextovej gramatiky gg_cs_cfdg. Je to modul pre cs_cfdg, ktorý dokáže napodobiť ľubovoľnú štruktúru. A to tak, že analyzuje naskenovaný obrázok (napr. uhlíkom nakreslená čiara) a prevedie ho do sady pravidiel bezkontextovej návrhovej gramatiky. S použitím všetkých troch aplikácií tak dokážeme v počítači „kresliť“ uhlíkom.

2 Použitie gramatík v grafických aplikáciách

V tejto kapitole najprv naznačíme teoretické pozadie formálnych gramatík, ich rozdelenie do tried, a bližšie rozoberieme bezkontextové gramatiky. Práve ich umelecké použitie nás zaujíma. Na bezkontextovej gramatike je založená grafická aplikácia cfdg, ktorá bude následne podrobnejšie popísaná aj s jej kontextovou nádstavbou cs_cfdg.

2.1 Gramatiky

Na objasnenie pojmu gramatika musíme najprv uviesť a vysvetliť niekoľko základných pojmov ako sú abeceda, slovo a jazyk. Chomského hierarchia rozdeluje formálne gramatiky do tried, z ktorých ďalej obsiahlejsie rozpíšeme len bezkontextové gramatiky.

2.1.1 Základné pojmy

Gramatiku nemôžme definovať bez vysvetlenia základných pojmov. Uvedieme ich definície, tak ako sú uvedené v [1] a [2], a jednoduché príklady.

Abeceda je ľubovoľná neprázdna konečná množina. Prvky abecedy nazývame symboly. Existujú aj nekonečné abecedy, tými sa však nebudeme v tomto texte zaoberať. Ako príklad môžme uviesť dvojprvkovú binárnu abecedu reprezentovanú množinou $\{0, 1\}$ alebo abecedy programovacích jazykov.

Reťazcom nad abecedou Σ rozumieme každú konečnú postupnosť symbolov z abecedy Σ. Prázdnu postupnosť budeme nazývať prázdny reťazec a označovať ho ε. Nad reťazcami môžme robiť operácie konkatenácia, reverzné reťazce, podreťazce, dĺžka reťazca a ďalšie. Ako príklad poslúžia niektoré reťazce nad binárnou abecedou $\Sigma = \{0, 1\}$ z predchádzajúceho príkladu:

$$\varepsilon \qquad 0 \qquad 1 \qquad 00 \qquad 01 \qquad 10 \qquad 11 \ldots$$

Poradie symbolov v reťazci je významné, reťazce *01* a *10* sú rôzne.

Definícia **jazyka** vychádza z predpokladu, že vieme čo je abeceda a reťazec, čiže vieme vybrať množinu reťazcov nad danou abecedou. Nech Σ je abeceda. Označme symbolom Σ^* množinu všetkých reťazcov nad abecedou Σ vrátane prázdneho reťazca, symbolom Σ^+ množinu všetkých reťazcov nad Σ okrem prázdneho reťazca, čiže $\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$. Množinu L, pre ktorú platí $L \subseteq \Sigma^*$ (prípadne $L \subseteq \Sigma^+$, ak $\varepsilon \notin L$), nazývame jazykom L nad abecedou Σ. Jazykom teda môže byť

ľubovoľná podmnožina reťazcov nad danou abecedou, vrátane $\emptyset$ a $\{\varepsilon\}$, čo sú jazyky nad každou abecedou. Jazyk $\emptyset$ neobsahuje žiadny reťazec a jazyk $\{\varepsilon\}$ obsahuje jeden reťazec (aj keď je to reťazec prázdny), takže platí $\emptyset \neq \{\varepsilon\}$. Poznáme jazyky konečné aj nekonečné. Nad jazykmi možme definovať dve skupiny operácií, ktoré vyplývajú z ich vlastností. Jazyk sme definovali ako množinu a môžme teda využívať obvyklé množinové operácie ako sú zjednotenie, prienik, rozdiel a doplnok. Druhú skupinu operácií umožňuje fakt, že prvky jazykov sú reťazce. Môžme definovať operácie súčinu a iterácie jazykov. Ako jednoduchý príklad môžme uviesť programovacie jazyky, sú nekonečné, vždy môžme napísať nový program. Program nie je nič iné ako množina reťazcov programovacieho jazyka nad jeho abecedou.

Gramatiky sú jeden z dvoch základných typov konečnej reprezentácie konečných, ale aj nekonečných jazykov (druhým typom sú automaty). Gramatika pomocou množiny pravidiel určuje, ktoré reťazce (postupnosti symbolov z abecedy) z jazyka sú platné, avšak neurčuje ich sémantiku. Teraz uvedieme formálnu definíciu gramatiky. Gramatika G je štvorica $G=(N, \Sigma, P, S)$, kde

- N je konečná množina (abeceda) nonterminálnych symbolov
- Σ je konečná množina (abeceda) terminálnych symbolov, $N \cap \Sigma = \emptyset$
- P je konečná podmnožina kartézskeho súčinu $(N \cup \Sigma)^* N (N \cup \Sigma)^* \times (N \cup \Sigma)^*$
- $\Sigma \in N$ je východzí (taktiež začiatočný) symbol gramatiky

Množina P je konečná množina pravidiel pomocou ktorých môžme daný jazyk generovať. Prvok (α, β) množiny P nazývame prepisovacím pravidlom (skrátene pravidlom) a budeme ho zapisovať v tvare $\alpha \rightarrow \beta$. Pravidlo určuje možnú substitúciu reťazca β za reťazec α, ktorý sa vyskytuje ako podreťazec generovaného reťazca. Reťazec α resp. β nazývame ľavou resp. pravou stranou prepisovacieho pravidla. Postupnou aplikáciou pravidiel na nonterminálne symboly generujeme reťazce tvorené iba terminálnymi symbolmi. Tieto reťazce potom predstavujú vety gramatikou definovaného jazyka.

Skôr ako sa dostaneme k príkladu, musíme uviesť čo je to **derivácia**. Uvažujme gramatiku G z predošlého odstavca. Nech λ a μ sú reťazce z $(N \cup \Sigma)^*$. Medzi reťazcami λ a μ platí relácia $\Rightarrow_G$, nazývaná priama derivácia, ak môžme reťazce λ a μ vyjadriť v tvare $\lambda = \gamma \alpha \delta$, $\mu = \gamma \beta \delta$, kde γ a δ sú ľubovoľné reťazce z $(N \cup \Sigma)^*$ a $\alpha \rightarrow \beta$ je nejaké prepisovacie pravidlo z P. Ak platí medzi reťazcami λ a μ relácia priamej derivácie, potom píšeme $\alpha \Rightarrow_G \mu$ a hovoríme, že reťazec μ je možné priamo generovať z reťazca λ v gramatike G.

Uvažujme gramatiku $G=(N, \Sigma, P, S)$ kde:

$$N=\{S, A, B, C, D\}$$

$$\Sigma=\{e, f, g, h\}$$

$$P=\{S \Rightarrow AhBC, C \Rightarrow \varepsilon, BC \Rightarrow D, AhA \Rightarrow eCg, D \Rightarrow Af\}$$

Uvažujme derivácie:

$$S \Rightarrow AhB\,\mathbf{C} \Rightarrow AhB$$

$$S \Rightarrow Ah\,\mathbf{BC} \Rightarrow Ah\,\mathbf{D} \Rightarrow \mathbf{AhA}\,f \Rightarrow e\,\mathbf{C}\,gf \Rightarrow egf$$

S pomocou pravidiel z množiny P sme rozgenerovali začiatočný symbol gramatiky. Pravidlá sme však použili v rôznom poradí (podreťazec, ktorý predstavuje ľavú stranu pravidla je hrubo vyznačený). V prvom prípade nemôžme ďalej derivovať, pretože v reťazci AhB neexistuje podreťazec, ktorý by bol rovný ľavej strane niektorého pravidla. V druhom prípade sme však postupnou deriváciou získali reťazec z jazyka $L(G)$.

2.1.2 Rozdelenie gramatík

V roku 1956 zaviedol Noam Chomsky definície štyroch základných typov gramatík. Gramatiky sú rozdelené podľa tvaru prepisovacích pravidiel a sú rozpoznateľné na rôznych teoretických výpočetných modeloch.

V tabulke 1 je toto rozdelenie uvedené, spolu s jazykmi ktoré tieto gramatiky generujú, a teoretický výpočetný model schopný tento jazyk interpretovať. Je dokázateľné, že trieda jazykov typu 3 je vlastnou podtriedou jazyka typu 2. Podobne ako sa pre triedy jazykov typu 2 resp. typu 1 dá dokázať, že sú vlastnou podtriedov jazyka typu 1 resp. typu 0. To však nie je predmetom tejto práce a nebudeme sa tým ďalej zaoberať.

Gramatiky		**Jazyky**	**Výpočetný model**
typ 0	neobmedzené	rekurzívne spočítateľné	Turingov stroj
typ 1	kontextové	kontextové	lineárne ohraničený Turingov stroj
typ 2	bezkontextové	bezkontextové	zásobníkový automat
typ 3	regulárne	regulárne	konečný automat

Tabulka 1: Rozdelenie gramatík podľa Chomského hierarchie

2.1.3 Bezkontextové gramatiky

Bezkontextové gramatiky definujú bezkontextové jazyky a veľmi často aj syntax programovacích jazykov. Sú spájané zo zásobníkovými automatmi, ktoré sú teoretickým modelom syntaktických analyzátorov.

Už sme naznačili, že gramatiky sa rozdeľujú podľa tvaru prepisovacích pravidiel. Budeme vychádzať z gramatiky G typu 0, ktorú už sme vyššie definovali ako štvoricu $G=(N, \Sigma, P, S)$. Gramatika je bezkontextová ak každé pravidlo v množine P je v tvare $X \rightarrow \gamma$, kde X je jeden nonterminálny symbol a γ je postupnosť terminálnych a nonterminálnych symbolov (tá môže byť aj prázdna).

2.2 CFDG

Autorom základnej myšlienky je Chris Coyne, ktorý vytvoril jednoduchý programovací jazyk, pre generovanie obrázkov na základe nedeterministických pravidiel. Cfdg je skratka pre Context Free Design Grammar, vo voľnom preklade je to bezkontextová návrhová gramatika. Cfdg je plne grafické prostredie pre upravovanie, vykresľovanie a skúmanie cfdg gramatík.

V tejto kapitole rozoberieme základné vlastnosti cfdg, stručne popíšeme syntax a použitie demonštrujeme na príklade. V závere zhrnieme dôležité ..

2.2.1 Základné vlastnosti

Aplikácia beží na všetkých platformách a je voľne šíritelná. Umožňuje sledovať priebeh generovania obrázku, vygenerované obrázky ukladá vo formáte PNG alebo SVG. V dokumentácii sa uvádza, že je schopná vykresliť veľmi veľké obrázky (až do 100 megapixelov) a zvládne vykresliť milióny tvarov. Pracuje v 2D afinnom zobrazení a umožňuje aj zložitejšie definície ako napr. rekurziu (využívané fraktálmi).

Cfdg berie ako vstup súbor pravidiel bezkontextovej gramatiky. Každé pravidlo určuje, ako sa nonterminálny symbol môže rozgenerovať do postupnosti iných terminálnych aj nonterminálnych symbolov. Novovzniknuté nonterminálne symboly môžu byť popísané ďalším pravidlom a byť znova rozgenerované. Terminálne symboly nemajú žiadne pravidlá, hneď sa vykreslia. V cfdg existuje len niekoľko terminálnych symbolov. Konkrétne sú to základné vykresľovacie tvary ako štvorec, kruh a trojuholník. Ak súbor pravidiel dokážeme zo začiatočného symbolu rozložiť na postupnosť terminálnych symbolov (postupne rozderivovať), môžme vykresliť obrázok zo základných tvarov. Vykreslené obrázky sú legálnymi vetami nad jazykom definovaným práve súborom pravidiel z gramatiky.

2.2.2 Použitie

Teraz vysvetlíme z čoho pozostáva súbor pravidiel cfdg. Jediným povinným príkazom je *startshape*, určenie začiatočného symbolu, z ktorého sa začne derivovať. Musí existovať aspoň jeden začiatočný symbol. Ak je ich viac, do úvahy sa berie len prvý v poradí. Farba pozadia výstupného obrázku sa nastavuje direktívou *background*, ak nie je uvedená, použije sa biele pozadie. Príkaz *tile* umožňuje opakovanie vytvoreného vzoru podľa deliacej mriežky, špecifikovanej týmto príkazom. Vkladanie ďalších cfdg súborov zabezpečuje direktíva *include*. Počet týchto príkazov je neobmedzený, vkladaný súbor môže obsahovať ďalšie z týchto príkazov. Táto vlastnosť dovoľuje deliť gramatiky na menšie časti a tie volať z rôznych programov. Ako si neskôr ukážeme, na tomto princípe sú prepojené aplikácie cs_cfdg a gg_cs_cfdg.

Najdôležitejšou časťou sú pravidlá tvarov, tzv. s*hapes rules*. Tie určujú ako má cfdg tieto tvary vykresliť, neexistuje žiadne obmedzenie na ich počet. Tvar je zložený z primitívnych tvarov ako je štvorec, kruh a obdĺžnik (terminálne symboly z bezkontextovej návrhovej gramatiky cfdg) a ďalších tvarov (neprimitívnych, neterminálnych symbolov). Pre každý tvar musí existovať aspoň jedno pravidlo, inak aplikácia cfdg skončí s chybovým hlásením. Pravidlo môže byť nasledované váhou pravidla (kladné reálne číslo). Ak váha nie je uvedená, ráta sa s implicitnou hodnotou 1. V zložených zátvorkách sú uvedené náhrady tvarov, tzv. s*hape replacements.* V podstate ide o postupnosť terminálnych a nonterminálnych symbolov, ktoré sa získajú rozložením nonterminálneho symbolu (v tomto prípade je to názov pravidla). Použitie týchto príkazov a pravidiel je znázornené v ilustrácii 1.

```
startshape example

background {b 1}

//include ./example2.cfdg

rule example {
        SQUARE   {}
        CIRCLE   {s 0.9 b 0.3}
        TRIANGLE {s 0.8 y -0.3 b 0.8}
        SQUARE   {s 1.2 z -1 b 0.6}
}
```

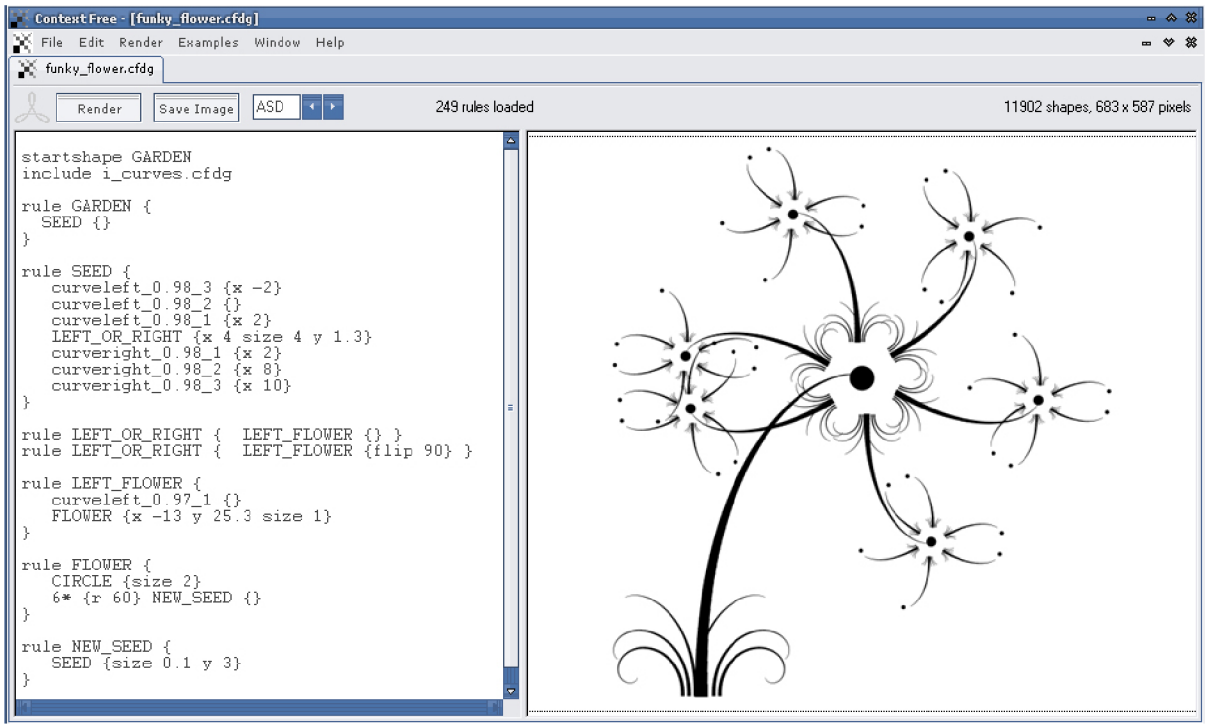
Ilustrácia 1: Jednoduchá cfdg gramatika a jej grafický výstup

Shapes adjustments (úpravy tvarov) prenášajú grafické vlastnosti na základné tvary cfdg. Rozdeľujeme ich na geometriu a farbu. Geometria tvarov je definovaná 2D afinným zobrazením. V tomto zobrazení rovnobežným priamkam zodpovedajú opäť rovnobežné priamky, tie však nemusia byť rovnobežné s pôvodnými priamkami. To umožňuje využívať posuvy, rotácie, zmeny mierky,

zkosenie alebo transformácie zložené z týchto operácií. Farbu určujú zložky farebného modelu HSB (tón, sýtosť, jas farby) a opacita. Viac informácií k načatej tématike o úprave tvarov nájdete v [3].

2.2.3 Zhrnutie

Cfdg je šikovný nástroj pre tvorbu obrázkov z jednoduchých bezkontextových návrhových gramatík. Stále prebieha intenzívny vývoj, aktuálna je verzia 2.1, ktorá má príjemné a jednoduché ovládanie. K dispozícii je dostatok ukážkových príkladov, jeden z nich je na obrázku 2. Bola vydaná prvá publikácia s názvom Community of Variation, prináša približne 60 ukážok jedných z najlepších bezkontextových návrhových gramatík za posledné dva roky. Tiež by som rád upozornil na plne funkčnú php variáciu cfdg od Yevgena Korshykova (http://korsh.com/cfdg/).



Ilustrácia 2: Grafické uživateľské rozhranie cfdg pre Windows

2.3 CS_CFDG

Je to kontextová nadstavba aplikácie cfgd. Dokáže inteligentne aplikovať už vytvorenú bezkontextovú návrhovu gramatiku na línie v obrázkoch. Princíp rozpoznania línií a využitie cs_cfdg hneď popíšeme.

2.3.1 Princíp

Veľmi zjednodušene môžme povedať, že cs_cfdg vytvorí z 8-bit bitmapy súbor pravidiel, ktoré popisujú líniu v danom obrázku. Lepšie však bude postup vysvetliť.

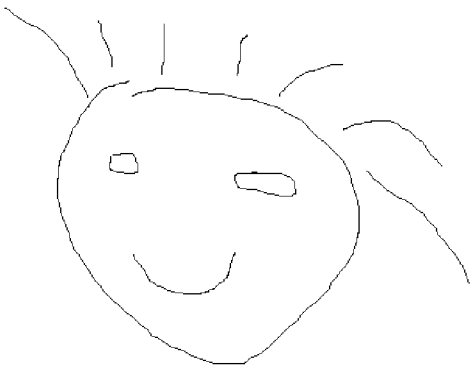
Prechádza sa celým obrázkom a každý nebiely pixel je predmetom skúmania. Pre všetky takéto pixely sa prehľadáva okolie určené dvoma „sústrednými kružnicami“ s rôznym polomerom (pracuje sa s rastrom, ide skôr o vzdialenosť pixelov). Polomer oboch kružníc sa dá nastaviť vstupnými parametrami. Pre každý nebiely pixel vo vymedzenej oblasti sa vypočíta uhol, ktorý zviera s horizontálnou osou prechádzajúcou skúmaným pixelom. Vypočítané uhly sa vkladajú do kontajneru. Pre jeden pixel sa v kontajnery môže nachádzať niekoľko rôznych uhlov. Vstupným parametrom je možné ovplyvniť ich počet. Tie sa postupne z kontajneru eliminujú, pričom sa nechávajú iba uhly s najväčším rozdielom.

Po analýze vstupného obrázku sa informácie z kontajneru zapíšu pomocou pravidiel návrhovej gramatiky cfdg do súboru. Vznikne tak súbor pravidiel, ktoré určujú z akého miesta a pod akým uholm sa má vykresliť nejaký ťah. Pravidlá jednotlivých ťahov sa vložia pomocou direktívy cfdg *include*.

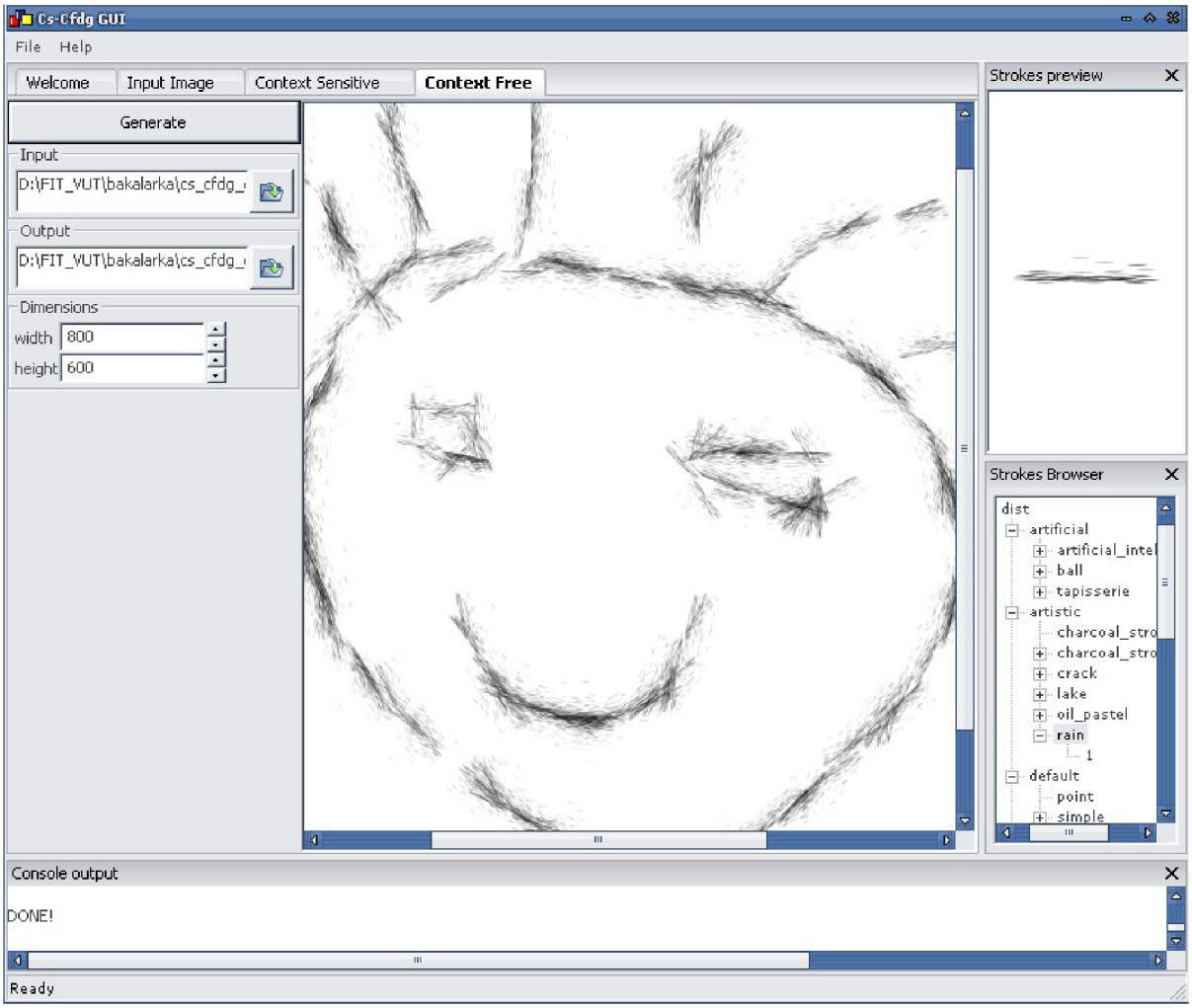
2.3.2 Využitie

Je zrejmé, že cs_cfdg vytvára nové a zaujímavé možnosti použitia bezkontextových gramatík v grafických aplikáciách. Vznikli podmienky pre vytvorenie generátoru bezkontextovej návrhovej gramatiky gg_cs_cfdg, ktorý sa opiera o súbor vytvorený aplikáciou cs_cfdg. Ilustrácia 3 zachytáva príklad vstupného obrázku a časť bezkontextovej návrhovej gramatiky vygenerovaný programom cs_cfdg. V grafickou uživateľskom rozhraní môžme, po vložení gramatiky popisujúcej ťah, rovno vidieť výsledok a aj výstup konzoly (islustrácia 4).

```
startshape strokes
include ./coloured_pencil.cfdg
rule strokes {
        stroke{x 6 y -11 r 341.565063 alpha 0.000000}
        stroke{x 7 y -11 r 326.309937 alpha 0.000000}
        stroke{x 8 y -12 r 326.309937 alpha 0.000000}
        stroke{x 9 y -12 r 161.565048 alpha 0.000000}
        stroke{x 10 y -13 r 153.434952 alpha 0.000000}
        stroke{x 11 y -14 r 143.130096 alpha 0.000000}
        stroke{x 12 y -15 r 143.130096 alpha 0.000000}
        stroke{x 13 y -16 r 135.000000 alpha 0.000000}
        stroke{x 14 y -17 r 135.000000 alpha 0.000000}
  .
  .
```



Ilustrácia 3: Časť gramatiky vygenerovanej zo vstupného obrázku pomocou cs_cfdg



Ilustrácia 4: Grafické uživateľské rozhranie cs_cfdg pre Windows

3 Generátor bezkontextovej gramatiky

Generátor bezkontextovej gramatiky gg_cs_cfdg (v ďalšom texte už len generátor) vznikol ako modul pre cs_cfdg, ktorý generuje jednoduchú bezkontextovú gramatiku, tak ako ju definoval Chris Coyne pre aplikáciu cfdg. V tejto kapitole podrobne rozoberieme návrh generátoru. Naznačíme niektoré implementačné problémy a budeme sa venovať použitelnosti aplikácie.

3.1 Návrh

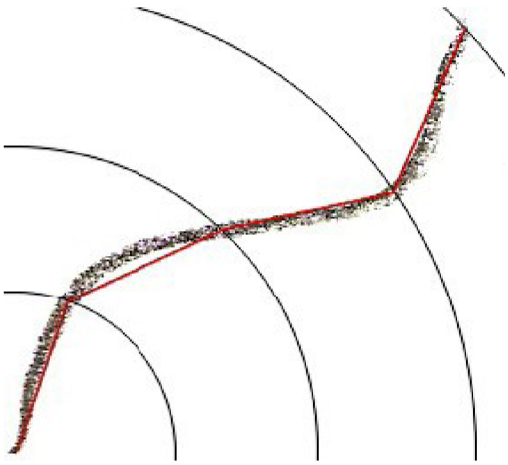
Budeme sa venovať návrhu generátoru bezkontextovej gramatiky. Uvedieme čo berieme ako vstup programu a vysvetlíme význam vstupných parametrov.

3.1.1 Vstupné dáta

Ako vstup programu bude slúžiť naskenovaný obrázok, napr. ťahu uhlíkom, vodovou farbou alebo sprejom. Tento obrázok program zanalyzuje a vygeneruje príslušnú bezkontextovú gramatiku, ktorá popisuje štruktúru obrázku a bude ju vediet napodobiť. Načrtneme dva varianty.

Prvou možnosťou je brať do úvahy celý naskenovaný obrázok, detekovať začiatok a koniec ťahu, ktorý následne aproximujeme splajnom. Splajn rozdeliť na segmenty pomocou sústredných kružníc a tie ďalej analyzovať (ilustrácia 5). Každý segment by sme popísali vhodným štatistickým modelom, ktorý by určoval rozloženie farebných pixelov, a na jeho základe generovať bezkontextovú gramatiku popisujúcu daný ťah. Tento spôsob je však náročný a vygenerovaná gramatika by bola veľmi rozsiahla.

V poradí druhou možnosťou je z naskenovaného obrázku vybrať vhodnú oblasť a tú použiť ako vstupné dáta (ilustrácia 6). Odpadajú tak ťažkosti s detekciou a aproximáciou ťahu,



Ilustrácia 5: Aproximácia ťahu splajnom

Ilustrácia 6: Výber vhodnej oblasti

predpokladajú sa však aspoň základné znalosti uživateľa v oblasti práce s grafickými editormi. Pri tomto spôsobe je možné z jedného ťahu generovať viac rôznych gramatík, stačí vybrať inú oblasť a použiť ju ako nový vstup.

Druhá varianta je z celkového hladiska elegantnejšia a efektívnejšia, preto som sa ju rozhodol implementovať. Ako formát vstupného obrázku som zvolil 24-bitovú bitmapu, je to rozšírený formát a dobre sa s ním pracuje.

3.1.2 Vstupné parametre

Jednou z priorít pri návrhu programu bola čo najväčšia možnosť variability výsledkov, čiže generovania rôznych bezkontextových gramatík. Modifikáciou vstupných parametrov získame nové výstupy aj bez zmeny vstupného obrázku. Táto vlastnosť zaručuje širšie použitie a viac možností pri generovaní gramatík. Uvedieme jednotlivé parametre s ich krátkym popisom, niektorým sa budeme bližšie venovať v kapitole 3.2.

`-i <cesta k/názov súboru>` vstupný obrázok, zdroj pre generátor

`-o <cesta k/názov súboru>` výstupný súbor, vygenerovaná gramtika

`-bgIC <0/1>` farba pozadia vstupného obrázku, zatiaľ podporovaná len biela a čierna

`-bgIT <0.0-1.0>` tolerancia saturácie a jasu pozadia vstupného obrázku

`-bgOC <0/1>` nastavenie farby pozadia výstupného obrázku (formou pravidla), zatiaľ podporovaná len biela a čierna

`-w <0-50>` šírka ťahu, ktorý bude gramatikou generovaný

`-ch <0.0-1.0>` pravdepodobnosť prechodu na nasledujúci segment (pravdepodobnosť pravidla)

`-l <1-20>` veľkosť skúmanej oblasti pri generovaní segmentu (v pixeloch)

`-pxT <1-255>` tolerancia intenzity farby susedného pixelu v skúmanej oblasti

3.2 Implementácia

Program je písaný v jazyku C a pre svoje spustenie nepotrebuje žiadne prídavné knižnice. Zdrojové kódy gg_cs_cfdg sú rozdelené do troch súborov. Hlavičkový súbor definuje makrá, štruktúry a prototypy funkcií. Funkcie sú implementované v samostatnom súbore a sú volané z hlavného programu.

Kapitolu rozdelíme podľa priebehu generovania bezkontextovej gramatiky. Najprv vysvetlíme spracovanie vstupných parametrov, potom uloženie grafickej informácie zo vstupného obrázku, a nakoniec samotné generovanie gramatiky. V texte budú uvedené niektoré názvy štruktúr a

funkcií z programu bez bližšieho popisu. Tieto špecifiká, ako argumenty funkcií a zložky štruktúr, sú uvedené v programovej dokumentácií, ktorú nájdete v prílohách.

3.2.1 Spracovanie vstupných parametrov

Vstupné parametre z príkazového riadku spracováva funkcia *getParameters()*. Ak nie je niektorý z nepovinných parametrov zadaný, použije sa prednastavená hodnota. Inak sa overí, či hodnoty spadajú do požadovaného intervalu. Pre veľký počet parametrov je vhodné uložiť ich do štruktúry. Funkcia pre spracovanie parametrov preto vráti buď štruktúru *S_INPUT_PARAM* naplnenú hodnotami, alebo ak nastane chyba, vypíše chybové hlásenie a ukončí program.

Vstupné parametre sú okrem samotného generovania použité aj vo funkcii *createNfo()*, ktorá k vygenerovanej gramatike vytvorí súbor zo základnými informáciami o gramatike.

3.2.2 Načítanie vstupného obrázku

V kapitole 3.1.1 sme naznačili, že ako vstupný obrázok akceptujeme len 24-bit bitmapu. V rámci prenositelnosti medzi platformami sme museli definovať dve štruktúry *BITMAPFILEHEADER* a *BITMAPINFOHEADER*. Obe štruktúry sú prevzaté z [5] a slúžia na uloženie informácií o bitmapách.

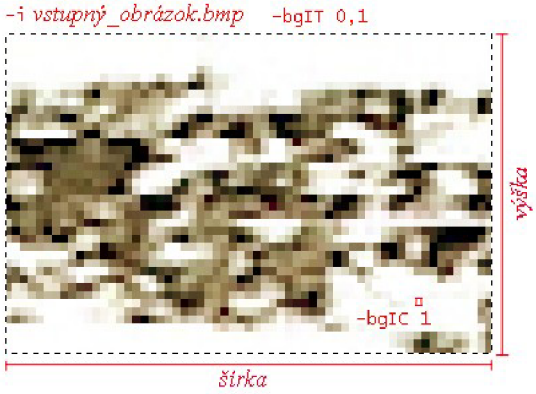
Venujme sa teraz formátu 24-bit bitmapy a s tým spojenými problémami. V každej bitmape sú na začiatku uvedené hlavičky popisujúce všetky dôležité parametre obrázku. Na rozdiel od 8-bit bitmapy, kde je za hlavičkami ešte uvedená paleta farieb, nasleduje informácia o farbách v jednotlivých pixeloch. Ako prvý sa udáva pixel v ľavom dolnom rohu a postupuje sa po riadkoch až do pravého horného pixelu. Používa sa farebný model RGB. Pre každú farbu je vyhradených 8 bitov, čo umožňuje definovať viac ako 16miliónov farieb. Farebné zložky v pixeloch sú uvedené v obrátenom poradí ako sa bežne udávajú a to modrá, zelená a červená.

Na spracovanie bitmáp nebola použitá žiadna externá knižnica, všetky tieto vlastnosti sú zohľadnené vo funkcii *load_bitmap()*. Táto funkcia si na základe informácií z hlavičiek bitmapy alokuje dostatok pamäti a vráti dvojrozmerné pole *S_RGB* hodnôt, s ktorým sa ďalej pracuje. Štruktúra *S_RGB* obsahuje informácie o farbe v konkrétnom pixely.

3.2.3 Generovanie bezkontextovej gramatiky

Teraz rozpíšeme princíp funkcie *generateCFG()*, ktorá vytvorí hlavný výstup programu. Funkcia analyzuje obrázok uložený v buffry pomocou funkcie *load_bitmap()*. Využijeme funkcie na výpočet intenzity pixelu *getIntensity()* a pre prevod farby z farebného modelu RGB na hodnoty HSB *rgb2hsb()*. Už sme vysvetlili ako fungujú pravidlá v bezkontextových gramatikách a v aplikácii cfdg. V ďalšom texte budeme pravidlom rozumieť práve pravidlo bezkontextovej gramatiky z cfdg.

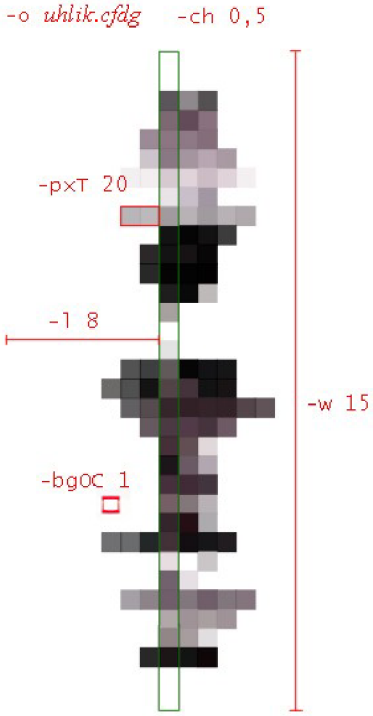
Najprv sa do výstupného súboru vygeneruje pravidlo, ktoré nastaví farbu pozadia na základe vstupného parametru *-bgOC*. Nasledujú pravidlá ťahov pomenované rovnako ako pravidlá vygenerované programom cs_cfdg. Každý segment je definovaný jedným z týchto pravidiel. Z kapitoly 2.3 o cs_cfdg už vieme, že ťahy budú nanášané po nejakej línii. Segmenty určujú, aká štruktúra sa po tejto línii bude vykresľovať. Ako sa tvorí vykresľovaná štruktúra popíšeme v ďalších odstavcoch.



Ilustrácia 7: Vstupný obrázok

Generuje sa toľko segmentov, aká je šírka obrázku. Každý segment má počet podsegmentov rovný výške vstupného obrázku a v každom podsegmente je zachytené okolie zadané parametrom *-l*. V obrazovej prílohe 3 je časť vygenerovanej gramatiky, na ktorej je vidieť ako tieto pravidlá a ich štruktúra vyzerá.

Ako sa teda analyzuje vstupný obrázok a ako sa z neho generujú pravidlá? Vysvetlíme to pomocou ilustrácií 7 a 8, sú na nich znázornené aj vstupné parametre s prednastavenými hodnotami. Základnou myšlienkou je vygenerovať jeden segment z každého stĺpca vstupného obrázku. Ako sme už povedali vyššie, segment sa skladá z podsegmentov. Podsegment budeme brať ako jeden riadok v stĺpci, tie sa začínajú generovať zo stredu stĺpca. Je to kvôli zarovnaniu ťahu na stred línie popísanej vygenerovanou gramatikou z cs_cfdg. V cs_cfdg tak môžme využívať detekciu smeru línie, bez toho aby sa ťahy rozhádzali na obe strany.



Ilustrácia 8: Vygenerovaný ťah

Podiel výšky vstupného obrázku a šírky ťahu zadanej vstupným parametrom *-w* je koeficient, ktorý udáva v akom pomere musia byť základné tvary cfdg (terminálne symboly SQUARE, CIRCLE a TRIANGLE) zmenšené resp. zväčšené. To znamená že v jednom stĺpci generovanej gramatiky je rovnaký počet riadkov ako vo vstupnom obrázku, upravených tak aby vyhovovali šírke ťahu. Na riadku sa negenerujú základné tvary, iba ak farba v danom pixely súhlasí s farbou uvedenou parametrom *-bgIC*. Prípadne ak jeho saturácia a jas sú tejto farbe podobné. Interval do ktorého hodnoty môžu patriť je určený pomocou parametru *-bgIT*.

Chceme zachytiť štruktúru vstupného obrázku a preto sa program „pozrie“ na okolité stĺpce až do vzdialenosti zo vstupného parametru *-l*. Postupuje sa do oboch strán, referenčná hodnota

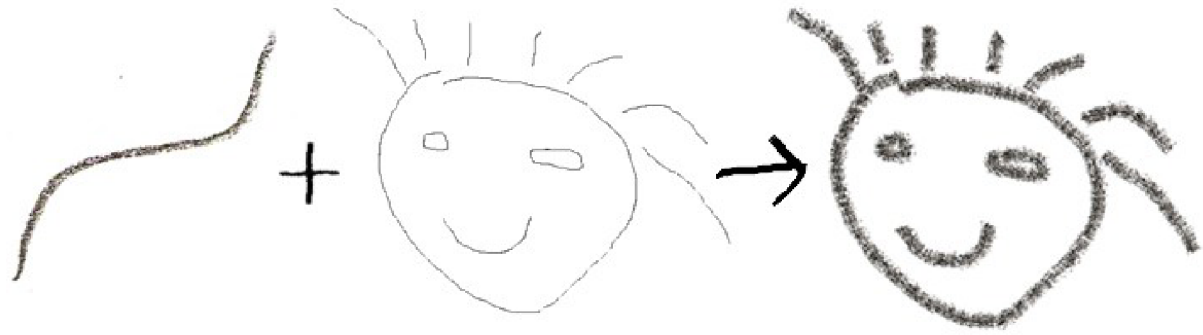
intenzity sa vypočíta zo stredu riadku. Štruktúra sa zachováva na základe porovnania susedných pixelov, rozdiel ich intenzít nesmie prekročiť hodnotu z parametru *-pxT*. Ak ju prekročia, prehľadávanie riadku do daného smeru končí. Ak sú intenzity priateľné, vygeneruje sa pravidlo, ktoré zaručí vykreslenie pixelu v správnej veľkosti, na dobrom mieste a farbou zo vstupného obrázku. Posunie sa na vedlajší pixel a proces sa zopakuje. Tentoraz sa za referenčnú hodnotu berie intenzita predchádzajúceho pixelu. Podsegment je teda skupina vykreslovacích pravidiel na jednom riadku. Zo stredu stĺpca pokračujeme po riadkoch hore aj dole, generujú sa ďalšie podsegmenty. Segment obsahuje všetky podsegmenty z jedného stĺpca a vytvorí tak základný vykreslovací element pre cfdg. S pravdepodobnosťou určenou parametrom *-ch* sa po vykreslení jedného segmentu vykreslí aj bezprostredne nasledujúci segment, zachová sa tak väčšia časť štruktúry zo vstupného obrázku.

Funkcia *generateCFG()* takto po stĺpcoch prejde celým vstupným obrázkom a vygeneruje súbor s cfdg gramatikou, ktorá dokáže napodobiť štruktúru obrázku. Na záver sa ešte volá funkcia *createNfo()*, tá vytvorí súbor o vygenerovanej gramatike. Ten obsahuje dátum vytvorenia, celkový počet pravidiel a parametre, s ktorými generátor pracoval.

3.3 Použitie

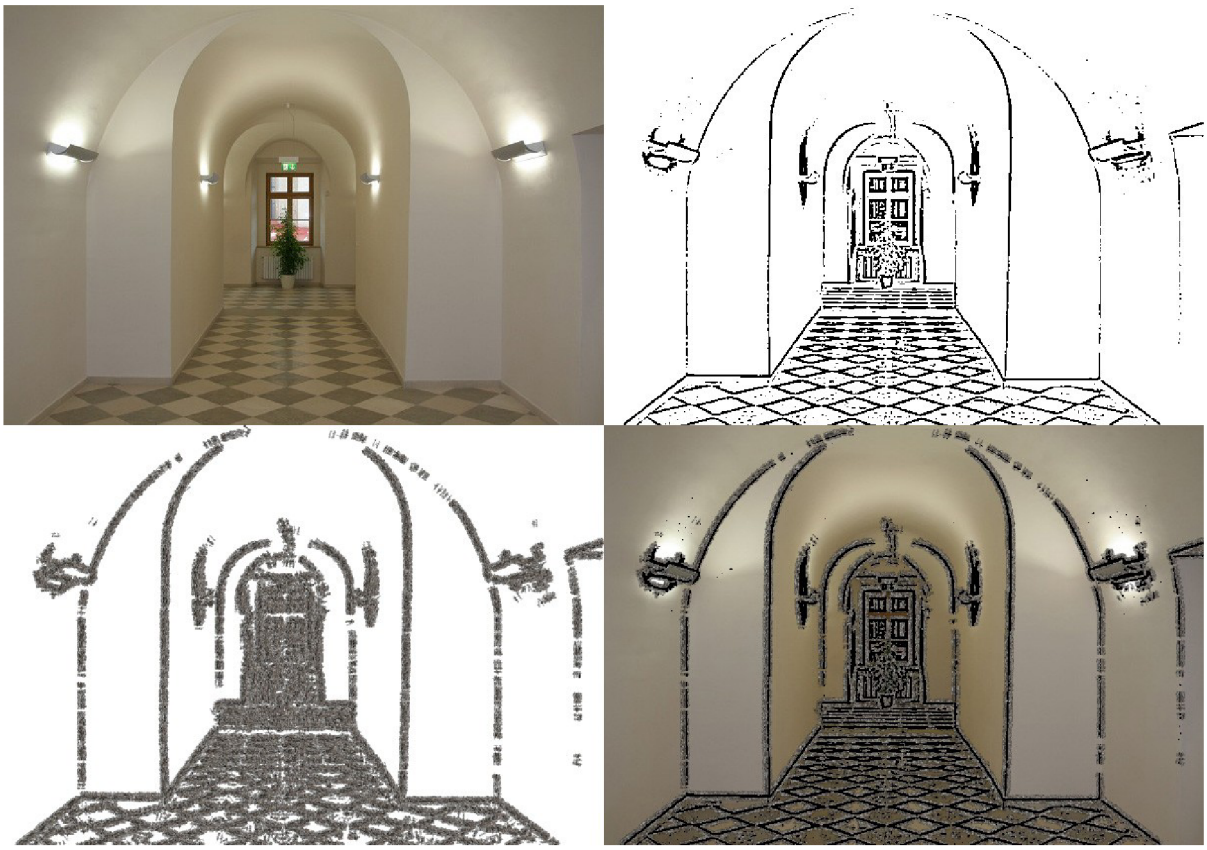
Automatické generovanie gramatiky, ktoré dokáže napodobiť takmer akúkoľvek štruktúru, rozširuje použitelnosť a možnosti aplikácie cs_cfdg. Spolu tvoria plnohodnotný modul pre cfdg a v budúcnosti sa očakáva spolupráca s komunitov zaoberajúcou sa touto tematikou. Projekt má svoje vlastné internetové stránky (vrátane dokumentácie) a je voľne šíritelný.

Základné použitie generátoru gg_cs_cfdg a cs_cfdg je znázornené na ilustrácii 9. Ide o jednoduché použitie vygenerovanej gramatiky na líniu. Ponúka sa jedna zaujímavá myšlienka použitia týchto nástrojov a to konkrétne vo videu. Pomocou jednoduchého skriptu „rozsekať“ video na sekvenciu záberov a tie modifikovať. Na každý záber postupne aplikovať detekciu hrán, vhodným programom je ImageMagick, bol použitý pri tvorbe ilustrácie 10. Obrázky s detekovanými hranami zanalyzovať programom cs_cfdg, použiť gramatiku vygenerovanú pomocou gg_cs_cfdg, a následne



Ilustrácia 9: Použitie aplikácií gg_cs_cfdg a cs_cfdg

vykresliť cez cfdg. Získame tak vrstvu kde sú hrany nakreslené bezkontextovou gramatikou z naskenovaného ťahu. Zostáva zložiť vrstvy do jedného obrázku a nakoniec zo všetkých modifikovaných obrázkov vytvoriť video.



Ilustrácia 10: Modifikácia fotografie

4 Záver

Generátor bezkontextovej návrhovej gramatiky gg_cs_cfdg je podľa všetkého zatiaľ jediný svojho druhu. Umožňuje automatické generovanie návrhových gramatík analýzou vstupného obrázku. Návrhová gramatika napodobňuje štruktúru ťahu a popisuje ju súborom vykresľovacích pravidiel.

Spolu s aplikáciou cs_cfdg, ktorú rozširuje, tvorí plnohodnotný modul pre aplikáciu cfdg a predpokladá sa ich ďalší spoločný vývoj. Generátor gramatiky a cs_cfdg budú v budúcnosti prezentované ako jeden spoločný projekt. Už existuje funkčná www stránka tohoto projektu (http://cscfdg.sourceforge.net/), vrátane galérie, odkazu na dokumentáciu a jednoduchý tutoriál, ako tieto nástroje používať. Po ohlásení projektu na diskusných fórach a stránkach k cfdg, sa predpokladá zapojenie ďalších ľudí do vývoja. Možností pre zdokonalenie generátoru je viacero, napr. prepísanie zdrojových kódov do c++, prípadne aj jeho priama integrácia do cs_cfdg. V novšej verzii sa ráta s odstránením prázdnych (zbytočných) pravidiel, ktoré by v bezkontextových gramatikách nemali ani existovať, a ďalšom zmenšovaní veľkosti generovanej návrhovej gramatiky. Veľmi vítané by bolo grafické uživateľské rozhranie, stačilo by pridať záložku do gui cs_cfdg.

Literatúra

[1] Česka Milan: Teoretická informatika, učební texty, FIT VUT v Brně, 2002

[2] Meduna A., Lukáš R.: Formální jazyky a překladače, studijní opora, FIT VUT v Brně, 2006

[3] WWW stránky. Context free art
http://www.contextfreeart.org/

[4] WWW stránky. Coyne Chris, Context free design grammar
http://www.chriscoyne.com/wordpress/index.php/cfdg-context-free-design-grammar/

[5] WWW stránky. Microsoft development network, Bitmaps
http://msdn.microsoft.com/en-us/library/ms532349(VS.85).aspx

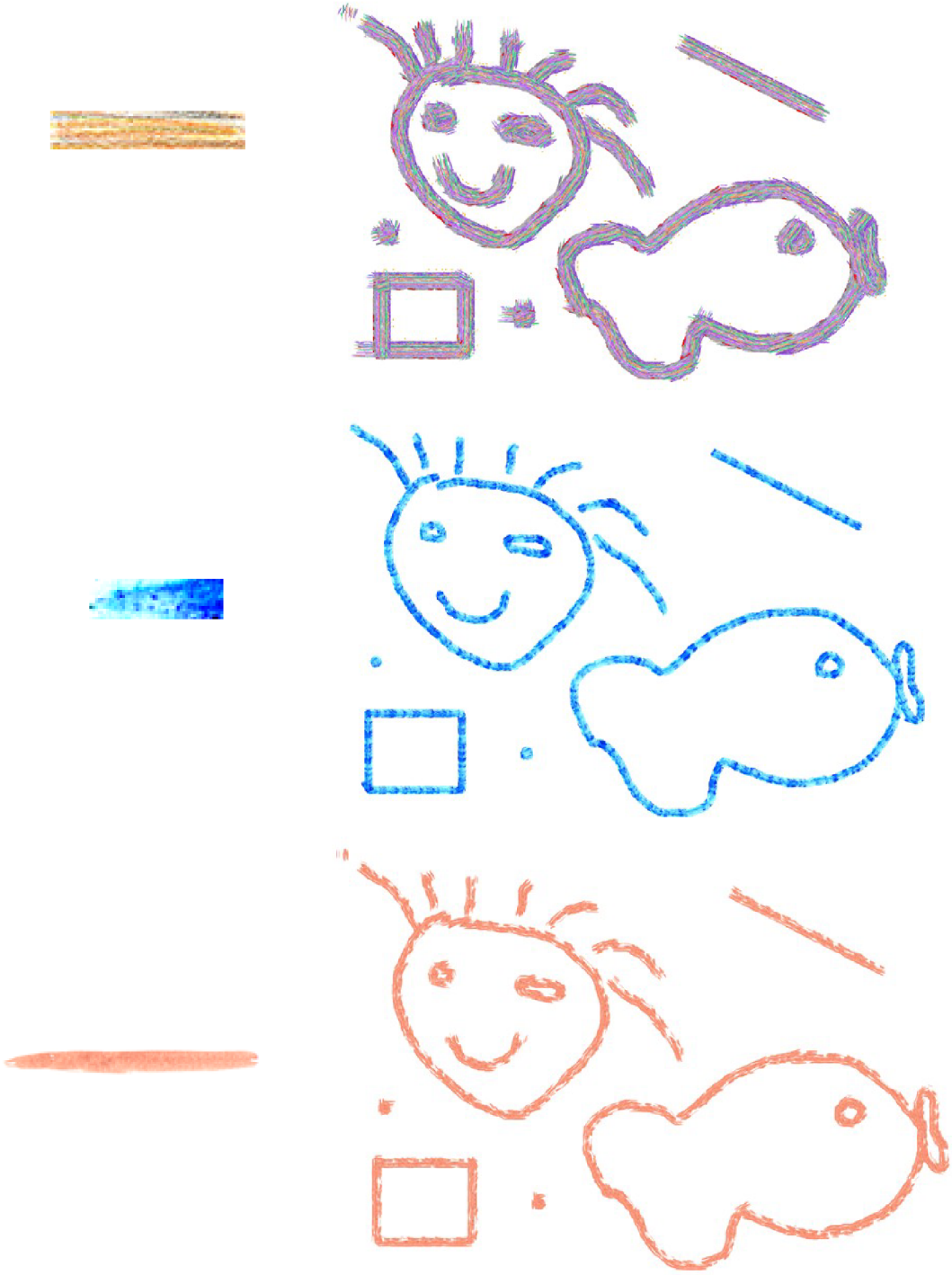
[6] WWW stránky. Wikipedia, the free encyclopedia.
http://en.wikipedia.org/wiki/Main_Page

Zoznam príloh

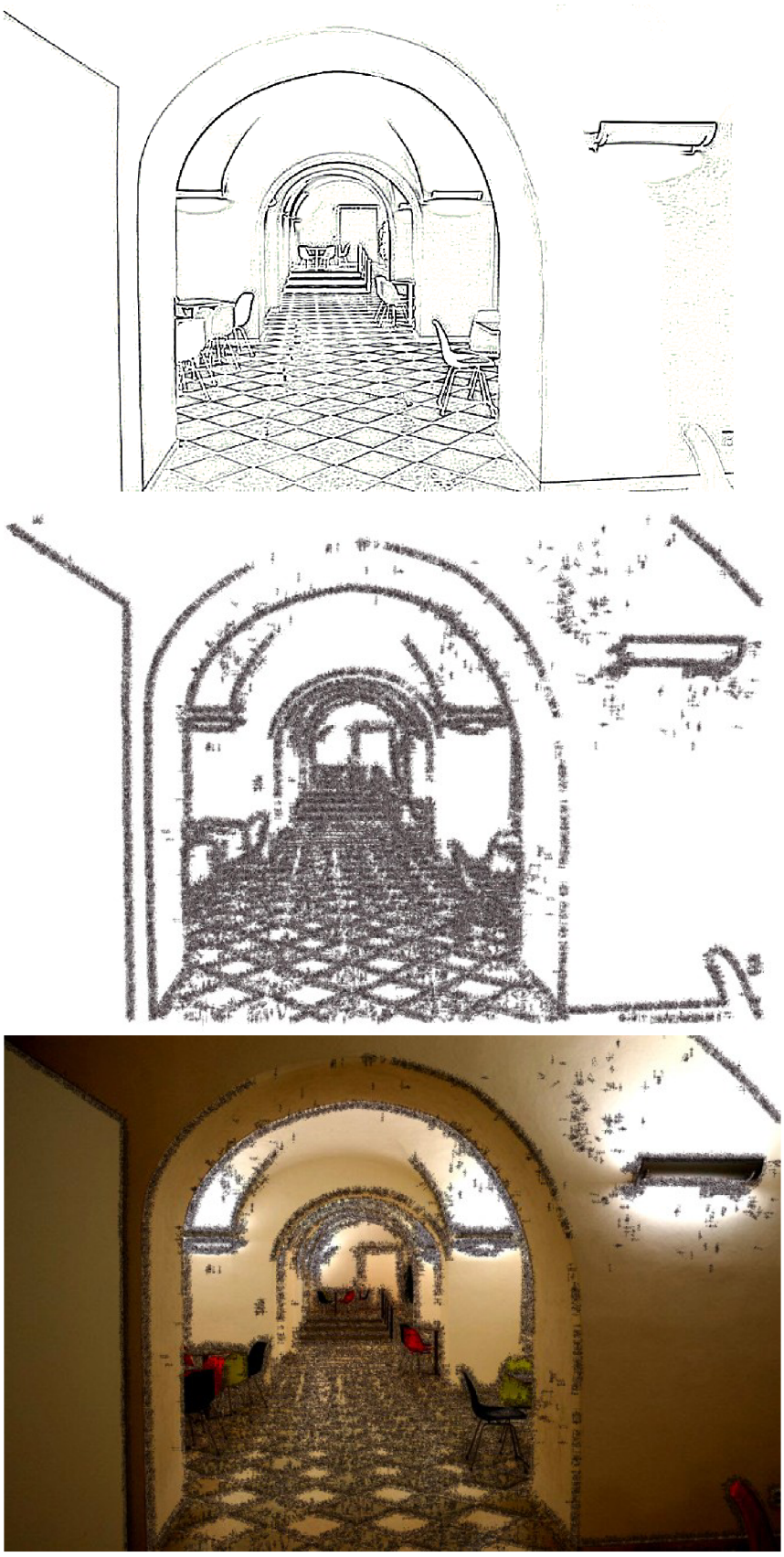
Príloha 1. CD s programom, zdrojovými súbormi, programovou dokumentáciou a ukážkovými výstupmi

Príloha 2. Obrazová príloha, ukážky použitia rozoberaných nástrojov

Obrazová príloha



Obrazová príloha 1: Aplikácia rôznych gramatík vygenerovaných generátorom gg_cs_cfdg

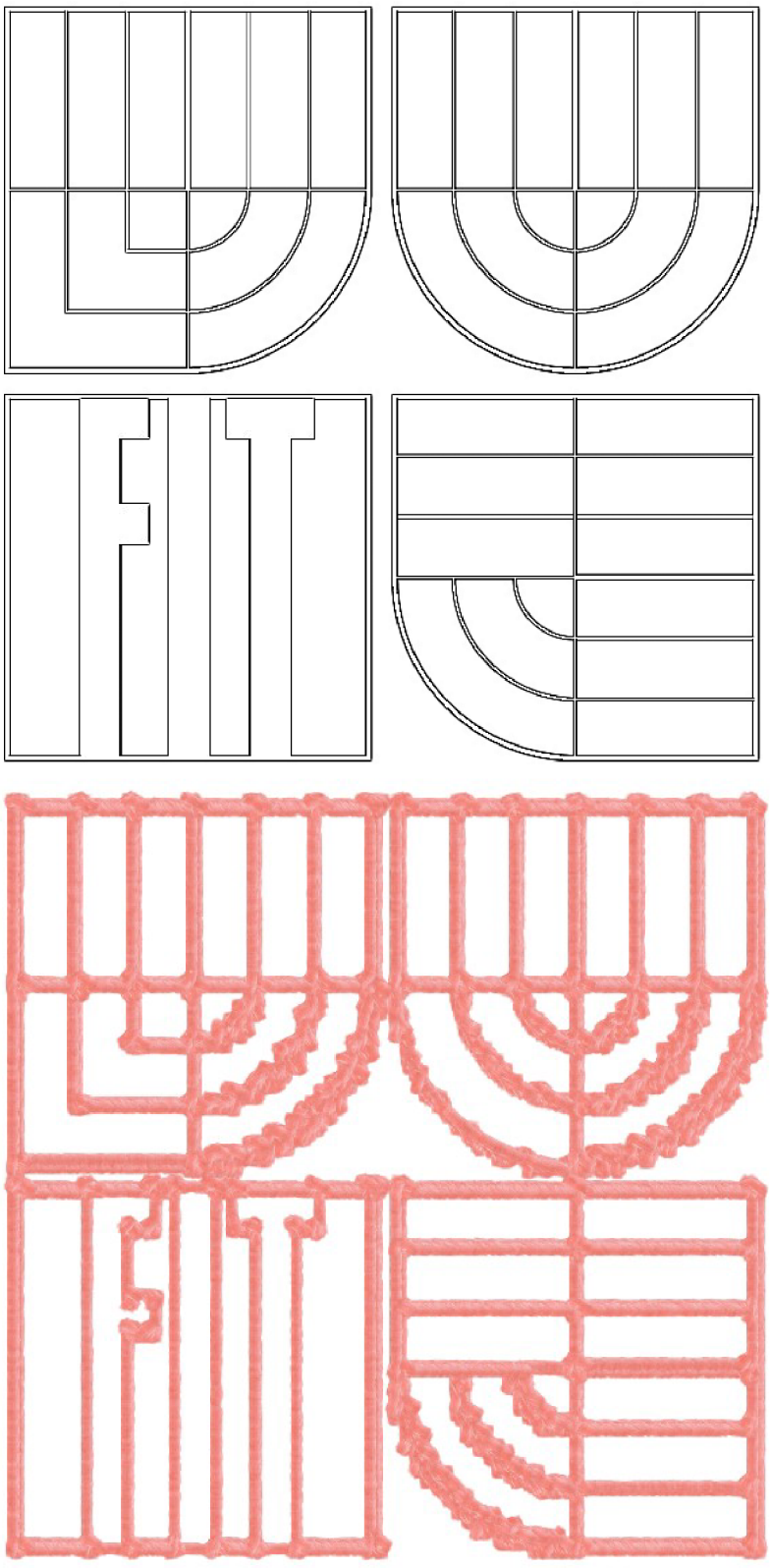


Obrazová príloha 2: Aplikácia nástrojov gg_cs_cfdg, cs_cfdg, cfdg na fotografiu

```
background { b 1 }                                  // pravidlo pozadia

rule stroke {                                       // pravidlá ťahov
       seg0 {}
}
rule stroke {
       seg1 {}
}
.
.
rule seg0 0.50 {                                    // pravidlá segmentov bez pokračovania
       seg0_0 {r 90.0}
}
rule seg0 0.50 {                                    // pravidlá segmentov s pokračovaním
       seg0_0 {r 90.0}
       seg1 {x 0.3750}
}
rule seg0_0 {                                       // pravidlá podsegmentov
       SQUARE{s 0.3750 x 0.0000 y 0.0000 hue 318 sat 0.06 b 0.63 a -0.08}
       seg0_1{}
}
rule seg0_1 {
       SQUARE{s 0.3750 x 0.3750 y 0.0000 hue 324 sat 0.08 b 0.70 a -0.08}
       SQUARE{s 0.3750 x 0.3750 y 0.3750 hue 326 sat 0.07 b 0.75 a -0.16}
       SQUARE{s 0.3750 x 0.3750 y 0.7500 hue 344 sat 0.06 b 0.78 a -0.24}
       SQUARE{s 0.3750 x 0.3750 y 1.1250 hue 349 sat 0.05 b 0.84 a -0.62}
       seg0_2{}
}
.
.
rule seg0_39 {
       seg0_40{}
}
rule seg0_40  {}                                    // pravidlo ukončujúce segment
.
.
```

Obrazová príloha 3: Časť gramatiky vygenerovanej aplikáciou gg_cs_cfdg



Obrazová príloha 4: Logo FIT VUT v Brně nakreslené červenou vodovou farbou