

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Bakalářská práce**

# **Modelování dřevin pomocí L-systémů**

# Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 4. května 2013

Eva Koubková

# Abstract

Modeling of trees using L-systems

This thesis is focused on familiarization with Lindenmayer systems (L-systems) and their subsequent use for modeling trees. L-systems are a type of rewriting grammars, created by Hungarian biologist and botanist Aristid Lindenmayer to model plant growth.

The practical part of this thesis is to develop grammars for selected groups of trees. For rulemaking was necessary to determine the possibilities of L-system and find a suitable tool for creating grammars and their subsequent visualizations. The resulting grammar, and 3D models of trees and shrubs created by these grammars are then compared. In addition to differences between the grammars also resulting models are compared with real trees.

# Abstrakt

Modelování dřevin pomocí L-systémů

Tato bakalářská práce je zaměřena na seznámení se s Lindenmayerovými systémy (L-systémy), a jejich následné využití pro modelování dřevin. L-systémy jsou typ přepisovacích gramatik, vytvořených maďarským biologem a botanikem Aristidem Lindenmayerem pro modelování růstu rostlin.

Praktickou částí této práce je vytvoření gramatik pro vybrané skupiny dřevin. Pro tvorbu předpisů bylo nejprve nutné zjistit možnosti L-systémů a najít vhodný nástroj pro tvorbu gramatik a jejich následnou vizualizaci. Vzniklé gramatiky, a pomocí nich vytvořené 3D modely stromů a keřů, jsou následně porovnávány. Kromě rozdílů mezi gramatikami jsou také srovnány vzniklé modely s reálnými dřevinami.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>4</b>
<b>2</b>	<b>L-systémy a gramatiky</b>	<b>5</b>
2.1	L-systémy . . . . .	5
2.2	Želví grafika . . . . .	6
2.3	Způsoby přepisování . . . . .	8
2.4	0L-systémy . . . . .	9
2.5	D0L-systémy . . . . .	9
2.6	Závorkové L-systémy . . . . .	9
2.7	Kontextové L-systémy . . . . .	10
2.8	Parametrické L-systémy . . . . .	12
2.9	Stochastické L-systémy . . . . .	13
<b>3</b>	<b>Aplikace pro vizualizaci</b>	<b>15</b>
3.1	Vyzkoušené nástroje . . . . .	15
3.1.1	Generate a Tree as L-System . . . . .	15
3.1.2	Lsystems Application . . . . .	16
3.1.3	WWW L-System Explorer . . . . .	17
3.2	Fractal Grower . . . . .	18
3.3	L-studio . . . . .	19
3.4	Vybrané nástroje . . . . .	20
3.4.1	Základ práce s L-studiem . . . . .	20
<b>4</b>	<b>Skupiny dřevin</b>	<b>24</b>
4.1	Základní rozdělení . . . . .	24
4.2	Rozdělení dle větvení stonku . . . . .	25
4.3	Rozdělení dle tvaru koruny . . . . .	27
<b>5</b>	<b>Návrh gramatik</b>	<b>28</b>
5.1	Přechod z Fractal Grower na L-studio . . . . .	29
5.2	Vytváření gramatik . . . . .	30

---

5.3	Společné vlastnosti gramatik . . . . .	30
<b>6</b>	<b>Vytvořené modely a jejich porovnání</b>	<b>33</b>
6.1	Rozdělení dle způsobu větvení . . . . .	33
6.1.1	Monopodiální větvení . . . . .	33
6.1.2	Sympodiální větvení . . . . .	35
6.2	Rozdělení dle tvaru koruny . . . . .	37
6.2.1	Pyramidální tvar . . . . .	37
6.2.2	Sloupovitý tvar . . . . .	39
6.2.3	Kulovitý tvar . . . . .	41
6.2.4	Deštníkovitý tvar . . . . .	43
6.2.5	Nezařazený tvar vrby . . . . .	44
6.3	Keře . . . . .	45
6.4	Zhodnocení vytvořených gramatik . . . . .	47
6.5	Srovnání modelů a reality . . . . .	48
<b>7</b>	<b>Závěr</b>	<b>52</b>
<b>A</b>	<b>Obsah příloženého CD</b>	<b>55</b>

# Seznam obrázků

2.1	Aplikace přepisovacích pravidel . . . . .	5
2.2	a) Kreslíci želva b) Změna polohy želvy . . . . .	6
2.3	Vektory a symboly udávající polohu želvy v prostoru, a rotační matice pro orientaci želvy ve 3D [1] . . . . .	7
2.4	Stejný výsledek pro dva způsoby přepisování . . . . .	8
2.5	Výstupy pro čtyři iterace závorkového L-systému . . . . .	10
2.6	Výstupy pro čtyři iterace kontextového L-systému . . . . .	11
2.7	Výstupy pro čtyři iterace parametrického L-systému . . . . .	12
2.8	Výstupy pro čtyři iterace stochastického L-systému - jedna z možností . . . . .	14
2.9	Příklad na stochastické L-systémy - další možnosti výstupu . . . . .	14
3.1	Generate a Tree as L-Systemn . . . . .	15
3.2	Lsystems Application . . . . .	16
3.3	Lsystems Application - srovnání použitých křivek . . . . .	17
3.4	WWW L-System Explorer . . . . .	18
3.5	Program Fractal Grower . . . . .	19
3.6	Záložka L-system programu L-studio a grafický výstup zapsané gramatiky . . . . .	21
3.7	Záložka View programu L-studio . . . . .	22
3.8	Záložka Animate programu L-studio . . . . .	23
3.9	Nastavování barvy v záložce Colors programu L-studio . . . . .	23
4.1	Růstové tvary dřevin [3] . . . . .	24
4.2	Větvení stonku . . . . .	25
4.3	Habity korun [4] . . . . .	27
5.1	Vytvořená gramatika pro cypřiš v programu Fractal Grower . . . . .	28
5.2	Vytvořená gramatika pro cypřiš v programu L-Studio . . . . .	29
6.1	Výsledek gramatiky pro monopodiální větvení . . . . .	34
6.2	Výsledek gramatiky pro sympodiální větvení . . . . .	35

6.3	Výsledek gramatiky pro sympodiální větvení s přidanými alternativami větvení . . . . .	35
6.4	Monopodiální jehličnan pyramidálního tvaru . . . . .	38
6.5	Monopodiální listnatý strom pyramidálního tvaru . . . . .	38
6.6	Monopodiální strom sloupovitého tvaru . . . . .	40
6.7	Monopodiální listnatý strom sloupovitého tvaru, vzniklý úpravou gramatiky pro monopodiální listnatý strom . . . . .	40
6.8	Monopodiální listnatý strom kulovitého tvaru . . . . .	42
6.9	Sympodiální strom kulovitého tvaru . . . . .	42
6.10	Sympodiální strom deštníkovitého tvaru . . . . .	43
6.11	Sympodiální strom tvaru vrby . . . . .	44
6.12	Vytvořený keř v programu L-Studio . . . . .	45
6.13	Srovnání monopodiálních stromů sloupovitého tvaru se skutečným stromem . . . . .	48
6.14	Srovnání monopodiálního jehličnanu se skutečným stromem . . . . .	49
6.15	Srovnání monopodiálního listnatého stromu s kulovitou korunou se skutečným stromem . . . . .	49
6.16	Srovnání monopodiálního listnatého stromu s pyramidální korunou se skutečným stromem . . . . .	50
6.17	Srovnání keřů . . . . .	50
6.18	Srovnání sympodiálního stromu s kulovitou korunou se skutečným stromem . . . . .	51
6.19	Srovnání sympodiálního stromu s rozložitou kulovitou korunou se skutečným stromem . . . . .	51



# Seznam tabulek

2.1	Základní symboly želví grafiky pro 2D vykreslování . . . . .	6
2.2	Základní symboly želví grafiky pro 3D vykreslování . . . . .	8
2.3	Rozšíření symbolů želví grafiky pro závorkové L-systémy . . . . .	9
2.4	Typy kontextových L-systémů a jejich zápis . . . . .	10
3.1	Příklad a význam zdrojového textu . . . . .	21
3.2	Příklad a význam zápisu v záložce View . . . . .	22

# 1 Úvod

Tématem této práce je využití L-systémů pro vytváření modelů dřevin, tedy stromů a keřů. Nejprve je ovšem nutné se důkladně seznámit s těmito systémy, které jsou obecně zaměřeny na modelování rostlin pomocí přepisovacích pravidel. Důležitým předpokladem pro modelování rostlin pomocí L-systémů jsou vlastnosti rostlin podobné fraktálům, tedy že malá část rostliny má podobný tvar jako celá rostlina.

Pro vizualizaci Lindenmayerových systémů již existuje více aplikací, na kterých se lze naučit základní principy. Menší výběr je pak mezi aplikacemi poskytujícími více možností pro modelování rostlin. Částí této práce je také najít takovou aplikaci, která bude schopna zobrazovat modely dřevin.

Hlavním cílem této práce je tvorba různorodých gramatik pro generování stromů a keřů pomocí L-systémů. Takový úkol není úplně jednoduchý, vzhledem k tomu, že každý strom je naprostý unikát. Nelze tedy generovat dřeviny pouze pomocí jednoduchých předpisů, musíme také zajistit určitou náhodnost při jejich růstu. To nám umožňují takzvané stochastické L-systémy, díky nimž může existovat pro jeden symbol více přepisovacích pravidel a mezi nimi se podle zadané pravděpodobnosti vybírá, které se použije.

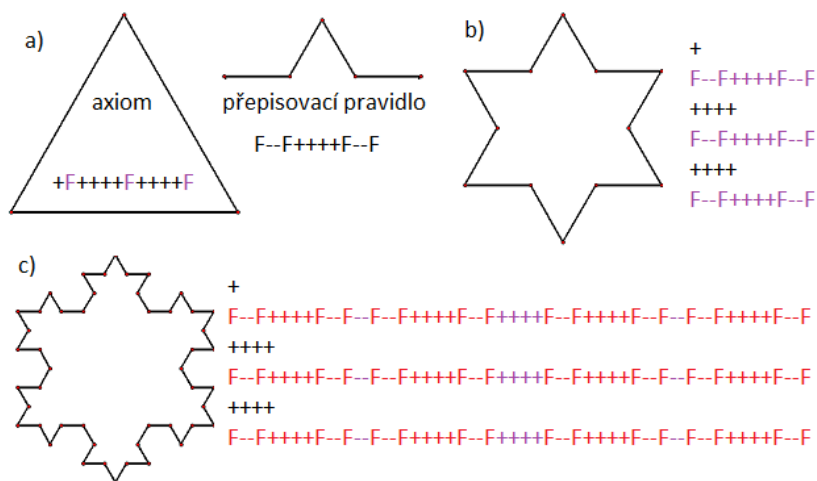
V rámci práce jsou dřeviny rozděleny do několika skupin podle jejich tvaru, který vzniká podle postupného větvení stonku. Pro některé vybrané skupiny jsou vytvořeny základní předpisy a pro další skupiny jsou z těchto základních předpisů různými úpravami odvozovány další předpisy. Vznikající předpisy jsou při odvozování mezi sebou porovnávány, podle toho co je třeba změnit, aby vznikl předpis pro jinou skupinu.

## 2 L-systémy a gramatiky

### 2.1 L-systémy

L-systém, neboli Lindenmayerův systém, je forma prepisovací gramatiky, využívá ke generování fraktálů. Kromě generování různorodých fraktálů se tyto systémy hodí také na generování rostlin, které mají často vlastnosti podobné fraktálům (menší část rostliny připomíná rostlinu celou). Za vznikem L-systémů stojí maďarský biolog Aristid Lindenmayer, který je také spoluautorem knihy [1], která je hlavním zdrojem informací v této práci.

Základem prepisovacích gramatik je axiom (startovací řetězec, který se bude dále prepisovat) a jedno nebo více prepisovacích pravidel. U L-systémů probíhá prepisování paralelně, tedy v jednom kroku prepíšeme najednou všechny symboly, pro které známe prepisovací pravidla. Příklad aplikace prepisovacích pravidel je na obrázku 2.1. V ukázce jsou použity tři základní symboly, a to F, představující vykreslení čáry předem dané velikosti. A znaky + a - udávající natočení o 30 stupňů doprava nebo doleva. Na obrázku 2.1.a) je definice gramatiky, tedy axiom a řetězec, kterým budeme prepisovat symbol F (prepisovací pravidlo). Na obrázcích 2.1.b a 2.1.c jsou ukázány vzniklé obrazce a řetězce po aplikaci dané gramatiky. Pro přehlednost jsou řetězce rozděleny na řádky a barevně odlišeny ty části, které byly právě přepsány.



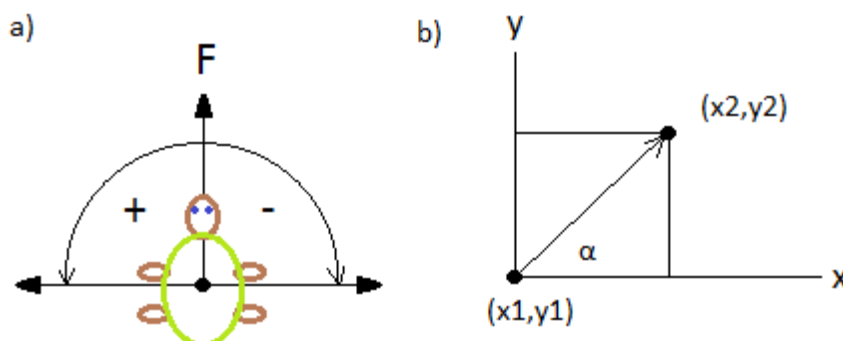
Obrázek 2.1: Aplikace prepisovacích pravidel

## 2.2 Želví grafika

O způsobu vykreslování L-systémů se hovoří jako o želví grafice. Želvou rozumíme bod, na kterém se zrovna nacházíme při vykreslování řetězce vzniklého po zadaném počtu prepisování. Kreslicí želva si uchovává určité parametry, podle kterých se kreslí, a to směr, jakým je natočená (respektive úhel natočení  $\alpha$ ), délku  $d$ , kterou urazí v jednom kroku, a aktuální souřadnice, na kterých se vyskytuje  $(x,y)$ .

Kreslicí želva s možnostmi změny směru je zobrazena na obrázku 2.2.a, vysvětlení základních symbolů je v tabulce 2.1. Na obrázku 2.2.b jsou zobrazeny dva stavy želvy:

1. stav  $(x_1,y_1)$ , kdy je želva na souřadnicích  $(x_1,y_1)$  a je natočená o úhel  $\alpha$ , tento stav bychom tedy mohli popsat jako  $(x_1,y_1, \alpha)$ .
2. stav  $(x_2,y_2)$ , je stav po provedení jednoho kroku želvy, kdy se první stav změnil na  $(x_2 = d\cos\alpha, y_2 = d\sin\alpha, \alpha)$ .

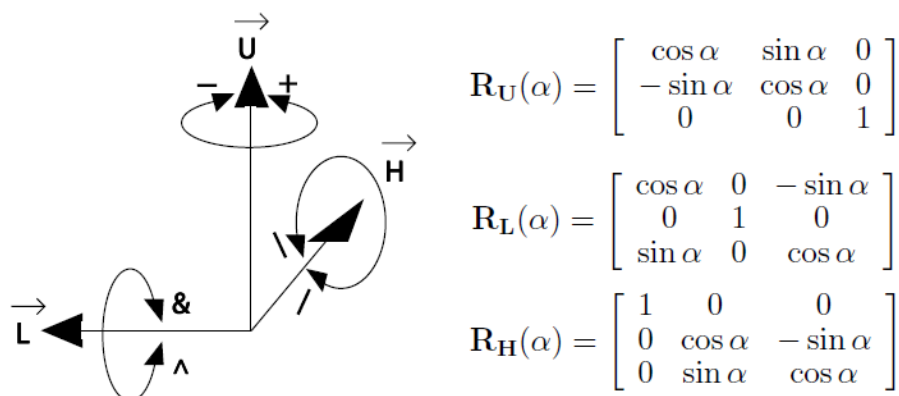


Obrázek 2.2: a) Kreslicí želva b) Změna polohy želvy

Symbol	význam
<b>F</b>	Vykreslí čáru o délce $d$ a změní stav želvy na $(d\cos\alpha, d\sin\alpha, \alpha)$ .
<b>f</b>	Posune želvu o délku $d$ , ale nic nevykreslí.
<b>+</b>	Natočí želvu doleva o zadaný úhel a změní stav želvy na (pozice $x$ , pozice $y$ , $\alpha +$ daný úhel).
<b>-</b>	Natočí želvu doprava o zadaný úhel a změní stav želvy na (pozice $x$ , pozice $y$ , $\alpha -$ daný úhel)

Tabulka 2.1: Základní symboly želví grafiky pro 2D vykreslování

V případě vykreslování ve 3D prostoru jsou možnosti želvy rošřeny o možnost natáčení do více stran, přibudou tedy další symboly pro vykreslování (tabulka 2.2). Pro orientaci želvy je třeba zavést tři vektory  $\vec{H}$ ,  $\vec{U}$ ,  $\vec{L}$  (na obrázku 2.3), podle kterých se bude želva různě natáčet. Směr pohledu želvy určuje vektor  $\vec{H}$ , orientaci krunýře určuje vektor  $\vec{U}$  a poslední vektor  $\vec{L}$  ukazuje levou stranu želvy. Rotaci želvy okolo těchto vektorů určují rotační matice na obrázku 2.3, celková rotace želvy se skládá ze všech tří rotací a je popsána rovnicí:  $[\vec{H}' \vec{L}' \vec{U}'] = [\vec{H} \vec{L} \vec{U}]R$



$$R_U(\alpha) = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_L(\alpha) = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix}$$

$$R_H(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}$$

Obrázek 2.3: Vektory a symboly udávající polohu želvy v prostoru, a rotační matice pro orientaci želvy ve 3D [1]

Kromě určení natočení želvy pomocí rotace, potřebujeme také želvou hýbat. K posunu želvy použijeme translační matici:

$$T(d) = \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

kde  $d$  je délka posunu želvy.

Symbol	význam
<b>F</b>	Vykreslí čáru o dané délce $d$ a změní stav želvy pomocí translační matice $T(d)$ .
<b>f</b>	Posune želvu o danou délku $d$ a změní stav želvy pomocí translační matice $T(d)$ , ale nic nevykreslí.
<b>+</b>	Natočí želvu doleva o úhel $\delta$ s využitím rotační matice $R_U(\delta)$ .
<b>-</b>	Natočí želvu doprava o úhel $\delta$ s využitím rotační matice $R_U(-\delta)$ .
<b>\</b>	Nakloní želvu doleva o úhel $\delta$ s využitím rotační matice $R_H(\delta)$ .
<b>/</b>	Nakloní želvu doprava o úhel $\delta$ s využitím rotační matice $R_H(-\delta)$ .
<b>^</b>	Natočí želvu nahoru o úhel $\delta$ s využitím rotační matice $R_L(-\delta)$ .
<b>&amp;</b>	Natočí želvu dolů o úhel $\delta$ s využitím rotační matice $R_L(\delta)$ .
<b> </b>	Obrátí želvu o 180 stupňů s využitím rotační matice $R_U(180)$ .

Tabulka 2.2: Základní symboly želví grafiky pro 3D vykreslování

## 2.3 Způsoby přepisování

Přepisování symbolů můžeme rozdělit na dva typy. Přepisování hran (edge rewriting) a přepisování bodů (respektive uzlů - node rewriting). V prvním případě přepisujeme takové symboly, které reprezentují hrany, tedy symboly  $F$  (a případně další symboly se stejným významem). V druhém případě přepisujeme body, tedy takové symboly, díky kterým se nic nevykresluje. Na obrázku 2.4 je ukázka dvou předpisů, z nichž jeden používá hranové přepisování a druhý uzlové přepisování, ale výsledek (dračí křivka po deseti iteracích) je stejný. Vzhledem k tomu, že symboly  $l$  a  $r$  nejsou definovány jako symboly želvy, jedná se pouze o body, které se nijak nevykreslují.



Obrázek 2.4: Stejný výsledek pro dva způsoby přepisování

## 2.4 0L-systémy

Základními L-systémy jsou takzvané bezkontextové (0 v názvu značí kontext délky 0). Označují se tak, protože nemají kontext (viz kontextové systémy). Pravidla mají tedy pouze jednoduchý tvar, a daný symbol se vždy přepíše zadaným řetězcem, bez ohledu na to, kde se vyskytuje. Tyto systémy, pro které platí vše co bylo dosud v této práci napsáno o L-systémech, nejsou tedy zatím ničím omezeny, a následující druhy L-systémů je dále rozšiřují.

## 2.5 D0L-systémy

Přidáním omezení v podobě determinismu k 0L-systémům, vznikají deterministické a bezkontextové L-systémy (D0 v názvu). Jedná se o nejjednodušší formu L-systémů. Determinismus se projevuje tím, že pro každý symbol existuje maximálně jedno přepisovací pravidlo. Příkladem těchto L-systémů je ukázka Kochovy vložky na obrázku 2.1, vzhledem k tomu že je zde použito pro symbol právě jedno přepisovací pravidlo:

Axiom: F++++F++++F

F -> F-F++++F-F

## 2.6 Závorkové L-systémy

Dosud popisované L-systémy neumožňovaly větvení, které je nezbytné pro tvorbu dřevin. Pro vytváření větvících se rostlin jsou proto L-systémy rozšířeny o dva znaky "[" a "]". Při vykreslování a nalezení znaku "[" je vše uloženo na zásobník, do té doby než se objeví ukončovací závorka "]". Obsah mezi těmito závorkami je pak větví rostliny.

Symbol	význam
[	Uloží aktuální stav želvy do zásobníku.
]	Vybere naposled uložený stav želvy ze zásobníku a nastaví jej jako aktuální stav.

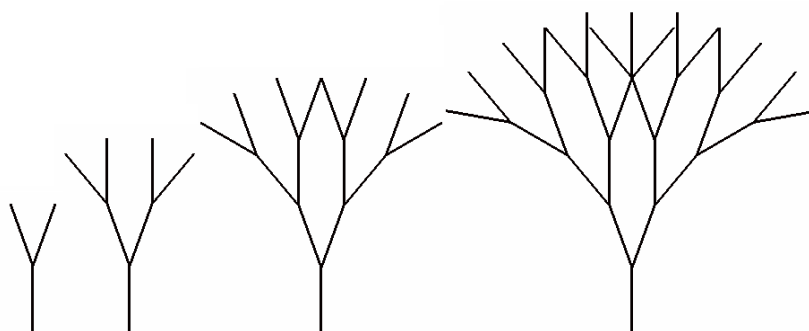
Tabulka 2.3: Rozšíření symbolů želví grafiky pro závorkové L-systémy

Příklad zápisu:

Axiom: F

F  $\rightarrow$  F[-F][+F]

V tomto příkladu se nejprve vykreslí větev směřující dopředu "F", poté je aktuální stav želvy uložen na zásobník a vykreslí se větev směřující doprava "-F", následně je kreslící želva vrácena na původní místo před kreslením této větve vybráním uloženého stavu ze zásobníku, a teprve pak je vykreslena větev směřující doleva "+F". Na obrázku 2.5 jsou ukázány výsledky aplikace tohoto pravidla po 1. až 4. iteraci s použitím úhlu větvení o velikosti 20 stupňů.



Obrázek 2.5: Výstupy pro čtyři iterace závorkového L-systému

## 2.7 Kontextové L-systémy

Oproti bezkontextovým systémům existují také kontextové systémy, u kterých použití přepisovacích pravidel závisí na zadaném kontextu. Kontext určuje předek anebo následník. U těchto systémů tedy záleží na posloupnosti symbolů. Rozlišujeme kontextové 2L-systémy, kde záleží zároveň na předkovi i následníkovi, a 1L-systémy, kde záleží pouze na následníkovi (pravý kontext), nebo pouze předkovi (levý kontext).

typ	zápis	popis
<b>L1</b>	$lk < A \rightarrow B$	A je přepsáno na B pokud leží za lk (levý kontext).
<b>L1</b>	$A > rk \rightarrow B$	A je přepsáno na B pokud leží před rk (pravý kontext).
<b>L2</b>	$lk < A > rk \rightarrow B$	A je přepsáno na B, pokud A leží mezi pravým a levým kontextem.

Tabulka 2.4: Typy kontextových L-systémů a jejich zápis



Bezkontextové systémy můžeme také zapsat jako kontextové, například pokud bychom chtěli zapsat příklad Kochovo vločky z obrázku 2.1 jako kontextový systém, vypadalo by prepisovací pravidlo takto:

$*\langle F \rangle* \rightarrow F-F++++F-F$

Hvězdička na místě pravého i levého kontextu značí že těmito kontexty může být cokoliv, následné prepisování tedy bude stejné jako u bezkontextových L-systémů.

Pro lepší ukázkou těchto systémů je uveden následující příklad, kde je využito pravého kontextu.

Axiom: FA

$* \langle F \rangle A \rightarrow FA [+F] [-FA]$

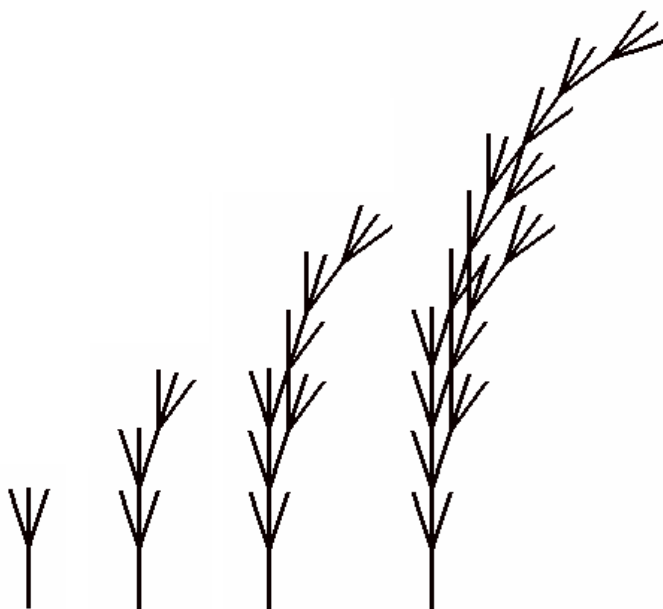
$A \rightarrow F$

Symbol F se v tomto případě prepíše pouze pokud je bezprostředně za ním symbol A, a starý symbol A se v každé iteraci prepíše na F. První dvě iterace budou tedy vypadat následovně:

1.iterace : FA [+F] [-FA] F

2.iterace : FA [+F] [-FA] F [+F] [-FA [+F] [-FA] F] F

Na obrázku 2.6 jsou zobrazeny grafické výstupy pro 4 iterace uvedeného příkladu, s použitým úhlem větvení o velikosti 18 stupňů.



Obrázek 2.6: Výstupy pro čtyři iterace kontextového L-systému

## 2.8 Parametrické L-systémy

Parametrické L-systémy vylepšují jednoduché L-systémy možností zadávat parametry jednotlivým částem vykreslovacího řetězce. Tedy odstraňují omezení, kdy velikost kreslené úsečky nebo úhlu natočení musela být vždy stejná.

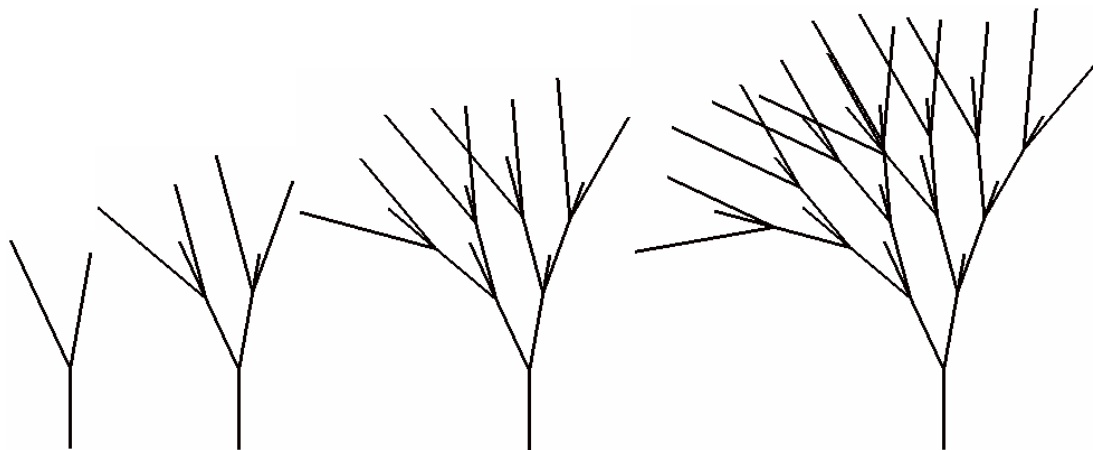
Parametry se zapisují do jednoduchých závorek za symbol ke kterému patří. U symbolů  $F(p)$  určuje parametr  $p$  délku vykreslené úsečky. U symbolů  $+(p)$  nebo  $-(p)$  označuje parametr  $p$  úhel natočení.

Příklad zápisu :

Axiom:  $F(p)$

$F(p) \rightarrow F(1) [-(10)F(1.5)] [(+25)F(1.8)]$

V tomto příkladě jsou v jednoduchých závorkách viděny jednotlivé parametry, náležející k symbolům psaným těsně nalevo. Natočení doprava  $-(10)$  bude tedy o  $10^\circ$  a následující úsečka  $F(1.5)$  bude dlouhá 1.5 jednotky, obdobně tomu bude i u druhé větve. Příklad je stejný jako příklad u závorkových systémů, až na zmíněné parametry, které způsobí změny, které jsou vidět na vyobrazení čtyř iterací na obrázku 2.7.



Obrázek 2.7: Výstupy pro čtyři iterace parametrického L-systému

Parametrů můžeme také využít pro určení za jakých podmínek se prepisovací pravidlo použije. Prepisovací pravidlo se pak zapisuje následovně:

předchůdce : podmínka  $\rightarrow$  následník

Například:

$F(p) : p > 2 \rightarrow F(1) [-(10)F(1.5)] [(+25)F(1.8)]$ ,  
 což znamená, že daný předchůdce  $F(p)$  se přepíše následníkem  
 $F(1) [-(10)F(1.5)] [(+25)F(1.8)]$ , jedině pokud je parametr  $p$  větší než 2.

V případě parametrických kontextových L-systémů můžeme v podmínce navíc porovnávat hodnoty kontextů.

levý kontext < předchůdce > pravý kontext : podmínka  $\rightarrow$  následník,

Například  $F(q) < F(p) : p < q \rightarrow F(1) [-(10)F(1.5)] [(+25)F(1.8)]$ , což znamená že se dané pravidlo použije pouze pokud je délka čáry předchůdce menší, než délka jeho levého kontextu.

## 2.9 Stochastické L-systémy

Stochastické L-systémy zajišťují princip náhody. V těchto systémech je pro jeden symbol zadáno více přepisovacích pravidel. U každého pravidla je pak zadáno s jakou pravděpodobností se použije.

Příklad zápisu :

Axiom:  $F(p)$

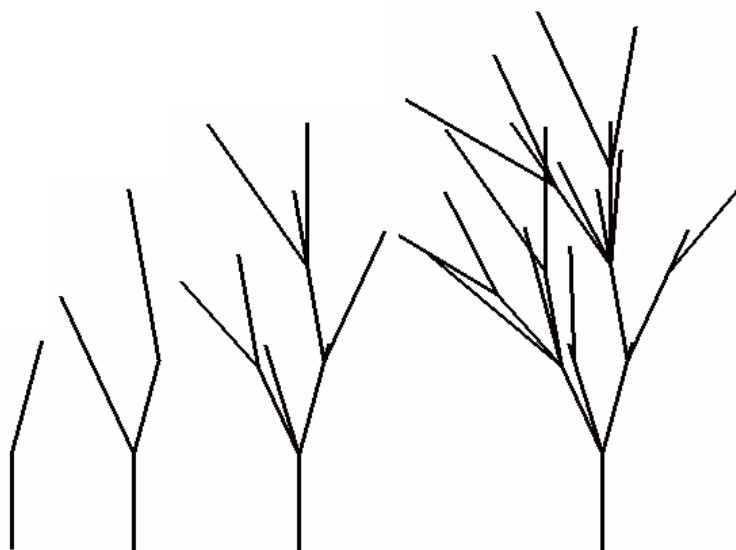
$F(p) \rightarrow F(1) [-(10)F(1.5)] [(+25)F(1.8)] : 0.25$

$F(p) \rightarrow F(1) [-(15)F(1.2)] [(+17)F(1.2)] : 0.25$

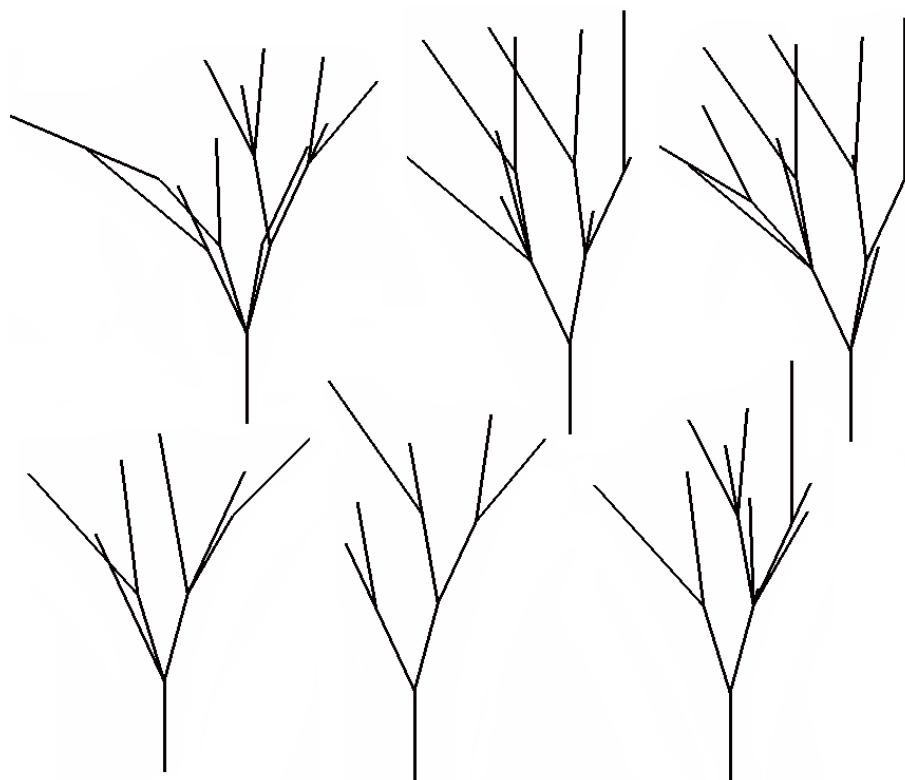
$F(p) \rightarrow F(1) [-(15)F(1.2)] : 0.25$

$F(p) \rightarrow F(1) [(+25)F(1.8)] : 0.25$

V tomto příkladě (jehož možné 4 iterace jsou zobrazeny na obrázku 2.8) je za dvojtečkou zadána pravděpodobnost s jakou se pravidlo použije. Vzhledem k tomu že jsou všechny pravděpodobnosti stejné (0.25), znamená to že všechna pravidla mají stejnou šanci že budou vybrána a použita. Pokud by byly pravděpodobnosti odlišné, mohlo by být některé pravidlo vybráno častěji než ostatní, anebo naopak. Použitím těchto pravidel s pravděpodobnostmi získáme pokaždé trochu jiný výsledek, jejich příklady jsou na obrázku 2.9.



Obrázek 2.8: Výstupy pro čtyři iterace stochastického L-systému - jedna z možností



Obrázek 2.9: Příklad na stochastické L-systémy - další možnosti výstupu

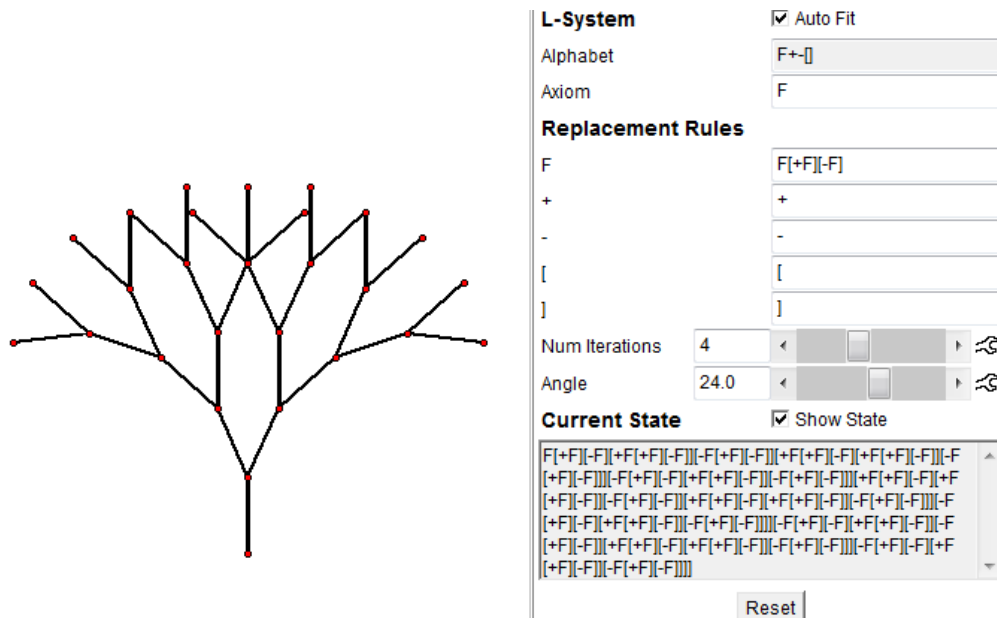
## 3 Aplikace pro vizualizaci

### 3.1 Vyzkoušené nástroje

Před započítím vytváření gramatik pro generování dřevin byly vyzkoušeny různé nástroje pro generování L-systémů. Nejprve byly zkušeny internetové applety, které nabízejí dostatečné možnosti pro první seznámení s L-systémy. Po seznámení se s L-systémy pomocí appletů ale bylo třeba vyzkoušet také programy, v kterých by šlo vytvářet a ukládat nové gramatiky.

#### 3.1.1 Generate a Tree as L-System

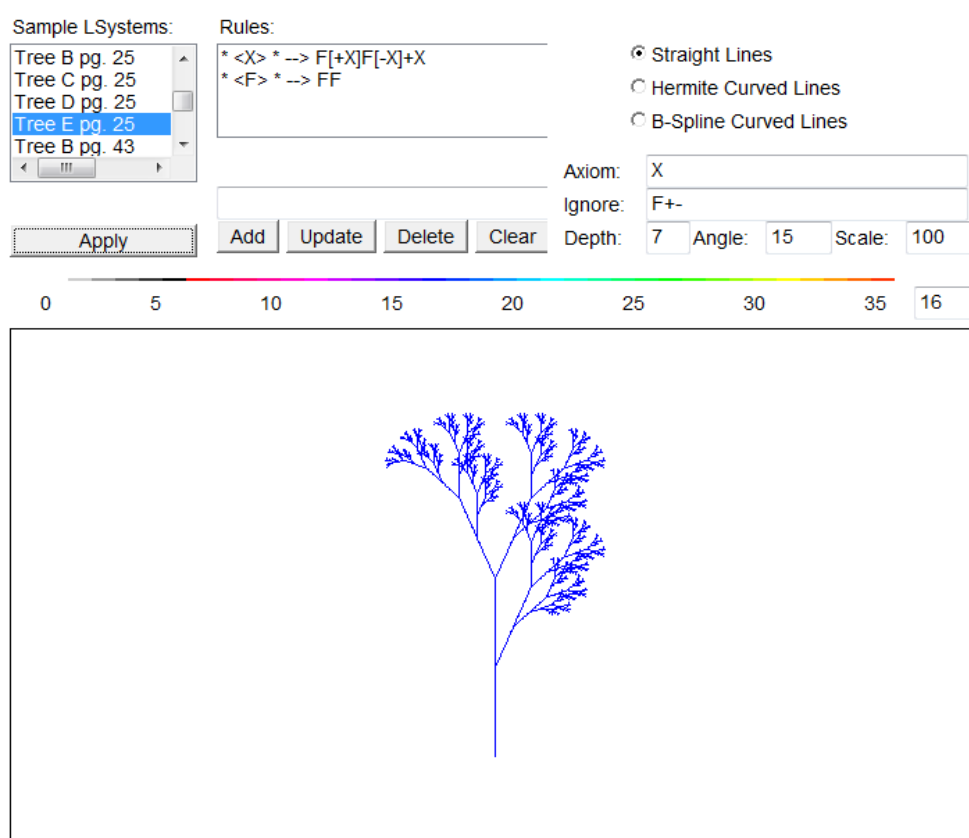
Jedná se o jednoduchý applet (na obrázku 3.1), ve kterém si lze vyzkoušet základní jednoduché gramatiky. Vše, co lze nastavit, je axiom, počet iterací, úhel natočení želvy a přepisovací pravidla jsou omezené na tyto symboly: F, +, -, [, ]. Pěknou vlastností tohoto appletu je možnost vypsání kreslicího řetězce po provedení všech přepisovacích iterací. Byl také využit pro vytvoření ilustračního obrázku 2.1.



Obrázek 3.1: Generate a Tree as L-Systemm

### 3.1.2 Lsystems Application

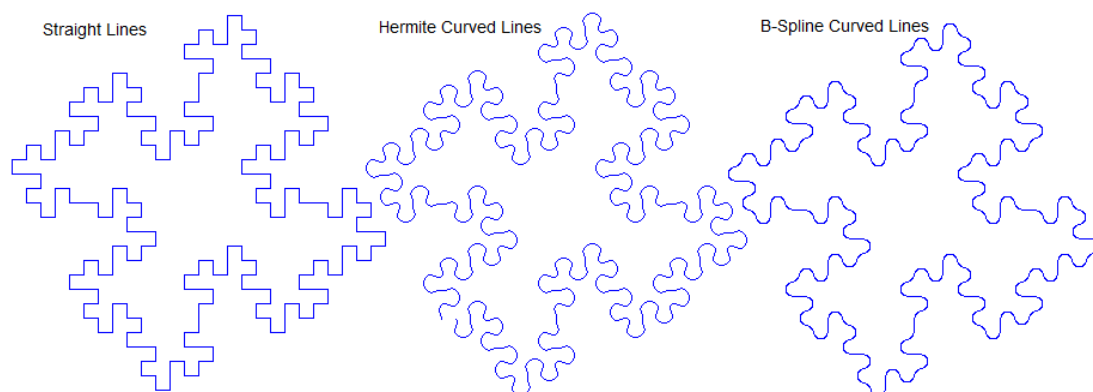
Tento applet [6] je o něco rozvinutější než předchozí, především jsou jeho možnosti rozšířeny o podporu kontextových L-systémů, a používání všech písmen abecedy jako prepisovacích symbolů. Jeho součástí je také několik zajímavých ukázkových příkladů fraktálů a také větvičích se struktur, mezi kterými je možné vybírat z nabídky v levém horním rohu appletu. Na obrázku 3.2 zobrazuje applet příklad větvičí struktury, mezi dalšími příklady jsou ale také například různé verze Kochovy křivky či dračí křivky.



Obrázek 3.2: Lsystems Application

Zajímavou možností je přepínání mezi způsobem zobrazení křivek (pravý horní roh 3.2), a to možnost vykreslení obyčejných rovných čar, hermitovských křivek, nebo b-spline křivek. Ukázka rozdílných výstupů pro různé křivky je na obrázku 3.3, při použití ukázkové gramatiky Kochova ostrova (Quadratic Koch Island 1):  $* <F> * \rightarrow F+F-F-FF+F+F-F$

Pod výběrem křivek se nastavuje axiom, ignorované symboly, počet iterací (Depth), úhel natočení (Angle) a měřítko (Scale). Mezi seznamem příkladů a výběrem typu křivek je pak okno, do kterého se zapisují přepisovací pravidla.



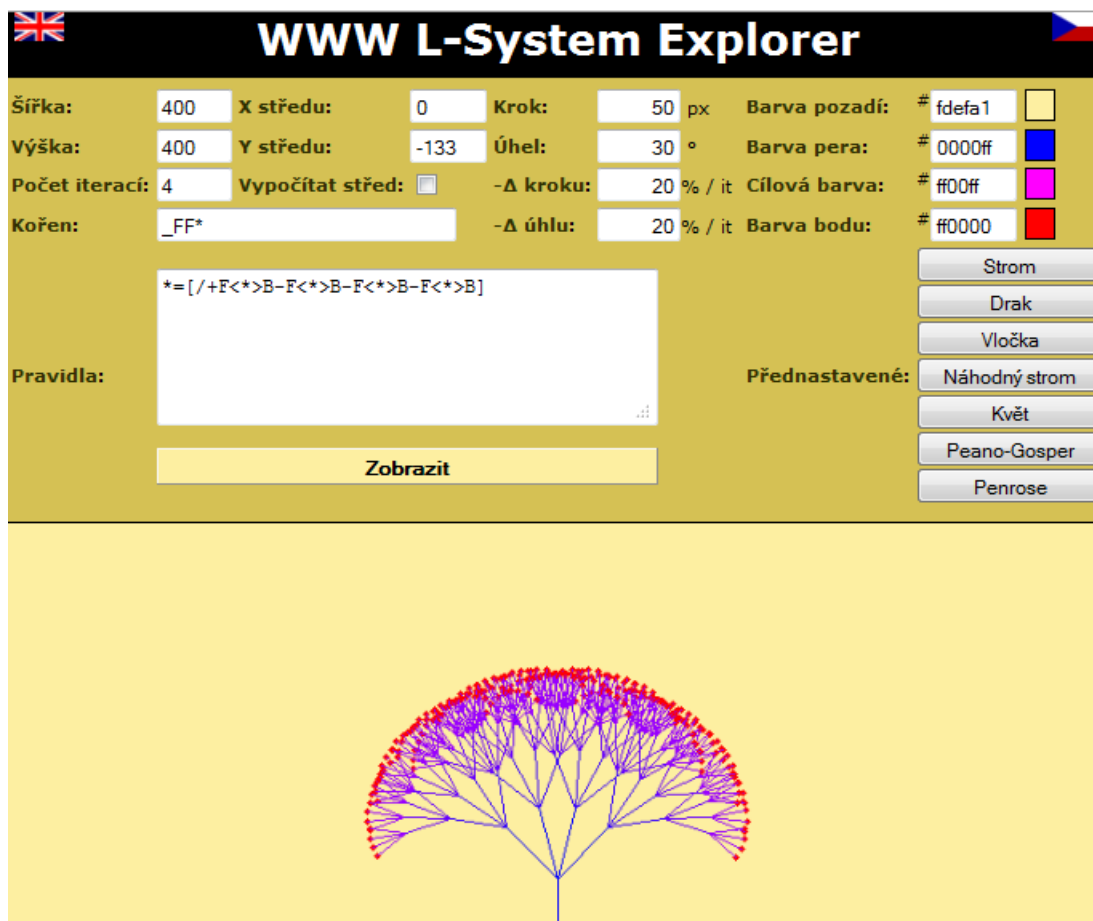
Obrázek 3.3: Lsystems Application - srovnání použitých křivek

### 3.1.3 WWW L-System Explorer

Český zástupce internetových nástrojů pro generování L-systémů z roku 2006 [7] se od předchozích aplikací liší především kompletní českou i anglickou lokalizací. Zahrnuje několik přednastavených ukázek L-systémů, a lze s ním také úpravou ukázkových předpisů vytvářet jednodušší gramatiky.

V návodu jsou popsány všechny používané parametry, způsob zadávání pravidel, a také jsou zde popsány všechny používané symboly. Některé používané znaky jsou přidány navíc k dosud používaným symbolům pro želví grafiku, jako například znak pro vykreslení velké tečky "\*", či symbol "B" pro krok želvou dozadu. Návod se zobrazí po kliknutí na kterýkoli název parametru (např. "Barva pozadí") při zobrazení otazníku u šipky.

Axiom se zapisuje do pole označeného jako "Kořen", přepisovací pravidla pak do pole "Pravidla". Nastavení dalších možných vlastností je díky české lokalizaci také intuitivní.

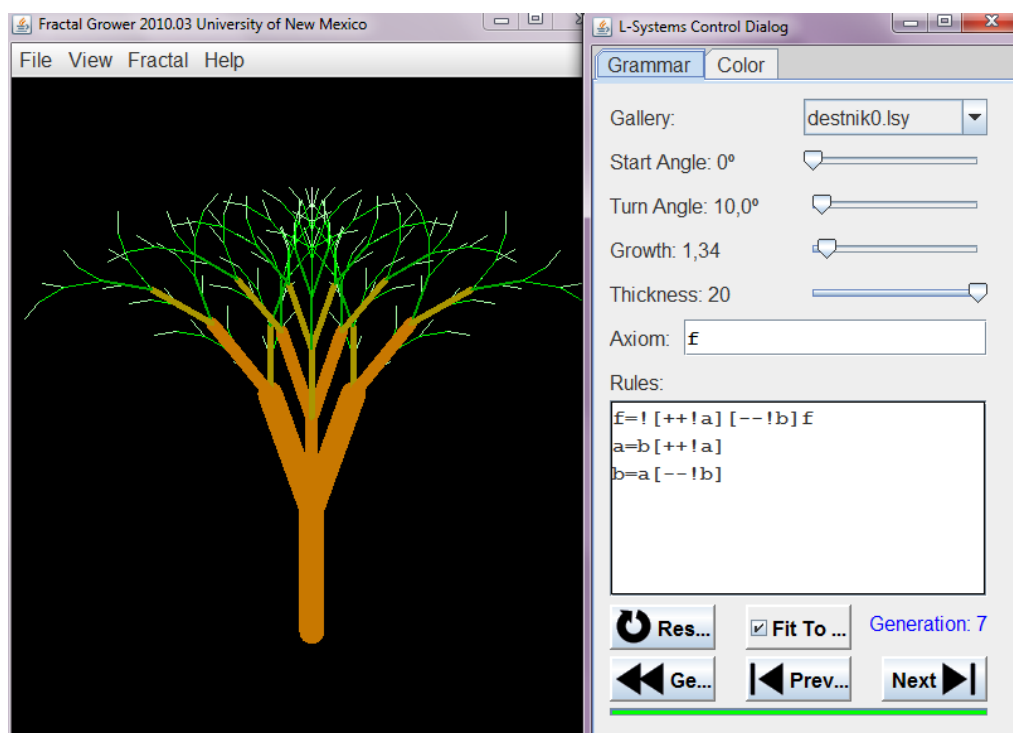


Obrázek 3.4: WWW L-System Explorer

## 3.2 Fractal Grower

Fractal Grower [8] (jehož grafické rozhraní můžeme vidět na obrázku 3.5) je pěkný program pro relativně jednoduché vytváření gramatik. V programu lze jednoduše nastavit počáteční hodnoty úhlu větvení (Start Angle a Turn Angle), konstantu růstu větví (Growth) a jejich šířku (Thickness). Tyto hodnoty lze plynule měnit a jejich změna se automaticky projevuje na vykresleném modelu. Vytváření gramatik je ale omezeno pouze na dvojrozměrné prostředí a neumožňuje používat stochastické systémy. Velmi se ale hodí do začátku práce s L-systémy.





Obrázek 3.5: Program Fractal Grower

### 3.3 L-studio

Narozdíl od předchozího programu je tento program [9] o hodně komplexnější a nabízí velké množství možností pro vytváření rostlin pomocí L-systémů. Umožňuje vytváření gramatik všech dříve zmíněných L-systémů a jejich vizualizaci jak ve 2D, tak ve 3D. Spoluautorem programu je Przemyslaw Prusinkiewicz, který je také jedním ze dvou hlavních autorů obsáhlé knihy o L-systémech [1].

Jediným mínusem tohoto programu je potřeba získání licence, která není zdarma. Na internetových stránkách [9] je ale možnost se zaregistrovat, a získat tak program k vyzkoušení. Takto získaný program začne po uplynutí zkušební lhůty (která je do konce roku), přidávat znehodnocující vodotisk přes vygenerované výstupy programu.

## 3.4 Vybrané nástroje

Pro vizualizaci vytvářených gramatik byl nejprve vybrán program Fractal Grower, který se hodí na počáteční zkoušení vytváření gramatik L-systémů. Vzhledem k jeho jednoduchosti, a chybějící podpoře stochastických L-systémů, ale bylo třeba později přejít na obsáhlejší program, kterým je L-studio. Z internetových appletů nebyl vybrán žádný zástupce, z důvodu jejich omezených možností, jako je podpora pouze malé části L-systémů a nemožnost ukládat vytvořené gramatiky.

Program Fractal Grower byl vybrán z důvodu jeho intuitivního ovládání a okamžitému zobrazování vizualizace gramatik, pro naučení práce s L-systémy. Byl využit převážně ze začátku pro vytvoření gramatik na generování různých stromů ve 2D. Následně ale bylo třeba přejít na komplexnější nástroj, který umožní modelování rostlin i ve 3D prostoru a nabídne větší možnosti tvorby gramatik. Na obrázku 3.5 je zobrazeno grafické uživatelské rozhraní tohoto programu, které tvoří dvě oddělená okna. V hlavním okně aplikace, které je nalevo, se zobrazuje vizualizace vytvořené gramatiky. Vlastní L-systém se nastavuje v okně napravo.

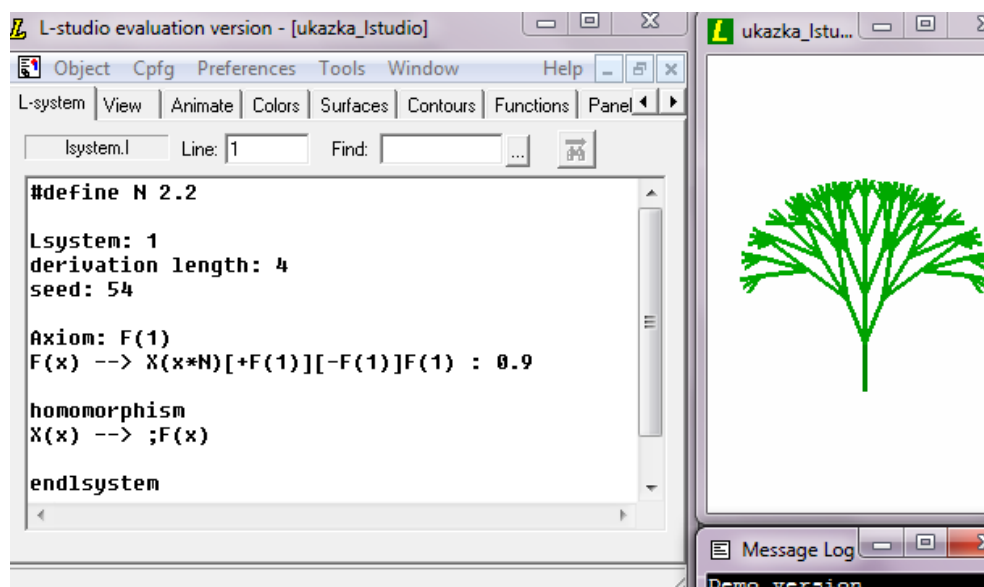
Program L-studio byl vybrán pro jeho komplexnost a velké množství možností pro vytváření rostlin pomocí L-systémů. Dalším důvodem pak byla shoda spoluautora tohoto programu a knížky, která je pro tuto práci hlavním zdrojem informací o L-systémech [1].

### 3.4.1 Základ práce s L-studiem

Pro práci s tímto programem je k jeho distribuci přiložena rozsáhlá dokumentace, kterou je vhodné si pročíst pro lepší seznámení s programem. Pro základní funkčnost ale stačí znát jenom malou část.

Pro začátek je vhodné prohlédnout si ukázkové příklady L-systémů, které jsou přiložené k programu ve složce oofs. Po prostudování ukázek je samozřejmě možné vytvořit vlastní L-systém od začátku. Ať už si jen prohlížíme ukázky, nebo vytváříme vlastní L-systém, potřebujeme především následující 4 záložky programu : L-System, View, Animate, a Colors, které budou popsány na následujících stránkách.

**L-system** - Zobrazení a úprava zdrojového textu vlastního L-systemu. Na obrázku 3.6 je zobrazena záložka L-system, společně s ukázkou grafického výstupu v okně napravo. V prostoru této záložky je zapsán vlastní předpis, podle kterého se bude vykreslovat. V tabulce 3.1 jsou pak vysvětleny některé často používané části zdrojového textu.

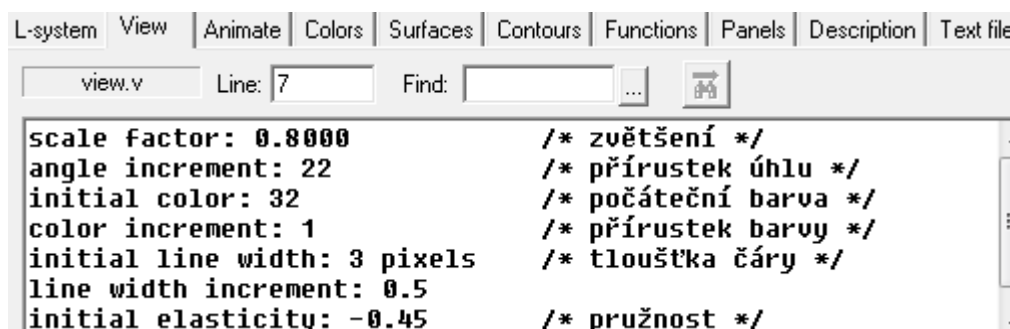


Obrázek 3.6: Záložka L-system programu L-studio a grafický výstup zapsané gramatiky

text	význam
<b>#define N 2.2</b>	Definuje konstantu N na hodnotu 2.2.
<b>Lsystem: 1</b>	Začátek zápisu L-systemu.
<b>derivation length: 4</b>	Nastaví délku derivace (počet iterací) na hodnotu 4.
<b>seed: 54</b>	Semínko náhodnosti potřebné pro funkčnost pravděpodobností použití pravidel.
<b>Axiom: F(1)</b>	Nastaví axiom na F(1).
<b>F(x) --&gt; X(x*N)[+F(1)]...</b>	Přepisovací pravidlo.
<b>homomorphism</b>	Blok pravidel udávající jak se ve výsledku interpretují symboly, které nejsou definovány programem (např. symbol X se vykreslí jako F).
<b>endlsystem</b>	Konec zápisu L-systemu.

Tabulka 3.1: Příklad a význam zdrojového textu

**View** - Zobrazení a úprava vlastností zobrazení L-systému. Tuto záložku (obrázek 3.7) tvoří také pouze prostor pro zápis textu, narozdíl od záložky L-system, zde nezapisujeme přepisovací pravidla, ale informace o tom, jak se má vytvořený řetězec zobrazit. Příklady nastavovaných hodnot jsou v tabulce 3.2.

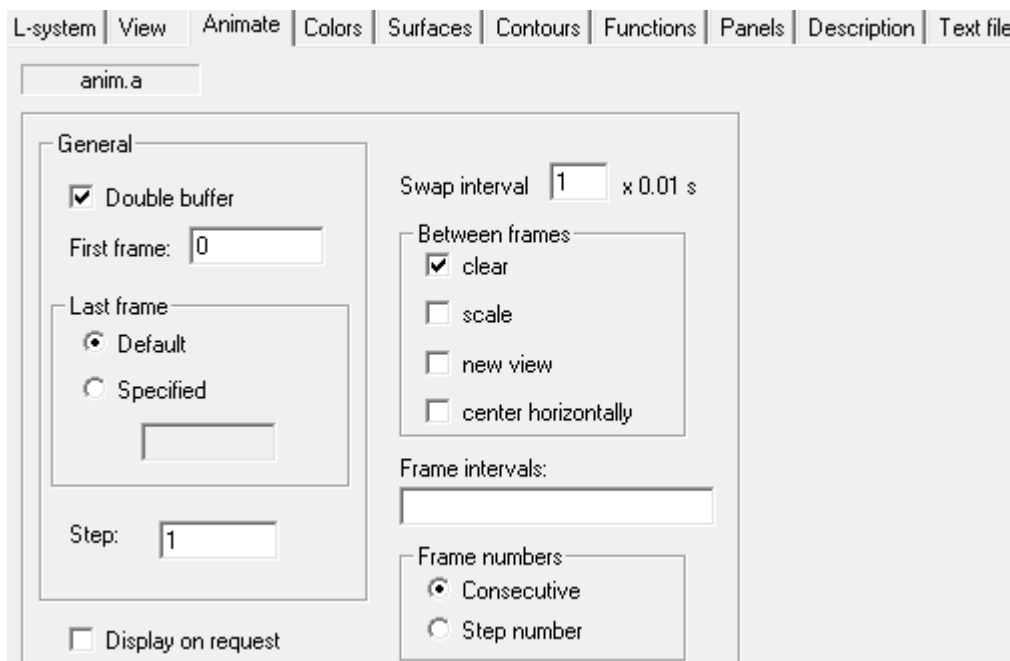


Obrázek 3.7: Záložka View programu L-studio

text	význam
<b>scale factor: 0.8000</b>	Zvětšovací faktor, udává jak velká se zobrazí vykreslená gramatika v okně vizualizace.
<b>angle increment: 22</b>	Nastaví konstantu zvětšování úhlu na 22.
<b>initial color: 32</b>	Nastaví počáteční barvu na číslo 32 (přiřazeno v záložce Colors).
<b>color increment: 1</b>	Posun v paletě barev.
<b>initial line width: 3 pixels</b>	Výchozí tloušťka čáry.
<b>line width increment: 0.5</b>	Násobitel šířky čáry pro postupnou změnu šířky čar v jednotlivých iteracích.
<b>initial elasticity: -0.45</b>	Pružnost čar, udávající jak moc se budou vytvořené větve ohýbat dolů (záporná hodnota) nebo nahoru (kladná hodnota).

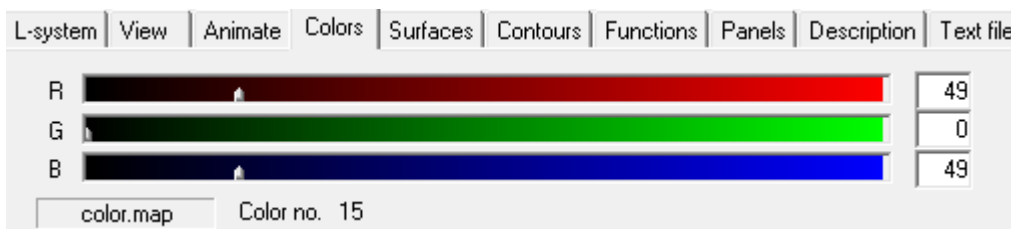
Tabulka 3.2: Příklad a význam zápisu v záložce View

**Animate** - Vlastnosti animace vykreslování. V této záložce (obrázek 3.8) se nastavuje například rychlost střídání snímků (swap interval) nebo možnost vyčištění obrazovky při kreslení další iterace (Between frames - clear).



Obrázek 3.8: Záložka Animate programu L-studio

**Colors** - Nastavení barev. Zde (obrázek 3.9) lze pomocí grafického rozhraní plynule nastavovat různé barvy. Tyto barvy se zařazují pod různá čísla do používané palety barev. Čísla jednotlivých barev v paletě se pak dají v záložce View použít pro nastavení počáteční barvy (initial color), nebo se dají používat i přímo v záložce L-system, jako parametr při změně barvy symbolem ”,(číslo barvy)” nebo ”;(číslo barvy)”.



Obrázek 3.9: Nastavování barvy v záložce Colors programu L-studio

## 4 Skupiny dřevin

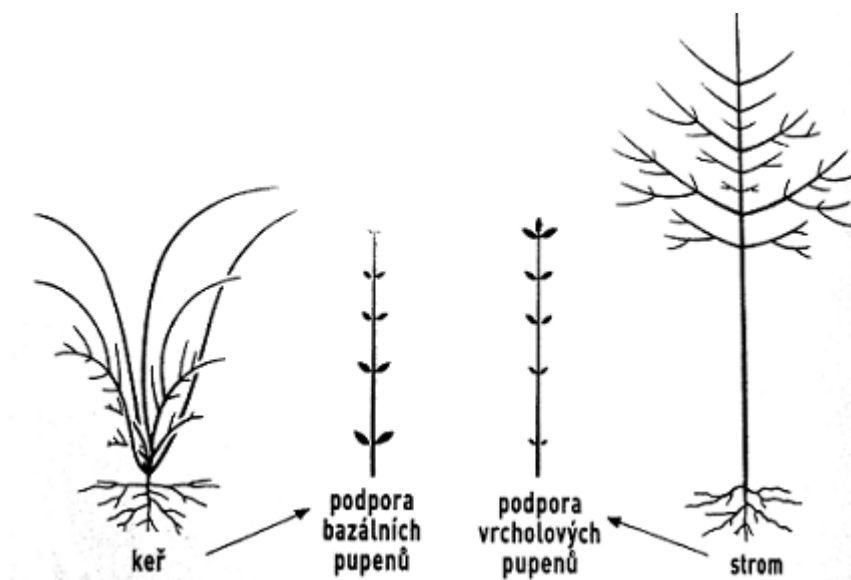
### 4.1 Základní rozdělení

Dřeviny můžeme rozlišovat podle více kategorií, například podle vzhledu kůry či tvaru listů. Tato dělení ale nemají příliš smysl pro rozdělení dřevin z hlediska jejich generování pomocí L-systémů. Proto byla z dostupné literatury vybrána jen některá dělení, která jsou dále popsána v této kapitole.

Rozdělení dřevin podle růstového tvaru dle [3] :

**Stromy** - Většinou mají jeden hlavní kmen, který se rozvětňuje v určité vzdálenosti od země a tvoří na kmeni korunu. Zpravidla dosahují výšky větší než 5 metrů. K tvorbě kmene a koruny dochází proto, že pupeny při vrcholu stonků rostou s větší podporou než pupeny při bázi (4.1).

**Keře** - Nemají jeden hlavní kmen, rozvětvuji se již od země, a dosahují menší výšky než stromy. Narozdíl od stromů rostou u keřů nejsilněji nejspodnější pupeny a směrem vzhůru se postupně zmenšují (4.1).



Obrázek 4.1: Růstové tvary dřevin [3]



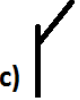
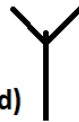
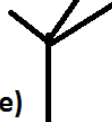
Rozdělení mezi stromy a keře podle výšky ovšem není úplně přesné, jelikož jak moc rostlina vyrostе závisí na prostředí v kterém roste. V některých oblastech tedy můžou být keře vzrostlé až tak, že vypadají jako stromy. V této práci bude tedy zanedbáno výškové kritérium pro stromy.

Pro konkrétnější rozdělení bylo vybráno rozdělení z botanického hlediska podle větvení stonku rostliny [2], a následně pro upřesnění podle výsledného tvaru koruny, neboli habitu (jak uvádí [2], habitus je charakteristickým vzhledem rostliny, který získá postupným rozvětčováním stonku a dalším vlivem okolního prostředí).

## 4.2 Rozdělení dle větvení stonku

Typy větvení stonku jsou obecné pro všechny rostliny mající stonek. U stromů je stonkem kmen, z kterého vyrůstají jednotlivé větve, díky tomu lze stromy rozlišovat podle způsobu větvení.

Na následujícím obrázku 4.2 jsou vypsány a zobrazeny všechny druhy větvení stonku pro úplnost, přestože se u dřevin vyskytují jen některé. U dřevin nás bude zajímat pouze postranní větvení, rozdělené na monopodiální a sympodiální.

<b>Vidličnaté</b>	 a)			
<b>Postranní</b>	<b>Monopodiální</b>	<b>Sympodiální</b>		
	 b)	<b>monocházium</b>  c)	<b>dicházium</b>  d)	<b>pleiocházium</b>  e)

Obrázek 4.2: Větvení stonku

**Vidličnaté větvení** (hemiblastické, dichotomické)

- Vlastnosti: Hlavní stoněk se rozděluje na dvě víceméně shodné části, nemá hlavní stoněk a místo větvení nezávisí na umístění listu. Jedná se o vývojově nejstarší typ větvení, u vyšších rostlin jako jsou stromy a keře se nevyskytuje (obr. 4.2 a) .

**Postranní větvení** (holoblastické, axilární)

- Vlastnosti: boční větev vzniká z dominantní větve, boční větve jsou ve vztahu k listu.
- Rozdělení podle dominance hlavního stonku nebo boční větve :

**monopodiální větvení** (monopodium jednonoží)

- Vlastnosti: Hlavní stoněk dominuje a boční větve jej nepřerůstají (obr. 4.2 b) .
- Příklady : smrk, dub, jasan.

**sympodiální větvení** (sympodium, sounoží)

- Vlastnosti: Boční větve zatlačují hlavní stoněk do strany.
- Rozděluje se na:

**monocházium** (obr.4.2c) v místě větvení vyrůstá 1 větev)

**dicházium** (obr.4.2d) vstřícně postavené 2 postranní větve)

**pleiocházium** (obr.4.2e) přerůstání hlavní větve třemi a více bočními větvemi)

**pseudichotomie** (obr.4.2f) u dichaziálního větvení hlavní vrchol zaniká, např. šerák, řešetlák)

- Příklady : lípa, dub, habr, jilm, kaštanovník jedlý.

Pokud se podíváme na kterýkoliv strom, zjistíme, že na jednom stromu se může vyskytovat více typů sympodiálního větvení. Z toho důvodu je bráno pro vytváření základních gramatik pouze rozdělení na monopodiální a sympodiální větvení. Další způsoby sympodiálního větvení jsou brány jako alternativy v přepisování symbolů (stochastická pravidla).



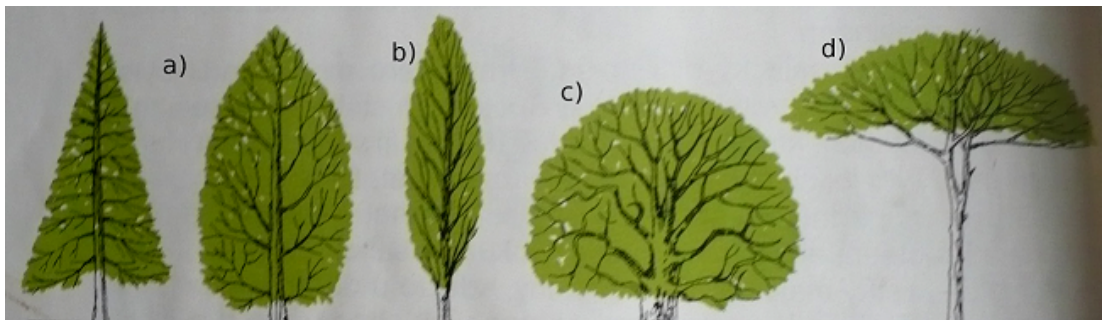
### 4.3 Rozdělení dle tvaru koruny

Již vzrostlé stromy můžeme rozlišovat také podle tvaru jejich koruny. Tyto tvary můžou být různě pojmenovány, v této práci bude použito dělení a názvy z [4].

Můžeme rozlišovat čtyři tvary korun stromů:

- a) pyramidální,
- b) sloupovitý,
- c) kulovitý,
- d) deštníkovitý.

Tyto tvary s písmenným označením jsou zobrazeny na obrázku 4.3.

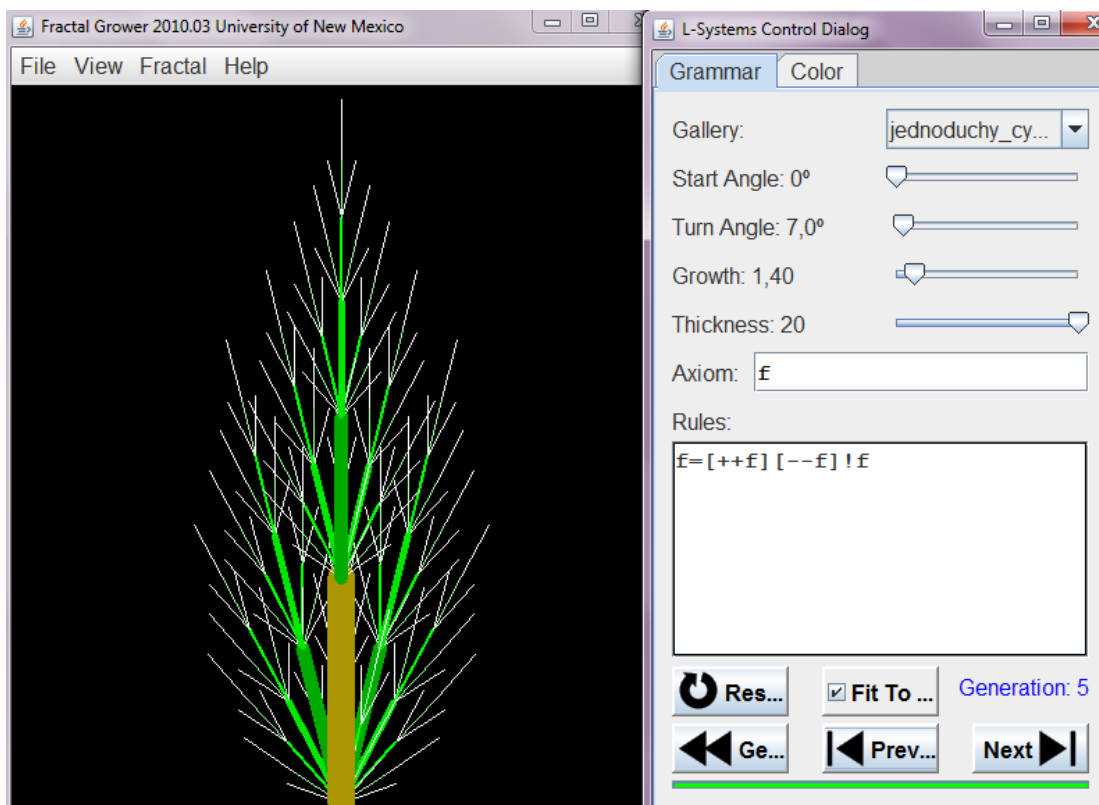


Obrázek 4.3: Habity korun [4]

## 5 Návrh gramatik

První pokusy o návrh gramatiky pro generování jednodušších stromů byly bez-parametrické a nestochastické, výsledné dřeviny tedy vypadaly nepřirozeně symetricky. Tyto první pokusy byly nejprve vytvářeny v programu Fractal Grower (například cypřiš na obrázku 5.1). Vzhledem k omezeným možnostem tohoto programu byly následně vybrány některé více povedené vytvořené gramatiky pro převod do programu L-studio, kde se dalo využít více možností.

Původní gramatiky vytvářené v programu Fractal Grower ve 2D, a jejich následné pokusy o převod do 3D v programu L-Studio jsou na příloženém CD ve složce "Gramatiky\_Projekt5".

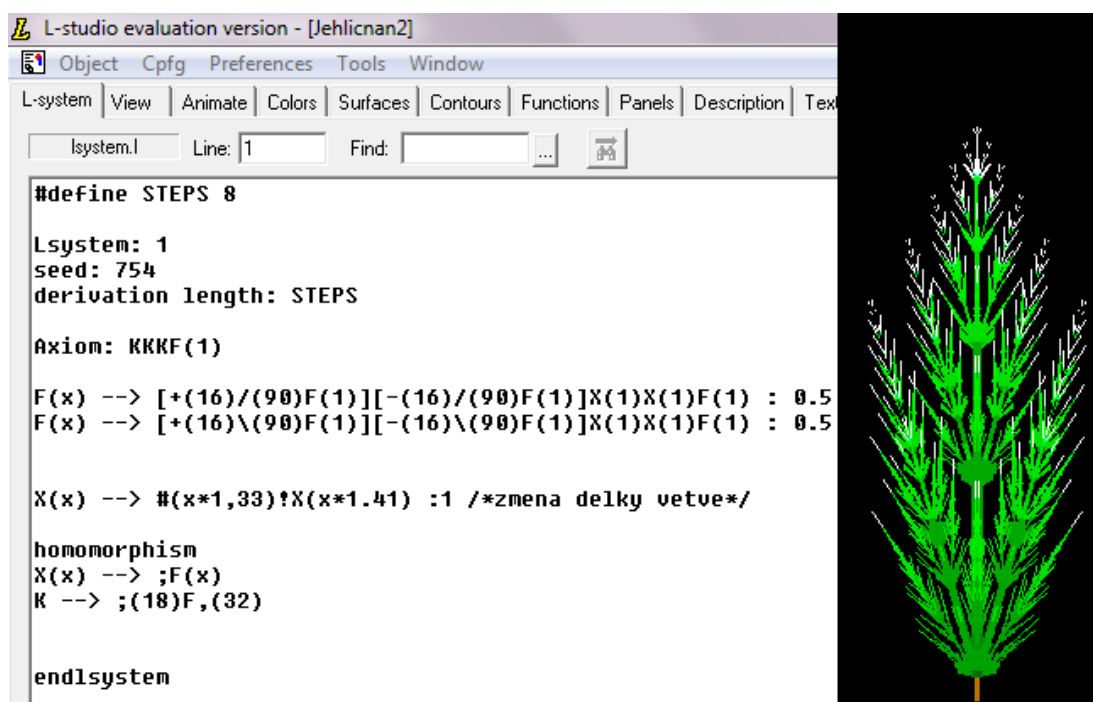


Obrázek 5.1: Vytvořená gramatika pro cypřiš v programu Fractal Grower

## 5.1 Přechod z Fractal Grower na L-studio

Při převádění gramatik do programu L-studio byly gramatiky změněné na parametrické a některé na stochastické. U některých vytvářených dřevin se ale ukázalo, že vypadají lépe bez použití náhodného vykreslování některých větví. Především u modelu podobnému cypřiši, nebo tůji. Vytvořený cypřiš totiž v případě použití náhody přišel o příliš mnoho větví a přestal mít typický tvar tůje.

Samotný převod bohužel nebyl úplně triviální, vzhledem k mírně odlišnému značení v programu Fractal Grower (čáry se vykreslují pomocí malých písmen "f" a zkrácené čáry vzhledem k hloubce zanoření pomocí symbolu "!"), a také zjednodušením v tomto programu (plynulou změnou některých vlastností, která se ihned projevovala na zobrazeném modelu, pomocí uživatelského rozhraní, což v programu L-studio nelze). Jedna z převedených gramatik do L-studia je na obrázku 5.2, přičemž původní v programu Fractal Grower je na obrázku 5.1.



Obrázek 5.2: Vytvořená gramatika pro cypřiš v programu L-Studio

## 5.2 Vytváření gramatik

Ze začátku bylo vytvořeno mnoho dále nepoužitelných gramatik, vzhledem k seznamování se se samotnými L-systémy a tvorbou gramatik. K prvním použitelným gramatikám jsem došla postupně převážně metodou "pokus omyl". Nejprve jsem zkoušela napsat jednoduchý předpis pro jednoduchou větvící se strukturu. Dále jsem předpis upravovala přidáváním různých symbolů (tedy změnami úhlu větvení, přidáváním větví a změnou jejich délky), a také přidáváním dalších přepisovacích pravidel a omezení.

Postupným vytvářením a upravováním gramatik v programu L-studio se přidávalo využívání různých možností L-systémů. Od jednoduchých gramatik ve 2D se postupně přešlo na 3D. Také byla využita možnost změny barvy, délky a tloušťky čar.

## 5.3 Společné vlastnosti gramatik

Základem pro tvorbu gramatik bylo určení potřebných symbolů a parametrů, jejich definování, a následné úpravy jejich hodnot. Většina dále uvedených předpisů má tedy společné následující :

### Přepisované symboly :

Následující přepisované symboly určují způsob větvení.

- K, k - symboly pro kmen
- P, Q - symboly pro pupeny z kterých budou vyrůstat větve, pokud jsou použity oba, liší se ve směru úhlu větvení
- V - symbol pro větev
- L - symbol pro list

### Konstanty :

V průběhu přepisování se nemění, mohou se ale použít například jako násobitelé níže uvedených parametrů. Násobitelé zajišťují postupné změny hodnot parametrů v každé iteraci, vynásobením s jejich aktuální hodnotou. Nastavováním různých hodnot konstant se může měnit tvar vznikajícího stromu.

- D1 - počáteční délka větve nebo kmene

- DNK - násobitel délky kmene
- DNV - násobitel délky větve
- B1 - číslo počáteční barvy určené v záložce Colors
- T1 - počáteční tloušťka kmene a větve
- TN - násobitel tloušťky pouze pro větve, nebo pro větve a kmen dohromady
- TNK - násobitel tloušťky kmene
- uhel1 - úhel prvního větvení od kmene
- uhel2 - úhel větvení - natočení želvy do stran, nebo i naklonění
- uhel3 - úhel větvení - naklonění želvy do stran
- NU2, NU3 - násobitelé úhlů
- listy - určuje mají-li se vykreslit jednoduché listy ( 1 = vykresli )

### Parametry :

Parametry jsou zadávané v závorkách za přepisovanými symboly. Mohou se měnit v průběhu přepisování. V gramatikách využívajících parametrické L-systémy se využívají také pro zadávání podmínek, za kterých bude použito přepisovací pravidlo (například je v předpisech v následující kapitole použita podmínka pro prodlužování větví jen dokud nedosáhnou určité délky).

- p - délka větve nebo kmene
- b - barva větve nebo kmene
- t - tloušťka větve nebo kmene
- vek - věk, nebo-li kolikrát byl symbol přepsán
- u,u2,u3 - úhly použité při větvení
- vzd - vzdálenost pupenu od kmene, v předpisech je použito pro zajištění maximální vzdálenosti větvení

V následujícím úseku kódu (5.1), jsou dále uvedeny části kódu společné pro většinu vytvořených gramatik v programu L-studio, a bez kterých by nešlo z gramatik vytvořit grafický výstup. Některé části ve výpisu se sice mění, nemají ale zásadní význam.

V části "konečné vykreslení symbolů" (začíná na řádce 18 ve výpisu 5.1) jsou uvedeny čísla, která nejsou mezi konstantami, tato čísla jsou použité barvy uvedené v záložce Colors.

```

1  /* definovani konstant */
2  #define D1 0.1 /* pocatecni delka */
3  #define B1 49 /* pocatecni barva */
4  #define T1 0.01 /* pocatecni tloustka (kmene a vetvi) */
5  ...
6  #define listy 0 /* listy.... 0 = ne , 1 = ano */
7  #define dcl 0.05 /* delka cary listu */
8
9  Lsystem: 1
10 derivation length: 16 /* pocet iteraci */
11 seed: 54
12
13 Axiom: K(1,D1,T1) /* koren/kmen */
14 /* prepisovaci pravidla : */
15 K(vek ,p, t) —> K(vek+1,p, t)
16 ...
17
18 /* konecne vykresleni symbolu */
19 homomorphism
20 V(p,b,t) : b < 56 —> ,(b)!(t)F(p) /* vetve se vykresli */
21 V(p,b,t) : b >= 56 —> ,(64)!(t)F(p) /* vetve se vykresli */
22 K(vek ,p, t) —> ,(64)!(t)F(p) /* kmen se vykresli */
23 k(vek ,p, t) —> ,(64)!(t)F(p) /* kmen se vykresli */
24 /* listy */
25 L : listy == 1 —> [, (49) { - f ( dcl) + f ( dcl) + f ( dcl) - | - f ( dcl) + f ( dcl) + f ( dcl) } ]
26
27 endlssystem

```

Listing 5.1: Opakující se části předpisů v programu L-studio

Další důležitou vlastností je hodnota "initial elasticity" v záložce "View" (výpis 5.2). Tento parametr určuje pružnost, pokud je jeho hodnota záporná, simuluje gravitaci a větve se ohýbají směrem k zemi. Naopak pokud je hodnota kladná, budou se větve ohýbat směrem nahoru. Změnou této vlastnosti lze tedy například snadno udělat z pyramidálního tvaru tvar sloupovitý.

```

1 scale factor: 0.8000 /* zvetsovaci faktor */
2 line style: cylinder /* vykreslovani cylindru misto car */
3 initial elasticity: 0.0 /* pruznost */

```

Listing 5.2: Důležité hodnoty v záložce View

V následující kapitole budou dále vypisovány jen ty části zdrojových textů z programu L-Studio, které se mění.

## 6 Vytvořené modely a jejich porovnání

Pro skupiny dřevin, popsanych v kapitole 4, bylo vytvořeno několik gramatik, které budou uvedeny dále. Následující skupiny se týkají pouze stromů. Příklad gramatiky pro keře je uveden až jako poslední.

Je třeba také zmínit že v uvedených předpisech se vyskytují čísla, která by mohla být zařazena mezi konstanty, ale pro jejich menší výskyt, nebo jednodušší zápis, jsou uvedena přímo v předpisech. Jedná se o hodnoty omezující rozrůstání dřevin. Jednou z těchto hodnot je maximální délka do které se prodlužují větve (např. řádka 16 ve výpisu 6.1, kde je dána podmínka prodloužení větve pouze pokud je  $p < 0.9$ ). Další využití je u nastavení maximální hodnoty vzdálenosti "vzd" u pupenů, kdy je ještě možné použít přepisovací pravidlo (např. řádka 7 ve výpisu 6.1).

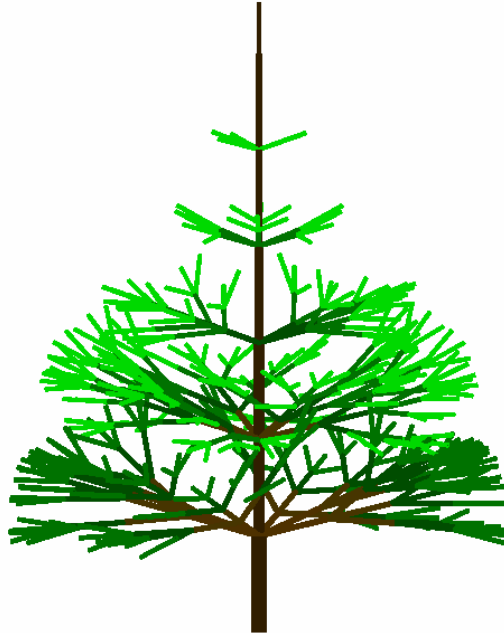
### 6.1 Rozdělení dle způsobu větvení

Jako základní rozdělení vytvářených předpisů bylo vybráno rozdělení podle způsobu větvení. Takto vznikly dva základní předpisy, od kterých se odvozovaly další.

#### 6.1.1 Monopodiální větvení

U monopodiálního větvení je třeba zajistit, aby měl strom jeden výrazný kmen a ostatní větve nepřesahovali jeho výšku. Typickými představiteli tohoto typu větvení jsou jehličnany, jako je například smrk. Z toho důvodu je jako první monopodiální gramatika vytvořen předpis, jehož výsledkem je strom podobný smrku (viz obr 6.1 ).

Pro zajištění nepřerůstání kmene větvemi je násobitel délky kmene ( $DNK=1.9$ ) o hodně větší než u větví ( $DNV=1.1$ ). V předpisu 6.1 je zajištěno neustálé prodlužování kmene a vyrůstání větví z kmene pod úhlem 70 stupňů. Pro každý pupen (P a Q) je určeno podobné přepisovací pravidlo, které zajišťuje růst větví do stran a dopředu, a zároveň vznik nových pupenů na koncích těchto větví. Z pupenů vyrůstají větve pouze do určené vzdálenosti od kmene ( $vzd < 5$ ), pro omezení šířky stromu. Jako další prvek tvarování je při každém přepsání pupenu zmenšen úhel větvení ( $u = u*0.8$ ).



Obrázek 6.1: Výsledek gramatiky pro monopodiální větvení

```

1 Axiom: K(1,D1,T1) /* koren/kmen */
2 /* kmen */
3 K(vek , p , t)
4     —> k(vek+1,p*DNK, t*TNK)[^( uhel1 )P(D1, uhel2 ,0)L]
5     [^( uhel1)Q(D1, uhel2 ,0)L]/(100)K(vek , p , t) : 1
6 /* vyrustani vetvi z pupenu */
7 P(p,u,vzd)L : {u=u*NU2; vzd=vzd+1;} vzd < 5
8     —> [-(u)\(u)V(p,B1,T1)P(D1,u,vzd)L]
9     [+(u)/(u)V(p,B1,T1)P(D1,u,vzd)L]
10    V(p,B1,T1)P(0.1,u,vzd)L : 1 /* pupen */
11 Q(p,u,vzd)L : {u=u*NU2; vzd=vzd+1;} vzd < 5
12    —> [-(u)/(u)V(p,B1,T1)Q(D1,u,vzd)L]
13    [+(u)\(u)V(p,B1,T1)Q(0.1,u,vzd)L]
14    V(p,B1,T1)Q(D1,u,vzd)L : 1 /* pupen2 */
15 /* preklesovani - prodruzovani vetvi a kmene*/
16 V(p,b,t) : p < 0.9 —> V(p*DNV,b+1,t*TN) : 1
17 k(vek , p , t) —> k(vek , p , t*TNK) : 1

```

Listing 6.1: Gramatika pro monopodiální větvení



```

1 #define DNK 1.9 /* nasobitel delky kmene */
2 #define DNV 1.1 /* nasobitel delky vetve */
3 #define TN 1.07 /* nasobitel tloustky */
4 #define TNK 1.2 /* nasobitel tloustky kmene */
5 #define uhel1 70 /* uhel vetveni od kmene */
6 #define uhel2 55 /* uhel vetveni */
7 #define NU2 0.8 /* nasobitel uhlu vetveni */

```

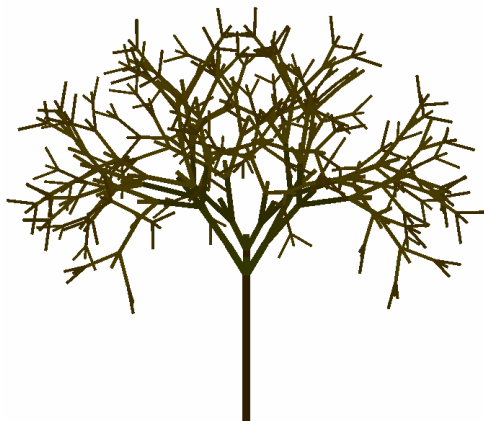
Listing 6.2: Konstanty pro monopodiální větvení

Základ v monopodiálním větvení mají dále následující tvary korun : pyramidální, sloupovitý a kulovitý.

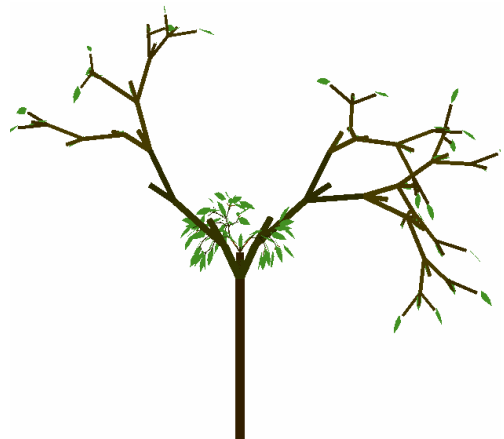
### 6.1.2 Sympodiální větvení

V případě sympodiálního větvení mohou vedlejší větve přesahovat hlavní kmen, který tímto ztrácí svoji výraznost a postupně zaniká. Způsobů sympodiálního větvení je více (viz kapitola 4), odlišnosti těchto způsobů ale spočívají pouze v počtu větví vznikajících ve větvicím místě. Z toho důvodu je pro sympodiální větvení uveden jeden předpis (6.3) s alternativními možnostmi větvení.

Na obrázku 6.2 je zobrazen výsledek gramatiky s menším počtem alternativních prepisovacích pravidel (bez pravidel na řádcích 18 až 25 ve výpisu 6.3. Na obrázku 6.3 je pak příklad výstupu s větším počtem alternativ.



Obrázek 6.2: Výsledek gramatiky pro sympodiální větvení



Obrázek 6.3: Výsledek gramatiky pro sympodiální větvení s přidáním alternativami větvení

Jako základní pravidlo (řádka 8 až 9 ve výpisu 6.3 pro pupen P a řádky 13 až 15 pro pupen Q) je dáno přepsání pupenu tak, aby z něj vyrostly dvě větve s novými pupeny do stran, a dopředu pokračovala krátká větev poloviční počáteční velikosti. Ostatní pravidla jsou pak alternativy pro větší různorodost větvení.

V počátečním axiomu je kromě kořenu  $K(1, D1, T1)$  také první pupen, který začne větvení pod úhlem 40 stupňů, a zkrácené pokračování kmene  $k(1, D1/5, T1)$ , u kterého je v následujících přepisovacích pravidel možnost dalšího větvení (řádek 5 ve výpisu 6.3)

```

1 Axiom:  $K(1, D1, T1) / (u_{hel1}) [P(D1, 1, u_{hel2}, u_{hel3})L] k(1, D1/5, T1)$ 
2
3  $K(v_{ek}, p, t) \longrightarrow K(v_{ek}+1, p*DNK, t*TN) : 0.7$ 
4  $k(v_{ek}, p, t) \longrightarrow k(v_{ek}+1, p*DNK, t*TN) : 0.7$ 
5  $k(v_{ek}, p, t)$ 
6  $\longrightarrow K(v_{ek}, p, t) / (u_{hel1}) [P(D1, 1, u_{hel2}, u_{hel3})L] k(1, p/5, T1) : 0.1$ 
7
8 /* vyrustani vetvi z pupenu */
9  $P(p, v_{zd}, u_2, u_3)L : \{u_2=u_2*NU_2; u_3=u_3*NU_3; v_{zd}=v_{zd}+1;\} v_{zd} < 9$ 
10  $\longrightarrow L[-(u_2) \setminus (u_3)V(p, B1, T1)Q(D1*2, v_{zd}, u_2, u_3)L]L$ 
11  $[(u_2)/(u_3)V(p, B1, T1)Q(D1*2, v_{zd}, u_2, u_3)L]LV(p/2, B1, T1) : 0.9$ 
12  $P(p, v_{zd}, u_2, u_3)L : v_{zd} < 9 \longrightarrow P(p, v_{zd}, u_2, u_3)L : 0.1$ 
13
14  $Q(p, v_{zd}, u_2, u_3)L : \{u_2=u_2*NU_2; u_3=u_3*NU_3; v_{zd}=v_{zd}+1;\} v_{zd} < 9$ 
15  $\longrightarrow L[+(u_2) \setminus (u_3)V(p, B1, T1)P(D1*2, v_{zd}, u_2, u_3)L]L$ 
16  $[-(u_2)/(u_3)V(p, B1, T1)P(D1*2, v_{zd}, u_2, u_3)L]LV(p/2, B1, T1) : 0.9$ 
17  $Q(p, v_{zd}, u_2, u_3)L : v_{zd} < 9 \longrightarrow Q(p, v_{zd}, u_2, u_3)L : 0.1$ 
18
19  $P(p, v_{zd}, u_2, u_3)L : \{u_2=u_2*NU_2; u_3=u_3*NU_3; v_{zd}=v_{zd}+1;\} v_{zd} < 9$ 
20  $\longrightarrow L[+(u_2)/(u_3)V(p, B1, T1)Q(D1*2, v_{zd}, u_2, u_3)L]LV(p/2, B1, T1) : 0.2$ 
21  $Q(p, v_{zd}, u_2, u_3)L : \{u_2=u_2*NU_2; u_3=u_3*NU_3; v_{zd}=v_{zd}+1;\} v_{zd} < 9$ 
22  $\longrightarrow L[-(u_2)/(u_3)V(p, B1, T1)P(D1*2, v_{zd}, u_2, u_3)L]LV(p/2, B1, T1) : 0.2$ 
23  $P(p, v_{zd}, u_2, u_3)L : \{u_2=u_2*NU_2; u_3=u_3*NU_3; v_{zd}=v_{zd}+1;\} v_{zd} < 9$ 
24  $\longrightarrow L[-(u_2) \setminus (u_3)V(p, B1, T1)Q(D1*2, v_{zd}, u_2, u_3)L]LV(p/2, B1, T1) : 0.2$ 
25  $Q(p, v_{zd}, u_2, u_3)L : \{u_2=u_2*NU_2; u_3=u_3*NU_3; v_{zd}=v_{zd}+1;\} v_{zd} < 9$ 
26  $\longrightarrow L[+(u_2) \setminus (u_3)V(p, B1, T1)P(D1*2, v_{zd}, u_2, u_3)L]LV(p/2, B1, T1) : 0.2$ 
27
28 /* prekleslovani - prodluzovani vetvi */
29  $V(p, b, t) \longrightarrow V(p*DNV, b+1, t*TN) : 0.7$ 

```

Listing 6.3: Gramatika pro sympodiální větvení

```

1 #define DNK 1.2 /* nasobitel delky kmene */
2 #define DNV 1.15 /* nasobitel delky vetve */
3 #define TN 1.15 /* nasobitel tloustky */
4 #define uhel1 40 /* uhel vetveni od kmene */
5 #define uhel2 30 /* uhel vetveni doprava/doleva*/
6 #define NU2 1.05 /* nasobitel uhlu 2 */
7 #define uhel3 90 /* uhel vetveni - nakloneni*/
8 #define NU3 1.0 /* nasobitel uhlu 3 */

```

Listing 6.4: Konstanty pro sympodiální větvení

Základ v sympodiálním větvení mají dále následující tvary korun : kulovitý, deštníkovitý a nezařazený tvar vrby.

## 6.2 Rozdělení dle tvaru koruny

Přestože často mají koruny stromů neurčité tvary, lze je alespoň přibližně zařadit do určité skupiny tvarů. Stromy s různým typem větvení mají předpoklady pro různé výsledné tvary, proto nejsou vytvořeny pokaždé dvojice předpisů pro tvar a typ větvení.

### 6.2.1 Pyramidální tvar

Pyramidální tvar má strom, jehož gramatika je uvedena jako první pro monopodiální větvení. Pouhou změnou pravděpodobnosti použití prepisovacích pravidel pro P a Q z hodnoty 1 na 0.8, přidáním možnosti přepsání pupenu znovu na pupen s jinou hodnotou úhlu a vzdálenosti (uvedených v ukázce kódu níže 6.5), a zvýšením počtu iterací na 16 vznikne strom zobrazený na obrázku 6.4.

```

1 P(p, u, vzd)L : {u=u*NU2; vzd=vzd+1;} vzd < 5
2               —> P(p, u, vzd)L : 0.3
3 Q(p, u, vzd)L : {u=u*NU2; vzd=vzd+1;} vzd < 5
4               —> Q(p, u, vzd)L : 0.3

```

Listing 6.5: Přidaná pravidla pro monopodiální větvení

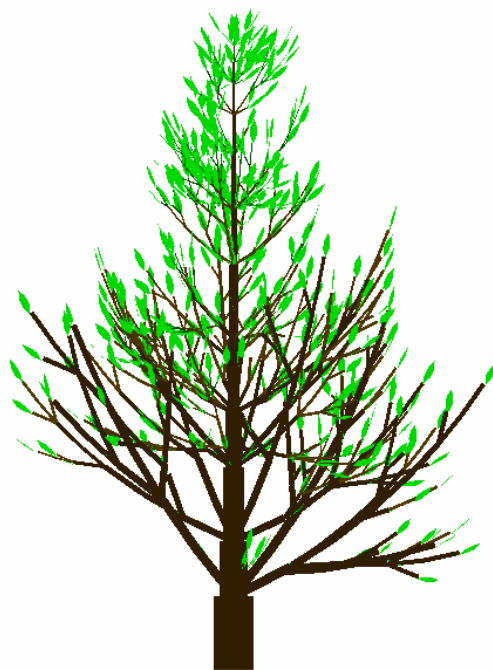
Další možný pyramidální tvar má listnatý strom na obrázku 6.5, který vzniknul rozsáhlejší úpravou gramatiky pro monopodiální větvení. Přepisovaný pupen byl použit jen jeden, a byly pro něj rozšířeny alternativy přepsání. Možnosti přepsání ve výpisu 6.7 na řádcích 9 až 20 jsou následující:

- 1) z pupenu vznikne list,
- 2) pupen zůstane pupenem,
- 3) vyroste jedna větev s pupenem doprava a dvojnásobná větev rovně,
- 4) vyroste jedna větev s pupenem doleva a dvojnásobná větev rovně,
- 5) vyrostou větve s pupeny doprava i doleva a dvojnásobná větev rovně.

K přepisovacím pravidlům pro kmen byla navíc přidána i možnost rozvětvení se pod původními větvemi (výpis 6.7, řádky 5 a 6).



Obrázek 6.4: Monopodiální jehličnan pyramidálního tvaru



Obrázek 6.5: Monopodiální listnatý strom pyramidálního tvaru

```

1 #define DNK 1.35 /* nasobitel delky kmene */
2 #define DNV 1.2  /* nasobitel delky vetve */
3 #define TN 1.25 /* nasobitel tloustky */
4 #define TNK 1.35 /* nasobitel tloustky kmene */
5 #define uhel1 45 /* uhel vetveni od kmene */
6 #define uhel2 66 /* uhel vetveni doprava/doleva*/
7 #define NU2 0.8 /* nasobitel uhlu 2 */
8 #define uhel3 90 /* uhel vetveni - nakloneni*/
9 #define NU3 0.8 /* nasobitel uhlu 3 */

```

Listing 6.6: Konstanty pro pyramidální listnatý strom

```

1 Axiom: K(1, D1, T1)
2 K(vek, p, t)  -> k(vek+1, p*DNK, t*TNK) [ ^ ( uhel1 ) P(D1, 0, uhel2, uhel3) L ]
3               [ & ( uhel1 ) P(D1, 0, uhel2, uhel3) L ] / ( 100 ) K(1, D1, T1) : 0.8
4 k(vek, p, t)  -> k(vek+1, p*DNK, t*TNK) [ ^ ( uhel1 ) P(D1, 0, uhel2, uhel3) L ]
5               [ & ( uhel1 ) P(D1, 0, uhel2, uhel3) L ] : 0.2
6
7 /* vyrustani vetvi z pupenu */
8 P(p, vzd, u2, u3) L : vzd < 5 -> L : 0.2
9 P(p, vzd, u2, u3) L : vzd < 5 -> P(p, vzd, u2, u3) L : 0.5
10 P(p, vzd, u2, u3) L : { u2=u2*NU2; u3=u3*NU3; vzd=vzd+1; } vzd < 5
11   -> [ -(u2) / (u3) V(p, B1, T1) LP(D1, vzd, u2, u3) L ]
12       V(p, B1, T1) V(p, B1, T1) P(D1, vzd, u2, u3) L ^ ( 30 ) : 0.7
13 P(p, vzd, u2, u3) L : { u2=u2*NU2; u3=u3*NU3; vzd=vzd+1; } vzd < 5
14   -> [ +(u2) \ (u3) V(p, B1, T1) LP(D1, vzd, u2, u3) L ]
15       V(p, B1, T1) V(p, B1, T1) P(D1, vzd, u2, u3) L ^ ( 30 ) : 0.7
16 P(p, vzd, u2, u3) L : { u2=u2*NU2; u3=u3*NU3; vzd=vzd+1; } vzd < 5
17   -> [ +(u2) / (u3) V(p, B1, T1) LP(D1, vzd, u2, u3) L ]
18       [ -(u2) \ (u3) V(p, B1, T1) LP(D1, vzd, u2, u3) L ]
19       V(p, B1, T1) V(p, B1, T1) P(D1, vzd, u2, u3) L ^ ( 30 ) : 0.3

```

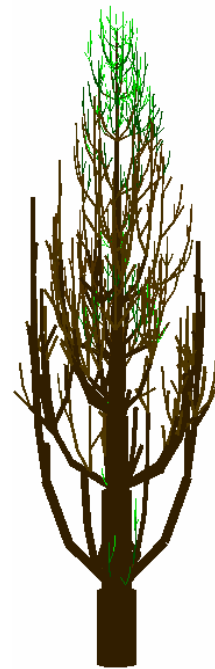
Listing 6.7: Gramatika pro pyramidální listnatý strom

## 6.2.2 Sloupovitý tvar

Sloupovitého tvaru dosáhneme snadno změnou hodnoty "initial elasticity" na hodnotu 0.6 u gramatiky pro listnatý strom pyramidálního tvaru, výsledkem je pak strom na obrázku 6.7. Ještě před existencí této gramatiky byla ale vytvořena jiná gramatika pro stromy sloupovitého tvaru, jejíž výstup je na obrázku 6.6. Liší se jak v nastavených konstantách (6.8), tak v prepisovacích pravidlech (6.9).



Obrázek 6.6: Monopodiální strom sloupovitého tvaru



Obrázek 6.7: Monopodiální listnatý strom sloupovitého tvaru, vzniklý úpravou gramatiky pro monopodiální listnatý strom

Základem pro tento tvar je znovu gramatika pro monopodiální větvení. V této gramatice (výpis 6.9) je už ale přidáno rozlišení úhlů pro natočení želvy do stran (uhel2) a naklánění želvy (uhel3), jejich zmenšování je ale zajištěno stejným násobitelem 0.8. Samotný předpis je ale stejný a sloupovitost zajišťuje především hodnota initial elasticity nastavená na 0.4, a změna konstant uvedených ve výpisu 6.8.

```

1 #define DNK 1.65 /* nasobitel delky kmene */
2 #define DNV 1.3 /* nasobitel delky vetve */
3 #define TN 1.3 /* nasobitel tloustky */
4 #define TNK 1.4 /* nasobitel tloustky kmene */
5 #define uhel1 30 /* uhel vetveni od kmene */
6 #define uhel2 36 /* uhel vetveni doprava/doleva*/
7 #define NU2 0.8 /* nasobitel uhlu 2 */
8 #define uhel3 90 /* uhel vetveni - nakloneni*/
9 #define NU3 0.8 /* nasobitel uhlu 3 */

```

Listing 6.8: Konstanty pro sloupovitý tvar

```

1 K(vek , p , t)
2   —> k(vek+1,p*DNK, t*TNK)[^( uhel1)P(D1,0 , uhel2 , uhel3)L]
3     [&( uhel1)Q(D1,0 , uhel2 , uhel3)L]/(100)K(vek , p , t) : 1
4 k(vek , p , t) : vek > 3
5   —> k(vek , p/2 , t)[^( uhel1)P(D1,0 , uhel2 , uhel3)L]
6     [&( uhel1)Q(D1,0 , uhel2 , uhel3)L]k(vek , p/2 , t) : 0.4
7
8 /* vyrustani vetvi z pupenu */
9 P(p, vzd , u2 , u3)L : {u2=u2*NU2; u3=u3*NU3; vzd=vzd+1;} vzd < 5
10  —> [(u2)/(u3)V(p, B1, T1)Q(D1, vzd , u2 , u3)L]
11     [-(u2)/(u3)V(p, B1, T1)P(D1, vzd , u2 , u3)L]V(p, B1, T1)V(p/2, B1, T1)
12     V(p, B1, T1)Q(D1, vzd , u2 , u3)L : 1
13 Q(p, vzd , u2 , u3)L : {u2=u2*NU2; u3=u3*NU3; vzd=vzd+1;} vzd < 5
14  —> [(u2)\(u3)V(p, B1, T1)P(D1, vzd , u2 , u3)L]
15     [-(u2)\(u3)V(p, B1, T1)Q(D1, vzd , u2 , u3)L]V(p, B1, T1)V(p/2, B1, T1)
16     V(p, B1, T1)P(D1, vzd , u2 , u3)L : 1
17
18 V(p, b , t) : p > 0.9
19  —> V(p/2, b , t)P(D1,0 , uhel2 , uhel3)LV(p/2, b , t) : 0.3
20 k(vek , p , t) : vek < 7 —> k(vek+1,p*DNK, t*TNK) : 1
21 k(vek , p , t) : vek > 6 —> k(vek+1,p, t*TNK) : 1

```

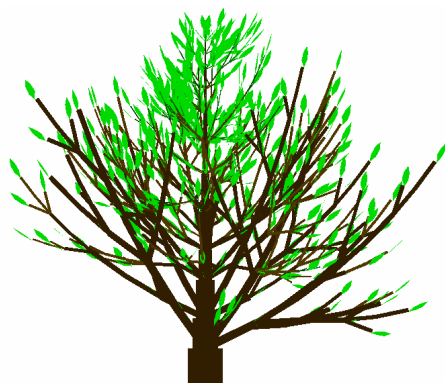
Listing 6.9: Gramatika pro sloupovitý tvar

### 6.2.3 Kulovitý tvar

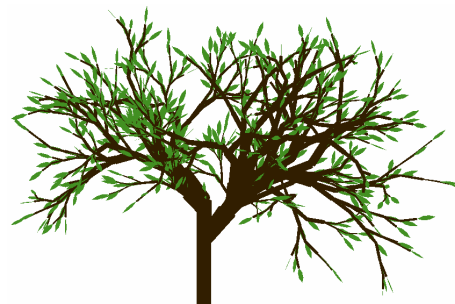
Kulovitý tvar lze získat jednoduchou úpravou gramatiky pro monopodiální listnatý strom pyramidálního tvaru (obr. 6.5), a to pouhou změnou násobitele délky kmene (DNK) na 1.25. Výsledkem úpravy je strom na obrázku 6.8.

V případě sympodiálního větvení má kulovitý tvar základní gramatika pro sympodiální větvení, s nastavenou hodnotou initial elasticity na -0.1. Dodržet přesný kulovitý tvar se ale příliš nepodařilo. Úpravami této gramatiky se přešlo na předpis, jehož konstanty (6.10) a gramatika (6.11) zajišťují vytváření realistějšího, ale více rozloženého stromu zobrazeného na obrázku 6.9.

V této vytvořené gramatice je pomocí uvedených konstant (výpis 6.10) zajištěna výrazná změna úhlu pro náklon větví, a naopak postupné zmenšování úhlu větvení do stran. V gramatice (výpis 6.11) je hlavní změnou oproti základní gramatice přidání vytvoření rovné větve mezi dvě větvící se větve do stran (ve výpisu 6.11 řádky 8 až 10 pro pupen P a řádky 16 až 18 pro pupen Q).



Obrázek 6.8: Monopodiální listnatý strom kulovitého tvaru



Obrázek 6.9: Sympodiální strom kulovitého tvaru

```

1 #define DNK 1.25 /* nasobitel delky kmene */
2 #define DNV 1.15 /* nasobitel delky vetve */
3 #define TN 1.3 /* nasobitel tloustky */
4 #define TNK 1.3 /* nasobitel tloustky kmene */
5 #define uhel1 20 /* uhel vetveni od kmene */
6 #define uhel2 50 /* uhel vetveni doprava/doleva*/
7 #define NU2 0.9 /* nasobitel uhlu 2 */
8 #define uhel3 60 /* uhel vetveni - nakloneni*/
9 #define NU3 1.9 /* nasobitel uhlu 3 */

```

Listing 6.10: Konstanty pro kulovitý tvar

```

1 Axiom :
2 K(1 ,D1 ,T1)/( uhel1 ) [P(D1,1 , uhel2 , uhel3 )L]/( uhel1 ) [Q(D1,1 , uhel2 , uhel3 )L]
3
4 K(vek , p , t)  —> K(vek , p*2 , t) [P(D1,1 , uhel2 , uhel3 )L]
5   [P(D1,1 , uhel2 , uhel3 )L]/( uhel1 )P(D1,1 , uhel2 , uhel3 ) : 0.02
6
7 /* vyrustani vetvi z pupenu */
8 P(p , vzd , u2 , u3)L : {u2=u2*NU2; u3=u3*NU3; vzd=vzd+1;} vzd < 9
9   —> L[-(u2)/(u3)V(p , B1 , T1)Q(p , vzd , u2 , u3)L]V(p , B1 , T1)L
10   [(u2)/(u3)V(p , B1 , T1)P(p , vzd , u2 , u3)L]L : 0.6
11 P(p , vzd , u2 , u3)L : {u2=u2*NU2; u3=u3*NU3; vzd=vzd+1;} vzd < 9
12   —> LV(p , B1 , T1)L[(u2)/(u3)V(p , B1 , T1)Q(p , vzd , u2 , u3)L]L : 0.1
13 P(p , vzd , u2 , u3)L : vzd < 9
14   —> Q(p/2 , vzd , u2 , u3)L : 0.2
15

```



```

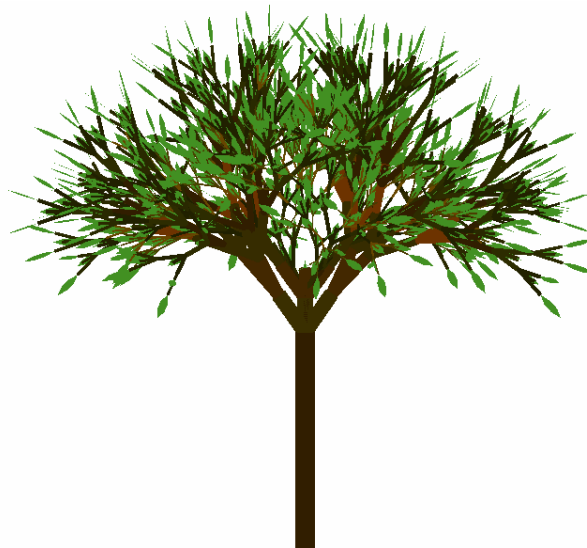
16 Q(p, vzd, u2, u3)L : {u2=u2*NU2; u3=u3*NU3; vzd=vzd+1;} vzd < 9
17   —> L[-(u2)\(u3)V(p, B1, T1)Q(p, vzd, u2, u3)L]V(p, B1, T1)L
18     [(u2)/(u3)V(p, B1, T1)P(p, vzd, u2, u3)L]L : 0.6
19 Q(p, vzd, u2, u3)L : {u2=u2*NU2; u3=u3*NU3; vzd=vzd+1;} vzd < 9
20   —> LV(p, B1, T1)L[-(u2)/(u3)V(p, B1, T1)P(p, vzd, u2, u3)L]L : 0.1
21 Q(p, vzd, u2, u3)L : vzd < 9
22   —> P(p*2, vzd, u2, u3)L : 0.2
23
24 /* preklesovani - prodluzovani vetvi a kmene */
25 V(p, b, t) : p < 1.5 —> V(p*DNV*DNV*DNV, b+1, t*TN) : 0.7
26 V(p, b, t) : p >= 1.5 —> V(p*DNV, b+1, t*TN) : 0.7
27 K(vek, p, t) —> K(vek+1, p*DNK, t*TNK) : 0.7

```

Listing 6.11: Gramatika pro kulovitý tvar

## 6.2.4 Deštníkovitý tvar

Přijatelný deštníkovitý tvar (6.10) získáme úpravou gramatiky pro sympodiální větvení. Pro získání požadovaného tvaru bylo třeba jen mírně pozměnit konstanty pro tuto gramatiku a initial elasticity nastavit na 0. Změny jsou uvedeny ve výpisu 6.12. Hlavním činitelem získání požadovaného tvaru je v tomto případě postupné zmenšování úhlu větvení do stran "uhel2", pomocí násobitele "NU2 = 0.9".



Obrázek 6.10: Sympodiální strom deštníkovitého tvaru

```

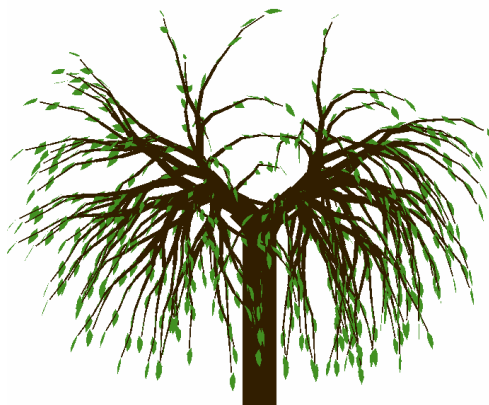
1 #define DNK 1.3 /* nasobitel delky kmene */
2 #define DNV 1.1 /* nasobitel delky vetve */
3 #define TN 1.25 /* nasobitel tloustky */
4 #define uhel1 20 /* uhel vetveni od kmene */
5 #define uhel2 40 /* uhel vetveni */
6 #define NU2 0.9 /* nasobitel uhlu 2 */
7 #define uhel3 90 /* uhel vetveni - do prostoru */
8 #define NU3 1.0 /* nasobitel uhlu 3 */

```

Listing 6.12: Konstanty pro deštníkovitý tvar

### 6.2.5 Nezařazený tvar vrby

Zkoušením různých úprav gramatik se sympodiálním větvením vznikl strom podobný vrbě, zobrazený na obrázku 6.11. Větve mířící k zemi jsou zajištěny hodnotou "initial elasticity" nastavenou na -0.2. Jejich jednoduchý dlouhý vzhled je pak zajištěn prepisovacím pravidlem na řádcích 10 až 13 ve výpisu gramatiky 6.14.



Obrázek 6.11: Sympodiální strom tvaru vrby

```

1 #define DNK 1.35 /* nasobitel delky kmene */
2 #define DNV 1.15 /* nasobitel delky vetve */
3 #define TN 1.35 /* nasobitel tloustky */
4 #define uhel1 20 /* uhel vetveni od kmene */
5 #define uhel2 50 /* uhel vetveni doprava/doleva*/
6 #define NU2 0.88 /* nasobitel uhlu 2 */
7 #define uhel3 60 /* uhel vetveni - nakloneni*/
8 #define NU3 1.6 /* nasobitel uhlu 3 */

```

Listing 6.13: Konstanty pro tvar vrby

```

1 Axiom: K(1,D1,T1)/(uhel1)[P(D1,1,uhel2,uhel3)L]
2 K(vek,p,t)  -> K(vek+1,p*DNK,t*TN): 0.7
3
4 /* vyrustani vetvi z pupenu */
5 P(p,vzd,u2,u3)L :
6     {u2=u2*NU2; u3=u3*NU3; p=p*vzd; vzd=vzd+1;} vzd < 9
7     -> L[-(u2)/(u3)V(p,B1,T1)P(D1,vzd,u2,u3)L]L
8     [(u2)/(u3)V(p,B1,T1)P(D1,vzd,u2,u3)L]L : 0.7
9 P(p,vzd,u2,u3)L :
10    {u2=u2*NU2; u3=u3*NU3; p=p*vzd; vzd=vzd+1;}
11    (vzd >= 9)&&(vzd < 13)
12    -> L[(u2)/(u3)V(p,B1,T1)P(D1,vzd,u2,u3)L]L : 0.3
13
14 /* prekleslovani - prodluzovani vetvi */
15 V(p,b,t) : p < 4.5  -> V(p*DNV*DNV*DNV,b+1,t*TN) : 0.7
16 V(p,b,t) : p >= 4.5 -> V(p,b+1,t*TN) : 0.7

```

Listing 6.14: Gramatika pro tvar vrby

## 6.3 Keře

Dalším typem dřevin jsou keře, které svým vzhledem často připomínají pouze koruny stromů bez kmene. Gramatika pro keř (gramatika 6.16, konstanty 6.15, výstup 6.12) byla tudíž vytvořena upravením gramatik pro stromy se sympodiálním větvením, tím způsobem, že bylo odstraněno vykreslení kmene a dále upravovány úhly větvení.



Obrázek 6.12: Vytvořený keř v programu L-Studio

```

1 #define DNK 1.4 /* nasobitel delky kmene */
2 #define DNV 1.35 /* nasobitel delky vetve */
3 #define TN 1.25 /* nasobitel tloustky */
4 #define uhel2 46 /* uhel vetveni doprava/doleva*/
5 #define NU2 0.9 /* nasobitel uhlu 2 */
6 #define uhel3 55 /* uhel vetveni - nakloneni*/
7 #define NU3 1.2 /* nasobitel uhlu 3 */

```

Listing 6.15: Konstanty pro keř

Aby nebyl keř symetrický, jsou pro přepsání pupenů dána 4 možná pravidla. Možnosti změny pupenu jsou tedy:

- 1) pupen zůstane pupenem,
- 2) začnou vyrůstat tři různé větve s novými pupeny,
- 3) začne vyrůstat jen jedna větev s novým pupenem a jeden pupen zůstane.

```

1 Axiom: K(1, D1, T1)
2 K(vek, p, t) : vek < 5
3 —> K(vek+1, p*DNK, t*TN)/(130)[+(10)P(D1, 0, uhel2, uhel3)L]: 0.3
4
5 /* vyrustani vetvi z pupenu */
6 P(p, vzd, u2, u3)L —> P(0.1, vzd, u2, u3)L : 0.3
7 P(p, vzd, u2, u3)L : {u2=u2*NU2; u3=u3*NU3; vzd=vzd+1;} vzd < 7
8 —> [-(u2)/(u3)V(p, B1, T1)P(D1, vzd, u2, u3)L]
9 [+(u2)/(u3)V(p, B1, T1)P(D1, vzd, u2, u3)L]
10 P(D1, vzd, u2, u3)LBV(0.01, B1, T1) : 0.3
11 P(p, vzd, u2, u3)L : {u2=u2*NU2; u3=u3*NU3; vzd=vzd+1;} vzd < 7
12 —> [+(u2)/(u3)V(p, B1, T1)P(D1, vzd, u2, u3)L]
13 P(D1, vzd, u2, u3)LB : 0.3
14 P(p, vzd, u2, u3)L : {u2=u2*NU2; u3=u3*NU3; vzd=vzd+1;} vzd < 7
15 —> [-(u2)/(u3)V(p, B1, T1)P(D1, vzd, u2, u3)L]
16 P(D1, vzd, u2, u3)LB : 0.3
17
18 V(p, b, t) : p > 0.9
19 —> V(p/2, b, t)P(0.1, 0, uhel2, uhel3)LV(p/2, b, t) : 0.3
20
21 /* prekleslovani - prodluzovani vetvi */
22 V(p, b, t) : p < 1.6 —> V(p*DNV, b+1, t*TN) : 0.7

```

Listing 6.16: Gramatika pro keř

## 6.4 Zhodnocení vytvořených gramatik

K vytvoření všech gramatik bylo využito možností parametrických a stochastických L-systémů. Díky nastavovaným parametrům pro délky větví a úhly větvení, lze měnit výsledný tvar vzrostlého stromu. Použití více alternativ prepisovacích symbolů pak zajišťuje různorodost větvení. Ani jednu gramatiku by ale nešlo vytvořit bez možnosti tvorby větví, kterou zajišťují závorkové L-systémy.

Při vytváření gramatik bylo zjištěno, že v některých případech stačí jen drobná změna, aby byl změněn výsledný tvar. Takovými změnami byla změna násobitele délky kmene u monopodiálního lisnatého stromu, pro změnu z pyramidálního tvaru na kulovitý tvar. Ze stejné gramatiky byl také změnou hodnoty initial elasticity získán sloupovitý tvar. V ostatních případech bylo třeba postupně upravovat více parametrů, přidávat a mírně upravovat prepisovací pravidla, a sledovat vliv jejich změn na výsledný tvar.

Grafické výstupy gramatik se podobají skutečným stromům, ale u některých je tato podobnost více vzdálená. Podobnost s realitou lze zhodnotit na obrázcích uvedených v následující kapitole.

## 6.5 Srovnání modelů a reality

Při tvorbě modelů stromů byly jako předloha použity různé stromy, které jsou obecně známy, a také takové stromy, jejichž vyobrazení lze nalézt v literatuře [3] [4]. Na následujících obrázcích jsou pro srovnání uvedeny grafické výstupy některých gramatik společně se skutečnými stromy.



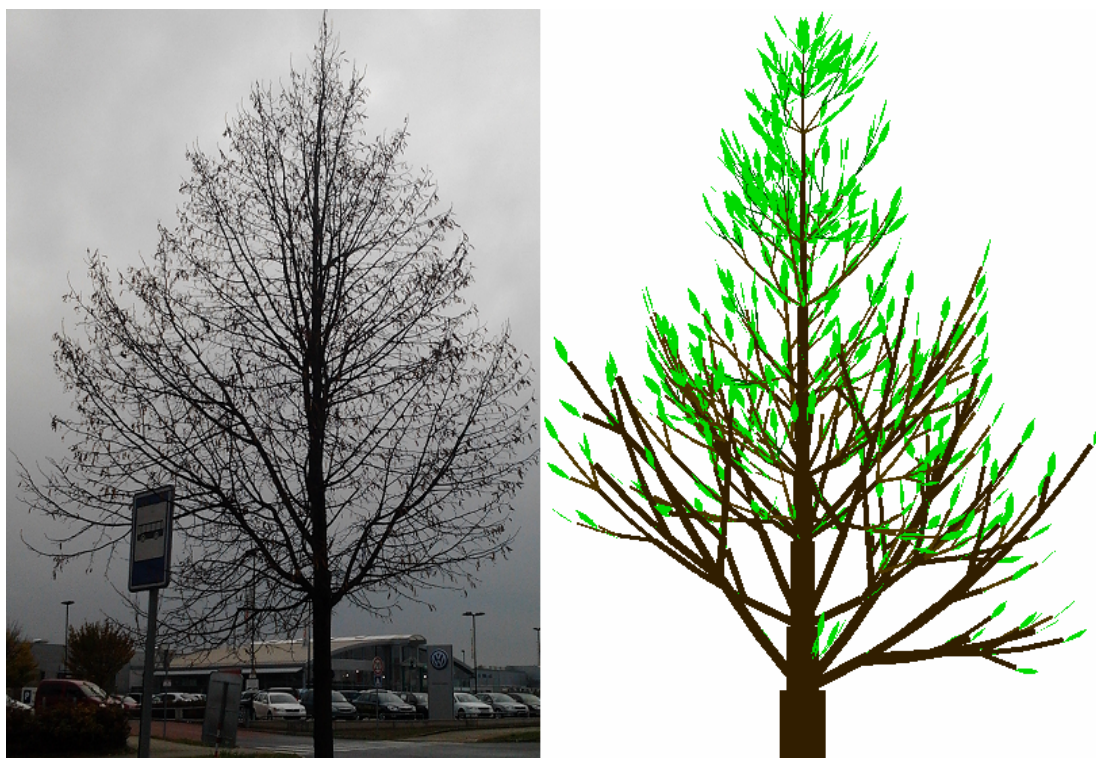
Obrázek 6.13: Srovnání monopodiálních stromů sloupovitého tvaru se skutečným stromem



Obrázek 6.14: Srovnání monopodiálního jehličnanu se skutečným stromem



Obrázek 6.15: Srovnání monopodiálního listnatého stromu s kulovitou korunou se skutečným stromem



Obrázek 6.16: Srovnání monopodiálního listnatého stromu s pyramidální korunou se skutečným stromem

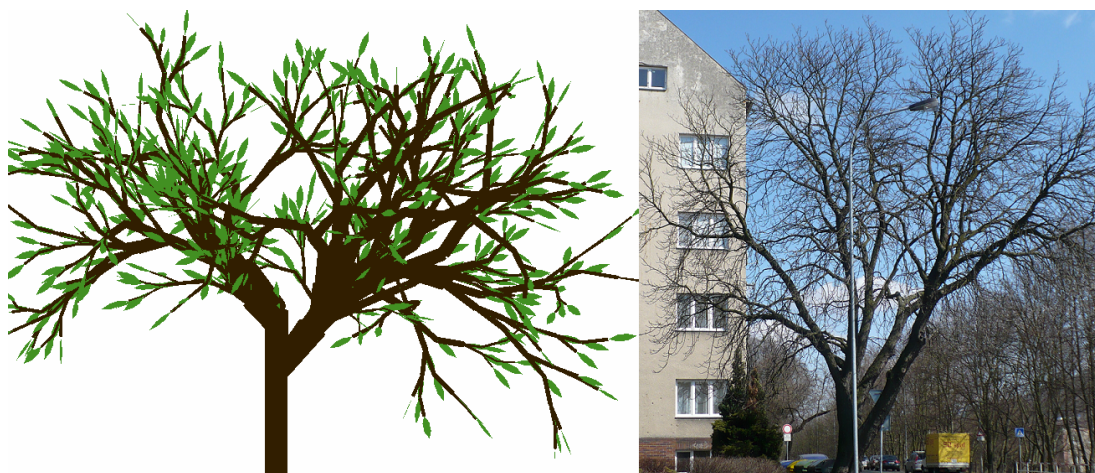


Obrázek 6.17: Srovnání keřů





Obrázek 6.18: Srovnání sympodiálního stromu s kulovitou korunou se skutečným stromem



Obrázek 6.19: Srovnání sympodiálního stromu s rozložitou kulovitou korunou se skutečným stromem

## 7 Závěr

V rámci práce na této bakalářské práci jsem se seznámila se zajímavou možností generování rostlin pomocí paralelních prepisovacích gramatik, nazývaných L-systémy. Po nastudování teoretických informací jsem vyzkoušela různé aplikace umožňující zobrazování L-systémů.

Následně jsem sama zkoušela vymýšlet různé předpisy, pomocí nichž by vznikali struktury podobné stromům. To se mi v rámci mezí podařilo nejprve ve 2D a následně i ve 3D prostoru. Výsledky tohoto zkoušení jsou na přiloženém CD, jak ve formě screenshotů, tak jako předpisy spustitelné v programech, v kterých byly vytvářeny (Fractal Grower, LStudio).

Výběr skupin dřevin pro tvorbu gramatik se ukázal být trochu komplikovanější, vzhledem k tomu, že jsem v žádné literatuře nenašla nějakou přehlednou tabulku s rozdělením, a při pozorování stromů v mém okolí jsem zjistila, že každý strom je originál.

Konečné gramatiky pro vybrané skupiny dřevin jsem nakonec vytvářela pouze ve 3D, v programu L-studio. Výsledné vygenerované modely dřevin se již více podobají reálným předlohám, ale stále by je šlo do budoucna vylepšovat.

# Literatura

- [1] PRUSINKIEWICZ, Przemyslaw a Aristid LINDENMAYER.  
*The algorithmic beauty of plants* [online].  
New York: Springer-Verlag,  
c1990, xii, 228 p. [cit. 2012-11-14].  
ISBN 35-409-7297-8.
- [2] NOVÁK, Jan a Milan SKALICKÝ.  
Botanika : cytologie, histologie, organologie a systematika.  
2. vyd. s. 94-95.  
ISBN 978-80-904011-5-0.
- [3] HECKER, Ulrich.  
Stromy a keře: klíč ke spolehlivému určování - 3 znaky.  
3. vyd. Editor Miroslav Volf. Čestlice: Rebo,  
2012, 238 s. Průvodce přírodou (Rebo).  
ISBN 978-80-255-0594-6.
- [4] BANFI, Enrico.  
Stromy: na zahradě, v parku a ve volné přírodě.  
3. vyd. Editor Miroslav Volf. Praha: Euromedia Group,  
2001, 223 s. Velký průvodce přírodou.  
ISBN 80-720-2807-3.
- [5] Generate a Tree as L-system. [online].  
Dostupné na: [cit. 2012-11-14].  
<http://www.javaview.de/vgp/tutor/lssystem/PaLSystem.html>

- 
- [6] L-systems Application. [online].  
Dostupné na: [cit. 2012-11-14].  
<http://www.biologie.uni-hamburg.de/b-online/lstjava/LSys.html>
- [7] WWW L-System Explorer. [online].  
Dostupné na: [cit. 2012-11-14].  
<http://zdeck.borg.cz/wlse/l-system.php>
- [8] Fractal Grower. [online].  
Dostupné na: [cit. 2012-11-14].  
<http://www.cs.unm.edu/~joel/PaperFoldingFractal/>
- [9] L-Studio. [online].  
Dostupné na: [cit. 2012-11-14].  
[http://algorithmicbotany.org/virtual\\_laboratory/](http://algorithmicbotany.org/virtual_laboratory/)

# A Obsah příloženého CD

## složka Dokumentace :

- složka latex-source - zdrojové soubory TeXu
- složka ukazkove\_gramatiky - zdrojové soubory gramatik použitých jako ukázky v teoretické části této práce
- BP\_A10B0601P\_Koubkova.pdf - bakalářská práce v PDF

## složka Gramatiky\_Bakalarska\_prace :

Obsahuje screenshoty grafických výstupů a soubory konečných gramatik pro program L-Studio.

## složka Gramatiky\_Projekt5 :

Obsahuje gramatiky vytvořené v rámci Projektu 5

- složka FractalGrower - gramatiky a screenshoty z programu Fractal Grower
- složka LStudio - počáteční gramatiky a screenshoty z programu Lstudio
- složka Srovnavaci\_obrazky - srovnání předchozích grafických výstupů se skutečnými stromy

## složka Pouzite\_programy :

- Applety\_odkazy.html - html soubor z odkazy na vyzkoušené applety a odkazy na stažení použitých programů
- složka FractalGrower - obsahuje spustitelný jar soubor
- složka Lstudio - obsahuje demo verze programu, získané registrací na [algorithmicbotany.com](http://algorithmicbotany.com)