

Vysoké učení technické v Brně
Fakulta informačních technologií

Doc. RNDr. Alexander Meduna, CSc.

FORMÁLNÍ JAZYKY MODELY A JEJICH APLIKACE

**FORMAL LANGUAGES: MODELS
AND THEIR APPLICATIONS**

Teze přednášky k profesorskému jmenovacímu řízení v oboru
„Výpočetní technika a informatika“



BRNO 2005

Klíčová slova

formální jazyky, formální modely, gramatiky, automaty, kooperace, distribuce, paralelismus, řízení, teoretické vlastnosti, zobecnění, uniformita, flexibilita, ekonomičnost, redukce, aplikace, bioinformatika

Key Words

Formal languages, computation, models, automata, grammars, language families, generative power, properties, parallelism, regulation, cooperation, integration, distribution, applications, bioinformatics

Abstrakt. Tyto teze prezentují základní informace o autorovi a jeho výzkumu formálních jazyků. Autor publikoval dvě knihy na mezinárodní úrovni (*Springer, Wiley*), čtyři knižní kapitoly a více než padesát článků v mezinárodních časopisech. Jeho výzkum se zaměřuje na studium formálních jazyků, definovaných moderními paralelními, řízenými a kooperujícími modely, které umožňují diskutovat o dnešních informačních technologiích s matematickou přesností. Ve svém výzkumu se autor soustředí na studium síly diskutovaných formálních modelů a na redukci jejich komponent. Teze kladou autorovy výsledky do souvislosti s jinými výsledky na mezinárodní úrovni. Ve svém závěru teze demonstrují aplikace zkoumaných formálních modelů.

Obsah

	strana
I. AUTOR	4
I.1 Základní informace o autorovi	4
I.2 Životopis	6
I.3 Publikační aktivity	7
II. FORMÁLNÍ JAZYKY A JEJICH MODELY	9
II.1 Úvod	9
II.2 Definice	10
II.3 Paralelní formální modely	12
II.4 Řízené formální modely	17
II.5 Kooperující formální modely	22
II.6 Odkazy	29
III APLIKACE	33
IV. PŘÍLOHY	37
IV.1 Vybrané publikace autora	37
IV.2 <i>English Summary</i>	43
IV.3 Obrázky	44

I. AUTOR

I.1 Základní informace o autorovi

Doc. RNDr. Alexander Meduna, CSc.
narozen 27.6.1957, Olomouc

Obor: informatika

Zaměření: matematicky orientovaná informatika

Specializace: teorie formálních jazyků

Ústav: UIFS FIT VUT v Brně

Kvalifikace:

1999: Habilitace v oboru výpočetní technika a informatika, VUT v Brně

1988: CSc. v oboru výpočetní technika a informatika, VUT v Brně

1982: RNDr. v oboru matematická informatika a teoretická informatika, Univerzita Palackého v Olomouci.

Profesionální kariéra:

- Zástupce vedoucího Ústavu informačních systémů FIT VUT v Brně, od 2002
- Docent na Ústavu informatiky a výpočetní techniky, FEI VUT v Brně, 1999 - 2002
- Samostatný vědecký pracovník na CVIS VUT v Brně, 1996 – 1999
- *Visiting assistant professor, University of Missouri--Columbia, USA*, 1988 – 1996
- Samostatný vědecký pracovník na CVIS VUT v Brně, 1985 – 1988
- Samostatný vědecký pracovník v Laboratoři výpočetní techniky UP Olomouc, 1982 – 1985

Současné zařazení:

Docent na FIT VUT v Brně, zástupce vedoucího Ústavu informačních systémů FIT VUT v Brně

V USA (*University of Missouri--Columbia*) garantoval předměty:

CS341 Automata Theory I

CS441 Automata Theory II

CS343 Compilers I

CS443 Compilers II

CS345 Principles of Programming Languages

CS352 Operating System Theory

V současnosti na FIT VUT garantuje předměty:

Bakalářské studium: Formální jazyky a překladače

Magisterské studium: Výstavba překladačů

Inženýrské studium: Základy překladačů

Doktorské studium: Moderní teoretická informatika

Vedení doktorandů: 13 doktorandů (dva úspěšní absolventi)

Současné výzkumné projekty:

- Vhodně integrované modely moderních informačních technologií, GAČR, GA201/04/0441, 2004-dosud

Publikační činnost:

- **2 knihy**

Automata and Languages, Springer, London, 2000, 986 stran

Grammars with Context Conditions and Their Applications, Wiley, 2005

(spoluautor M. Švec, v tisku)

- **4 knižní kapitoly**
- **52 recenzovaných článků** v angličtině v mezinárodních vědeckých časopisech
- **3 vysokoškolská skripta**
- **12 příspěvků na mezinárodních konferencích**

Rodinný stav: ženatý

Znalost jazyků: anglicky (aktivně), polsky a rusky (pasivně)

I.2 Životopis

1957

- narozen 27.6.1957 v Olomouci v rodině lékaře

1972-1976

- studium na gymnáziu ve Šternberku, úspěšně ukončeno maturitou.

1976-1981

- studium matematické informatiky a teoretické kybernetiky na Přírodovědecké fakultě Univerzity Palackého v Olomouci; úspěšně ukončeno státní závěrečnou zkouškou

1981-1985

- práce systémového programátora v Laboratoři počítačích strojů na Univerzitě Palackého v Olomouci
- úspěšné složení státní rigorózní zkoušky a získání titulu RNDr. (1982)
- výuka kompilátorů na Přírodovědecké fakultě Univerzity Palackého v Olomouci

1985-1988

- práce systémového programátora na OVC VUT v Brně
- studium vědecké aspirantury v oboru informatika na FEE VUT pod vedením Prof. RNDr. Jiřího Kopřivy, CSc; uspěně ukončení vědecké aspirantury a získání titulu CSc. (1988)
- výuka automatů a kompilátorů v postgraduálním studiu pořádaném OVC VUT, Brno (1985-1988)

1988-1996

- hostující profesor na *University of Missouri--Columbia, USA*

1996-1999

- vědecký pracovník na CVIS VUT Brno

1999-2000

- odborný asistent na UIVT FEE VUT Brno

2000-dosud

- docent na UIVT FEE VUT Brno

I.3 Publikační aktivity

Knihy: 2 (1 samostatně, 1 kolektivně)

Meduna, A.: *Automata and Languages: Theory and Applications*, Springer, London, 2000

Výtah. Tato kniha obsahuje kompletní základy teorie formálních jazyků. Je podán přehled výsledků o nejvýznamnějších typech automatů a gramatik. Práce objasňuje velmi podrobně všechny formalismy týkající se těchto základních formálních modelů a ilustruje je příklady. Je kladen důraz na algoritmy, které provádějí transformace diskutovaných modelů. Kniha také obsahuje mnoho vypracovaných příkladů ilustrujících diskutované modely. Je zde rovněž zahrnuta teorie vyčíslitelnosti. Aplikace v kompilátorech jsou diskutovány podrobně. Kniha je určena zejména studentům bakalářského a magisterského stupně.

Meduna, A. and Švec, M.: *Grammars with Context Conditions and Their Applications*, Wiley, 225 stran, 2005 (v tisku)

Výtah. Tato kniha diskutuje gramatiky, které jsou založeny na pravidlech, jejichž kontextové podmínky jsou výrazně zjednodušeny proti přísným podmínkám klasických kontextových pravidel. Je podán přehled o nejvýznamnějších typech těchto gramatik, jejichž alternativní kontextové podmínky mohou být rozděleny na kontextové podmínky kladené na (1) definiční obor relace přímé derivace, (2) používání pravidel a (3) podřetězce sousedící s přepisovanými symboly. Pro všechny tyto gramatiky jsou zavedeny a následně studovány sekvenční a paralelní verze, které reprezentují dva dnešní nejzákladnější přístupy ke gramatickému přepisování. Během sekvenčního derivačního kroku gramatika přepíše jeden symbol ve stávající větné formě, zatímco paralelním derivačním krokem přepíše symboly všechny. Nejvýznamnějšími zástupci sekvenčních a paralelních gramatik jsou bezkontextové a EOL gramatiky, na nichž je celá diskuse sekvenčního i paralelního přepisování založena. Práce předkládá všechny formalismy týkající se gramatik s takovou matematickou přesností, aby byly předkládané výsledky správné a zároveň srozumitelné. Většina důkazů dosažených výsledků obsahuje transformace gramatik, při nichž je kladen důraz především na algoritmický přístup k diskutovaným gramatickým modelům a jejich praktickému využití. Kniha rovněž uvádí několik vypracovaných příkladů ilustrujících teoretické pojmy a jejich aplikace. Kniha je určena především odborníkům v oblasti formálních jazyků, kteří se zabývají gramatickými modely diskutovanými dnešní teoretickou informatikou. Dále je určena jako doprovodný text pro pokročilé kurzy teoretické informatiky na vědecké úrovni.

Knižní kapitoly v knihách na mezinárodní úrovni: 4

Recenzovaných články v mezinárodních vědeckých časopisech: 52

Mezi vědecké časopisy, do nichž autor nejčastěji publikuje, patří

Acta Cybernetika

Acta Informatica

Fundamenta Informaticae

Information Processing Letters

International Journal of Computer Mathematics

Journal of Automata, Languages and Combinatorics

Theoretical Computer Science

Poznámka: Vzhledem k relativně rozsáhlé časopisecké publikační činnosti autora tyto teze uvádějí v příloze IV.1 pouze seznam jeho nejdůležitějších článků na mezinárodní úrovni. Úplný seznam publikací autora byl předán hodnotící komisi a je rovněž uveden na <http://www.fit.vutbr.cz/~meduna>.

II. FORMÁLNÍ JAZYKY A JEJICH MODELY

II.1 Úvod

„Hranice mého jazyka znamenají hranice mého světa“ píše Wittgenstein ve svém *Tractatus Logico-Philosophicus*. Ve světě informatiky toto smělé, ba provokativní prohlášení nepochybně platí, neboť jazyk je prostředkem, kterým informatik vyjadřuje informaci, a tím pádem je to právě jazyk, který klade meze vědeckému působení informatika. Tato ústřední role jazyka v informatice vedla hned při jejím zrodu ke vzniku teorie jazyků, která formalizuje jakékoliv jazyky a získává o nich poznatky s matematickou přesností. Není tedy divu, že poznatky této *teorie formálních jazyků* jsou signifikantní pro celou řadu oblastí informatiky sahající od překladačů programovacích jazyků přes matematickou lingvistiku až po genetiku.

Každý výzkum formálních jazyků je zahájen výběrem vhodných modelů pro jejich reprezentaci. Existují dva fundamentální typy takových modelů – gramatiky a automaty. Gramatiky jsou založeny na pravidlech, které vytvářejí správné věty jazyků. Automaty naopak pomocí svých pravidel rozpoznávají věty definovaného jazyka.

Přesněji řečeno, *gramatika* generuje svůj jazyk prováděním derivačních kroků, které mění řetězce symbolů, nazývané větné formy, na jiné řetězce podle svých gramatických pravidel. Během jednoho derivačního kroku gramatika ve stávající větné formě přepíše část této formy řetězcem tak, jak má předepsáno některým ze svých pravidel. Pokud gramatika provede posloupnost derivačních kroků ze svého startujícího symbolu až do větné formy skládající se z tzv. terminálních symbolů, pak tuto výslednou větnou formu nazýváme větou, která patří do generovaného jazyka. Množina všech takto generovaných vět tvoří jazyk daný gramatikou.

Zatímco gramatika jazyk generuje, *automat* představuje velmi formalizovaný zápis algoritmu, který definuje jazyk tím, že rozpoznává správně vytvořené vstupní řetězce symbolů. Automat čte vstupní řetězec symbol po symbolu a s takto načtenými symboly provádí jisté přechody podle svých pravidel, aby zjistil, zda má být vstupní řetězec přijat a zařazen do definovaného jazyka. Množina všech takto přijatých řetězců pak tvoří jazyk, který automat definuje.

Teorie formálních jazyků vždy zavádí své modely tak, aby adekvátním způsobem formalizovaly diskutované informační technologie a tím umožnily jejich rigorózní výzkum. V dnešních multiprocesorových architekturách vzájemně propojených počítačů jsou současně informační technologie založeny na *paralelním* způsobu výpočtu, který je *řízen* komplikovanými algoritmy. Tento vysoce efektivní způsob zpracování informace mnohdy využívá celé řady vzájemně *kooperujících* metod. Současná teorie formálních jazyku proto věnuje zásadní pozornost paralelním, řízeným a spolupracujícím gramatikám a automatům. Předkládané teze prezentují přehled autorových nejdůležitějších poznatků o těchto modelech.

II.2 Definice

V této podkapitole uvádíme pouze klíčové pojmy teorie formálních jazyků, které jsou nutné pro porozumění zbývajících částí těchto tezí. Autorova kniha [38] obsahuje kompletní teorii formálních jazyků, na kterou zde čtenáře odkazujeme.

Pro libovolnou abecedu V , V^* označuje množinu všech řetězců nad V , přičemž ε označuje prázdný řetězec. Položme $V^+ = V^* - \{\varepsilon\}$. Pro každý řetězec $w \in W^*$, $|w|$ označuje délku řetězce w a $\text{alph}(w)$ označuje množinu symbolů, které se v řetězci w vyskytují.

Definice: *Bezkontextová gramatika* G je čtveřice

$$G = (V, T, P, S),$$

kde:

- V je úplná abeceda gramatiky G
- $T \subseteq V$
- P je konečná množina přepisovacích pravidel tvaru $A \rightarrow x$, kde $A \in V - T$ a $x \in V^*$
- S je startovací symbol, $S \in V - T$

Symbole v T se nazývají *terminální symboly* nebo *terminály*. Symbole z množiny $V - T$ se nazývají *neterminální symboly* nebo *neterminály*.

Jestliže $A \rightarrow x \in P$ a $u, v \in V^*$, pak řekneme, že G provede *přímý derivační krok* z uAv do uxv , což zapisujeme $uAv \Rightarrow uxv$. Matematicky, \Rightarrow je binární relace nad V^* . Symbolem \Rightarrow^* budeme značit tranzitivní a reflexivní uzávěr této relace. Pokud $S \Rightarrow^* w$, kde $w \in V^*$, potom řekneme, že w je *větná forma* derivovaná v gramatice G . *Jazyk generovaný gramatikou* G , $L(G)$, je definován

$$L(G) = \{ w : S \Rightarrow^* w \text{ a } w \in T^* \}$$

Definice: *Konečný automat* je pětice

$$M = (Q, \Sigma, R, s, F),$$

kde:

- Q je konečná množina stavů
- Σ je abeceda vstupních symbolů
- R je konečná množina přepisovacích pravidel tvaru $pa \rightarrow q$, kde $p, q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$
- $s \in Q$ je počáteční stav
- $F \subseteq Q$ je množina koncových stavů

Konfigurace M je libovolný řetězec tvaru qv , kde $q \in Q$ a $v \in \Sigma^*$. Necht' K označuje množinu všech konfigurací M . Jestliže $pa \rightarrow q \in R$ a $v \in \Sigma^*$, pak M provede *přechod* z konfigurace pav do qv , což zapisujeme $pav \Rightarrow qv$. Matematicky, \Rightarrow je binární relace na K . Symbolem \Rightarrow^* označujeme tranzitivní a reflexivní uzávěr této relace. *Jazyk přijímaný* M , $L(M)$, je definován

$$L(M) = \{ w: sw \Rightarrow^* f, w \in \Sigma^*, f \in F \}$$

Definice: *Zásobníkový automat* je sedmice

$$M = (Q, \Sigma, W, R, s, S, F),$$

kde:

- Q je konečná množina stavů
- Σ je abeceda vstupních symbolů
- W je abeceda zásobníkových symbolů
- R je konečná množina prepisovacích pravidel tvaru $Apa \rightarrow wq$, kde $A \in W, p, q \in Q, a \in \Sigma \cup \{\epsilon\}$ a $w \in W^*$
- $s \in Q$ je počáteční stav
- $S \in W$ je počáteční zásobníkový symbol
- $F \subseteq Q$ je množina koncových stavů

Konfigurace M je libovolný řetězec tvaru uqv , kde $u \in W^*, q \in Q$ a $v \in \Sigma^*$. Necht' K označuje množinu všech konfigurací M . Pokud $Apa \rightarrow wq \in P, u \in W^*, a \in \Sigma^*$, pak M provede *přechod* z konfigurace $uApav$ do $uwqv$, což zapisujeme jako $uApav \Rightarrow uwqv$. Matematicky, \Rightarrow je binární relace na K . Symbolem \Rightarrow^* budeme značit tranzitivní a reflexivní uzávěr této relace. *Jazyk přijímaný* M , $L(M)$, je definován

$$L(M) = \{ w: Ssw \Rightarrow^* f, w \in \Sigma^*, f \in F \}$$

Následující čtyři třídy jazyků mají klíčový význam v teorii formálních jazyků. Třída *regulárních jazyků* je označena **REG** a je definovaná konečnými automaty. Třída *bezkontextových jazyků* je označena **CF** a je definovaná zásobníkovými automaty a bezkontextovými gramatikami. Další důležitou třídou jazyků je třída *kontextových jazyků*, která je označena **CS**. Třída *rekurzivně spočetných jazyků* je označena **RE** a definovaná Turingovými stroji. Pro tyto čtyři třídy je zavedena *Chomského hierarchie*, pro kterou platí následující vztah:

$$\mathbf{REG} \subset \mathbf{CF} \subset \mathbf{CS} \subset \mathbf{RE}$$

II.3 Paralelní formální modely

V těchto tezích diskutujeme o následujících dvou typech paralelních formálních modelů založených na gramatikách, které jsou podobné gramatikám bezkontextovým:

- gramatiky s rozptýleným kontextem
- multigramatiky

(A) Gramatiky s rozptýleným kontextem jsou založeny na sekvencích bezkontextových pravidel, které simultánně přepisují více neterminálů během jednoho derivačního kroku (viz [5], [27], [28], [30], [37], [38]).

(B) Multigramatiky používají bezkontextová pravidla, která mají terminál nebo neterminál na jejich levých stranách. Použitím extrémně jednoduchých regulárních jazyků, které se nazývají *selektory*, tyto gramatiky specifikují sekvence symbolů přepisovaných během derivačního kroku a navíc kladou jisté omezení na kontext vyskytující se mezi přepisovanými symboly. Jinak pracují analogicky jako bezkontextové gramatiky (viz [4], [11], [35]).

Gramatiky s rozptýleným kontextem

Definice: Gramatika s rozptýleným kontextem je čtveřice $G = (V, T, P, S)$, kde V označuje úplnou abecedu, T abecedu terminálních symbolů, S je startující symbol a P konečnou množinu pravidel tvaru

$$(A_1, A_2, \dots, A_n) \rightarrow (x_1, x_2, \dots, x_n),$$

kde n je kladné celé číslo, $A_i \in V - T$ a $x_i \in V^*$ pro všechna $i = 1, \dots, n$. Jestliže každé pravidlo $(A_1, A_2, \dots, A_n) \rightarrow (x_1, x_2, \dots, x_n)$ gramatiky splňuje podmínku $x_i \in V^+$ pro všechna $i = 1, \dots, n$, mluvíme o gramatikách s rozptýleným kontextem bez ε -pravidel. Použitím pravidla $(A_1, A_2, \dots, A_n) \rightarrow (x_1, x_2, \dots, x_n)$ v derivačním kroku nahradí gramatika G simultánně všechny symboly A_1, A_2, \dots, A_n za x_1, x_2, \dots, x_n . Formálně,

$$u_1 A_1 u_2 A_2 u_3 \dots u_n A_n u_{n+1} \Rightarrow u_1 x_1 u_2 x_2 u_3 \dots u_n x_n u_{n+1}$$

pro libovolné $u_1, \dots, u_{n+1} \in V^*$. Symbol \Rightarrow označuje relaci přímé derivace. Obvyklým způsobem je definován tranzitivní a relexivní uzávěr relace přímé derivace označený symbolem \Rightarrow^* .

Jazyk generovaný gramatikou G je definován

$$L(G) = \{ w : S \Rightarrow^* w, w \in T^* \}.$$

Příklad: Uvažujme gramatiku s rozptýleným kontextem, G , s následujícími třemi pravidly:

$$(S) \rightarrow (AA), (A, A) \rightarrow (aA, bAa) \text{ a } (A, A) \rightarrow (\varepsilon, \varepsilon)$$

G generuje např. řetězec $aabbaa$ takto:

$$S \Rightarrow AA \Rightarrow aAbAa \Rightarrow aaAbbAaa \Rightarrow aabbaa$$

G generuje následující jazyk, který není bezkontextový:

$$L(G) = \{ a^i b^i a^i : i \geq 0 \}$$

Výsledky: Autor se zabýval problematikou gramatik s rozptýleným kontextem v dvaadvaceti článcích a do značné míry i v monografii Meduna, A. a Švec, M.: *Grammars with Context Conditions and Their Applications*, Wiley, New York, 2005 (v tisku). V roce 2001 autor prezentoval přehled svých výsledků o těchto gramatikách v rámci hlavní vyžádané přednášky každoročně konané mezinárodní konference *Descriptive Complexity of Formal Models* ve Vídni. Patrně nejhodnotnějším výsledkem na toto téma je jeho věta o ekvivalenci Turingových strojů a gramatik s rozptýleným kontextem publikovaná v [27]. Připomeňme, že ještě v roce 1989 zařadili profesori Dassow a Paun do své známé monografie *Regulated Rewriting in Formal Language Theory* (Akademie-Verlag, Berlin, 1989) otázku, zda tato ekvivalence platí, do seznamu nejdůležitějších otevřených problémů (viz. *Open Problem 2.4.3* na str. 145 zmíněné monografie [4]). Redukce gramatik tvoří velmi živé současné téma teorie formálních jazyků a autor jej diskutoval již v jedenácti článcích a v monografii Meduna, A. a Švec, M.: *Grammars with Context Conditions and Their Applications*, Wiley, New York, 2005 (v tisku). Zcela unikátní výsledky jsou obsaženy ve třech studiích [37], [51] a [52]. První ze zmíněných studií dokázala, že gramatiky s rozptýleným kontextem, které obsahují pouze tři neterminální symboly, jsou stále ekvivalentní s Turingovými stroji, čímž byl vylepšen analogický výsledek pro maticové gramatiky se šesti neterminálními symboly dokázaný Prof. Paunem (Rumunsko) v roce 1984. Jako první v historii teorii formálních jazyků, druhá z uvedených studií zredukovala počet pravidel v gramatikách s rozptýleným kontextem. Snad ještě významnější je poslední z uvedených studií, ve které se jako první podařilo zredukovat souběžně několik gramatických komponent (počet pravidel a počet neterminálních symbolů).

Multigramatiky

Definice: Multigramatika G je pětice

$$G = (V, T, P, S, K),$$

kde V , T a S mají stejný význam jako v bezkontextových gramatikách. P je konečná množina pravidel tvaru $a \rightarrow x$, kde $a \in V$ a $x \in V^*$. K je konečná množina selektorů tvaru

$$X_1 \mathbf{active}(Y_1) X_2 \dots X_n \mathbf{active}(Y_n) X_{n+1}$$

kde

n je kladné celé číslo,

pro všechna $i = 1, \dots, n + 1$, $X_i \in \{Z^* : Z \subseteq V\}$,

pro všechna $j = 1, \dots, n$, $Y_j \subseteq V$ a $Y_j \neq \emptyset$

G provádí *derivační krok* ve tvaru

$$u_1 a_1 u_2 a_2 u_3 \dots u_n a_n u_{n+1} \Rightarrow u_1 x_1 u_2 x_2 u_3 \dots u_n x_n u_{n+1}$$

jestliže K obsahuje selektor $X_1 \mathbf{active}(Y_1) X_2 \dots X_n \mathbf{active}(Y_n) X_{n+1}$ splňující:

pro všechna $i = 1, \dots, n + 1$ $u_i \in X_i$

pro všechna $j = 1, \dots, n$, $a_j \in Y_j$ a $a_j \rightarrow x_j \in P$

Jazyk generovaný G , $L(G)$, je definován

$$L(G) = \{ w : S \Rightarrow^* w \text{ a } w \in T^* \},$$

kde \Rightarrow^* označuje tranzitivní a reflexivní uzávěr \Rightarrow .

Příklad: Uvažujme multigramatiku, G , s následujícími pravidly:

$$S \rightarrow AA, S \rightarrow BB, A \rightarrow Aa, A \rightarrow Ba, B \rightarrow Ab, B \rightarrow Bb, A \rightarrow \varepsilon, B \rightarrow \varepsilon$$

a těmito třemi selektory:

$$\mathbf{active}(\{S\}), \mathbf{active}(\{A\})\{a, b\}^* \mathbf{active}(\{A\})\{a, b\}^*, \mathbf{active}(\{B\})\{a, b\}^* \mathbf{active}(\{B\})\{a, b\}^*$$

Například G derivuje $abab$ následovně:

$$S \Rightarrow BB \Rightarrow AbAb \Rightarrow AabAab \Rightarrow abab$$

G generuje následující jazyk:

$$L(G) = \{ ww : w \in \{a, b\}^* \}$$

Definice: Rozšířená multigramatika G je pětice

$$G = (V, T, P, S, K),$$

kde V , T , P a S mají stejný význam jako v multigramatice. K je konečná množina selektorů tvaru:

$$X_1 \mathbf{active}(Y_1) X_2 \dots X_n \mathbf{active}(Y_n) X_{n+1},$$

kde

n je kladné celé číslo

pro všechna $i = 1, \dots, n+1, X_i \in \{Z^* : Z \subseteq V\}$

pro všechna $j = 1, \dots, n, Y_j \in \{Z^* : Z \subseteq V \text{ a } Z \neq \emptyset\}$.

Pro každé $v \in V^+$, kde $v = a_1 \dots a_{|v|}$ s $a_i \in V$ pro $i = 1, \dots, |v|$, definujeme jazyk $ContinuousRewriting(v) \subseteq V^*$ následující ekvivalencí pro každé $z \in V^*$,

$$z \in ContinuousRewriting(v)$$

tehdy a jen tehdy

$$a_i \rightarrow x_i \in P \text{ pro všechna } i = 1, \dots, |v|, \text{ a } z = x_1 \dots x_{|v|}$$

G provádí derivační krok tvaru:

$$u_1y_1u_2y_2u_3 \dots u_ny_nu_{n+1} \Rightarrow u_1z_1u_2z_2u_3 \dots u_nz_nu_{n+1},$$

jestliže K obsahuje $X_1 \mathbf{active}(Y_1) X_2 \dots X_n \mathbf{active}(Y_n) X_{n+1}$ takové, že

pro všechna $i = 1, \dots, n, y_i \in Y_i$ a $z_i \in \mathit{ContinuousRewriting}(y_i)$

pro všechna $i = 1, \dots, n + 1, u_i \in X_i$

Jako obvykle jazyk generovaný $G, L(G)$, je definován

$$L(G) = \{ w : S \Rightarrow^* w \text{ a } w \in T^* \},$$

kde \Rightarrow^* označuje tranzitivní a reflexivní uzávěr \Rightarrow .

Příklad: Uvažujme rozšířenou multigramatiku, G , s následujícími pravidly:

$$S \rightarrow bcb, b \rightarrow bb, c \rightarrow cc,$$

a těmito třemi selektory:

$$\mathbf{active}(\{S\}), \mathbf{active}(\{b\}^+) \{c\}^* \mathbf{active}(\{b\}^+), \text{ a } \mathbf{active}(\{b\}^+) \mathbf{active}(\{c\}^+) \mathbf{active}(\{b\}^+)$$

G generuje $bbbbccbbbb$ následovně:

$$S \Rightarrow bcb \Rightarrow bcbcb \Rightarrow bbbccbbbbb$$

Všimněme si, že G generuje následující jazyk, který není bezkontextový

$$L(G) = \{ b^j c^k b^j : j = 2^i \text{ a } k = 2^r \text{ tak, že } i \geq r \geq 0 \}$$

Výsledky: Autor dokázal, že multigramatiky s šesti neterminály jsou ekvivalentní s Turingovými stroji (viz [32]). Analogický výsledek dokázal i pro rozšířené multigramatiky (viz [35]).

II.4 Řízené formální modely

Řízené formální modely označují skupinu formálních modelů, které jsou založeny na běžných pravidlech, jejich aplikace však podléhá určitým omezujícím matematickým prostředkům, které určují výběr použitého pravidla v každém kroku modelovaného výpočtu. Řízené gramatiky si probereme zcela neformálně, zatímco o řízených automatech budeme diskutovat formálněji.

Řízené gramatiky

Řízené gramatiky patří dnes již ke klasickým tématům teorie formálních jazyků (viz [4], [65]) a autor se touto problematikou zabývá již dvacet let. Věnoval tomuto tématu osmnáct článků a diskutuje o něm rovněž ve svých monografiích Meduna, A.: *Automata and Languages* (Springer, London, 2000) a Meduna, A. a Švec, M.: *Grammars with Context Conditions and Their Applications*, Wiley, New York, 2005 (v tisku).

Hlavní tři typy řízených gramatik jsou maticové gramatiky, programované gramatiky a gramatiky s nahodilým kontextem. V *programovaných gramatikách* je derivace řízena dvěma množinami návěstí přiřazených ke každému pravidlu, přičemž první určuje, která pravidla použít v případě úspěšné aplikace daného pravidla, a druhá specifikuje, kterými pravidly lze pokračovat v případě, kdy se dané pravidlo nepodařilo aplikovat a prepisovaný řetězec tedy zůstává beze změny. V *gramatikách s nahodilým kontextem* je každé pravidlo doplněno dvěma množinami neterminálů, které se nazývají povolující a zakazující kontext. Takovéto pravidlo pak lze na řetězec aplikovat za předpokladu, že daný řetězec obsahuje všechny symboly z povolujícího kontextu a žádný symbol ze zakazujícího kontextu. K paralelní verzi těchto gramatik se vrátíme v aplikacích, o nichž budeme diskutovat v části III těchto tezí. Maticové gramatiky si nyní probereme podrobněji, i když ne zcela formálně.

Maticová gramatika je tvořena dvěma komponentami. První komponentou je obyčejná bezkontextová gramatika G . Druhou komponentou je konečná množina tzv. *matic*, což jsou řetězce nad množinou prepisovacích pravidel v G . Jednotlivé matice se zapisují buď jako řetězce návěstí označujících jednotlivá bezkontextová pravidla, nebo přímo jako řetězce pravidel gramatiky G . Množina matic se obvykle označuje písmenem M . Jeden derivační krok v maticové gramatice je proveden postupnou aplikací všech prepisovacích pravidel v některé matici v tom pořadí, v jakém jsou v matici uvedena. Pokud některé pravidlo matice aplikovat nelze, nelze aplikovat ani matici jako celek.

Příklad: Uvažujme maticovou gramatiku se startujícím neterminálem S , množinou pravidel

$$1: S \rightarrow AB, 2: A \rightarrow aA, 3: B \rightarrow bBc, 4: A \rightarrow a, 5: B \rightarrow bc$$

a množinou matic $M = \{1, 23, 45\}$. Nejprve bude neterminál S přepsán na řetězec AB . Jestliže nyní přepíšeme AB pomocí pravidla 2, musíme následně přepsat aAB pomocí pravidla 3 na $aAbBc$, protože pravidlo 2 se vyskytuje jako první jedině v matici 23 a každá matice musí být vždy aplikována jako celek. Po přepsání symbolu S na AB maticí 1 je řetězec AB dále rozvíjen opakovanou aplikací matice 23, přičemž je neustále zachováván stejný počet symbolů a , b a c . Derivace je ukončena aplikací matice 45, což vede k řetězci tvaru $a^n b^n c^n$, pro nějaké $n \geq 1$. Generovaný jazyk je tedy

$$L(G) = \{ a^i b^i c^i : i \geq 0 \}.$$

Všeobecně řečeno, *kanonická gramatická derivace* znamená, že se přepisuje vždy nejlevější či nejpravější výskyt neterminálů ve stávající větne formě. U maticových gramatik rozlišujeme tři základní typy nejlevějších kanonických derivací (nejpravější kanonické derivace jsou definovány analogicky):

- *Typ 1:* Je aplikována taková matice, jejíž každé pravidlo přepíše vždy nejlevější neterminál ve stávající větne formě.
- *Typ 2:* Z matic, které jsou aplikovatelné na aktuální větne formu, se vybere ta matice, jejíž první pravidlo přepíše nejlevější možný neterminál. Každé následující pravidlo této matice přepíše nejlevější výskyt neterminálu, který tvoří jeho levou stranu.
- *Typ 3:* Použije se libovolná matice, avšak každé pravidlo vždy přepíše nejlevější výskyt neterminálu, který tvoří jeho levou stranu.

Využití kanonických typů derivací má vliv na generativní sílu maticových gramatik. Zatímco maticové gramatiky, které provádí kanonické derivace typu 1, generují **CF**, maticové gramatiky, které provádí kanonické derivace typu 2, generují **RE**. Otevřeným problémem zůstává, zda maticové gramatiky, které provádí kanonické derivace typu 3, generují rovněž **RE**.

Výsledky: V oblasti kanonických derivací maticových gramatik dosáhl autor celé řady výsledků (viz [24], [25]). Byly zavedeny *LR* maticové gramatiky a *RL* maticové gramatiky kombinující v jedné matici pravidla aplikovaná na nejlevější a nejpravější neterminál. Dále byly formulovány a dokázány věty o síle redukovaných maticových gramatik využívajících levou derivaci typu 2 a o síle maticových gramatik kombinujících využití levé derivace typu 1 a 3. Bylo dokázáno, že

každý jazyk v **RE** lze generovat *LR*-maticovou gramatikou či *RL*-maticovou gramatikou obsahující nejvýše dvě matice se dvěma pravidly a délkou dvě, přičemž všechny ostatní matice mají pouze jedno pravidlo. Maticové gramatiky, které aplikují jednu matici pomocí nejlevější derivace typu 1 a všechny ostatní matice pomocí nejlevější derivace typu 3 generují rovněž **RE**. Otevřený problém, zda maticové gramatiky používající levou derivaci typu 3, generují **RE**, nebyl vyřešen, ale bylo dokázáno, že stačí přidat jedinou matici, která bude aplikována pomocí levé derivace typu 1, aby tyto maticové gramatiky definovaly **RE**.

Řízené automaty

Uvažujme zásobníkový automat, $M = (Q, \Sigma, W, R, s, S, F)$, definován v II.2. Necht' opět K označuje množinu všech konfigurací M . Necht' $x, y \in K$ jsou libovolné dvě konfigurace. Jestliže M provede $x \Rightarrow^* y$ podle řetězce pravidel $\rho \in R^*$, píšeme $x \Rightarrow^* y [\rho]$. Necht' $D \subseteq R^*$. M řízený D přijímá jazyk $L(M, D)$, který je definován

$$L(M, D) = \{ w : Ssw \Rightarrow^* f[\rho], w \in \Sigma^*, f \in F, \rho \in D \}$$

Definice: Řízený zásobníkový automat je sedmice

$$M = (Q, \Sigma, \Gamma, R, s, S, F),$$

kde

- Q je konečná množina stavů,
- Σ je vstupní abeceda,
- Γ je zásobníková abeceda,
- R je konečná množina pravidel tvaru $Apa \rightarrow wq$, kde $A \in \Gamma, p, q \in Q, a \in \Sigma \cup \{\epsilon\}$ a $w \in \Gamma^*$,
- $s \in Q$ je počáteční stav,
- $S \in \Gamma$ je počáteční symbol zásobníku,
- $F \subseteq Q$ je množina koncových stavů.

Navíc je požadováno, aby množiny Q, Σ a Γ byly po dvou disjunktní. Necht' Ψ je abeceda *návěští pravidel* taková, že $\text{card}(\Psi) = \text{card}(R)$ a ψ je bijekce z R do Ψ . Skutečnost, že ψ zobrazuje pravidlo $Apa \rightarrow wq \in R$ na $\rho \in \Psi$, bude pro jednoduchost zapisována $\rho.Apa \rightarrow wq \in R$. Jinými slovy, $\rho.Apa \rightarrow wq$ znamená, že $\psi(Apa \rightarrow wq) = \rho$. Konfigurace řízeného zásobníkového automatu M je řetězec Apa , kde $A \in \Gamma^*, p \in Q$ a $a \in \Sigma^*$. Jestliže $x \in \Gamma^*, y \in \Sigma^*$, $xApay$ a $xwqy$ jsou konfigurace M a $\rho.Apa \rightarrow wq \in R$ je pravidlo, pak M provede přechod z konfigurace $xApay$ do konfigurace $xwqy$ podle ρ , symbolicky $xApay \Rightarrow xwqy [\rho]$. Necht' χ je libovolná konfigurace

M . M provede nula přechodů z χ do χ podle ε , což zapíšeme jako $\chi \Rightarrow^0 \chi [\varepsilon]$. Necht' existuje posloupnost konfigurací $\chi_0, \chi_1, \dots, \chi_n$ pro nějaké $n \geq 1$ taková, že $\chi_{i-1} \Rightarrow \chi_i [\rho_i]$, kde $\rho_i \in \Psi$ pro $i = 1, \dots, n$, potom M provede n přechodů z χ_0 do χ_n podle $\rho_1 \dots \rho_n$, což symbolicky zapíšeme jako $\chi_0 \Rightarrow^n \chi_n [\rho_1 \dots \rho_n]$. Necht' Ξ je řídicí jazyk nad Ψ tj. $\Xi \subseteq \Psi^*$. M řízený Ξ definuje jazyk označen $L(M, \Xi)$ a definován

$$L(M, \Xi) = \{ w: Ssw \Rightarrow^* f[\sigma], w \in \Sigma^*, f \in F, \sigma \in \Xi \},$$

kde \Rightarrow^* označuje transitivní a reflexivní uzávěr relace \Rightarrow .

Výsledky: V autorově výzkumu řízených automatů mají zcela zásadní význam články [40] a [47]. Tyto práce poprvé zavedly a zkoumaly řízené automaty, které přirozeným způsobem doplňují výzkum řízených gramatik, o kterých existuje mnoho studií publikovaných asi od roku 1960. V tomto smyslu tedy otvírají výše uvedené práce novou kapitolu teorie formálních jazyků. Autor dokázal, že tyto automaty řízené regulárními jazyky nezvyšují svou sílu. Na druhé straně zásobníkové automaty řízené lineárními jazyky definují **RE** (viz [41]).

Další zásobníkový formální model, kterému autor věnuje pozornost v současnosti, je sebereprodukcující zásobníkový převodník, který vychází ze zásobníkového převodníku a rozšiřuje jej pouze o možnost vícenásobného překladu vstupního řetězce. Prakticky tedy nezasahuje do vnitřní struktury převodníku a zachovává tak jeho snadnou praktickou implementaci. Sebereprodukcující zásobníkový převodník zpracovává vstupní řetězec stejným způsobem jako běžný zásobníkový převodník. Zpracování vstupního řetězce začíná v počáteční konfiguraci. Vstupní řetězec se překládá obvyklým způsobem. Po úspěšném překladu však převodník nemusí ukončit svoji činnost, ale může vytvořený překlad přesunout zpět na vstup a tento řetězec znovu přeložit. Tento přesun vygenerovaného řetězce zpět na vstup se nazývá *sebereprodukcující krok*. Provedení sebereprodukcujícího kroku je však podmíněno tím, že stav převodníku musí patřit do množiny takzvaných sebereprodukcujících stavů. Dojde-li tedy k překladu vstupního řetězce, ale převodník se nenachází v sebereprodukcujícím stavu, je překlad ukončen a vytvořený výstupní řetězec je překladem původního vstupního řetězce.

Definice: Sebereprodukcující zásobníkový převodník je osmice

$$M=(Q, \Gamma, \Sigma, \Omega, R, s, S, O),$$

kde

- Q je konečná množina stavů,
- Γ je abeceda taková, že $Q \cap \Gamma = \emptyset$,
- $\Sigma \subseteq \Gamma$ je vstupní abeceda,
- $\Omega \subseteq \Gamma$ je výstupní abeceda,
- R je konečná množina *překladových pravidel* tvaru $xqw \rightarrow ypv$, kde $x, y, w, v \in \Gamma^*$ a $q, p \in Q$,
- $s \in Q$ je počáteční stav,
- $S \in \Gamma$ je počáteční symbol zásobníku,
- $O \subseteq Q$ je množina sebereprodukcujících stavů.

Nechť $\$ \notin (Q \cup \Gamma)$ je speciální oddělovací symbol. Konfigurací sebereprodukcujícího zásobníkového převodníku M nazveme libovolný řetězec tvaru $\$zqy\x , kde $x, y, z \in \Gamma^*$ a $q \in Q$.
 Nechť M je sebereprodukcující zásobníkový převodník, $u_1qw \rightarrow u_2pv \in R$ a $y = \$hu_1q wz\t a $x = \$hu_2pz\tv dvě konfigurace, kde $h, u_1, u_2, w, t, v, z \in \Gamma^*$, $q, p \in Q$. Potom M provádí *překladový krok* z y do x v M , což symbolicky zapíšeme jako $y \xrightarrow{t} x [u_1qw \rightarrow u_2pv]$ nebo jednoduše $y \xrightarrow{t} x$ v M . Pokud $y = \$hq\t a $x = \$hqt\$$, kde $t, h \in \Gamma^*$ a $q \in O$, potom M provádí *sebereprodukcující krok* z y do x v M , což symbolicky zapíšeme $y \xrightarrow{r} x$. Řekneme, že M provádí krok z konfigurace y do konfigurace x , což zapíšeme $y \Rightarrow x$, pokud $y \xrightarrow{t} x$ nebo $y \xrightarrow{r} x$. Nechť \Rightarrow^* značí tranzitivní a reflexivní uzávěr relace \Rightarrow . Nechť M je sebereprodukcující zásobníkový převodník a $w, v \in \Gamma^*$. Potom M *překládá* w na v jestliže $\$Ssw\$ \Rightarrow^* \$q\$v$ v M . *Překlad definovaný* M označíme $T(M)$ a je definován

$$T(M) = \{(w, v) : \$Ssw\$ \Rightarrow^* \$q\$v, \text{ kde } w \in \Sigma^*, v \in \Omega^*, q \in Q\}.$$

Položme $domain(T(M)) = \{w : (w, x) \in T(M)\}$ a $range(T(M)) = \{x : (w, x) \in T(M)\}$.

Výsledky: Autor dokázal, že pro každý jazyk v **RE** existuje sebereprodukcující zásobníkový převodník M takový, že $domain(T(M)) = L$ a $range(T(M)) = \{\varepsilon\}$. Tento výsledek byl přijat pro publikaci v časopise *Kybernetika*.

II.5 Kooperující formální modely

Základními kooperujícími formálními modely jsou gramatické systémy (viz [2]), které jsou založeny na množině gramatik, které spolu jednoduchým způsobem komunikují. Rozlišujeme CD (*cooperating distributed*) gramatické systémy a PC (*parallel communicating*) gramatické systémy.

CD gramatické systémy pracují sekvenčně. Tyto systémy se skládají z komponent, které představují gramatiky. Každý derivační krok je proveden pouze jednou gramatikou. Řízení takového systému spočívá ve výběru komponenty, která následně provede derivační krok. Například pomocí jedné gramatiky se provede k kroků a po té v derivaci pokračuje jiná gramatika, která opět provede k kroků atd. Rovněž je stanovena podmínka ukončující činnost celého systému. Takové ukončení může být provedeno v okamžiku, kdy žádná gramatika nemůže vykonat další derivační krok.

PC gramatické systémy pracují paralelně. Každá komponenta, kterou je opět gramatika, má svoji vlastní větnou formu, nad kterou provádí derivační kroky. Řízení komponent je realizováno pomocí komunikačních symbolů. Na základě těchto symbolů gramatika určí, kde dojde k vložení větné formy z jiné gramatiky. Jazyk, který generuje první komponenta systému, je výsledný jazyk celého systému.

V posledních letech se autor intenzivně zabývá studiem PC gramatických systémů. Autor zavedl variantu nazývanou dvousměrný PC gramatický systém, který provádí tři výpočetní kroky – derivaci, redukci a komunikaci. Dvousměrný PC gramatický systém provádí derivační krok standardním způsobem, tedy přepisuje levou stranu pravidla na stranu pravou. Během redukčního kroku naopak přepisuje pravou stranu pravidla na stranu levou. Komunikační krok je prováděn způsobem obvyklým pro PC gramatické systémy; po jeho provedení je však výpočetní mód změněn z derivace na redukci a naopak.

Tyto systémy samozřejmě zasluhují pozornost teorie formálních jazyků z hlediska teoretického. Praktický pohled je ale neméně zajímavý. Dvousměrné PC gramatické systémy formalizují modely, kombinují redukční a výpočetní kroky, které se často vyskytují v aplikované informatice. Tak např. syntaktický analyzátor překladače obvykle kombinuje přístup zdola-nahoru pro zpracování výrazů a přístup shora-dolů pro zpracování základních programových konstrukcí pro řízení toku programu (cykly ap.). Dalším příkladem z této oblasti může být generování tříadresného kódu. Nejprve se provádí syntaxí řízené generování abstraktního syntaktického stromu metodou shora-dolů a následně překlad tohoto stromu na požadovaný tříadresný kód metodou zdola-nahoru. Opět se celý proces výpočtu skládá z derivační a redukční části. Existují tedy jak teoretické, tak praktické důvody pro zkoumání dvousměrných PC gramatických systémů.

Autor věnuje zásadní pozornost centralizovaným dvousměrným metalineárním PC gramatickým systémům pracujícím v módu bez návratu (viz [57]). V centralizovaném systému si pouze první komponenta může vyžádat provedení komunikačního kroku. Metalineární systém je založen na metalineárních bezkontextových gramatikách. Práce systému v módu bez návratu vyjadřuje, že jednotlivé komponenty po provedení komunikačního kroku pokračují ve zpracovávání aktuální větné formy a nevrací se ke svému axiomu. Autor se zaměřuje na diskuzi o redukcí těchto systémů, což je významný současný trend v teorii formálních jazyků. Jeho hlavní výsledek ukazuje, že každý jazyk v **RE** je generován centralizovaným dvousměrným PC gramatickým systémem, který má pouze dvě metalineární komponenty, a během každého výpočtu provede jediný komunikační krok. Nadto první komponenta obsahuje pouze tři nonterminály, jediné pravidlo obsahující komunikační symbol a její libovolná větná forma neobsahuje více než dva nonterminály.

Definice: Necht' k a n jsou dvě přirozená čísla. Dvousměrná k -lineární PC komponenta je čtveřice

$$G = (N, T, P, S),$$

kde N a T jsou vzájemně disjunktí abecedy nazývané nonterminály resp. terminály a $S \in N$ je startovací symbol. Označme $M = N - \{S\}$. P je konečná množina pravidel takových, že každé pravidlo má jeden z následujících tvarů:

- $S \rightarrow x$, kde $x \in (M \cup T)^*$ a x obsahuje maximálně k výskytů symbolů z M ,
- $A \rightarrow x$, kde $A \in M$ a $x \in T^*MT^* \cup T^*$.

Necht' $u, v \in (N \cup T)^*$. Pro všechny $A \rightarrow x \in P$ píšeme $uAv \xrightarrow{d} uxv$ a $uxv \xrightarrow{r} uAv$, kde d a r označuje přímou *derivaci*, respektive přímou *redukcí*. Pokud chceme vyjádřit, že G provede derivaci $uAv \xrightarrow{d} uxv$ podle pravidla $A \rightarrow x$, potom píšeme $uAv \xrightarrow{d} uxv [A \rightarrow x]$. $uxv \xrightarrow{r} uAv [A \rightarrow x]$ má analogický význam pro redukcí. Dvousměrný k -lineární n -PC gramatický systém je $(n + 1)$ -tice

$$\Gamma = (Q, G_1, \dots, G_n),$$

kde

- $Q = \{q_i : i = 1, \dots, n\}$ je množina komunikačních symbolů,
- $G_i = (Q \cup N_i, T, P_i, S_i)$, je dvousměrná k -lineární PC komponenta taková, že $Q \cap (N_i \cup T) = \emptyset$ pro všechna $i = 1, \dots, n$.

Nechť $q\text{-}P_i \subseteq P_i$ označuje množinu všech pravidel z P_i obsahujících komunikační symbol. Konfigurace je n -tice tvaru (x_1, \dots, x_n) , kde $x_i \in (Q \cup N_i \cup T)^*$, $1 \leq i \leq n$. Počáteční konfigurace, s , je definována jako $s = (S_1, \dots, S_n)$. Nechť Θ označuje množinu všech konfigurací Γ . Pro všechna $x \in \Theta$ a $i = 1, \dots, n$ $i\text{-}x$ značí i -tou komponentu. Jestliže tedy $x = (x_1, \dots, x_i, \dots, x_n)$, potom $i\text{-}x = x_i$. Pro všechna $x \in \Theta$ definujeme mapování ${}_x\theta$ nad množinou $\{i\text{-}x : 1 \leq i \leq n\}$ jako ${}_x\theta(i\text{-}x) = z_1 z_2 \dots z_{|i\text{-}x|}$, kde pro všechna $1 \leq h \leq |i\text{-}x|$ platí: jestliže pro nějaké $q_j \in Q$, $i = 1, \dots, n$, $\mathbf{sym}(i\text{-}x, h) = q_j$ a zároveň $\mathbf{alph}(j\text{-}x) \cap Q = \emptyset$, potom $z_h = j\text{-}x$; jinak $(\mathbf{sym}(i\text{-}x, h) \notin Q$ nebo $\mathbf{alph}(j\text{-}x) \cap Q \neq \emptyset$) $z_h = \mathbf{sym}(i\text{-}x, h)$.

Nechť $y, x \in \Theta$. Píšeme:

- $y \xrightarrow{d} x$ v Γ jestliže $i\text{-}y \xrightarrow{d} i\text{-}x$ v G_i nebo $i\text{-}y = i\text{-}x$ pro $i\text{-}y, i\text{-}x \in T^*$, pro všechna $i = 1, \dots, n$;
- $y \xrightarrow{r} x$ v Γ jestliže $i\text{-}y \xrightarrow{r} i\text{-}x$ v G_i nebo $i\text{-}y = i\text{-}x$ pro $i\text{-}y, i\text{-}x \in \{S_i\} \cup T^*$, pro všechna $i = 1, \dots, n$;
- $y \xrightarrow{q} x$ v Γ jestliže $i\text{-}x = {}_y\theta(i\text{-}y)$ v G_i , pro všechna $i = 1, \dots, n$.

Neformálně můžeme říci, že Γ pracuje ve třech výpočetních módech $d \Rightarrow, r \Rightarrow, q \Rightarrow$, které reprezentují přímou derivaci, redukci a komunikaci. Formálně, nechť $k \geq 1$, $\alpha_j \in \Theta$, $1 \leq i \leq k$ a $\alpha_0 \xrightarrow{*} \alpha_1 \xrightarrow{k_2} \alpha_2 \dots \alpha_{k-1} \xrightarrow{k} \alpha_k$, kde $k_m \in \{d, r, q\}$, $1 \leq m \leq k$; píšeme $\alpha_0 \xrightarrow{*} \alpha_k$, jestliže $k_1 = d$ a každé $k_p \in \{d, r, q\}$, $1 \leq p \leq k-1$, splňuje:

- jestliže $k_p = q$, pak $k_{p+1}, k_{p-1} \in \{d, r\}$ a $k_{p+1} \neq k_{p-1}$,
- jestliže $k_p \in \{d, r\}$, pak $k_{p+1} \in \{q, k_p\}$.

Neformálně řečeno, po provedení komunikačního kroku, Γ změní svůj výpočetní mód derivace na redukci a naopak. Jazyk $L(\Gamma)$ je definován

$$L(\Gamma) = \{z \in T^* : s \xrightarrow{*} \alpha \text{ v } \Gamma, \text{ kde } z = \mathbf{del}(1\text{-}\alpha, S_1) \text{ pro } \alpha \in \Theta\}.$$

$L(\Gamma)$ obsahuje takové řetězce $z \in T^*$, které vzniknou výpočtem $s \xrightarrow{*} \alpha$ v Γ a následným vymazáním symbolů S_1 z výsledné větné formy první komponenty. Takový výpočet budeme nazývat úspěšným. Dvousměrnými metalineárními PC gramatickými systémy budeme označovat všechny dvousměrné k -lineární n -PC gramatické systémy, kde $k \geq 1$.

Nyní zavedeme několik speciálních pojmů pro dvousměrné k -lineární PC gramatické systémy.

Konečný index: Nechť $s \xrightarrow{*} \alpha$ je úspěšný výpočet v Γ , kde $\alpha \in \Theta$ a nechť $i \in \{1, \dots, n\}$. Pomocí $i\text{-index}(s \xrightarrow{*} \alpha)$ budeme označovat maximální počet nonterminálů ve větných formách vznikajících během úspěšného výpočtu v G_i . Jestliže pro každý úspěšný výpočet $s \xrightarrow{*} \xi$ v Γ , kde $\xi \in \Theta$, existuje $k \geq 1$ takové, že $i\text{-index}(s \xrightarrow{*} \xi) \leq k$, potom G_i je *konečného indexu*.

Jestliže je G_i konečného indexu, $index(G_i)$ značí minimální číslo h splňující $i-index(s \Rightarrow^* \xi) \leq h$, pro všechny úspěšné výpočty $s \Rightarrow^* \varpi$ v Γ , kde $\varpi \in \Theta$; $index(G_i) = \infty$ znamená, že G_i není konečného indexu. Jestliže G_j je konečného indexu pro všechna $j = 1, \dots, n$, potom je Γ konečného indexu a $index(\Gamma)$ označuje minimální číslo g splňující $index(G_l) \leq g$, pro všechna $l = 1, \dots, n$; $index(\Gamma) = \infty$ znamená, že Γ není konečného indexu.

Stupeň komunikace: Pro $s \Rightarrow^* x$ v Γ , kde $x \in \Theta$, $q-degree(s \Rightarrow^* x)$ označuje počet komunikačních kroků (${}_q \Rightarrow$) ve výpočtu $s \Rightarrow^* x$. Jestliže pro každý výpočet $s \Rightarrow^* \xi$ v Γ , kde $\xi \in \Theta$, existuje $k \geq 1$ takové, že $q-degree(s \Rightarrow^* \xi) \leq k$, potom má Γ konečný stupeň komunikace. Jestliže je Γ konečného stupně komunikace, $q-degree(\Gamma)$ označuje minimální číslo h splňující $q-degree(s \Rightarrow^* \xi) \leq h$, pro každý výpočet $s \Rightarrow^* \xi$ v Γ ; $q-degree(\Gamma) = \infty$ znamená, že Γ nemá konečný stupeň komunikace.

Centralizovaná verze: Γ je centralizovaný, pokud se v pravidlech P_i v $G_i = (N_i, T_i, P_i, S_i)$, pro všechna $i = 2, \dots, n$, nevyskytuje žádný komunikační symbol. Jinak řečeno, pouze P_1 obsahuje komunikační symboly, G_1 se nazývá *hlavní* a je jedinou komponentou, která vykonává komunikační krok.

Výsledky: Autor dokázal, že každý neunární rekurzivně spočetný jazyk je definován pomocí centralizovaného dvousměrného 3-lineárního 2-PC gramatického systému, $\Gamma = (\{Q_2\}, G_1, G_2)$ takového, že $index(G_1) = 2$, $index(G_2) = 3$ a $q-degree(\Gamma) = 1$ (viz [57]). Navíc první komponenta G_1 obsahuje pouze tři neterminály a pouze jediné pravidlo obsahující komunikační symbol. Toto tvrzení autor získal tak, že postupně dokázal následující tvrzení:

(1) Pro každý rekurzivně spočetný jazyk L existuje levě-rozšířená frontová gramatika Q splňující $L(Q) = L$.

(2) Necht' Q' je levě-rozšířená frontová gramatika. Potom existuje levě-rozšířená frontová gramatika, $Q = (V, T, W, F, s, R)$, taková, že $L(Q') = L(Q)$, $W = X \cup Y \cup \{1\}$, kde X, Y a $\{1\}$ jsou po dvou disjunktní množiny a každé pravidlo $(a, b, x, c) \in R$ splňuje buď $a \in V - T, b \in X, x \in (V - T)^*, c \in X \cup \{1\}$, nebo $a \in V - T, b \in Y \cup \{1\}, x \in T^*, c \in Y$.

(3) Necht' $Q = (V, T, W, F, s, R)$ je levě-rozšířená frontová gramatika taková, že $card(\mathbf{alph}(L(Q))) \geq 2$. Potom existuje centralizovaný dvousměrný 3-lineární 2-PC gramatický systém, $\Gamma = (\{Q_2\}, G_1, G_2)$, takový, že $L(\Gamma) = L(Q)$, $index(G_1) = 2$, $index(G_2) = 3$, $index(\Gamma) = 3$, $q-degree(\Gamma) = 1$. Navíc první komponenta, $G_1 = (\{Q_2\} \cup N_1, T, P_1, S_1)$ splňuje $card(N_1) = 3$ a $q-P_1 = \{A \rightarrow Q_2\}$.

(4) Necht' L je rekurzivně spočetný jazyk takový, že $\text{card}(\mathbf{alph}(L)) \geq 2$. Potom existuje centralizovaný dvousměrný 3-lineární 2-PC gramatický systém, $\Gamma = (\{Q_2\}, G_1, G_2)$, takový, že $L(\Gamma) = L$, $\text{index}(G_1) = 2$, $\text{index}(G_2) = 3$, $\text{index}(\Gamma) = 3$, $q\text{-degree}(\Gamma) = 1$ a hlavní komponenta, $G_1 = (\{Q_2\} \cup N_1, T, P_1, S_1)$ splňuje $\text{card}(N_1) = 3$ a $q\text{-}P_1 = \{A \rightarrow Q_2\}$.

Budoucí výzkum

Všechny výše uvedené gramatické systémy jsou většinou navrženy tak, že sice každá komponenta generuje jiný řetězec, ale pro generovaný jazyk je důležitý pouze řetězec generovaný první komponentou, ostatní slouží pouze jako kontrolní elementy. V této části práce je gramatický systém navržen tak, že pro výsledný jazyk jsou podstatné vygenerované řetězce ze všech komponent, přičemž je na závěr s nimi provedena jistá operace, například konkatence, a tím je získán výsledný řetězec.

Přesněji bychom mohli specifikovat takový gramatický systém následovně: Pro každé přirozené číslo n , n -generativní gramatický systém se skládá z n bezkontextových gramatik, které generují řetězce pomocí nejlevější derivace, což znamená, že v každém kroku je v dané větě formě přepsán vždy neterminální symbol, který je nejvíce vlevo. Těchto n nejlevějších derivací je řízeno n -tici neterminálů nebo n -tici pravidel. Tímto řízením gramatický systém generuje n -tici řetězců, se kterými jsou provedeny jisté základní operace a pomocí výsledku operace je definován generovaný jazyk. Mezi tyto operace patří především sjednocení, konkatence a výběr řetězce, který generuje první komponenta. Popíšme si nyní tyto gramatické systémy přesněji.

Definice: n -multigenerativní neterminálově synchronizovaný gramatický systém (n -MGN) je $n+1$ -tice $\Gamma = (G_1, G_2, \dots, G_n, Q)$, kde:

- $G_i = (N_i, T_i, P_i, S_i)$ je bezkontextová gramatika pro všechna $i = 1, \dots, n$,
- Q je konečná množina *kontrolních n -tic neterminálů* tvaru (A_1, A_2, \dots, A_n) , kde $A_i \in N_i$ pro všechna $i = 1, \dots, n$.

Necht' $\Gamma = (G_1, G_2, \dots, G_n, Q)$ je n -MGN. Potom *multiforma* je n -tice $\chi = (x_1, x_2, \dots, x_n)$, kde $x_i \in (T_i \cup N_i)^*$ pro všechna $i = 1, \dots, n$.

Necht' $\Gamma = (G_1, G_2, \dots, G_n, Q)$ je n -MGN, necht' $\chi = (u_1A_1v_1, u_2A_2v_2, \dots, u_nA_nv_n)$, $\bar{\chi} = (u_1x_1v_1, u_2x_2v_2, \dots, u_nx_nv_n)$, jsou dvě multiformy, kde $A_i \in N_i$, $u_i \in T_i^*$, $v_i, x_i \in (N_i \cup T_i)^*$ pro všechna $i = 1, \dots, n$. Dále necht' $A_i \rightarrow x_i \in P_i$ pro všechna $i = 1, \dots, n$ a $(A_1, A_2, \dots, A_n) \in Q$. Pak říkáme, že χ *přímo derivuje* $\bar{\chi}$ a zapisujeme $\chi \Rightarrow \bar{\chi}$. Standardním způsobem je přímý derivační krok \Rightarrow rozšířen na \Rightarrow^* , \Rightarrow^+ a \Rightarrow^k pro libovolné $k \geq 0$.

Nechť $\Gamma = (G_1, G_2, \dots, G_n, Q)$ je n -MGN. Potom n -jazyk generovaný Γ , n - $L(\Gamma)$, je definován

$$n-L(\Gamma) = \{(w_1, w_2, \dots, w_n) : (S_1, S_2, \dots, S_n) \Rightarrow^* (w_1, w_2, \dots, w_n), w_i \in T_i^* \text{ pro } i = 1, \dots, n\}$$

Nechť $\Gamma = (G_1, G_2, \dots, G_n, Q)$ je n -MGN. Jazyk generovaný Γ v módu *sjednocení*, $L_{union}(\Gamma)$, je definován

$$L_{union}(\Gamma) = \{w : (w_1, w_2, \dots, w_n) \in n-L(\Gamma), w \in \{w_i : i = 1, \dots, n\}\}$$

Nechť $\Gamma = (G_1, G_2, \dots, G_n, Q)$ je n -MGN. Jazyk generovaný Γ v módu *konkatenace*, $L_{conc}(\Gamma)$, je definován jako:

$$L_{conc}(\Gamma) = \{w_1 w_2 \dots w_n : (w_1, w_2, \dots, w_n) \in n-L(\Gamma)\}$$

Nechť $\Gamma = (G_1, G_2, \dots, G_n, Q)$ je n -MGN. Jazyk generovaný Γ v módu *první komponenty*, $L_{first}(\Gamma)$, je definován jako

$$L_{first}(\Gamma) = \{w_1 : (w_1, w_2, \dots, w_n) \in n-L(\Gamma)\}$$

Příklad: $\Gamma = (G_1, G_2, Q)$, kde

- $G_1 = (\{S_1, A_1\}, \{a, b, c\}, \{S_1 \rightarrow aS_1, S_1 \rightarrow aA_1, A_1 \rightarrow bA_1c, A_1 \rightarrow bc\}, S_1)$
- $G_2 = (\{S_2, A_2\}, \{d\}, \{S_2 \rightarrow S_2A_2, S_2 \rightarrow A_2, A_2 \rightarrow d\}, S_2)$
- $Q = \{(S_1, A_1), (S_2, A_2)\}$

je 2-multigenerativní neterminálově synchronizovaný gramatický systém (2-MGN).

Poznamenejme, že pro daný 2-MGN platí:

- $2-L(\Gamma) = \{(a^n b^n c^n, d^n) : n \geq 1\}$
- $L_{union}(\Gamma) = \{a^n b^n c^n : n \geq 1\} \cup \{d^n : n \geq 1\}$
- $L_{conc}(\Gamma) = \{a^n b^n c^n d^n : n \geq 1\}$
- $L_{first}(\Gamma) = \{a^n b^n c^n : n \geq 1\}$

Definice: n -multigenerativní pravidlově synchronizovaný gramatický systém (n -MGR) je $n+1$ -tice

$$\Gamma = (G_1, G_2, \dots, G_n, Q), \text{ kde}$$

- $G_i = (N_i, T_i, P_i, S_i)$ je bezkontextová gramatika pro všechna $i = 1, \dots, n$,
- Q je konečná množina *řídících n -tic pravidel* tvaru (p_1, p_2, \dots, p_n) , kde $p_i \in P_i$ pro všechna $i = 1, \dots, n$.

Multiforma pro n-MGR je definována shodně s multiformou pro n-MGN.

Nechť $\Gamma = (G_1, G_2, \dots, G_n, Q)$ je n-MGR, necht' $\chi = (u_1A_1v_1, u_2A_2v_2, \dots, u_nA_nv_n)$, $\bar{\chi} = (u_1x_1v_1, u_2x_2v_2, \dots, u_nx_nv_n)$, jsou dvě multiformy, kde $A_i \in N_i$, $u_i \in T_i^*$, $v_i, x_i \in (N_i \cup T_i)^*$ pro všechna $i = 1, \dots, n$. Dále necht' $p_i: A_i \rightarrow x_i \in P_i$ pro všechna $i = 1, \dots, n$ a $(p_1, p_2, \dots, p_n) \in Q$. Pak říkáme, že χ přímo derivuje $\bar{\chi}$ a zapisujeme $\chi \Rightarrow \bar{\chi}$.

Definice n-jazyka generového n-MGR je shodná s definicí jazyka generovaným n-MGS. Stejně tak jazyky v různých módech pro n-MGR jsou definovány shodně s jazyky generovanými pomocí n-MGS.

Příklad: $\Gamma = (G_1, G_2, Q)$, kde:

- $G_1 = (\{S_1, A_1\}, \{a, b, c\}, \{\mathbf{1}: S_1 \rightarrow aS_1, \mathbf{2}: S_1 \rightarrow aA_1, \mathbf{3}: A_1 \rightarrow bA_1c, \mathbf{4}: A_1 \rightarrow bc\}, S_1)$
- $G_2 = (\{S_2\}, \{d\}, \{\mathbf{1}: S_2 \rightarrow S_2, \mathbf{2}: S_2 \rightarrow S_2S_2, \mathbf{3}: S_2 \rightarrow d\}, S_2)$
- $Q = \{(\mathbf{1}, \mathbf{1}), (\mathbf{2}, \mathbf{2}), (\mathbf{3}, \mathbf{3}), (\mathbf{4}, \mathbf{3})\}$

je 2-multigenerativní pravidlově synchronizovaný gramatický systém (2-MGR). Poznamenejme, že pro daný 2-MGR platí:

- $2-L(\Gamma) = \{(a^n b^n c^n, d^n): n \geq 1\}$
- $L_{union}(\Gamma) = \{a^n b^n c^n: n \geq 1\} \cup \{d^n: n \geq 1\}$
- $L_{conc}(\Gamma) = \{a^n b^n c^n d^n: n \geq 1\}$
- $L_{first}(\Gamma) = \{a^n b^n c^n: n \geq 1\}$

Výsledky: Ve spolupráci se svým doktorandem R. Lukášem autor dokázal následující výsledky, které se nyní připravují pro publikaci:

- Libovolný n-MGN lze převést na ekvivalentní n-MGR a naopak.
- Necht' $L(n\text{-MGR})_X$ označuje třídu všech jazyků generovaných pomocí n-MGR v módu X , kde $X \in \{union, conc, first\}$ a $L(n\text{-MGN})_X$ třídu všech jazyků generovaných pomocí n-MGN v módu X , kde $X \in \{union, conc, first\}$. Mezi těmito třídami platí vztah

$$RE = L(2\text{-MGR})_X = L(2\text{-MGN})_X \text{ pro libovolné } X \in \{union, conc, first\}$$

II.6 Odkazy

- [1] Beigel, R. and Floyd, R. W.: *The Language of Machines*; Freeman, New York, (1994) ISBN 0-7167-8266-9.
- [2] Csuhaj-Varju, E., Dassow, J., Kelemen, J., and Gh. Paun, Gh.: *Grammar Systems: A Grammatical Approach to Distribution and Cooperation*, Gordon and Breach, London, 1994 ISBN 803-9-80266-1.
- [3] Csuhaj-Varju, E., Dassow, J., Kelemen, J., and Gh. Paun, Gh.: Eco-Grammar Systems: A grammatical framework for life-like interactions, *Artificial Life* 3 (1996), 27-38.
- [4] Dassow, J. and Paun, Gh. : *Regulated Rewriting in Formal Language Theory*, Springer, Berlin, 1989 ISBN 3-05-500408-6.
- [5] Gonczarowski, J. and Warmuth, M.K.: Scattered and Context-Sensitive Rewriting, *Acta Informatica* 20 (1989), 391-411.
- [6] Habel, A. : Hyperedge Replacement: Grammars and Languages, *LNCS* 643, Springer, Berlin, 1992.
- [7] Hein, J. L.: *Discrete Structures, Logic, and Computability*. Jones and Bartlett, London, (1995) ISBN 0-86720-477-x.
- [8] Fernau, H.: Scattered Context Grammars with Regulation, *Analele Univ. Bucuresti: Matematica-Informatica*, XLV, 1 (1996).
- [9] Greibach, S. and Hopcroft, J.E.: Scattered Context Grammars, *J. Comp. Syst. Sci.* 3 (1969), 233-247.
- [10] M. Ito (ed.): *Words, Languages, and Combinatorics*, World Scientific, Singapore, 1992 ISBN 872-5-10675-1.
- [11] Kleijn, H. C. M. and Rozenberg, G.: Multi Grammars, *Inter. J. Comput. Math* 12 (1983), 177-201.
- [12] Mayer, O.: Some restrictive Devices for Context-Free Grammars, *Information and Control* 20 (1972), 69-92.
- [13] Meduna, A. : A Note on Exponential Density of ETOL Languages, *Kybernetika* 22 (1986), 514-518.
- [14] Meduna, A. : Evaluated Grammars, *Acta Cybernetika* 8 (1987), 169-176.
- [15] Meduna, A. : Characterization of the Chomsky Hierarchy through Sequential-Parallel Grammars, *Rostock. Math. Kolloq.* 32 (1987), 4 - 14.
- [16] Meduna, A. and Horvath, G. : On State Grammars, *Acta Cybernetica* 8 (1988), 237 -245.
- [17] Meduna, A. : Context Free Derivations on Word Monoids, *Acta Informatica* 27, 781 - 786, 1990.
- [18] Meduna, A. : Generalized Forbidding Grammars, *International Journal of Computer Mathematics* 36 (1990), 31 - 38.

- [19] Meduna, A. : Global Context Conditional Grammars, *J. Inform. Process. Cybern.* 27 (1991), 159 - 165.
- [20] Meduna, A. : Symbiotic EOL Systems , *Acta Cybernetica* 12 (1992), 164 -172.
- [21] Meduna, A. : A Formalization of Sequential, Parallel, and Continuous Rewriting, *International Journal of Computer Mathematics* 39 (1993), 24 -32.
- [22] Meduna, A. : Canonical Scattered Rewriting, *International Journal of Computer Mathematics* 51 (1993), 122 - 129.
- [23] Meduna, A. and Csuhaaj-Varju, E.: Grammars with Context Conditions, *EATCS Bulletin* 32 (1993), 112 - 124.
- [24] Meduna, A. : Matrix Grammars under Leftmost and Rightmost Restrictions, in vol. *Mathematical Linguistics and Related Topics* (Gh. Paun, ed.), The Publ. House of the Romanian Academy, Bucharest, (1994), 243 - 257.
- [25] Meduna, A., Crooks, C., and Sarek, M.: Syntactic Complexity of Regulated Rewriting, *Kybernetika* 30(1994), 177 - 186.
- [26] Meduna, A., and Gopalaratnam, M. : On Semi-Conditional Grammars with Productions Having either Forbidding or Permitting Conditions, *Acta Cybernetica* 11 (1994), 309 - 323.
- [27] Meduna, A. : Syntactic Complexity of Scattered Context Grammars, *Acta Informatica* 32 (1995), 285 - 298.
- [28] Meduna, A. : A Trivial Method of Characterizing the Family of Recursively Enumerable Languages by Scattered Context Grammars, *EATCS Bulletin* 56 (1995), 104 - 106.
- [29] Meduna, A. : Syntactic Complexity of Context-Free Grammars over Word Monoids, *Acta Informatica* 33 (1996), 457 - 462.
- [30] Meduna, A. : Four-Nonterminal Scattered Context Grammars Characterize the Family of Recursively Enumerable Languages, *International Journal of Computer Mathematics* 63 (1997), 67-83.
- [31] Meduna, A. : On the Number of Nonterminals in Matrix Grammars with Leftmost Derivations, *LNCS* 1217 (1997), 27 - 38.
- [32] Meduna, A. : Six-Nonterminal Multi-Sequential Grammars Characterize the Family of Recursively Enumerable Languages, *International Journal of Computer Mathematics* 65, 1997, 179 -189.
- [33] Meduna, A. : Economical Transformation of Phrase-Structure Grammars to Scattered Context Grammars, *Acta Cybernetica* 13, 1998, 225 - 242.
- [34] Meduna, A. : Uniform Rewriting Based on Permutations, *International Journal of Computer Mathematics* 69, 1998, 57 - 74.
- [35] Meduna, A. : Descriptive Complexity of Multi-Continues Grammars, *Acta Cybernetica* 14, 1998, 375-384.

- [36] Meduna, A. : Prefix Pushdown Automata, *International Journal of Computer Mathematics* 71, Gordon and Breach, London, 1999, s. 215-228.
- [37] Meduna, A. : Generative Power of Three-Nonterminal Scattered Context Grammars, *Theoretical Computer Science* 246, 2000, 276-284
- [38] Meduna, A. : Automata and Languages: Theory and Applications, Springer, London, (2000)
- [39] Meduna, A. and Kolář, D.: Descriptive complexity of multi-parallel grammars with respect to the number of nonterminals, In: *Grammars and Automata for String Processing: from Mathematics and Computer Science to Biology, and Back*, Francis and Taylor, 2000, 724 – 732
- [40] Meduna, A. and Kolář, D.: Regulated Pushdown Automata, *Acta Cybernetica* 18, 2000, 653-664
- [41] Meduna, A.: Generative Power of Three-Nonterminal Scattered Context Grammars, *Theoretical Computer Science*, 2000, 625-631
- [42] Meduna, A.: Terminating Left-Hand Sides of Scattered Context Grammars, *Theoretical Computer Science*, 2000, 423-427
- [43] Meduna, A.: Uniform Generation of Languages by Scattered Context Grammars, *Fundamenta Informaticae* 44, 2001, 231-235
- [44] Meduna, A. and Kolář, D.: Homogenous Grammars with a Reduced Number of Non-Context-Free Productions, *Information Processing Letters* 81, 2002, 253-257
- [45] Meduna, A. and Vurm, P.: Multisequential Grammars with Homogeneous Selectors, *Fundamenta Informaticae* 34, 2001, 1 - 7
- [46] Meduna, A. and Kolář, D.: One-Turn Regulated Pushdown Automata and Their Reduction, *Fundamenta Informaticae*, Vol. 2002, No. 16, Amsterdam, NL, p. 399-405
- [47] Meduna, A. and Švec, M.: Reduction of Simple Semi-Conditional Grammars with Respect to the Number of Conditional Productions, *Acta Cybernetica* 15, 2002, 353-360
- [48] Meduna, A.: Descriptive Complexity of Scattered Rewriting and Multirewriting: An Overview, *Journal of Automata, Languages and Combinatorics* 7, 2002, 571-577
- [49] Meduna, A.: Simultaneously One-Turn Two-Pushdown Automata, *International Journal of Computer Mathematics* 80, 2003, 679-687
- [50] Meduna, A. and Fernau, H.: A Simultaneous Reduction of Several Measures of Descriptive Complexity in Scattered Context Grammars, *Information Processing Letters* 86, 2003, 235-240
- [51] Meduna, A. and Fernau, H.: On the Degree of Scattered Context-Sensitivity, *Theoretical Computer Science* 290, 2003, 2121-2124
- [52] Meduna, A. and Švec, M.: Descriptive Complexity of Generalized Forbidding Grammars, *International Journal of Computer Mathematics* 80, 2003, 11-17

- [54] Meduna, A. and Švec, M.: Forbidding EOL Systems, *Theoretical Computer Science* 54, 2003, 256-276
- [55] Meduna Alexander: Coincidental Extention of Scattered Context Languages, *Acta Informatica* 39, 2003, 307-314
- [56] Meduna Alexander: Simultaneously One-Turn Two-Pushdown Automata, *International Journal of Computer Mathematics* 80, 2003, 679-687
- [57] Meduna, A.: Two-Way Metalinear PC Grammar Systems and Their Descriptive Complexity, *Acta Cybernetica* 16, 2004, 385 – 397
- [58] Meduna, Alexander and Vitek, M.: New Language Operations in the Formal Language Theory, *Schedae Informaticae* 13, 2004, 31-42
- [59] Meduna, A. and Švec, M.: *Grammars with Context Conditions and Their Applications*, Wiley, 225 stran, 2005 (v tisku)
- [60] Milgram, D. and Azriel Rosenfeld, A.: A Note on Scattered Context Grammars, *Information Processing Letters*, 1(2), 1971, 47-50.
- [61] Paun, Gh.: On simple Matrix Languages versus Scattered Context Languages, *RAIRO* 16 (1982), 245-253.
- [62] Paun, Gh.: *Artificial Life: Grammatical Models*, Black Sea University Press, Bucharest, Romania (1995) ISBN 973-0-00131-6.
- [63] Paun, Gh.: *Mathematical Linguistics and Related Topics*, The Publ. House of the Romanian Academy, Bucharest, Romania (1995) ISBN 502-0-10675-1.
- [64] Rozenberg, G. : Selective Substitution Grammars (Towards a Framework for Rewriting Systems), Part I: Definitions and Examples, *J. Inform. Process. Cybern.* 13 (1977), 455-463.
- [65] Rozenberg, G., and Salomaa, A.: *The Mathematical Theory of L Systems*, Academic Press, New York, (1980) ISBN 0-12-597140-0.
- [66] Rozenberg, G. and Salomaa, A. (eds.): *Handbook of Formal Languages*, Volume 1 through 3, Springer, (1997) ISBN 3-540-60649-1.
- [67] Salomaa, A., *Computation and Automata*, Cambridge University Press, Cambridge, England, (1985) ISBN 0-521-30245-5.
- [68] H. J. Shyr, *Free Monoids and Languages*, Hon Min Book Comp., Taichung, 1991.
- [69] Ullman, J. D.: Computational Aspects of VLSI, *Computer Science Press*, Rockville, Maryland, (1984) ISBN 3-523-00233-1.
- [70] Virkkunen, V.: On Scattered Context Grammars, *Acta Universitatis Ouluensis*, Series A, Mathematica 6 (1973), 75-82.

III. APLIKACE

Tato část tezí demonstruje aplikace formálních modelů diskutovaných teoreticky v části II. Tyto modely se využívají v řadě oblastí informatiky jako např. v matematické lingvistice, ekonomicky orientované informatice či informatické technologii využívané informatikou. Matematická lingvistika pomocí těchto modelů specifikuje komplikované syntaktické struktury přirozených jazyků. Ekonomicky orientovaná informatika využívá těchto modelů k simulování nejrůznějších ekonomických modelů, aby předvíдалa vývoje na trhu v závislosti na různých podmínkách. Využití v biologii si nyní probereme podrobněji.

Uvažujme typický molekulární organismus tvořený z několika částí, jejichž chování se mírně odlišuje. Za krátký časový úsek se každá z těchto skupin vyvíjí svým specifickým způsobem. Během této doby je tedy jejich vývoj relativně nezávislý. Občas však mezi sebou tyto skupiny komunikují. Tato komunikace obvykle ovlivňuje budoucí chování organismu jako celku. Simulace takového organismu vede k použití odpovídajícího gramatického systému, o kterém se diskutovalo v podkapitole II.3. Takové gramatické modely nám umožňují zcela obecně simulovat několik skupin molekul. Pokud je některý molekulární organismus modifikován přidáním nebo odebráním jedné ze skupin, můžeme stále využít stejný gramatický systém pro simulaci takto vzniklého organismu. Výsledky, které se týkají redukce takových systémů (viz II.5), nám umožňují modelovat takový molekulární organismus velmi úsporným způsobem.

Zavedme si nyní L gramatiky s kontextovými podmínkami (srov. II.4) a demonstrujme, jak je lze v biologii využít třemi zcela konkrétními studii.

Definice: L gramatika s kontextovými podmínkami je trojice

$$G = (T, P, S),$$

kde

- T je úplná abeceda gramatiky G ,
- P je konečná množina přepisovacích pravidel tvaru $(A \rightarrow x, U, V)$, kde $A \in T$, $x \in T^*$, $U, V \subseteq T$
- S je startovací řetězec, $S \in T^*$

Nechť pro nějaké $n \geq 1$, $(A_i \rightarrow x_i, U_i, V_i) \in P$, $1 \leq i \leq n$, $U_i \subseteq \text{alph}(A_1A_2\dots A_n)$, $V_i \cap \text{alph}(A_1A_2\dots A_n) = \emptyset$, pro každé $i = 1, \dots, n$. Pak řekneme, že G provede a *přímý derivační krok* z $A_1A_2\dots A_n$ do $x_1x_2\dots x_n$, což zapisujeme $A_1A_2\dots A_n \Rightarrow^* x_1x_2\dots x_n$. Jako obvykle symbolem \Rightarrow^* označujeme tranzitivní a reflexivní uzávěr \Rightarrow . Jazyk generovaný gramatikou G , $L(G)$, je definován

$$L(G) = \{ w : S \Rightarrow^* w \}$$

Studie 1: Buněčný organismus. Uvažujme buněčný organismus, ve kterém se každá buňka rozdělí na dvě během jednoho kroku zdravého vývoje (viz Obrázek 1). Kdykoliv virus infikuje některé buňky, vývoj organismu stagnuje, dokud není úplně vyléčen. Během této periody každá buňka reprodukuje jen sebe sama bez vytváření nových buněk, dokud se celý organismus opět neuzdraví (viz Obrázek 2). Pro formální zápis takového vývoje použijeme L gramatiku s kontextovými podmínkami $G = (\{A, B\}, P, A)$, ve které označíme zdravou buňku A a infikovanou buňku B , přičemž chování tohoto organismu popíšeme následujícími pravidly z P

$(A \rightarrow AA, \emptyset, \{B\}), (A \rightarrow A, \{B\}, \emptyset), (A \rightarrow B, \emptyset, \emptyset), (B \rightarrow B, \emptyset, \emptyset), (B \rightarrow A, \emptyset, \emptyset)$

Definice: Nechť $G = (T, P, S)$ je L gramatika s kontextovými podmínkami. G se nazývá L gramatika se zakazujícími podmínkami, jestliže každé $(A \rightarrow x, U, V) \in P$ splňuje $U = \emptyset$. V těchto gramatikách pro jednoduchost píšeme $(A \rightarrow x, V) \in P$ místo $(A \rightarrow x, \emptyset, V) \in P$.

Studie 2: Červená řasa. Uvažujme L gramatiku bez kontextových podmínek, $G = (V, P, 1)$, kde $V = \{1, 2, 3, 4, 5, 6, 7, 8, [,]\}$ a P obsahuje pravidla

$1 \rightarrow 23, 2 \rightarrow 2, 3 \rightarrow 24, 4 \rightarrow 54, [\rightarrow [, 5 \rightarrow 6, 6 \rightarrow 7, 7 \rightarrow 8[1], 8 \rightarrow 8,] \rightarrow]$

Zdravý vývoj: Z biologického hlediska tato gramatika G popisuje vývoj červené řasy. Výrazy v závorkách reprezentují větve, jejichž pozice je označena 8. Tyto větve jsou zobrazeny jako střídající se části vyrůstající ze základní větve. Obrázek 3 popisuje biologickou interpretaci vývojových stádií formálně popsanych následující derivací, která obsahuje 13 řetězců odpovídající stavům (a) až (m) na obrázku

$1 \Rightarrow_G 23$
 $\Rightarrow_G 224$
 $\Rightarrow_G 2254$
 $\Rightarrow_G 22654$
 $\Rightarrow_G 227654$
 $\Rightarrow_G 228[1]7654$
 $\Rightarrow_G 228[23]8[1]7654$
 $\Rightarrow_G 228[224]8[23]8[1]7654$
 $\Rightarrow_G 228[2254]8[224]8[23]8[1]7654$
 $\Rightarrow_G 228[22654]8[2254]8[224]8[23]8[1]7654$
 $\Rightarrow_G 228[227654]8[22654]8[2254]8[224]8[23]8[1]7654$
 $\Rightarrow_G 228[228[1]7654]8[227654]8[22654]8[2254]8[224]8[23]8[1]7654$

Částečné odumření: Nyní předpokládejme, že se červená řasa dostane do nezdravého prostředí, ve kterém jen některé její části přežijí a zbytek odumře. Tento proces začíná od nejnovějších buněk na koncích větví, které jsou příliš mladé a slabé, aby přežily, a dostává se až k vnitřním buňkám označeným 2 a 8, které jsou dostatečně silné pro přežití v nových podmínkách.. Tento vývoj je popsán L gramatikou se zakazujícími podmínkami, jejíž pravidla jsou

$(1 \rightarrow 23, W), (1' \rightarrow 2', \{3', 4', 5', 6', 7'\}), (2 \rightarrow 2, W), (2' \rightarrow 2', \emptyset), (3 \rightarrow 24, W),$
 $(3' \rightarrow \varepsilon, \{4', 5', 6', 7'\}), (4 \rightarrow 54, W), (4' \rightarrow \varepsilon, \emptyset), (5 \rightarrow 6, W), (5' \rightarrow \varepsilon, \{4'\}), (6 \rightarrow 7, W),$
 $(6' \rightarrow \varepsilon, \{4', 5'\}), (7 \rightarrow 8[1], W), (7' \rightarrow \varepsilon, \{4', 5', 6'\}), (8 \rightarrow 8, W), ([\rightarrow [, \emptyset), ([\rightarrow], \emptyset),$
a pro každé $a \in V, (a \rightarrow a', \emptyset)$ a $(a' \rightarrow a', \emptyset)$, kde $W = \{a' : a \in V\}$.

Obrázek 4 graficky interpretuje proces odumírání korespondující k této derivaci

$$\begin{aligned}
1 &\Rightarrow_G^* 228[228[1]7654]8[227654]8[22654]8[2254]8[224]8[23]8[1]7654 \\
&\Rightarrow_G 2'2'8'[2'2'8'[1']7'6'5'4']8'[2'2'7'6'5'4']8'[2'2'6'5'4']8'[2'2'5'4']8'[2'2'4'] \\
&\quad 8'[2'3']8'[1']7'6'5'4' \\
&\Rightarrow_G 2'2'8'[2'2'8'[1']7'6'5']8'[2'2'7'6'5']8'[2'2'6'5']8'[2'2'5']8'[2'2']8'[2'3'] \\
&\quad 8'[1']7'6'5' \\
&\Rightarrow_G 2'2'8'[2'2'8'[1']7'6']8'[2'2'7'6']8'[2'2'6']8'[2'2']8'[2'2']8'[2'3']8'[1']7'6' \\
&\Rightarrow_G 2'2'8'[2'2'8'[1']7']8'[2'2'7']8'[2'2']8'[2'2']8'[2'2']8'[2'3']8'[1']7' \\
&\Rightarrow_G 2'2'8'[2'2'8'[1']]8'[2'2']8'[2'2']8'[2'2']8'[2'2']8'[2'3']8'[1'] \\
&\Rightarrow_G 2'2'8'[2'2'8'[1']]8'[2'2']8'[2'2']8'[2'2']8'[2'2']8'[2']8'[1'] \\
&\Rightarrow_G 2'2'8'[2'2'8'[2']]8'[2'2']8'[2'2']8'[2'2']8'[2'2']8'[2']8'[2']
\end{aligned}$$

Degenerace: Uvažujme nyní situaci, kdy červená řasa degeneruje. Během tohoto procesu se mohou pouze buňky hlavní větve dělit do větví, zatímco ostatní buňky tuto vlastnost postrádají. Takové chování je možné popsat L gramatikou se zakazujícími podmínkami obsahující pravidla

$(1 \rightarrow 23, \emptyset), (2 \rightarrow 2, \emptyset), (3 \rightarrow 24, \emptyset), (4 \rightarrow 54, \emptyset), (5 \rightarrow 6, \emptyset), (6 \rightarrow 7, \emptyset), (7 \rightarrow 8[1],$
 $\{D\}), (8 \rightarrow 8, \emptyset), ([\rightarrow [, \emptyset), ([\rightarrow], \emptyset), (7 \rightarrow 8[D], \emptyset), (D \rightarrow ED, \emptyset), (E \rightarrow E, \emptyset).$

Obrázek 5 popisuje degenerační proces popsaný následující derivací

$$\begin{aligned}
1 &\Rightarrow_G^* 227654 \\
&\Rightarrow_G 228[D]7654 \\
&\Rightarrow_G 228[ED]8[D]7654 \\
&\Rightarrow_G 228[E^2D]8[ED]8[D]7654 \\
&\Rightarrow_G 228[E^3D]8[E^2D]8[ED]8[D]7654 \\
&\Rightarrow_G 228[E^4D]8[E^3D]8[E^2D]8[ED]8[D]7654 \\
&\Rightarrow_G 228[E^5D]8[E^4D]8[E^3D]8[E^2D]8[ED]8[D]7654
\end{aligned}$$

$$\begin{aligned} &\Rightarrow_G 228[E^6D]8[E^5D]8[E^4D]8[E^3D]8[E^2D]8[ED]8[D]7654 \\ &\Rightarrow_G 228[E^7D]8[E^6D]8[E^5D]8[E^4D]8[E^3D]8[E^2D]8[ED]8[D]7654 \\ &\Rightarrow_G 228[E^8D]8[E^7D]8[E^6D]8[E^5D]8[E^4D]8[E^3D]8[E^2D]8[ED]8[D]7654 \end{aligned}$$

Studie 3: Implementace gramatik modelující růst rostlin. V této závěrečné studii prezentujeme ukázkou z nové knihy *Grammars with Context Conditions and Their Applications*, Wiley, 225 stran (v tisku), kterou autor napsal spolu se svým doktorandem M. Švecem. Tato ukázkou popisuje implementaci L gramatik s kontextovými podmínkami, které simulují vývoj některých rostlin. Růst rostlin do značné míry závisí na množství vody a minerálů absorbovaných kořeny a přepravených do části nad zemí. L gramatiky kontextovými podmínkami kontrolují tok zdrojů, které jsou získány u kořenů a jako substance jsou distribuovány až k vrcholu. Vrchol přijímá substance a pokud je množství nashromážděných zdrojů větší než předdefinovaný práh, vrchol se rozdvojí a vznikne tak nová postranní větev. Distribuce substancí závisí na množství vrcholů, které je nutno vyživovat, a na typu části rostliny. Ve zmíněné knize je navržen jednoduchý programovací jazyk, který umožňuje takové chování rostliny naprogramovat. Obrázky 6 a 7 prezentují ukázky vytvořené tímto programovacím jazykem, přičemž Obrázek 7 je výsledkem tohoto programu.

$$\begin{aligned} \text{axiom: } &N(1)I(1, \text{straight}, 0, 1)A(1) \\ p_1: &N(k) \rightarrow N(k+1) \\ p_2: &I(id, t, e, c) ? N(k), A(id) \\ &: id == 1 \\ &\rightarrow I(id, t, \sigma_0 2^{(k-1)\eta^k}, 1) \\ p_3: &I(id, t, e, c) ? N(k), I(id_s, t_s, e_s, c_s), I(id_l, t_l, e_l, c_l) \\ &: id == 1 \wedge id_s == 2 * id \wedge id_l == 2 * id + 1 \\ &\rightarrow I(id, t, \sigma_0 2^{(k-1)\eta^k}, c_s + c_l) \\ p_4: &I(id, t, e, c) ? I(id_p, t_p, e_p, c_p), I(id_s, t_s, e_s, c_s), I(id_l, t_l, e_l, c_l) \\ &: id_p == \lfloor id / 2 \rfloor \wedge id_s == 2 * id \wedge id_l == 2 * id + 1 \\ &\rightarrow I(id, t, \delta(t, e_p, c_p, c), c_s + c_l) \\ p_5: &I(id, t, e, c) ? I(id_p, t_p, e_p, c_p), A(id_a) \\ &: id_p == \lfloor id / 2 \rfloor \wedge id_a == id \\ &\rightarrow I(id, t, \delta(t, e_p, c_p, c), 1) \\ p_6: &A(id) ? I(id_p, t_p, e_p, c_p) \\ &: id == id_p \wedge e_p \geq e_{th} \\ &\rightarrow [+ (\alpha) I(2 * id + 1, \text{lateral}, e_p * (1 - \lambda), 1) A(2 * id + 1)] \\ &/(\pi) I(2 * id, \text{straight}, e_p * \lambda, 1) A(2 * id) \end{aligned}$$

IV PŘÍLOHY

IV.1 Vybrané publikace autora

Následující seznam obsahuje pouze vybrané autorovy knižní kapitoly a časopisecké recenzované publikace na mezinárodní úrovni, které autor sám považuje za důležité. Úplný seznam publikací autor předložil hodnotící komisi a je také uveden na <http://www.fit.vutbr.cz/~meduna>.

[Meduna, 86]

Meduna, A. : A Note on Exponential Density of ETOL Languages, *Kybernetika* 22 (1986), 514-518

[Meduna, 87a]

Meduna, A. : Evaluated Grammars, *Acta Cybernetika* 8 (1987), 169-176

[Meduna, 87b]

Meduna, A. : Characterization of the Chomsky Hierarchy through Sequential-Parallel Grammars, *Rostock. Math. Kolloq.* 32 (1987), 4 - 14

[Meduna and Horvath, 88]

Meduna, A. and Horvath, G. : On State Grammars, *Acta Cybernetica* 8 (1988), 237 -245

[Meduna, 90a]

Meduna, A. : Context Free Derivations on Word Monoids, *Acta Informatica* 27, 781 - 786, 1990

[Meduna, 90b]

Meduna, A. : Generalized Forbidding Grammars, *International Journal of Computer Mathematics* 36 (1990), 31 - 38

[Meduna, 91]

Meduna, A. : Global Context Conditional Grammars, *J. Inform. Process. Cybern.* 27 (1991), 159 - 165

[Meduna, 92]

Meduna, A. : Symbiotic EOL Systems , *Acta Cybernetika* 12 (1992), 164 -172

[Meduna, 93a]

Meduna, A. : A Formalization of Sequential, Parallel, and Continuous Rewriting, *International Journal of Computer Mathematics* 39 (1993), 24 - 32

[Meduna, 93b]

Meduna, A. : Canonical Scattered Rewriting, *International Journal of Computer Mathematics* 51 (1993), 122 - 129

[Meduna and Csuhaj-Varju, 93]

Meduna, A. and Csuhaj-Varju, E.: Grammars with Context Conditions, *EATCS Bulletin* 32 (1993), 112 - 124

[Meduna, 94]

Meduna, A. : Matrix Grammars under Leftmost and Rightmost Restrictions, in vol. *Mathematical Linguistics and Related Topics* (Gh. Paun, ed.), The Publ. House of the Romanian Academy, Bucharest, (1994), 243 - 257

[Meduna, Crooks, and Sarek, 94]

12. Meduna, A., Crooks, C., and Sarek, M.: Syntactic Complexity of Regulated Rewriting, *Kybernetika* 30(1994), 177 - 186

[Meduna and Gopalaratnam, 94]

Meduna, A., and Gopalaratnam, M. : On Semi-Conditional Grammars with Productions Having either Forbidding or Permitting Conditions, *Acta Cybernetica* 11 (1994), 307 - 323

[Meduna, 95a]

Meduna, A. : Syntactic Complexity of Scattered Context Grammars, *Acta Informatica* 32 (1995), 285 - 298

[Meduna, 95b]

Meduna, A. : A Trivial Method of Characterizing the Family of Recursively Enumerable Languages by Scattered Context Grammars, *EATCS Bulletin* 56 (1995), 104 - 106

[Meduna, 96]

Meduna, A. : Syntactic Complexity of Context-Free Grammars over Word Monoids, *Acta Informatica* 33 (1996), 457 - 462

[Meduna, 97a]

Meduna, A. : Four-Nonterminal Scattered Context Grammars Characterize the Family of Recursively Enumerable Languages, *International Journal of Computer Mathematics* 63 (1997), 67-83

[Meduna, 97b]

Meduna, A. : On the Number of Nonterminals in Matrix Grammars with Leftmost Derivations, *LNCS 1217* (1997), 27 - 38

[Meduna, 97c]

Meduna, A. : Six-Nonterminal Multi-Sequential Grammars Characterize the Family of Recursively Enumerable Languages, *International Journal of Computer Mathematics* 65, 1997, 179 -189

[Meduna, 98b]

Meduna, A. : Economical Transformation of Phrase-Structure Grammars to Scattered Context Grammars, *Acta Cybernetica* 13, 1998, 225 - 242

[Meduna, 98c]

Meduna, A. : Uniform Rewriting Based on Permutations, *International Journal of Computer Mathematics* 69, 1998, 57 - 74

[Meduna, 98d]

Meduna, A. : Descriptive Complexity of Multi-Continues Grammars, *Acta Cybernetica* 13, 1998, 375-384

[Meduna99a]

Meduna, A. : Prefix Pushdown Automata, *International Journal of Computer Mathematics* 71, 1999, 215-228

[Meduna99b]

Meduna, A. : Terminating Left-Hand Sides of Scattered Context Productions, *Theoretical Computer Science* 237, 1999, 567-601

[Meduna00a]

Meduna, A. : Generative Power of Three-Nonterminal Scattered Context Grammars, *Theoretical Computer Science* 246, 2000, 276-284

[Meduna00b]

Meduna, A. : Automata and Languages: Theory and Applications, Springer, London, (2000)

[Meduna and Kolář, 00a]

Meduna, A. and Kolář, D.: Descriptive complexity of multi-parallel grammars with respect to the number of nonterminals, Grammars and Automata for String Processing: from Mathematics and Computer Science to Biology, and Back, Francis and Taylor, 2000, 724 – 732

[Meduna and Kolář, 00b]

Meduna, A. and Kolář, D.: Regulated Pushdown Automata, Acta Cybernetica 18, 2000,. 653-664

[Meduna00c]

Meduna, A.: Generative Power of Three-Nonterminal Scattered Context Grammars, Theoretical Computer Science, 2000, 625-631

[Meduna00d]

Meduna, A.: Terminating Left-Hand Sides of Scattered Context Grammars, Theoretical Computer Science, 2000, 423-427

[Meduna01d]

Meduna, A.: Uniform Generation of Languages by Scattered Context Grammars, Fundamenta Informaticae 44, 2001, 231-235

[Meduna and Kolář, 01]

Meduna, A. and Kolář, D.: Homogenous Grammars with a Reduced Number of Non-Context-Free Productions, Information Processing Letters 81, 2002, 253-257

[Meduna and Vurm, 01]

Meduna, A. and Vurm, P.: Multisequential Grammars with Homogeneous Selectors, Fundamenta Informaticae 34, 2001, 1 - 7

[Meduna and Kolář, 02]

Meduna, A. and Kolář, D.: One-Turn Regulated Pushdown Automata and Their Reduction, Fundamenta Informaticae, Vol. 2002, No. 16, Amsterdam, NL, p. 399-405

[Meduna and Švec, 02]

Meduna, A. and Švec, M.: Reduction of Simple Semi-Conditional Grammars with Respect to the Number of Conditional Productions, *Acta Cybernetica* 15, 2002, 353-360

[Meduna02a]

Meduna, A.: Descriptive Complexity of Scattered Rewriting and Multirewriting: An Overview, *Journal of Automata, Languages and Combinatorics* 7, 2002, 571-577

[Meduna03b]

Meduna, A.: Simultaneously One-Turn Two-Pushdown Automata, *International Journal of Computer Mathematics* 80, 2003, 679-687

[Meduna and Fernau, 03a]

Meduna, A. and Fernau, H.: A Simultaneous Reduction of Several Measures of Descriptive Complexity in Scattered Context Grammars, *Information Processing Letters* 86, 2003, 235-240

[Meduna and Fernau, 03b]

Meduna, A. and Fernau, H.: On the Degree of Scattered Context-Sensitivity, *Theoretical Computer Science* 290, 2003, 2121-2124

[Meduna and Švec, 03a]

Meduna, A. and Švec, M.: Descriptive Complexity of Generalized Forbidding Grammars, *International Journal of Computer Mathematics* 80, 2003, 11-17

[Meduna and Švec, 03b]

Meduna, A. and Švec, M.: Forbidding EOL Systems, *Theoretical Computer Science* 54, 2003, 256-276

[Meduna, 03a]

Meduna Alexander: Coincidental Extension of Scattered Context Languages, *Acta Informatica* 39, 2003, 307-314

[Meduna, 03b]

Meduna Alexander: Simultaneously One-Turn Two-Pushdown Automata, *International Journal of Computer Mathematics* 80, 2003, 679-687

[Meduna, 04a]

Meduna, A.: Two-Way Metalinear PC Grammar Systems and Their Descriptive Complexity, *Acta Cybernetica* 16, 2004, 385 – 397

[Meduna and Vitek, 04b]

Meduna, Alexander and Vitek, M.: New Language Operations in the Formal Language Theory, *Schedae Informaticae* 13, 2004, 31-42

[Meduna and Švec, 05]

Meduna, A. and Švec, M.: *Grammars with Context Conditions and Their Applications*, Wiley, 225 stran, 2005 (v tisku)

IV.2 English Summary

About the Author

Alexander Meduna was born June 27, 1957 in Olomouc, the Czech Republic. From 1978 to 1982, he attended the Palacký University, receiving his M.S. in Computer Science in 1982. He received his Ph.D. in Computer Science at the Brno University of Technology in 1988. He defended his habilitation thesis and became an associate professor at the Brno University of Technology in 1999.

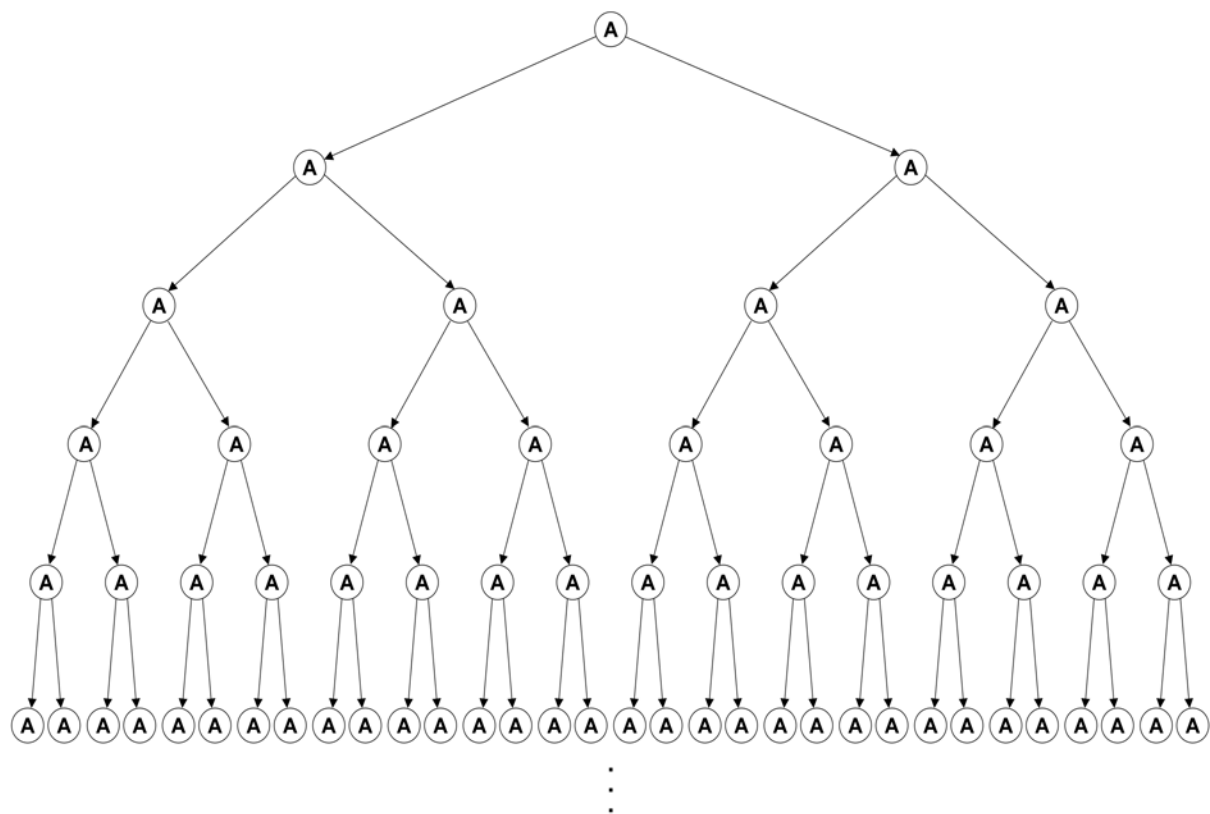
In 1988, based on some of his publications, he was invited to the University of Missouri—Columbia, USA, where he taught several computer science classes for almost a decade. Most of these classes dealt with automata, formal languages, computability, and compilers. In 1996, he returned to the Czech Republic to do research at the Computing Center of the Brno University of Technology. From there, he moved to the Faculty of Information Technology at the Brno University of Technology, where he has since taught various computer science classes, such as compilers and theoretical computer science.

About the Author's Research

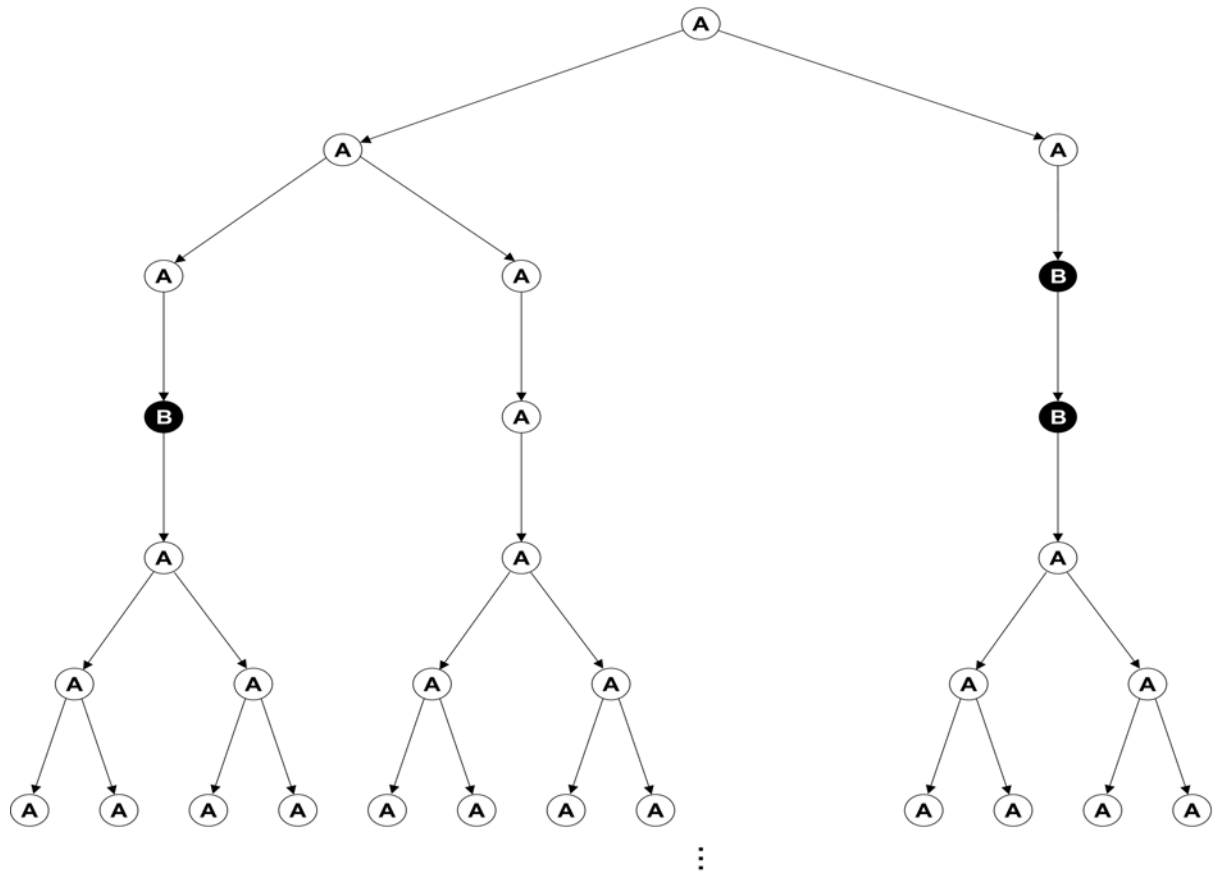
The present thesis discusses the author's investigation of formal languages and their models. The author's results concerning parallel, regulated, and cooperating formal models are given. This thesis describes various modifications of the formal models, which significantly increase the power of context-free grammars; in fact, most of these models are as powerful as the Turing machines. Thus, it comes as no surprise that the author pays a special attention to them.

Perhaps most importantly, this thesis concentrates its investigation on the reduction of various components in the formal models under discussion. Most significantly, it studies how to achieve this reduction without any decrease of the generative power. By doing so, it actually makes these models more succinct and economical, and this economization is obviously highly appreciated both from a theoretical and a practical viewpoint as demonstrated in the conclusion.

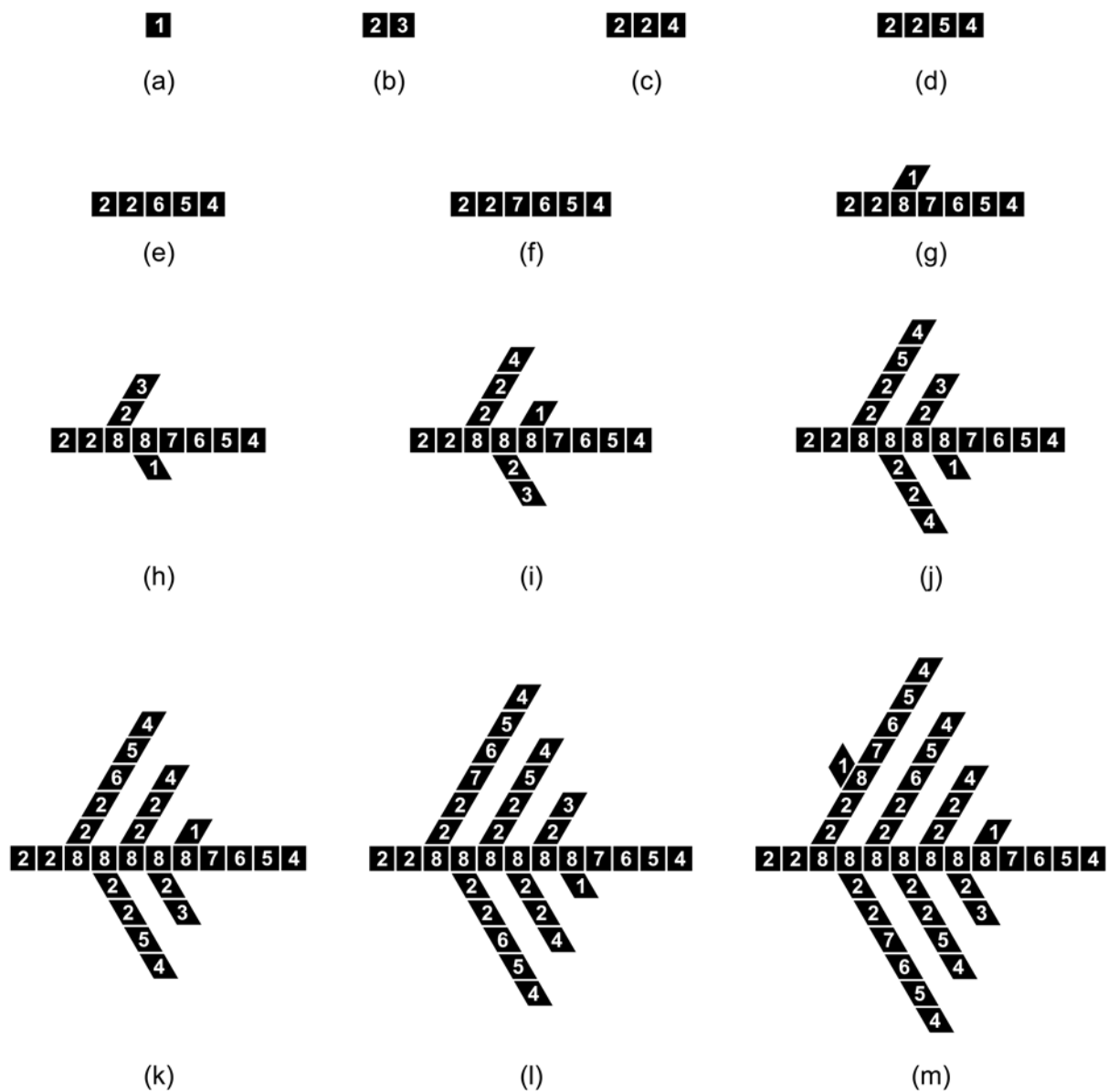
IV.3 Obrázky



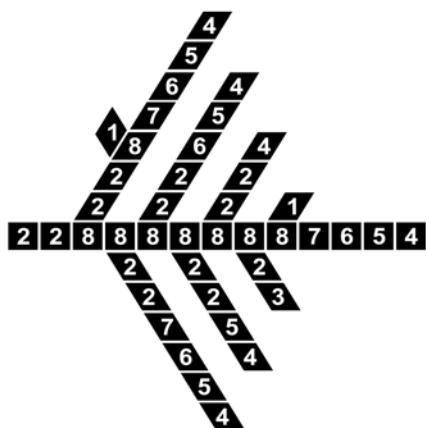
Obrázek 1. Zdravý vývoj buněčného organismu



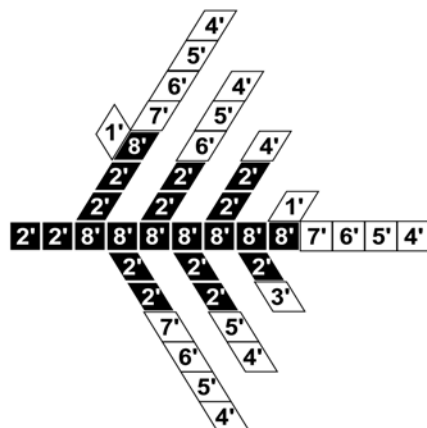
Obrázek 2. *Vývoj buněčného organismu infikovaného virem*



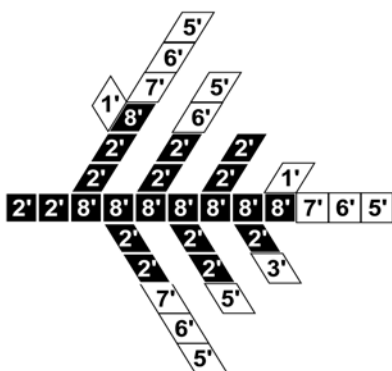
Obrázek 3. Zdravý růst řasy



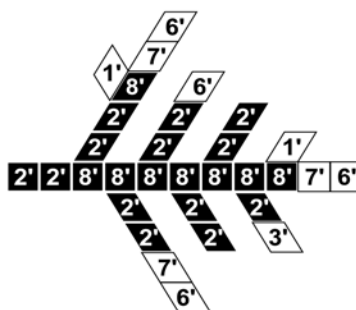
(a)



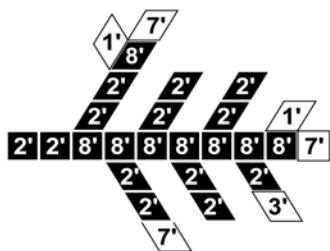
(b)



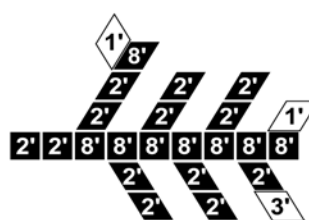
(c)



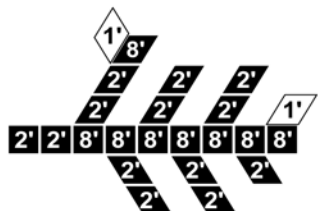
(d)



(e)



(f)



(g)



(h)

Obrázek 4. Částečné odumření řasy

2 2 7 6 5 4

(a)

2 2 8 7 6 5 4

(b)

2 2 8 8 7 6 5 4

(c)

2 2 8 8 8 7 6 5 4

(d)

2 2 8 8 8 8 7 6 5 4

(e)

2 2 8 8 8 8 8 7 6 5 4

(f)

2 2 8 8 8 8 8 8 7 6 5 4

(g)

2 2 8 8 8 8 8 8 8 7 6 5 4

(h)

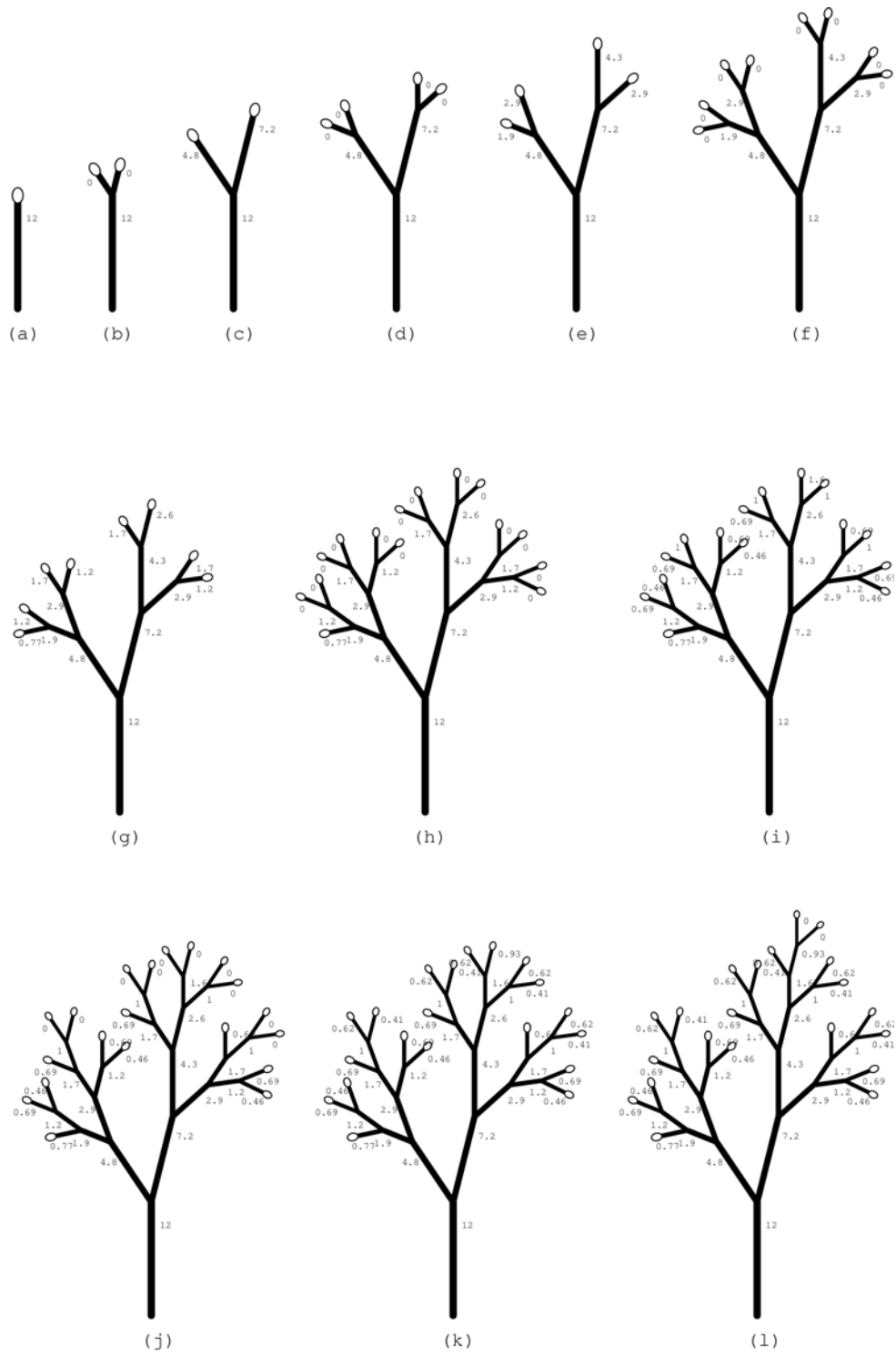
2 2 8 8 8 8 8 8 8 8 7 6 5 4

(i)

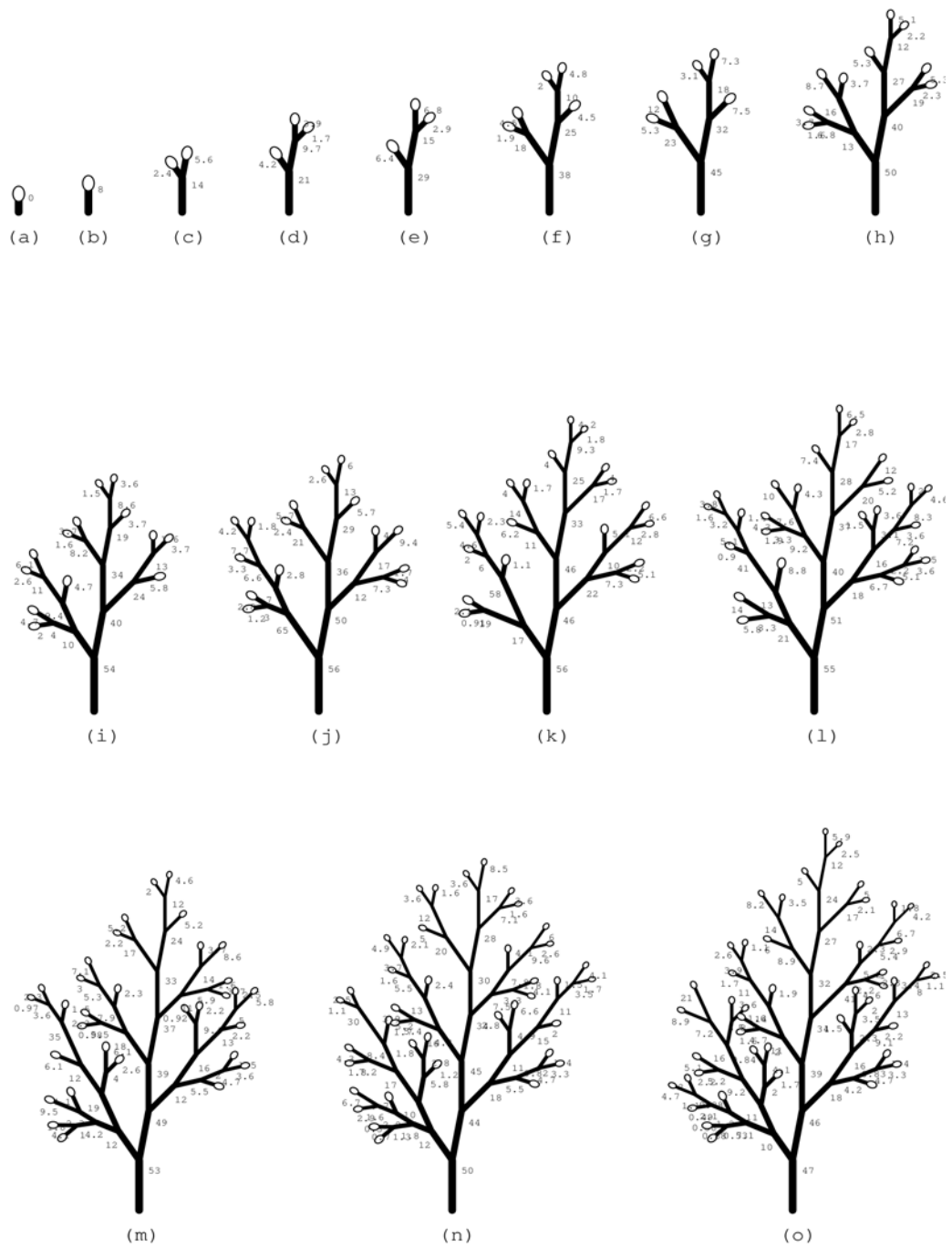
2 2 8 8 8 8 8 8 8 8 8 7 6 5 4

(j)

Obrázek 5. Degenerace řasy



Obrázek 6. Ukázka růstu rostliny



Obrázek 7. *Růst rostliny vytvořený programem ze závěru části III*