



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

MULTIDIMENSIONÁLNÍ AUTOMATY A JEJICH JAZYKY

MULTIDIMENSIONAL FINITE AUTOMATA AND THEIR LANGUAGES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

ZDENĚK HLADÍK

VEDOUcí PRÁCE

SUPERVISOR

prof. RNDr. ALEXANDER MEDUNA, CSc.

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů

Akademický rok 2016/2017

Zadání diplomové práce

Řešitel: **Hladík Zdeněk, Bc.**

Obor: Bezpečnost informačních technologií

Téma: **Multidimensionální automaty a jejich jazyky**
Multi-Dimensional Automata and Their Languages

Kategorie: Teoretická informatika

Pokyny:

1. Dle pokynů vedoucího se seznamte s pokročilou teorií n-dimensionálních jazyků a automatů, které je přijímají.
2. Zaveďte nový typ n-dimensionálních konečných automatů dle pokynů vedoucího.
3. Studujte vlastnosti automatů z bodu 2 dle pokynů vedoucího.
4. Aplikujte a implementujte automaty z bodu 2 dle pokynů vedoucího.
5. Diskutujte další vývoj projektu.

Literatura:

- Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages, Volume 1-3, Springer, 1997, ISBN 3-540-60649-1
- Aho, A.V., Lam, M.S., Sethi, R., Ullman, J.D.: Compilers: Principles, Techniques, and Tools (2nd Edition), Pearson Education, 2006, ISBN 0-321-48681-1
- Dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

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

Vedoucí: **Meduna Alexander, prof. RNDr., CSc., UIFS FIT VUT**

Datum zadání: 1. listopadu 2016

Datum odevzdání: 24. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Tato práce obsahuje stručný přehled teorie dvoudimensionálních jazyků, která byla tématem předchozí práce autorů. Dále jsou za pomoci této teorie představeny jazyky třídimensionální a multidimensionální. Pro každý druh těchto formálních jazyků je představen jejich ekvivalent řetězce a operace s danými řetězci. Především jsou také pro každý druh těchto jazyků představeny také příslušné vyvinuté formální gramatiky a především automaty. Spolu s těmito pojmy je součástí práce i demonstrace vyvinuté aplikace simulující představené multidimensionální automaty.

Abstract

This work includes brief overview of the theory behind two-dimensional languages which was the main theme examined in previous work. Based upon this theory there are next chapters introducing a new concept of three-dimensional and multi-dimensional languages. For each type of language there is a definition of corresponding type of strings and string operations. Finally, for each type of language there are definitions of developed formal gramatics and automata. Along with theory, in work there is an demonstration of developed program, which provides simulation of introduced multidimensional automata.

Klíčová slova

multidimensionální jazyky, multidimensionální automaty, třídimensionální jazyky, třídimensionální automaty, multidimensionální buňka, formální gramatika, multidimensionální Turingův stroj

Keywords

multi-dimensional languages, multi-dimensional automata, three-dimensional languages, three-dimensional automata, multi-dimensional cell, formal gramatic, multi-dimensional Turing machine

Citace

HLADÍK, Zdeněk. *Multidimensionální automaty a jejich jazyky*. Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Meduna Alexander.

Multidimensionální automaty a jejich jazyky

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana profesora Alexandra Meduny. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Zdeněk Hladík
1. května 2017

Obsah

1 Úvod	4
2 Dvoudimensionální jazyky	6
2.1 Rozšíření dimensionality	6
2.2 Jazyky pro dvě dimenze	7
2.2.1 Obrazy	7
2.2.2 Operace s obrazy	8
2.2.3 Jazyky obrazů	10
2.3 Gramatiky	11
2.3.1 Gramatika s vertikálními a horizontálními pravidly	11
2.3.2 Bezkontextová dvoudimensionální gramatika - jednofázový přístup	13
2.3.3 Čistá (pure) bezkontextová dvoudimensionální gramatika	13
2.4 Regulární výrazy	15
3 Dvoudimensionální automaty	17
3.1 Čtyřsměrný automat	17
3.2 On-line teselační automat	20
3.3 Zapomínající dvoudimensionální automat	21
3.4 Sgraffito automat	22
3.5 Vyvinuté automaty	24
3.6 Praktické využití	26
4 Třídimensionální jazyky a automaty	29
4.1 Jazyky a tělesa	29
4.1.1 Tělesa	29
4.1.2 Operace pro tělesa	31
4.1.3 Třídimensionální jazyk	33
4.2 Gramatiky	34
4.2.1 Spatial grammars	35
4.2.2 Třífázová gramatika	35
4.2.3 Bezkontextová třídimensionální gramatika	37
4.3 Třídimensionální automaty	38
4.3.1 Třídimensionální Turingův stroj	38
4.3.2 Šestisměrný konečný automat	39
4.3.3 Třídimensionální celulární teselační automat	40
4.4 Praktické využití automatů	41

5	Multidimensionální jazyky	43
5.1	Řetězce jako buňky	43
5.1.1	Velikost buňky	44
5.1.2	Prázdná buňka	44
5.1.3	Omezení pro buňky	44
5.2	Multidimensionální operace	45
5.2.1	Operace konkatence	45
5.2.2	Operace rotace	46
5.2.3	Další operace	46
5.3	Multidimensionální jazyky	47
5.3.1	Operace nad jazyky	47
5.3.2	Uzávěry jazyků	48
5.4	Multidimensionální gramatiky	48
5.4.1	Multidimensionální hierarchická gramatika	48
5.4.2	Bezkontextová multidimensionální gramatika	49
6	Multidimensionální automaty	51
6.1	Pravidla pro automaty	51
6.2	Multidimensionální Turingův stroj	52
6.3	Vícsměrný multidimensionální konečný automat	53
6.4	Multidimensionální celulární automaty	55
6.5	Multidimensionální zapomínající automat	56
6.6	Využití multidimensionálních automatů	57
7	Simulátor multidimensionálních automatů	59
7.1	O aplikaci	59
7.1.1	Struktura projektu	59
7.1.2	Metriky projektu	61
7.2	Rozhraní aplikace	61
7.2.1	Editace automatu	61
7.2.2	Ovládání simulace	64
7.3	Implementované automaty	65
7.3.1	Multidimensionální Turingův stroj	66
7.3.2	Vícsměrný multidimensionální automat	66
8	Závěr	68
	Literatura	70
	Přílohy	75
A	Obsah CD	76
B	Příklady pro simulátor	77
B.1	Postup při testování příkladu	77
B.2	Příklady - Vícsměrný multidimensionální automat	77
B.2.1	Příklad 1	77
B.2.2	Příklad 2	78
B.2.3	Příklad 3	79

B.3	Příklady - Multidimensionální Turingův stroj	80
B.3.1	Příklad 1	80
B.3.2	Příklad 2	81
B.3.3	Příklad 3	82

Kapitola 1

Úvod

Tato práce se zabývá představením obecně nového pojmu pro teoretickou informatiku, jímž jsou multidimensionální jazyky a automaty. Konkrétně práce obsahuje teorii postupující od dvoudimensionálních jazyků až po n -dimensionální pro libovolné n .

V práci jsou postupně představeny dvoudimensionální jazyky, třídimensionální jazyky a multidimensionální jazyky, přičemž je kladen důraz na jednotnost pojmů a demonstraci postupu mezi počty dimenzí. S každým druhem jazyka jsou představeny řetězce s danou dimensionalitou, operace nad těmito řetězci a především formální prostředky pro definici daných jazyků, tedy gramatiky a automaty. Pro každý druh těchto automatů je také uvedeno několik potenciálních praktických oblastí, kde by automaty pracující nad řetězci s danou dimensionalitou mohly najít uplatnění.

Nejdříve je stručně představen pojem dvoudimensionálních řetězců, které jsou už déle diskutovaným tématem teoretické informatiky. Zároveň byla teorie dvoudimensionálních jazyků a automatů s důrazem na jejich praktické využití představena již v dřívější práci autorů. V příslušné kapitole je zaveden termín *obrazy*, označující dvoudimensionální podobu řetězců a operace pracující nad těmito obrazy.

Dále jsou uvedeny dvoudimensionální jazyky jako množiny obrazů. Ty mohou být definovány pomocí vybraných zástupců dvoudimensionálních gramatik, uvedených ve stejné kapitole. Zároveň s gramatikami jsou definovány regulární výrazy, které lze používat nad obrazy. Dalším představeným mechanismem jsou dvoudimensionální automaty. Pro ty je vyhrazena samostatná kapitola, kde je uvedeno pět zástupců, volících rozdílné přístupy k analýze obrazů. Jeden z těchto automatů je výsledkem dřívější práce autorů a demonstruje praktické využití.

V následující kapitole je použito poznatků získaných u dvoudimensionálních řetězců a jazyků pro definici jejich třídimensionálních obdůb. Tato transformace byla provedena autory a je prvním krokem k multidimensionálním jazykům. Jako obdoba řetězce je tedy ustanoveno *těleso*. Po vzoru předchozích kapitol jsou pro tato tělesa definovány některé operace a zavedeny příslušné jazyky. Dále jsou definováni a demonstrováni zástupci gramatik a automatů vyvinutých autory pro tuto práci. Jedná se o tři typy gramatik určených pro generování těles a tři typy automatů.

Jako obecně n -dimensionální řetězec je v následující kapitole představena *buňka*. Pro tuto buňku jsou poté definovány operace upravené pro libovolnou dimensionalitu buňky. U těchto operací je poznamenáno, že jejich druhů existuje nekonečně mnoho, jelikož během postupu mezi dimensionalitami byl v práci získán poznatek, že počet operací je úměrný počtu dimenzí, a to v jakémkoliv prostoru.

Tyto operace pak mohou vytvořit dále představené multidimensionální jazyky obsahující prvky s libovolnou dimensionalitou. Multidimensionální jazyky jsou tedy definovány jako nadřazené množiny všem jazykům s pevnou dimensionalitou, jelikož mohou obsahovat prvky kteréhokoliv z těchto podřazených jazyků.

Prostředkem ke generování těchto jazyků pak mohou být dále představení zástupci multidimensionálních gramatik. Takové gramatiky byly vyvinuty autory za účelem možnosti generování řetězců různé dimensionality. K analýze buněk s různou dimensionalitou pak může být využit některý z vyvinutých multidimensionálních automatů, které jsou v této práci definovány čtyři.

Pro demonstraci některých multidimensionálních automatů byla v rámci práce vyvinuta grafická aplikace, sloužící jako simulátor těchto automatů. V poslední kapitole je tato aplikace demonstrována, včetně popisu jejího rozhraní a rozboru implementace vybraných automatů. Aplikace umožňuje definici automatů dvou druhů pomocí grafického prostředí a simulaci jejich chodu na multidimensionálních buňkách.

Kapitola 2

Dvoudimensionální jazyky

Tato kapitola seznamuje čtenáře s teorií dvoudimensionálních jazyků, na které je postavena pozdější definice n -dimensionálních jazyků a automatů. V následující kapitole pak budou k této teorii připojeny i dvoudimensionální automaty. Tato látka není standardně přednášena v předmětech magisterského programu Fakulty informačních technologií VUT v Brně, jelikož se jedná o pokročilou část teoretické informatiky.

Látka je ale hlavní náplní předchozí práce autorů [11], z níž je tedy převzata většina definic a příkladů. Dalším významným zdrojem jsou články [9] a práce [8].

Kapitola je stěžejní pro další diskuzi o vícedimensionálních řetězcích a teorii jim příbuzné, jelikož se zatím jedná o jediné rozšíření dimensionalit řetězců. Teorie tří a vícedimensionálních řetězců je novým přínosem této práce. Proto je zde představeno velké množství nových teorémů a principů, které ukazuje ještě větší potenciální možnosti v n -dimensionálních jazycích.

2.1 Rozšíření dimensionalit

Jednoduchá definice řetězce a definice jazyků těchto řetězců byly v práci [1] rozšířeny o další dimenzi. Definice abecedy pak zůstává stejná, ze znaků této abecedy jsou ale tvořeny útvary s pokročilejší strukturou. Obecně lze počet dimenzí navyšovat libovolně, jak bude demonstrováno v dalších kapitolách.

Spolu s úpravou podoby prvků jazyků je nutné provádět i úpravy operací pro tyto prvky. Konkrétním příkladem může být operace *konkatenace*, která má úzký vztah k počtu dimenzí konkrétního jazyka. Je možné říci, že standardní jednodimensionální řetězec má dvě strany, k nimž lze pomocí konkatenace připojit další řetězec. Ve dvou a více dimenzích se ovšem tento počet stran navyšuje a je nutné vytvoření více druhů konkatenace.

Společně se vznikem více druhů konkatenace pak narůstá počet jazykových operací odvozených od konkatenace, stejně jako vznikají nové operace, které nemají obdobu v jednodimensionálních jazycích.

Je příhodné podotknout, že přidáváním dalších dimenzí do teorie formálních jazyků vzniká pomyslná hierarchie. Jednorozměrný řetězec je totiž i vícerozměrným řetězcem, pokud uvažujeme, že všechny dimenze řetězce kromě jedné byly omezeny na velikost 1.

2.2 Jazyky pro dvě dimenze

Pokud stanovíme počet dimenzí pro jazyky rovný dvěma, prvky těchto jazyků pak budou dvourozměrné matice symbolů použité abecedy. Pro tyto dvoudimensionální řetězce zavádíme název *obrazy*. Tento název plyne ze skutečnosti, že řetězce opravdu tvoří jakýsi obraz.

Pokud vezmeme libovolnou kresbu či fotografii, můžeme pak říci, že je daný objekt dvoudimensionálním řetězcem, složeným z bodů různých barev, které zde zastávají funkci symbolů. Stejně tak může být pochopena i klasická tabulka, jež obsahuje prvky abecedy.

Jelikož dvoudimensionální obrazy jsou člověku známé a jednoduché pro představení, komplexnost operací a definic je oproti obecně n -dimensionálním řetězcům značně menší, stejně jako pro třídimensionální řetězce, které budou analyzovány v 4.1.1.

Jak lze odvodit již ze znalosti podoby dvoudimensionálních jazyků, automaty pro dvoudimensionální jazyky lze využít především pro analýzu obrazu.

2.2.1 Obrazy

V tomto oddíle uvádíme definice a popisy převzaté z práce [11], které stanovují základní pojmy týkající se dvoudimensionálních řetězců.

Podotýkáme, že se jedná o jeden z více přístupů ke dvoudimensionálním řetězcům, konkrétně užívající *obdélníkové obrazy*. Existují i práce zabývající se například *lichoběžníkovými obrazy* definovanými odlišně.

Definice 1. *Dvoudimensionální řetězec* (nebo *obraz*) nad abecedou Σ je dvoudimensionální obdélníkové pole skládající se z elementů abecedy Σ .

Definice 2. Množina všech dvoudimensionálních řetězců nad abecedou Σ je značena Σ^{**} .

Pro daný obraz $p \in \Sigma^{**}$, necht $\ell_1(p)$ značí počet řádků obrazu p a $\ell_2(p)$ značí počet sloupců obrazu p . Dvojice $(\ell_1(p), \ell_2(p))$ je pak nazývána *velikost obrazu p* . *Prázdný obraz* je jediný obraz velikosti $(0,0)$ a je poté značen λ (pozn.: analogie prázdného řetězce ε). Obrazy velikosti $(n,0)$ a $(0,n)$ nejsou definovány.

Množina všech obrazů nad Σ o velikosti (m,n) , kde $m, n > 0$, je značena $\Sigma^{m \times n}$.

Dále, pokud $1 \leq i \leq \ell_1(p)$ a $1 \leq j \leq \ell_2(p)$, necht $p(i,j)$, nebo $p_{i,j}$ značí symbol v obrazu p se souřadnicemi (i,j) .

a	b	b	b
d	a	d	c
a	a	b	c
d	d	c	c
d	a	d	c

Tabulka 2.1: Obraz A

Obraz A, který je zobrazen tabulkou 2.1, je obraz velikosti $(5,4)$, tedy $\ell_1(A) = 5$ a $\ell_2(A) = 4$. Pak například $A_{1,2} = b$. Obraz je sestaven nad abecedou $\Sigma = \{a, b, c, d\}$.

Definice 3. Necht p je obraz velikosti (m,n) . *Blok* obrazu p je obraz p' , který je podmnožinou pole p . Pokud (m',n') je velikostí obrazu p' , pak $m' \leq m$ a $n' \leq n$ a existují celá čísla h,k ($h \leq m - m', k \leq n - n'$) taková, že $p'(i,j) = p(i+h, j+k)$ pro všechna $0 \leq i \leq m'$ a $0 \leq j \leq n'$.

Obraz B zobrazený tabulkou 2.2 o rozměrech $(3, 2)$ je *blokem* obrazu A. Hodnoty h a k z definice 3 jsou následující: $h, k = 1$.

Blok obrazu je tedy speciálním případem podřetězce. Podobu podřetězce je ve více dimenzích také vhodné diskutovat.

Například pro třídimensionální řetězce, tedy *tělesa* (viz 4.1.1), je pak podřetězec, či jinak řečeno *podtěleso*, reálnou třírozměrnou částí celku, tedy hmotnou komponentou.

a	d
a	b
d	c

Tabulka 2.2: Obraz B

2.2.2 Operace s obrazy

Jak bylo uvedeno v sekci 2.1, rozšíření dimensionalit řetězců se velmi projeví na definici operace konkaténace. Pro dvě dimenze je nutno zavést dva druhy konkaténace. Je zřejmé, že tento počet druhů konkaténací bude růst úměrně s počtem dimenzí.

Pro demonstraci *řádkové* a *sloupcové* konkaténace uvádíme příklady obrazů nad abecedou $\Sigma = \{a, b, c, d\}$ v tabulkách 2.3 a 2.4. Definice jsou opět převzaty z práce [11].

a	c	a
d	b	c
d	c	c

Tabulka 2.3: Obraz C

a	c	b
d	a	c
a	b	a

Tabulka 2.4: Obraz D

Definice 4. *Řádková konkaténace* obrazů C a D (značena $C \ominus D$) je dílčí operace, definovaná pouze pro $\ell_1(C) = \ell_1(D)$. Výsledek konkaténace je zobrazen tabulkou 2.5.

Definice 5. *Sloupcová konkaténace* obrazů C a D (značena $C \odot D$) je dílčí operace, definovaná pouze pro $\ell_2(C) = \ell_2(D)$. Výsledek konkaténace je zobrazen tabulkou 2.6.

a	c	a
d	b	c
d	c	c
a	c	b
d	a	c
a	b	a

Tabulka 2.5: Obraz $C \ominus D$

a	c	a	a	c	b
d	b	c	d	a	c
d	c	c	a	b	a

Tabulka 2.6: Obraz $C \odot D$

Jak je vidět, konkaténace mají logický vztah k rozměrům obrazů, kdy $\ell_1(p \ominus q) = \ell_1(p) + \ell_1(q)$ a $\ell_2(p \odot q) = \ell_2(p) + \ell_2(q)$.

Řádkovou a sloupcovou konkaténaci lze snadno najít v počítačové (či jiné) grafice. Jedná se o jednoduché spojování obrazů do větších celků. Prostřednictvím konkaténace jednotlivých pixelových bodů vznikají obrazce na monitoru.

Operací, jež nemá u jednodimensionálních řetězců obdobu, je operace *rotace*. Ta je definována a demonstrována následovně.

Definice 6. Necht C je obraz. *Rotace* (ve směru ručiček) obrazu C , značená jako C^R , je definována v tabulce 2.7.

d	d	a
c	b	c
c	c	a

Tabulka 2.7: Obraz C^R

Pro obrazy také zavádíme operaci *transpozice*, která je používána v matematické teorii týkající se matic. Jelikož je dvoudimensionální řetězec v podstatě maticí, lze tuto operaci použít.

Definice 7. Necht C je obraz. *Transpozice* obrazu C , značená jako C^T , je definována v tabulce 2.8.

a	d	d
c	b	c
a	c	c

Tabulka 2.8: Obraz C^T

V práci [33] byla zmíněna velmi zajímavá unární operace, značená symbolem \oplus .

Tato operace je definována nad množinou matic, jejichž prvky jsou obrazy nad danou abecedou.

Definice 8. Necht $P_{i,j}, i = 1, \dots, m, j = 1, \dots, n$ jsou obrazy nad abecedou Σ takové, že $\forall i \in \{1, \dots, m\} : \ell_1(P_{i1}) = \ell_1(P_{i2}) = \dots = \ell_1(P_{in})$ a zároveň $\forall j \in \{1, \dots, n\} : \ell_2(P_{1j}) = \ell_2(P_{2j}) = \dots = \ell_2(P_{mj})$. $\oplus[P_{ij}]_{m,n}$ je definováno jako $P_1 \oplus P_2 \oplus \dots \oplus P_m$, kde $P_k = P_{k1} \odot P_{k2} \odot \dots \odot P_{kn}$.

Výsledkem operace je tedy matice obsahující všechny obrazy stejných rozměrů, které byly zadány jako vstup operace.

Dále uvádíme speciální operaci definovanou v [9], a to operaci *projekce mapování*. Tato operace může sloužit k převodům do jiných abeced, nebo například k úpravě obrazu, jež by umožnila lepší případnou *komprimaci*. Dále lze pomocí projekce zachytit podobnost mezi obrazy či zdůraznit zákonitosti ve vzorech obrazu.

Pro následující definici necht Γ a Σ jsou dvě konečné abecedy a $\pi : \Gamma \rightarrow \Sigma$ je mapování, dále odkazované jako projekce.

Definice 9. Necht Γ a Σ jsou dvě konečné abecedy. Dále necht $p \in \Gamma^{**}$ je obraz. *Projekce mapování* π obrazu p je obraz $p' \in \Sigma^{**}$ takový, že $p'(i, j) = \pi(p(i, j))$, pro každé $1 \leq i \leq \ell_1(p)$ a $1 \leq j \leq \ell_2(p)$.

Výsledek operace projekce pro abecedu $\Delta = \{0, 1, 2, 3\}$ a mapování $\eta : \Sigma \rightarrow \Delta$, kde platí: $\eta(a) = 0, \eta(b) = 1, \eta(c) = 2, \eta(d) = 3$, použité na obraz C z tabulky 2.3, zapisujeme $\eta(C)$, je v tabulce 2.9

0	2	0
3	1	2
3	2	2

Tabulka 2.9: Obraz $\eta(C)$

2.2.3 Jazyky obrazů

Dvoudimensionální jazyky budou, jak lze očekávat, konečné či nekonečné množiny výše definovaných obrazů. Definice jsou převzaty z [9] a [11].

Podle [9], [36] a několika dalších prací ovšem zde nabývá *Chomského hierarchie* jistých, často radikálních, obměn. Například v článku [33] je předložen důkaz, že ne všechny jazyky obrazů přijímaných dvoudimensionálním konečným automatem jsou *bezkontextové*. Zároveň jsou v práci [32] podány důkazy o jiných uzávěrových vlastnostech jednotlivých tříd. Z toho lze vyvodit, že klasická *Chomského hierarchie* ve více dimenzích neplatí a nebudeme její existenci předpokládat.

Definice 10. *Dvoudimensionální jazyk* nad abecedou Σ je podmnožinou Σ^{**} .

Pro dvoudimensionální jazyky je samozřejmě možné použití *řádkové* a *sloupcové konkatenace*. Platí zde stejné zákonitosti jako pro klasickou operaci konkatenace a jednodimensionální řetězce a jazyky. Definice jsou následující:

Definice 11. Necht L_1, L_2 jsou dvoudimensionální jazyky nad abecedou Σ , pak *řádková konkatenace* L_1 a L_2 (značeno $L_1 \ominus L_2$) je definována následovně: $L_1 \ominus L_2 = \{p \ominus q | p \in L_1 \text{ a } q \in L_2\}$.

Definice 12. Necht L_1, L_2 jsou dvoudimensionální jazyky nad abecedou Σ , pak *sloupcová konkatenace* L_1 a L_2 (značeno $L_1 \odot L_2$) je definována následovně: $L_1 \odot L_2 = \{p \odot q | p \in L_1 \text{ a } q \in L_2\}$.

Stejně tak operace rotace provádí pro jazyk rotaci všech obrazů obsažených v jazyce. Je možno poznamenat, že lze dosáhnout cykličnosti libovolné množiny jazyka L , protože pro množinu (jazyk) $\{L \cup L^R \cup L^{RR} \cup L^{RRR}\}$ je operace *rotace* generátorem.

Zobrazení mezi množinami mající za operaci *rotaci* je bijektivní zobrazení.

Stejně tak pro operaci *transpozice* (značíme L^T) platí, že tuto operaci použijeme pro všechny obrazy v jazyce. Opět lze pro množinu $\{L \cup L^T\}$ a generátor operace transpozice tvrdit, že jde o cyklickou množinu. I zde se jedná o bijektivní zobrazení.

Obdobou operace *iterace jazyka*, známé z klasických jednodimensionálních jazyků, jsou operace *řádkového* a *sloupcového uzávěru jazyka*.

Definice 13. Necht L je jazyk. *Řádkový uzávěr* jazyka L , značen $L^{*\ominus}$ je definován jako $L^{*\ominus} = \bigcup_{i \geq 0} L^{i\ominus}$, kde $L^{0\ominus} = \lambda$, $L^{1\ominus} = L$, $L^{n\ominus} = L \ominus L^{(n-1)\ominus}$.

Definice 14. Necht L je jazyk. *Sloupcový uzávěr* jazyka L , značen $L^{*\odot}$ je definován jako $L^{*\odot} = \bigcup_{i \geq 0} L^{i\odot}$, kde $L^{0\odot} = \lambda$, $L^{1\odot} = L$, $L^{n\odot} = L \odot L^{(n-1)\odot}$.

Podotkněme možnost použití obou operací uzávěru v libovolném pořadí. Můžeme definovat následující: $L^{**} = (L^{*\ominus})^{*\odot} = (L^{*\odot})^{*\ominus}$. Tato notace je použitelná vzhledem k faktu, že Σ^{**} značí množinu všech obrazů nad abecedou Σ .

Vezměme pro příklad jazyk $L1$, jenž obsahuje všechny obrazy nad danou abecedou s přesně třemi sloupci. Pak $L1^{*\odot}$ je jazyk, jenž obsahuje veškeré obrazy s počtem sloupců rovným násobku tří.

2.3 Gramatiky

V současné době bylo již představeno mnoho způsobů generování jazyků obrazů pomocí gramatik. Toto téma se velmi úzce pojí s počítačovou grafikou a rozeznáváním vzorů v obraze. Existují proto možnosti použití například paralelních generujících pravidel, explicitních omezení izometričnosti obrazů a dalších vylepšení klasického pojmu gramatiky.

Převážně jsou ovšem probírány gramatiky *bezkontextové*, jelikož síla gramatik a automatů se ve dvou dimenzích různí od klasické *Chomského hierarchie*, jak bylo zmíněno v 2.2.3. Dále tedy budou probírány převážně bezkontextové gramatiky a jim podobné či jim napomáhající systémy.

Je příhodné zmínit, že tyto gramatiky jsou děleny na *izometrické* a *neizometrické*. Izometrické gramatiky vytvářejí stabilní geometrické tvary, v našem případě tedy obdélníkové obrazy, které byly představeny dříve. Neizometrické gramatiky ovšem mohou vytvořit a při derivacích měnit tvar vznikajícího řetězce, je tedy velmi mnoho možností geometrických tvarů řetězců ve výsledném jazyku příslušícím dané gramatice. V našem případě je ovšem většinou zajištěno generování obdélníkových řetězců, a to různými pomocnými principy.

Praktickou a zajímavou ukázkou, jak mohou být gramatiky použity pro grafiku, je jejich použití pro generování a rozeznávání vzorů v mapách, jak je představeno v [38]. V práci [31] je použito gramatik k rozeznávání matematických formulí, jde tedy o jakousi syntaktickou kontrolu a interpretaci obrazového vzorce.

Jelikož existuje velké množství dvoudimensionálních gramatik, poukazujeme pouze na některé práce ([37], [43], [42] a [44]) a uvádíme pouze několik příkladů, které dostatečně nastíní danou problematiku.

2.3.1 Gramatika s vertikálními a horizontálními pravidly

Jedná se o *pravou lineární gramatiku*, jež disponuje dvěma druhy *proměnných* (též *non-terminálů*) a *pravidel*. Gramatika je tedy obdobou klasické jednodimensionální *CFG*, ovšem s druhou dimenzí přináší, stejně jako operace konkatenace v 2.2.2, specializaci, a to konkrétně přechodových pravidel. Gramatika byla představena v článku [9].

Je jasné, že tato gramatika by pro n -dimensionální jazyky byla nezměrně složitá, jelikož s každou dimenzí navíc by musela být rozšířena n -tice, jež gramatiku zastupuje.

Typický je také způsob generování řetězců (obrazů) pomocí této gramatiky. Nejdříve je generován jednodimensionální řetězec σ pomocí horizontálních pravidel. Poté následuje generování obdélníkového obrazu pomocí vertikálních pravidel, zanechávající řetězec σ jako vrchní řádek výsledného obrazu. Je tak využito paralelního generování, což je princip použitý ve velké části výše uvedených prací.

Gramatika tedy postupuje skrz hierarchii dimenzí, proto ji dále označujeme pro odlišení jako *hierarchickou* gramatiku.

Definice 15. *Dvoudimenzionální hierarchická pravá lineární gramatika (2RLHG)* je definována jako sedmice $G = (V_h, V_v, \Sigma_I, \Sigma, S, R_h, R_v)$, kde:

V_h je konečná množina *horizontálních proměnných*,

V_v je konečná množina *vertikálních proměnných*,

$\Sigma_I \subseteq V_v$ je konečná množina *přechodných symbolů*,

Σ je konečná množina *terminálů*,

$S \in V_h$ je *počáteční symbol*,

R_h je konečná množina *horizontálních pravidel* ve tvaru $V \rightarrow AV'$ nebo $V \rightarrow A$, kde $V, V' \in V_h$ a $A \in \Sigma_I$,

R_v je konečná množina *vertikálních pravidel* ve tvaru $W \rightarrow aW'$ nebo $W \rightarrow a$, kde $W, W' \in V_v$ a $a \in \Sigma$.

Derivace je pak provedena ve dvou krocích. Nejdříve řetězcová gramatika $G_h = (V_h, \Sigma_I, S, R_h)$, která je podmnožinou dané gramatiky, generuje jazyk řetězců $H(G)$ nad abecedou přechodných symbolů Σ_I . Řetězce přítomné v jazyce $H(G)$ se stanou horními řádky obrazů výsledného jazyka.

V druhém kroku je každý přechodný symbol považován za samostatný *počáteční symbol* a je prováděno paralelní generování ve vertikálním směru. To se děje pomocí pravidel obsažených v R_v . Výsledkem jsou obrazy obdélníkového tvaru.

Je důležité, aby vertikální generování bylo prováděno *paralelně*, a to z toho důvodu, aby byla současně aplikována ukončující pravidla tvaru $W \rightarrow a$ v každém sloupci (aby všechny sloupce obrazu byly stejně dlouhé). Tím je zaručena zmíněná *izometričnost* gramatiky, jelikož jsou stále generovány pouze obdélníky.

Následuje příklad demonstrující generování obrázku pomocí výše definované gramatiky.

Příklad 1. Necht $G_1 = (V_h, V_v, \Sigma_I, \Sigma, S, R_h, R_v)$ je gramatika, kde:

- $V_h = \{S, T\}$;
- $V_v = \{A, B, C, D\}$;
- $\Sigma_I = \{A, B\}$;
- $\Sigma = \{\square, \blacksquare\}$;
- $R_h = \{S \rightarrow AT; T \rightarrow BS; T \rightarrow B\}$;
- $R_v = \{A \rightarrow \square C; C \rightarrow \blacksquare A; C \rightarrow \blacksquare; B \rightarrow \blacksquare D; D \rightarrow \square B; D \rightarrow \square\}$.

Výsledek je jazyk obsahující „šachovnicové“ obrazy, příklad těchto obrazů je znázorněn v tabulce 2.12.

■	□	■	□
□	■	□	■
■	□	■	□
□	■	□	■

Tabulka 2.10: Obraz $P \in L(G_1)$

Studie uzávěrových vlastností rodiny jazyků $\mathcal{L}(2RLG)$ pak jsou probírány například ve článku [28].

2.3.2 Bezkontextová dvoudimensionální gramatika - jednofázový přístup

Tato gramatika nepoužívá dvoufázový proces zmíněný u předchozí gramatiky, naopak generování probíhá pouze v jednom kroku.

Zároveň gramatika nabývá standardní podoby *čtveřice*. Tímto způsobem lze tedy později definovat s jistými úpravami i *vícedimensionální gramatiky* bez růstu velikosti n -tice gramatiky na neúnosnou velikost.

Definice gramatiky a příklad jsou převzaty z práce [33]. Také je možné podotknout, že ve spojitosti s autorovým jménem jsou v tomto oboru známé tzv. *Průša grammars*.

Definice 16. *Dvoudimensionální bezkontextová gramatika* G je čtveřice (V_N, V_T, S_0, P) , kde:

V_N je konečná množina *neterminálů*,

V_T je konečná množina *terminálů*,

$S_0 \in V_N$ je *počáteční symbol*,

P je konečná množina *přechodových pravidel* ve tvaru $N \rightarrow W$ nebo, kde $N \in V_N$ a $W \in (V_N \cup V_T)^{**} \setminus \{\lambda\}$.

Lze vidět, že jeden *non-terminál* lze pomocí derivačního kroku gramatiky přepsat na *blok obrazu*. Tento způsob by mohl být použit i u n -dimensionálních jazyků, kde by speciální *non-terminál* by přepsán na n -dimensionální těleso.

Tuto gramatiku můžeme dále demonstrovat na příkladu.

Příklad 2. Nechť L_1 je jazyk obrazů nad abecedou $\Sigma = \{0, 1\}$.

$$L_1 = \{O \mid O \in \{0, 1\}^{**} \wedge \exists i, j \in N : 1 < i < \ell_1(O) \wedge 1 < j < \ell_2(O) \wedge \forall x \in \{1, \dots, \ell_1(O)\}, y \in \{1, \dots, \ell_2(O)\} : O(x, y) = 0 \Leftrightarrow x \neq i \wedge y \neq k\}$$

L_1 je *bezkontextový*. Je generován pomocí bezkontextové gramatiky $G_a = (V_N, \Sigma, S, P)$, kde $V_N = \{S, A, V, H, M\}$ a množina P se skládá z následujících pravidel:

$$\begin{array}{c} A \quad V \quad A \\ S \rightarrow H \quad 1 \quad H, \quad A \rightarrow M, \quad A \rightarrow AM, \quad M \rightarrow 0, \quad M \rightarrow \begin{array}{c} 0 \\ M \end{array}, \\ A \quad V \quad A \\ V \rightarrow 1, \quad V \rightarrow \begin{array}{c} 1 \\ V \end{array}, \quad H \rightarrow 0, \quad H \rightarrow 0H. \end{array}$$

Non-terminál A pak generuje jazyk $\{0\}^{**} \setminus \{\lambda\}$, M generuje jednosloupčové obrazy obsahující pouze číslici 0 , V generuje jednosloupčové obrazy obsahující pouze číslici 1 a H generuje jednořádkové obrazy složené pouze z číslic 1 .

Je vhodné zmínit *neizometričnost* gramatiky, bez kontroly by výsledný obraz mohl nabývat jakéhokoliv tvaru a ne jen obdélníkového, jehož chceme dosáhnout.

2.3.3 Čistá (pure) bezkontextová dvoudimensionální gramatika

Jedná se o další přístup, v tomto případě uplatňující teorii *čistých (pure) gramatik* v dvoudimensionální oblasti. Čisté gramatiky byly představeny v práci [27].

Původní práce zabývající se touto teorií [45] představila kombinaci znalosti dvoudimensionálních řetězců a k jejich generování použitých upravených čistých gramatik. Látka

představená v této práci je ovšem převzata z novější práce [21], jež dále diskutuje tuto teorii.

Možné rozšíření těchto typů gramatik stejně jako v 2.3.1 znesnadňuje fakt, že n-tice představující gramatiku má svou velikost závislou na počtu použitých dimenzí. Opět je použito odlišné množiny pravidel (tzv. *tabulek pravidel*) pro každou dimenzi, což by u obecně n-dimensionální gramatiky vedlo k neurčitelné velikosti n-tice. Jedná se ovšem o zajímavý příklad gramatik, jenž by i přes jisté obtíže mohl být použit.

Dle práce [21] se jedná o zástupce neizometrických gramatik, tvar vznikajících řetězců se tedy může měnit. Z principu čistých gramatik a zmíněné neizometričnosti pak plyne, že jazyky odpovídající gramatikám mohou teoreticky obsahovat řetězce různých tvarů.

Tomu je ovšem předcházeno paralelním přístupem, stejně jako ve dříve představených gramatikách. Také jsou využity tabulky pravidel obsahující pravidla rozdělená podle délky generovaného řetězce, což umožňuje kontrolovat paralelní generování stejně dlouhých bloků obrazu.

Definice 17. *Čistá (pure) dvoudimensionální bezkontextová gramatika (P2DCFG) je čtveřice $G = (\Sigma, P_1, P_2, \mathcal{M}_0)$, kde:*

Σ je konečná *abeceda symbolů*,

$P_1 = \{c_i | 1 \leq i \leq s_c\}$, kde c_i je *tabulka sloupcových pravidel* a s_c je libovolné kladné celé číslo, každé c_i je konečná množina bezkontextových pravidel ve tvaru $a \rightarrow \alpha$, $a \in \Sigma$, $\alpha \in \Sigma^*$ taková, že pro libovolná dvě pravidla $a \rightarrow \alpha$, $b \rightarrow \beta$ v c_i platí $|\alpha| = |\beta|$, jinak řečeno α a β mají stejnou délku,

$P_2 = \{r_j | 1 \leq j \leq s_r\}$, kde r_j je *tabulka řádkových pravidel* a s_r je libovolné kladné celé číslo, každé r_j je konečná množina bezkontextových pravidel ve tvaru $c \rightarrow \gamma^T$, $c \in \Sigma$, $\gamma \in \Sigma^*$, taková, že pro libovolná dvě pravidla $c \rightarrow \gamma^T$, $d \rightarrow \delta^T$ v r_j , platí $|\gamma| = |\delta|$,

$\mathcal{M}_0 \subseteq \Sigma^{**} \setminus \{\lambda\}$ je konečná množina *axiomatických polí*.

Průběh generování obrazů je pak následující. Obraz je generován z jiného obrazu (konkrétně tedy i z jednoho počátečního symbolu) dvěma způsoby:

- pomocí pravidel v konkrétní tabulce sloupcových pravidel, kdy je paralelně přepsán každý symbol ve sloupci původního obrazu, či
- pomocí pravidel v tabulce řádkových pravidel, kdy je paralelně přepsán každý symbol v řádku původního obrazu.

Všechna použitá pravidla pro přepsání řádku, resp. sloupce, musí náležet do stejné tabulky. Tím je zaručen výsledný obdélníkový tvar vzniklého obrazu bez narušení rovnosti rozměrů jednotlivých řádků či sloupců.

Jazyk obrazů generovaný pomocí gramatiky G pak je množina obrazů $L(G) = \{M \in \Sigma^{**} | M_0 \Rightarrow^* M, M_0 \in \mathcal{M}_0\}$. Následuje příklad převzatý z práce [21] pro demonstraci postupu.

Příklad 3. Předpokládejme gramatiku $G_2 = (\Sigma, P_1, P_2, \{M_0\})$, kde $\Sigma = \{a, b, e\}$, $P_1 = \{c\}$, $P_2 = \{r\}$, kde $c = \{a \rightarrow bab, e \rightarrow aea\}$, $r = \{e \rightarrow \begin{smallmatrix} e \\ a \end{smallmatrix}, a \rightarrow \begin{smallmatrix} a \\ b \end{smallmatrix}\}$, a $M_0 = \begin{smallmatrix} a e a \\ b a b \end{smallmatrix}$.

Gramatika generuje jazyk L_2 skládající se z obrazů p velikosti $(m, 2n+1)$, $m \geq 2$, $n \geq 1$, kde $p(1, j) = p(1, j+n+1) = a$, pro $1 \leq j \leq n$; $p(1, n+1) = e$, $p(i, n+1) = a$ a pro $2 \leq i \leq n$; $p(i, j) = b$. Obraz, jenž patří do jazyka L_2 , je zobrazen v tabulce 2.11.

a	a	a	e	a	a	a
b	b	b	a	b	b	b
b	b	b	a	b	b	b
b	b	b	a	b	b	b
b	b	b	a	b	b	b

Tabulka 2.11: Obraz $P_1 \in L_2$

V této sekci byly představeny různé druhy gramatik generujících dvoudimensionální jazyky. Každá z gramatik se pak lišila od ostatních způsobem generování obrazů, přičemž bylo zmíněno, že častým prvkem takového generování je jeho paralelní průběh. Bylo zmíněno, že otázkou těchto gramatik se zabývá velké množství prací, tudíž jsou zajímavým a potenciálně využitelným tématem teoretické informatiky. Lze se domnívat, že tato skutečnost bude platit i u obecně multidimensionálních gramatik.

2.4 Regulární výrazy

Stejně jako pro jednodimensionální jazyky lze pomocí operací představených v 2.2.2 vytvořit mechanismus *regulárních výrazů*, jak byl definován v [8]. Pro zájemce, regulární jazyky ve dvou dimenzích jsou pak hlavním námětem práce [26], ve které je níže představená látka probrána důkladněji. Tyto regulární výrazy tvoří další možnost, jak vytvářet dvoudimensionální jazyky, v tomto případě *regulární*.

Mějme abecedu Σ a jazyky, z nichž každý obsahuje právě jeden prvek a , přičemž platí $a \in \Sigma$, a prázdný jazyk \emptyset . Tyto jazyky pak nazýváme *atomické jazyky*. Dále definujeme množinu $\mathcal{R} = \{\emptyset, \odot, * \ominus, * \odot, \cup, \cap, ^c\}$. Prvky množiny \mathcal{R} jsou nazývány *regulární operace*.

Jazyk L je jazykem *regulárním*, pokud je získán z atomických jazyků pomocí konečného množství aplikování regulárních operací. *Regulární výraz* je pak zápisem, jenž popisuje vytvoření konkrétního jazyka z atomických jazyků regulárními operacemi. Následuje formální definice.

Definice 18. *Regulární výraz* (RE) nad abecedou Σ je rekurzivně definován následovně:

- \emptyset a každý symbol $a \in \Sigma$ jsou regulárními výrazy.
- Pokud jsou α a β regulárními výrazy, pak $(\alpha) \cup (\beta)$, $(\alpha) \cap (\beta)$, $^c(\alpha)$, $(\alpha) \ominus (\beta)$, $(\alpha) \odot (\beta)$, $(\alpha)^{* \odot}$, $(\alpha)^{* \ominus}$ jsou regulárními výrazy

Všechny regulární výrazy nad abecedou Σ pak vyznačují dvoudimensionální jazyky nad Σ při standardní notaci daných operací. Podotýkáme, že operace projekce, rotace a transpozice nebyly zmíněny mezi regulárními operacemi. Výsledky těchto operací nad daným regulárním obrazem je ovšem logicky možné získat pomocí jiného pořadí regulárních operací, tudíž tato skutečnost nijak neovlivňuje vyjadřovací schopnost regulárních výrazů.

Definice 19. Dvoudimensionální jazyk $L \subseteq \Sigma^{**}$ je *regulární*, pokud jej lze vyjádřit pomocí regulárního výrazu nad Σ .

Následuje demonstrace použití regulárních výrazů:

Příklad 4. Mějme abecedu $\Sigma_1 = \{\square, \blacksquare\}$. Pak regulární výraz $((\blacksquare \ominus \square)^{* \ominus}) \odot ((\square \ominus \blacksquare)^{* \ominus})^{* \odot}$ značí jazyk „šachovnicových“ obrazů se stranami o sudých rozměrech, podobný tomu v příkladu 1.

■	□	■	□
□	■	□	■
■	□	■	□
□	■	□	■
■	□	■	□
□	■	□	■

Tabulka 2.12: Obraz z jazyka značeného r. výrazem

Bylo demonstrováno, že lze využít regulárních výrazů i ve dvou dimenzích, tedy je poskytnut další důkaz, že rozšíření dimensionalit řetězců nijak neomezuje prostředky, jenž jsou použity pro klasické řetězce. Naopak, jak bylo představeno v 2.3 a zároveň bude demonstrováno v následující sekci, toto zvýšení počtu dimenzí přináší více možností, jak definovat jazyky a jim náležející řetězce.

V této kapitole bylo zahájeno představení teorie vícedimensionálních jazyků. Čtenáři byl představen pojem *obraz*, který reprezentuje roli řetězce ve dvou dimenzích. Spolu s obrazem byly představeny i základní koncepty známé pro řetězce, jako jsou velikost a podřetězec (v tomto případě *podobraz*).

Dále byly uvedeny a demonstrovány rozšířené a nové operace pro dvoudimensionální řetězce. Bylo uvedeno, že s počtem dimenzí bude počet těchto operací logicky stoupat, a tak se pro vícedimensionální řetězce naskýtá více možností úprav.

Za pomoci těchto obrazů a operací byly definovány dvoudimensionální jazyky, operace s těmito jazyky a jejich uzávěry. Tyto jazyky pak lze definovat pomocí různých dvoudimensionálních gramatik, z nichž byli vybráni tři zástupci také představení v této kapitole. Nakonec byl uveden i princip dvoudimensionálních regulárních výrazů, definující regulární jazyky ve dvou dimenzích.

Kapitola 3

Dvoudimensionální automaty

V této kapitole představíme jedno z hlavních témat práce ve dvou dimenzích – automaty pracující nad maticovými systémy znaků, přičemž se může jednat i o pixely, a tedy automaty plní roli grafických analyzátorů. V sekci je diskutována tato skutečnost v rámci možných využití těchto automatů. To, že využití je možné, bylo dokázáno v předchozí práci (tedy [11]), kde byl vyvinut speciální automat (viz 3.5) spadající mezi dvoudimensionální automaty, který byl prakticky nasazen v analytickém programu.

Na úvod je nutné říci, že standardně se automaty ve dvou dimenzích pohybují v různých směrech, jelikož nelze zpracovávat vstupní obraz pixel po pixelu bez možnosti změny směru procházení. Jedná se tedy o obdobu Turingova stroje, který pracuje nad dvoudimensionální páskou reprezentovanou obrazem.

Společně s nárůstem počtu dimenzí analyzovaného „řetězce“ geometrickou řadou narůstá počet směrů, jimiž se může příslušný automat pohybovat. Standardní automat se v jedné dimenzi pohybuje v řetězci pouze jedním směrem. Obdobou pro tento postup může být dvousměrný dvoudimensionální automat, jenž se v obraze pohybuje od levého horního rohu pouze směry doprava a dolů.

Nemusíme se ovšem omezovat pouze na minimální počet směrů postupu automatu. Ve dvou dimenzích je standardně využíván automat používající 4 směry, což zdaleka není maximum, uvědomíme-li si, že ve dvou dimenzích má každý bod maticové struktury 8-okolí. Počet používaných směrů je u jednotlivých zástupců automatů proměnlivý, jak si lze všimnout v následujících podsekcích.

Automaty pro dvě dimenze nabízejí mnoho možností pro modifikaci. Nabízí se například zřejmá podobnost s celulárními automaty či grafovými analyzátoři. Uvedme některé zajímavé práce, týkající se tohoto tématu, například [7], [16], [32], [10], [34] a [17]. Následuje analýza několika vybraných zástupců dvoudimensionálních automatů, definice a příklady pro demonstraci rozdílných postupů automatů.

3.1 Čtyřsměrný automat

Jedná se o základní model dvoudimensionálního automatu. Je jednoduchou obdobou automatu pohybujícího se po pásce ve dvou směrech, samozřejmě ale s obměnami pro dvě dimenze. Stejně jako klasický Turingův stroj, jenž disponuje směry L a R, tento automat navíc přidává směry U a D pro pohyby směrem nahoru a dolů.

Oproti klasickému Turingovu stroji ovšem neprovádí automat změny v obrazu a jediná změna při získání nového znaku může být změna stavu automatu. Definice byla převzata z předchozí práce [11] a práce [9].

Definice 20. Čtyřsměrný automat, označován jako 4DFA, je sedmice $\Lambda = (\Sigma, Q, \Delta, q_0, q_a, q_r, \delta)$, kde:

- Σ je vstupní abeceda,
- Q je konečná množina stavů,
- $\Delta = \{R, L, U, D\}$ je množina směrů,
- $q_0 \in Q$ je počáteční stav,
- $q_a, q_r \in Q$ jsou akceptující a odmítající stav,
- $\delta : Q \setminus \{q_a, q_r\} \times \Sigma \cup \{\#\} \rightarrow Q \times \Delta$ je přechodová funkce.

Tento automat pak prochází obrazem po krocích ve směru dvou os od počátečních souřadnic $(1, 1)$. V každém kroku výpočtu pak může dojít ke změně stavu automatu a také k posunu hlavy automatu nad obrazem ve zmíněných směrech. Stejně jako u Turingových strojů nazvěme přepadnutí hlavy automatu, tedy pohyb hlavy přes hranice obrazu, *nestandardním* ukončením. Při nepřepadnutí hlavy pak může automat obraz přijmout či odmítnout přechodem do stavu q_a , resp. q_r . Pro pomoc s problémem přepadnutí pak můžeme vyznačovat hranice obrazu speciálním symbolem $\#$, kde $\# \notin \Sigma$.

Podotýkáme, že obrazy jsou v základu o jistých rozměrech, tedy nejsou nekonečné a automat nemění obsah obrazu, existuje tudíž konečně mnoho konfigurací automatu. Problém ukončení je tedy zde *rozhodnutelný*, i pokud je automat nedeterministický. To je společným prvkem tohoto a klasického konečného automatu.

Příklad 5. Mějme automat $\Lambda_1 = (\Sigma, Q, \Delta, q_0, q_a, q_r, \delta)$, kde $\Sigma = \{a, b, c\}$, $Q = \{q_0, q_a, q_r\}$ a $\delta = \{q_0a \rightarrow d_1D, \quad q_0b \rightarrow q_rD, \quad q_0c \rightarrow q_rD, \quad d_1\# \rightarrow r_1U, \quad d_1a \rightarrow q_0L,$
 $d_1b \rightarrow q_0L, \quad d_1c \rightarrow q_0L, \quad r_1a \rightarrow r_1L, \quad r_1b \rightarrow r_1L, \quad r_1c \rightarrow r_1L,$
 $r_1\# \rightarrow u_1R, \quad u_1a \rightarrow u_2U, \quad u_1b \rightarrow q_rU, \quad u_1c \rightarrow q_rU, \quad u_2a \rightarrow u_1R,$
 $u_2b \rightarrow u_1R, \quad u_2c \rightarrow u_1R, \quad u_2\# \rightarrow q_aD\}$.

Pro jednoduchost při použití automatu předpokládejme pouze čtvercové obrazy. Pak automat Λ_1 ověřuje, zda se na hlavní a vedlejší diagonále obrazu vyskytují pouze znaky a . Díky použití hranic nebude možné přepadnutí hlavy automatu, automat tedy vždy určí, zda obraz patří (resp. nepatří) do jazyka obrazů, které splňují zmíněné kritérium.

Mějme obrazy A_1, A_2 o rozměrech $(3, 3)$ nad abecedou Σ s označením hranic obrazu pomocí znaku $\#$ stejně, jako je označena prázdná část pásky pro Turingův stroj. Tyto obrazy jsou zachyceny tabulkami 3.1 a 3.2. Jak je vidět, obraz A_1 nebude automatem Λ_1 přijat a obraz A_2 přijat bude. Postup automatu pro A_1 je znázorněn v tabulce 3.3, pozice hlavy automatu je zvýrazněna červenou barvou a je zanesen stav spolu s použitým pravidlem.

#	#	#	#	#
#	a	a	a	#
#	b	b	c	#
#	b	b	a	#
#	#	#	#	#

Tabulka 3.1: Obraz A_1

#	#	#	#	#
#	a	b	a	#
#	c	a	c	#
#	a	b	a	#
#	#	#	#	#

Tabulka 3.2: Obraz A_2

#	#	#	#	#
#	a	a	a	#
#	b	b	c	#
#	b	b	a	#
#	#	#	#	#

Stav: q_0
 $\delta: q_0a \rightarrow d_1D$

#	#	#	#	#
#	a	a	a	#
#	b	b	c	#
#	b	b	a	#
#	#	#	#	#

Stav: d_1
 $\delta: d_1b \rightarrow q_0L$

#	#	#	#	#
#	a	a	a	#
#	b	b	c	#
#	b	b	a	#
#	#	#	#	#

Stav: q_0
 $\delta: q_0b \rightarrow q_rD$

Tabulka 3.3: Analýza obrazu A_1

Postup automatu pro obraz A_2 obsahuje velké množství konfigurací, proto není zanesen graficky. Použitá pravidla automatu budou v řadě po sobě následující:

- | | | | |
|------------------------------|-------------------------------|--------------------------------|-------------------------------|
| 1. $q_0a \rightarrow d_1D$, | 5. $q_0a \rightarrow d_1D$, | 9. $r_1a \rightarrow r_1L$, | 13. $u_1a \rightarrow u_2U$, |
| 2. $d_1c \rightarrow q_0L$, | 6. $d_1\# \rightarrow r_1U$, | 10. $r_1\# \rightarrow u_1R$, | 14. $u_2b \rightarrow u_1R$, |
| 3. $q_0a \rightarrow d_1D$, | 7. $r_1a \rightarrow r_1L$, | 11. $u_1a \rightarrow u_2U$, | 15. $u_1a \rightarrow u_2U$, |
| 4. $d_1b \rightarrow q_0L$, | 8. $r_1b \rightarrow r_1L$, | 12. $u_2c \rightarrow u_1R$, | 16. $u_2\# \rightarrow q_aD$ |

Automat tedy prochází diagonálně obraz od souřadnice $(1, 1)$, přičemž samozřejmě neexistuje směr pro pohyb ve směru diagonály, pro každý posun je tedy použito dvou pravidel. Při ověření, že celá první úhlopříčka je tvořena znaky a , což je indikováno přechodem automatu na okraj obrazu, se automat vrátí na prostor obrazu a začne postupovat vlevo.

Při nalezení levého okraje (opět pomocí přepadnutí na hranici obrazu) je postupováno obdobným způsobem po druhé úhlopříčce. Pokud i ta vyhovuje podmínkám, je obraz přijat přechodem automatu do stavu q_a .

Postup automatů tohoto druhu je zřejmě nejvíce intuitivní a pochopitelný, vychází z původního návrhu, představeného v práci [1]. Zde bylo použito i ohraničení obrazu tak, jak bylo demonstrováno. Dle práce patří mezi možnosti automatu i nalezení obdélníkového tvaru v obrazu, automat ovšem nedokáže zjistit symetričnost obrazu dle zvolené osy.

To je podobné jako hledání palindromu pomocí klasického konečného automatu, což také není splnitelný úkol. Tato podobnost ale není zavazující, dle práce je mnoho problémů řešitelných jednodimensionálním automatem nemožné vyřešit automatem dvoudimensionálním. Tento výrok platí i naopak.

Ve článku [9] je uveden důkaz, že platí $\mathcal{L}(4DFA) \subset \mathcal{L}(4NFA)$, nedeterministický automat tohoto typu je tedy silnější než automat deterministický. Ohledně problému uzavřenosti, rodiny jazyků $\mathcal{L}(4DFA)$ a $\mathcal{L}(4NFA)$ nejsou uzavřeny nad sloupcovou ani řádkovou konkatenací. Zároveň $\mathcal{L}(4DFA)$ a $\mathcal{L}(4NFA)$ nejsou uzavřeny nad spojením a průnikem. Ovšem $\mathcal{L}(4DFA)$ je uzavřena nad doplňkem.

Co se týče závislostí rodin jazyků generovaných pomocí gramatik a pomocí automatů, platí, že $\mathcal{L}(2RLG) \subset \mathcal{L}(4DFA)$, kde $2RLG$ je gramatika ze sekce 2.3.1. Důkazy uzávěrových vlastností a závislostí rodin jazyků se nacházejí ve zmíněném článku.

3.2 On-line teselační automat

Tento automat je velmi odlišný od klasického automatu, jako je například již představený $4DFA$. Původní návrh byl představen v práci [15], odkud je také čerpáno. Automat neprochází analyzovaný obraz (matici symbolů) krok po kroku, neanalyzuje totiž pouze jeden symbol v jeden čas.

Princip automatu spočívá ve využití principu *celulárních automatů*. Automat pracuje paralelně na celé pásce a představuje pole *buněk*. Každá buňka je paralelně s ostatními měněna dle zadaných pravidel na základě svého okolí, stavu automatu a obsahu odpovídající části obrazu.

Dvoudimenzionální on-line teselační automat je zvláštním případem takových automatů. Rozdílem je to, že všechny buňky nemění v každém kroku svůj stav. Polem buněk procházejí takzvané *přechodové vlny*, jež celé pole procházejí úhlopříčně zleva dolů. Každá buňka pak mění svůj stav na základě dvou sousedních, a sice té vlevo a té nahoře od dané buňky. Uvedená definice a příklad jsou převzaty z [9] a práce [11].

Definice 21. *Dvoudimenzionální on-line teselační automat*, značíme $2 - DOTA$, je definován jako $A = (\Sigma, Q, I, F, \delta)$, kde:

- Σ je vstupní abeceda,
- Q je konečná množina stavů,
- $I \subseteq Q$ je množina počátečních stavů,
- $F \subseteq Q$ je množina konečných stavů,
- $\delta : Q \times Q \times \Sigma \rightarrow Q$ je přechodová funkce.

Při běhu automatu je každému symbolu obrazu $p_a \in \Sigma^{**}$ na souřadnicích (i, j) přiřazován v mřížce buněk stav $q_a \in Q$. Během chodu automatu jsou tyto stavy jednotlivých buněk měněny na základě funkce δ , kde vstupem této funkce jsou stavy buněk na pozicích $(i - 1, j)$ a $(i, j - 1)$ a symbolu $p_a(i, j)$.

Na začátku jsou přiřazeny počáteční stavy q_0 buňkám v prvním řádku a sloupci obrazu p_a . Délka chodu výpočtu pak je $\ell_1(p_a) + \ell_2(p_a) - 1$ kroků automatu. Při prvním kroku je přiřazen stav buňce odpovídající pozici $(1, 1)$ jako výsledek funkce $\delta(q_0, q_0, p_a(1, 1))$. Ve druhém kroku je stav přiřazen paralelně buňkám na pozicích $(1, 2)$ a $(2, 1)$ atd. V libovolném kroku automatu s pořadím k je pak paralelně přiřazován stav buňkám na souřadnicích (i, j) , kde $i + j - 1 = k$. Přijmutí obrazu automatem $2 - DOTA$ je pak dosaženo přiřazením stavu q_f , kde $q_f \in F$ buňce na pozici $(\ell_1(p_a), \ell_2(p_a))$. Následující příklad převzatý z předchozí práce demonstruje tento chod.

Příklad 6. Necht jazyk L_{\square} nad abecedou $\Sigma = \{a\}$ obsahuje pouze čtvercové obrazy, tedy pro libovolný $p_{\square} \in L_{\square}$ platí $\ell_1(p_{\square}) = \ell_2(p_{\square})$.

Dále necht máme automat $A_1 = (\Sigma, Q, I, F, \delta)$, kde:

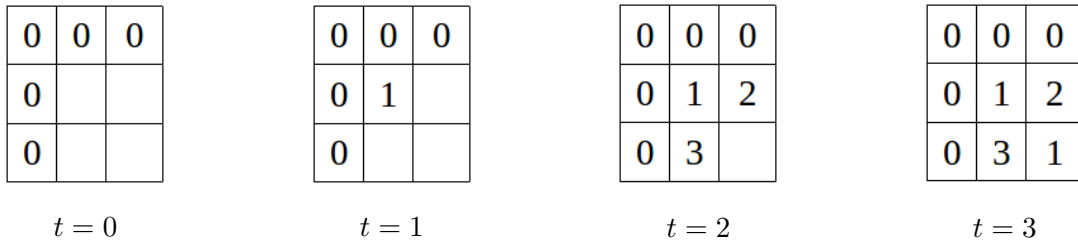
- $Q = \{0, 1, 2, 3\}$
- $I = \{0\}$
- $F = \{1\}$
- $\delta(0, 0, a) = \delta(2, 3, a) = 1$

$$\begin{aligned}\delta(0, 1, a) &= \delta(0, 2, a) = \delta(2, 1, a) = \delta(2, 2, a) = 2 \\ \delta(1, 0, a) &= \delta(3, 0, a) = \delta(1, 3, a) = \delta(3, 3, a) = 3\end{aligned}$$

Automat A_1 pak dokáže určit, zda obraz $p \in L_{\square}$. Činnost automatu (stavy buněk ve mřížce) je demonstrována na obrázku 3.1, pod každým krokem automatu je vyznačeno jeho pořadí.

Je vidět, že automat pracuje velmi efektivně, ovšem méně intuitivně než automat $4DFA$. Spojení automatu s principy celulárních automatů je ovšem skvělá a nabízející se myšlenka, jelikož tyto automaty byly pro práci s mřížkou vytvořeny.

Mohli bychom se pak bavit o jazycích *systemů*, jelikož ty jsou nejčastěji analyzovány pomocí celulárních automatů. Opět se dá polemizovat o velikosti okolí, které má vliv na stav buňky. Tato modifikace je u celulárních automatů význačným prvkem.



Obrázek 3.1: Postup automatu $2-DOTA$
(zdroj: [11])

Uzávěrové vlastnosti rodin jazyků $\mathcal{L}(2-DOTA)$ a $\mathcal{L}(2-OTA)$ jsou pak tyto. Obdobně jako u automatu $4NFA$ platí $\mathcal{L}(2-DOTA) \subset \mathcal{L}(2-OTA)$. Rodina jazyků $\mathcal{L}(2-OTA)$ je pak uzavřena vůči projekci, oběma typům konkatenace a jejich jazykovým uzávěrům.

Zajímavý je vztah mezi těmito automaty a automaty $4NFA$ a $4DFA$, totiž platí $\mathcal{L}(4NFA) \subset \mathcal{L}(2-OTA)$, což poukazuje na jistou hierarchii. Ovšem zároveň platí, že existují jazyky $L \subset \mathcal{L}(4DFA)$, pro které ale také platí $L \not\subset \mathcal{L}(2-DOTA)$. To poukazuje na velmi složitou strukturu hierarchie jazyků. Důkazy těchto závislostí rodin jazyků mohou být nalezeny v práci [15].

3.3 Zapomínající dvoudimensionální automat

Nyní představíme speciální obdobu Turingova stroje, kterou je *zapomínající automat*. Tento typ automatu byl rozebrán v práci [18], ovšem my se budeme zabývat jeho dvoudimensionální obdobou vytvořenou v práci [33].

Tato podmnožina Turingových strojů je omezena možností zapsání pouze jednoho konkrétního znaku na pásku. V praxi to znamená, že tento automat umožňuje vymazávat části pásky. Následuje formální definice.

Definice 22. *Dvoudimensionální zapomínající automat* (NFA) je dvoudimensionální Turingův stroj $(Q, \Sigma, \Sigma_0, q_0, \delta, Q_F)$, kde $\Sigma = \Sigma_0 \cup \{\#, @\}$, $@ \notin \Sigma_0$ je speciální symbol, jenž nazýváme symbol vymazání. Dále, pokud $(a, q) \rightarrow (\bar{a}, \bar{q}, d)$ je prvek přechodů daných funkcí δ , pak $a = \bar{a}$ nebo $\bar{a} = @$.

Takový deterministický automat pak může rozdělit svou dvoudimensionální pásku do bloků a pomocí zapomínání symbolů (tedy poznačení jejich výskytů symbolem @) může získávat informace o obrazu.

V práci [33] je pak uveden zdlouhavý důkaz mocnosti těchto automatů, jenž je nad rámec této práce.

Zapomínající automaty jsou silný nástroj pro analýzu jazyků v jednodimensionálním rozměru a mohly by být bez překážek zobecněny pro n-dimensionální účely, ovšem ne bez dopadu na jejich sílu.

Pro klasické jazyky je totiž v práci [18] dokázáno, že lze definovat třídu bezkontextových jazyků pomocí výše zmíněných automatů. Pro dvě dimenze tomu ale tak podle důkazu, podaného prací [33], není.

V práci [20] bylo dokázáno, že platí $\mathcal{L}(4-NFA) \subset \mathcal{L}(DFA)$, což poukazuje na zmíněnou sílu tohoto typu automatů.

3.4 Sgraffito automat

V případě tohoto automatu se jedná se o novější přístup oproti dříve představeným. Automat není úpravou žádného jednodimensionálního modelu a je specializován pro použití ve dvoudimensionálním měřítku.

Jak autor původní práce [32] (z níž jsou převzaty následující definice) uvádí, *sgraffito* (převzato z italštiny) znamená *škrábaný* či *poškrábaný* a je používán ve spojení s uměním, kde se jedná o techniku dvouvrstvé dekorace soch, keramiky či skla. První vrstva je nanášena společně se vzorem barev a je překryta druhou vrstvou. Dekorace a tvorba vzorů pak vzniká seškrabováním částí druhé vrstvy.

Tato technika je velmi podobná postupu automatu. Jedná se o automat s konečným stavovým prostorem, pracujícím nad obrazem složeným ze symbolů, z nichž každý má svou váhu. Dle autora si můžeme představit, že mřížka s hodnotami vah pro jednotlivé symboly je druhou vrstvou, nacházející se pod obrazem.

Automat se pak po obraze pohybuje (stejně jako *4NFA*) ve čtyřech směrech. Přitom musí být analyzovaný symbol vždy přepsán, a to pouze symbolem, jenž má nižší váhu než aktuální symbol. Obraz je přijat přechodem automatu do přijímajícího stavu.

Před definicí samotného sgraffito automatu definujme dvoudimensionální Turingův stroj, ačkoliv byl již použit v 3.3 (kde bylo předpokládáno intuitivní pochopení), definice tohoto automatu je na tuto definici přímo vázaná a proto je zde uvedena.

Nejdříve upravme myšlenku hranic obrazu tak, jak byla uvedena v 3.1, kde bylo použito pouze jednoho znaku #. Přidejme další znaky \vdash , \dashv , \top , \perp které budou značit hranice obrazu P tak, jak je zobrazeno v tabulce 3.4. Dohromady tyto symboly tvoří množinu S.

#	$\top \top \dots \top \top$	#
\vdash	P	\dashv
\vdots		\vdots
\vdash		\dashv
#	$\perp \perp \dots \perp \perp$	#

Tabulka 3.4: Obraz P s modifikovaným ohraničením

Definujme pak množinu $\mathcal{H} = \{R, L, U, D, Z\}$. Prvky této množiny nazýváme *pohyby hlavy*. Prvky R,L,U,D mají stejný význam, jako měly v sekci 3.1, a pohyb Z znamená žádný pohyb. Na základě této množiny pohybů definujme funkci $\nu : \mathcal{S} \rightarrow \mathcal{H}$ takovou, že $\nu(\vdash) = R, \nu(\dashv) = L, \nu(\top) = D, \nu(\perp) = U$ a $\nu(\#) = Z$.

Definice 23. *Dvoudimenzionální Turingův stroj (2TM)* je šestice $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, Q_F)$, kde:

- Σ je vstupní abeceda,
- Γ je pásková abeceda, kde $\Sigma \subseteq \Gamma$,
- Q je konečná, neprázdná množina stavů,
- $q_0 \in Q$ je počáteční stav,
- $Q_F \subseteq Q$ je množina koncových stavů a
- $\delta : (Q \setminus Q_F) \times (\Gamma \cup \mathcal{S}) \rightarrow 2^{Q \times (\Gamma \cup \mathcal{S}) \times \mathcal{H}}$ je přechodová funkce.

Dále, pro každou dvojici $(q, a) \in Q \times (\Gamma \cup \mathcal{S})$, každá $(q', a', d) \in \delta(q, a)$ platí následující:

- pokud $a \in \mathcal{S}$, pak $d = \nu(a) \wedge a' = a$, a
- pokud $a \notin \mathcal{S}$, pak $a' \notin \mathcal{S}$.

Definice sgraffito automatu je pak následující.

Definice 24. *Dvoudimenzionální sgraffito automat (2SA)* je sedmice $\mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_0, Q_F, \mu)$, kde:

- $(Q, \Sigma, \Gamma, \delta, q_0, Q_F,)$ je 2TM (dle definice 23),
- $\mu : \Gamma \rightarrow \mathbb{N}$ je váhová funkce a pro přechodovou funkci platí:

$$\forall q, q' \in Q, d \in \mathcal{H}, a, a' \in \Gamma ((q', a', d) \in \delta(q, a) \Rightarrow \mu(a') < \mu(a))$$

Vstupem automatu je klasicky obraz $P \in \Sigma^{**}$, počáteční pozice hlavy automatu je levý horní roh obrazu. Je dokázatelné, že pro jednorozměrné řetězce sgraffito automat vymezuje právě množinu *regulárních jazyků*.

Rodiny jazyků $\mathcal{L}(2SA)$ a $\mathcal{L}(2DSA)$ mají následující uzávěrové vlastnosti. Obě rodiny jsou uzavřeny nad množinovými operacemi průniku a spojení a obrazovou operací rotace. Navíc jsou uzavřeny nad operací zrcadlení, která nebyla demonstrována, ovšem její výsledek je představitelný.

U obrazových operací je rodina $\mathcal{L}(2SA)$ uzavřena nad oběma druhy konkatenace a projekcí a není uzavřena nad operací doplňku. Naopak rodina $\mathcal{L}(2DSA)$ není uzavřena ani nad jedním z druhů konkatenace či projekcí a je uzavřena nad operací doplňku. Důkazy těchto tvrzení jsou v práci [32].

Co se týče vztahu k jiným automatům, díky důkazům poskytnutých v pracích [32] a [20] pak lze tvrdit, že $\mathcal{L}(DSA) \subset \mathcal{L}(DFA)$, a tedy tranzitivně i $\mathcal{L}(DSA) \subset \mathcal{L}(4-NFA)$. Další zajímavé poznatky o uzávěrových vlastnostech a hierarchii jazykových rodin příslušících těchto automatům byly uvedeny v navazující práci autorů [35]. Zároveň je síla sgraffito automatů oproti ostatním typům automatů demonstrována v článku [34].

3.5 Vyvinuté automaty

Zde představujeme automaty, které byly vyvinuty v předchozí práci [11]. Cílem práce a vyvinutého automatu bylo představení dvoudimensionálních automatů a demonstrace jejich použití na praktických úkolech. Jelikož práce byla vyvíjena v oboru teoretické informatiky, nejednalo se o použití pro zpracování počítačové grafiky, jak by se nabízelo.

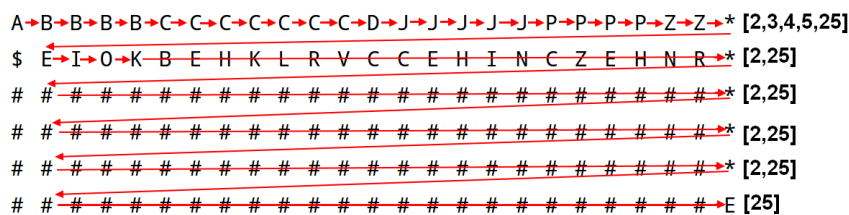
Práce se zabývala analýzou státní příslušnosti registračních značek vozidel, tedy určením množiny možných států, z nichž by zadaná registrační značka mohla pocházet. Praktického použití pro takový mechanismus existuje mnoho, primárním cílem ale bylo použití automatu v automatizovaném vyhodnocování mýtného pomocí mýtných bran a kamer.

Pro použití automatu byla vyvinuta databáze tvarů evropských registračních značek, aplikace pro editaci této databáze a aplikace s grafickým rozhraním pro používání automatu koncovými uživateli. Ukázka této aplikace je na obrázku 3.2.



Obrázek 3.2: Rozhraní aplikace (zdroj: [11])

Celkem bylo v práci postupně vyvinuto 5 automatů, jejichž vstupem byla dvoudimensionální tabulka pravidel, automat byl vždy konfigurován dle zadaného tvaru registrační značky a přijímal takové tabulky, jež v sobě obsahovaly jedno či více pravidel, která odpovídala zadané značce. Ukázky aplikace pro editaci databáze a tvaru tabulky pravidel použité v aplikaci jsou na obrázcích 3.3 a 3.4.



Obrázek 3.5: Postup třisměrného automatu (zdroj: [11])

3.6 Praktické využití

V této sekci se zabýváme otázkou praktického nasazení dvoudimensionálních automatů v reálných výpočetních situacích. Praktičnost gramatik byla dokázána v pracích uvedených v sekci 2.3. Zde se tedy soustředíme konkrétně na automaty. Možnost praktického použití dvoudimensionálních automatů byla prokázána v práci [11] způsobem, jaký byl uveden v sekci 3.5.

Nabízí se samozřejmě nasazení ve zpracování grafiky, tedy obrázků a fotografií, což bude nejspíše nejdiskutovanější téma, lze ovšem analyzovat i textové matice, tedy například textové soubory či tabulky.

Nasazení v biometrickém zabezpečení

Pokud se budeme zabývat použitím automatů v oblasti počítačové bezpečnosti, nabízí se použití pro analýzu biometrických dat. Uvažme obraz otisku prstu, který byl získán pomocí čtečky, jako matici symbolů, které reprezentují prázdný prostor či výskyt části papilární linie v daném segmentu obrazu.

Pak lze tuto matici symbolů analyzovat pomocí dvoudimensionálního automatu, který je konfigurován tak, aby se pohyboval v obrazu a umožnil potvrzení identity majitele otisku. Takový systém pak může být uplatněn například v přístupovém terminálu.

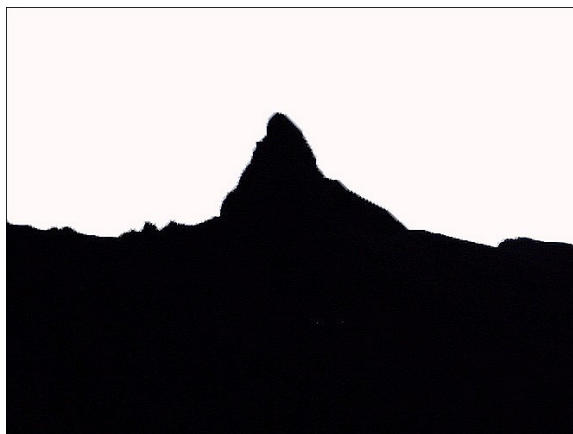
Dále lze pomocí kamery získat obraz tváře osoby, který je jako matice pixelů taktéž analyzovatelný pomocí zmíněného automatu. Tento automat pak může sloužit k potvrzení identity či vyhledávání konkrétní osoby ve snímcích.

Samozřejmě na abstraktní úrovni je využití takového automatu realitou, jelikož snímače v zabezpečovacích prvcích používají podobný přístup, který ovšem není s modelem dvoudimensionálního automatu spojován.

Analýza obrysu panoramat

Využití, které bylo diskutováno v rámci ÚPGM Fakulty Informačních Technologií VUTBR, nabízelo zjištění zeměpisné polohy místa, z něhož byla zachycena fotografie pohoří. Pokud se omezíme na pohoří Alpy, na němž byl systém testován, můžeme například uvést za příklad fotografie hory Matterhorn. Tento vrchol má velmi specifický tvar, z jeho fotografií lze sestavit model a na základě tohoto modelu vytvořit automat, jenž pro danou fotografii potvrdí, zda se jedná o fotografii daného vrcholu.

Abstraktně by se pak abeceda obrazu opět skládala ze dvou symbolů, které by reprezentovaly výskyt pixelů zachycujících pohoří a pixelů zachycujících oblohu. Automat by pak jednotlivé pixely analyzoval a zkoumal tvar vyfoceného pohoří. Ukázka takového obrazu je na obrázku 3.6.



Obrázek 3.6: Obrys hory Matterhorn (zdroj: hikr.org)

Pokud by pak byla známa ohnisková vzdálenost fotografie a další specifikace, bylo by možné podle úhlu zachycení daného vrcholu možné odhadnout místo pořízení fotografie. Část práce byla publikována v článku [2], ovšem bez použití zmíněného automatu.

Analýza databázových tabulek

Nyní diskutujeme využití automatů při dolování dat. Základním prvkem relačních databází je relace neboli tabulka. Relační databáze jsou jedním z nejpoužívanějších konceptů databází a samozřejmě tabulka je dvoudimenzionální pole údajů.

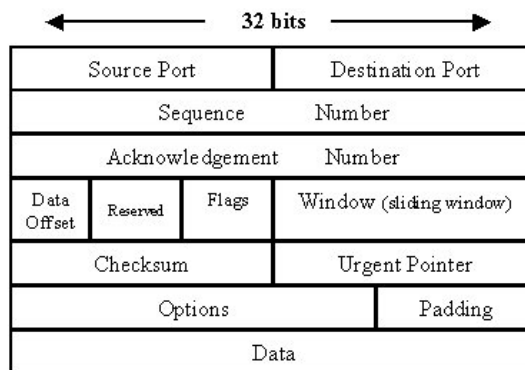
Získávání informací a znalostí z databáze, neboli *dolování dat*, je pak v současné době zajímavou disciplínou. Používá se analýza pomocí shlukování, fuzzy logiky a jiných přístupů. Pro analýzu samostatné tabulky nám opět může posloužit i dvoudimenzionální automat.

Uvažme tabulku agregovaných dat s výkazy prodejů produktů za jednotlivé měsíce. Taková tabulka může disponovat pro jednoduchost třemi sloupci: *produkt*, *období* a *počet*. Příslušný automat pak pracuje nad abecedou, jež obsahuje všechny unikátní údaje v tabulce, a prochází tabulku směrem dolů. Pokud nalezne výskyt produktu, jehož prodej uživatele zajímá, analyzuje údaj o počtu prodaných kusů a postupuje dál. Po dokončení analýzy tabulky pak může být ze stavu automatu zjištěn například stoupající trend prodejnosti produktu.

Analýza paketu

Podobně jako databázová tabulka může být datový paket zachycen jako dvoudimenzionální pole symbolů, jak je zachyceno na obrázku 3.7. Takové pole pak může dvoudimenzionální automat analyzovat z ohledu zdroje nebo cíle paketu. Také lze analyzovat přenášená data a podle toho rozhodnout, zda paket obsahuje škodlivý obsah.

Takový automat by pak mohl být implementován ve směrovačích nebo především ve firewallech, u nichž by docházelo k analýze paketu. Výsledný stav automatu, který by analyzoval zdrojový a cílový port a IP (případně obsah paketu), by určil, zda paket bude přijat, či zahozen.



Obrázek 3.7: Dvoudimensionální reprezentace paketu (zdroj: techrepublic.com)

V této kapitole byl představen základní princip fungování dvoudimensionálních automatů pracujících nad obrazy. K těmto principům byli pro demonstraci připojeni vybraní zajímaví zástupci těchto automatů.

Nejdříve byl definován a demonstrován základní model dvoudimensionálního automatu, pohybující se po obrazu ve čtyřech směrech. Tento automat je odvozen od Turingova stroje, neprovádí ovšem v obrazu žádné zápisy.

Dále byl za pomoci dalšího zástupce demonstrován úzký vztah mezi diskutovanými automaty a automaty celulárními, které jsou využívány převážně v oblasti simulací. Dodatečně byli představeni ještě dva speciální zástupci automatů, každý volící originální přístup při analýze obrazů.

Nakonec byly předvedeny příklady pro podporu tvrzení, že dvoudimensionální automaty mohou být použity pro praktické řešení problémů, což potvrzuje i jejich použití v předchozí práci autorů, také popsané v této kapitole. Následující kapitoly mají za účel tuto látku zobecnit a použít pro multidimensionální jazyky.

Kapitola 4

Třídimensionální jazyky a automaty

Tato kapitola má za úkol seznámit čtenáře s teorií třídimensionálních jazyků, gramatik a automatů. Tato teorie nebyla ve vědeckých pracích prozatím hlouběji prozkoumána, přinejmenším při srovnání s teorií v minulých kapitolách. Jedná se tedy z velké části o příspěvek autorů, přičemž v textu jsou zmíněny některé práce přímo či nepřímo související s představenými termíny.

Pokud se zaměříme obecně na koncept třídimensionálních jazyků, jedná se o velmi zajímavou látku už pouze z důvodu, že reálný svět je třídimensionální. Tyto jazyky jsou tedy jazyky reálných těles a jim příslušející automaty a gramatiky jsou pak prostředky popisující tyto nám známé útvary. Samozřejmě v této kapitole také uvádíme možná využití těchto prostředků, kterých vzhledem k výše zmíněné skutečnosti není málo.

Automaty a gramatiky budou v této kapitole probírány hlavně ve vztahu k jejich dvoudimensionálním obdobám, aby byl nalezen přístup, jakým alespoň některé zástupce zobecnit pro multidimensionální jazyky, což si tato práce klade za cíl. Úpravy gramatik a automatů jsou opět originálním příspěvkem autorů a autorům není známa jiná práce, která se těmito úpravami zabývá.

4.1 Jazyky a tělesa

V této sekci jsou stejně jako v předchozí kapitole představeny pojmy vymezující teorii formálních jazyků, zde konkrétně pro tři dimenze, spolu s operacemi a pojmy náležejícími k řetězcům.

4.1.1 Tělesa

Jak už bylo zmíněno v úvodu kapitoly, základním prvkem pro třídimensionální jazyky bude *těleso*, tak, jako byl obraz základem pro jazyky dvoudimensionální. Těleso je třídimensionální řetězec obsahující symboly z příslušné abecedy. Jelikož dle klasické definice řetězce i jeden symbol je řetězcem, platí, že symbol je i těleso. Navigaci v tělese pak ustanovíme dle standardizované *kartézské soustavy souřadnic*, jak je zobrazeno na obrázku 4.1. Příslušné formální definice budou následovat.

Definice jsou obdobou definic zmíněných v předchozí kapitole, a to z důvodu poukázání na návaznost těchto definic pro získání jednotného prostředku pro definici pojmů u multidimensionálního jazyka.

Budeme dále uvažovat tělesa jako *kvádry*, stejně, jako jsme uvažovali obdélníkové dvou-dimensionální řetězce. Pokud chceme zobrazit nepravidelný tvar těmito prostředky, použijeme kvádrové těleso vyznačující prostor zabraný zobrazovaným tvarem. Dále použijeme výskyt symbolu vyhrazeného pro značení prázdného prostoru, aby byl zobrazen zmíněný nepravidelný tvar.

Definice 25. *Třídimensionální řetězec*, neboli *těleso*, je objekt o tvaru kvádrů, skládající se z elementů abecedy Σ .

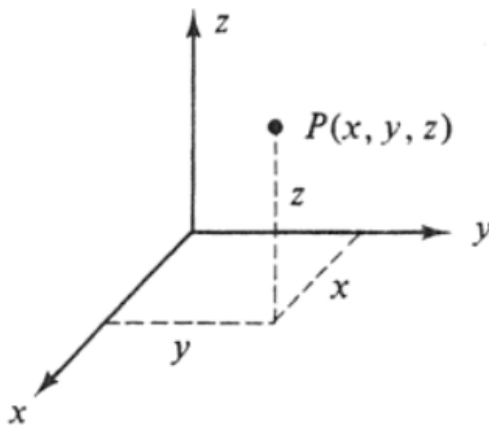
Definice 26. *Množina všech těles nad abecedou Σ je pak značena Σ^{***} .*

Pro dané těleso $o \in \Sigma^{***}$ nechť $\ell_1(o)$ značí *výšku tělesa o* (rozměr ve směru osy z), $\ell_2(o)$ značí *šířku tělesa o* (rozměr ve směru osy y) a $\ell_3(o)$ značí *hloubku tělesa o* (rozměr ve směru osy x). Trojici $(\ell_1(o), \ell_2(o), \ell_3(o))$ pak nazýváme *velikost tělesa o* . Jediné těleso o velikosti $(0, 0, 0)$, tedy *prázdné těleso*, zde značíme jako μ .

Pak také pro $i > 0, j \geq 0$ neexistuje žádné těleso s jedním z následujících rozměrů: $(i, j, 0), (i, 0, j), (j, i, 0), (0, i, j), (j, 0, i), (0, j, i)$.

Množina všech těles o rozměrech (k, l, m) nechť je značena: $\Sigma^{k \times l \times m}$.

Symbol v tělesu o na souřadnicích (k, l, m) , kde platí $1 \leq k \leq \ell_1(o)$, $1 \leq l \leq \ell_2(o)$ a $1 \leq m \leq \ell_3(o)$, pak můžeme značit $o(k, l, m)$ nebo $o_{k,l,m}$.



Obrázek 4.1: Kartézský systém souřadnic (zdroj: learnaboutelectronics.com)

Při popisu tělesa se ovšem dostáváme k problému jeho zachycení na dvojrozměrném médiu, nabízí se tudíž popisovat těleso po dvoudimensionálních vrstvách, tedy již definovaných *obrazech*, tak, jak je zobrazeno na obrázku 4.2. Tento způsob zachycení ovšem není příliš úsporný na využitý prostor a tato náročnost by se exponenciálně zvětšovala s počtem dimenzí pro multidimensionální řetězce. Volíme tedy matematický zápis demonstrováný v příkladu 7.

a	b	a	a
c	c	a	b
b	c	a	a
a	b	c	b

Vrstva pro $z = 1$

c	b	a	b
a	a	a	c
b	c	a	a
a	a	b	b

Vrstva pro $z = 2$

c	c	c	b
c	b	b	b
b	b	c	a
a	b	c	b

Vrstva pro $z = 3$

Obrázek 4.2: Vrstvy tělesa o_1

Příklad 7. Těleso o_1 zobrazené na obrázku 4.2 lze zapsat následovně:

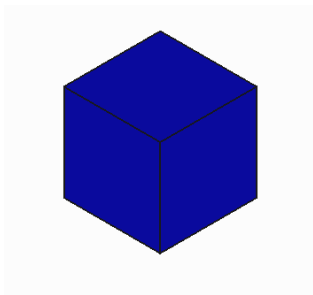
$$\begin{aligned}
o_1(1, 1, 1) &= a, & o_1(1, 1, 2) &= b, & o_1(1, 1, 3) &= a, & o_1(1, 1, 4) &= a, \\
o_1(1, 2, 1) &= c, & o_1(1, 2, 2) &= c, & o_1(1, 2, 3) &= a, & o_1(1, 2, 4) &= b, \\
o_1(1, 3, 1) &= b, & o_1(1, 3, 2) &= c, & o_1(1, 3, 3) &= a, & o_1(1, 3, 4) &= a, \\
o_1(1, 4, 1) &= a, & o_1(1, 4, 2) &= b, & o_1(1, 4, 3) &= c, & o_1(1, 4, 4) &= b, \\
o_1(2, 1, 1) &= c, & o_1(2, 1, 2) &= b, & o_1(2, 1, 3) &= a, & o_1(2, 1, 4) &= b, \\
o_1(2, 2, 1) &= a, & o_1(2, 2, 2) &= a, & o_1(2, 2, 3) &= a, & o_1(2, 2, 4) &= c, \\
o_1(2, 3, 1) &= b, & o_1(2, 3, 2) &= c, & o_1(2, 3, 3) &= a, & o_1(2, 3, 4) &= a, \\
o_1(2, 4, 1) &= a, & o_1(2, 4, 2) &= a, & o_1(2, 4, 3) &= b, & o_1(2, 4, 4) &= b, \\
o_1(3, 1, 1) &= c, & o_1(3, 1, 2) &= c, & o_1(3, 1, 3) &= c, & o_1(3, 1, 4) &= b, \\
o_1(3, 2, 1) &= c, & o_1(3, 2, 2) &= b, & o_1(3, 2, 3) &= b, & o_1(3, 2, 4) &= b, \\
o_1(3, 3, 1) &= b, & o_1(3, 3, 2) &= b, & o_1(3, 3, 3) &= c, & o_1(3, 3, 4) &= a, \\
o_1(3, 4, 1) &= a, & o_1(3, 4, 2) &= b, & o_1(3, 4, 3) &= c, & o_1(3, 4, 4) &= b
\end{aligned}$$

Znázorněné těleso o_1 pak má rozměry $(3, 4, 4)$, tedy kvádr skládající se ze 48 symbolů nad abecedou $\Sigma = \{a, b, c\}$. Dále následuje definice *podtělesa*, tedy třídimensionálního podřetězce.

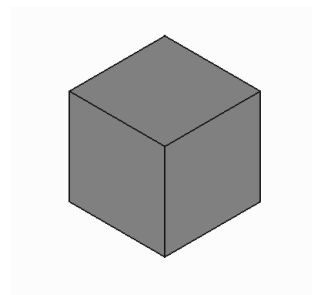
Definice 27. Necht' máme těleso O o rozměrech (a, b, c) . *Podtěleso (komponenta) tělesa O* , označme O' , je pak těleso obsažené uvnitř tělesa O . Takové těleso pak má rozměry (a', b', c') , pro které platí $a' \leq a$, $b' \leq b$ a $c' \leq c$ a existují nezáporná celá čísla x, y, z , pro která platí $O'(i, j, k) = O(i + x, j + y, k + z)$ pro všechna $0 \leq i \leq a'$, $0 \leq j \leq b'$ a $0 \leq k \leq c'$.

4.1.2 Operace pro tělesa

Stejně jako u dvoudimensionálních řetězců i zde nacházíme více operací pro manipulaci s tělesy oproti méně dimenzím. Konkrétně si tuto změnu můžeme demonstrovat na konkatenci, která získává tři obdoby (opět jedna na každou dimenzi). Definice jsou demonstrovány na obrázcích vytvořených v modelářském softwaru, zachycených v axonometrickém pravotočivém systému. Vstupem konkatencí budou tělesa O_1 a O_2 , zachycená na obrázcích 4.4 a 4.4. Jedná se o primitiva těles, konkrétně tedy o tělesa s rozměry $(1, 1, 1)$ obsahující „symboly“: šedou a modrou barvu.



Obrázek 4.3: Těleso O_1

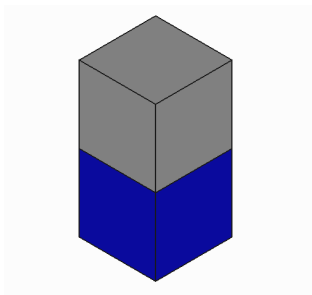


Obrázek 4.4: Těleso O_2

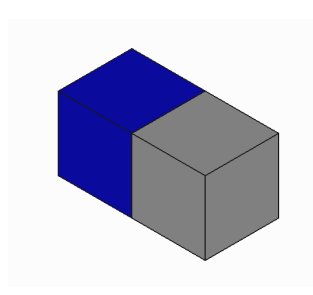
Definice 28. Výsledek *výškové konkatence* (ve směru osy z) těles O_1 a O_2 , značíme $O_1 \oplus_z O_2$, je zobrazen na obrázku 4.5.

Definice 29. Výsledek *šířkové konkatence* (ve směru osy y) těles O_1 a O_2 , značíme $O_1 \oplus_y O_2$, je zobrazen na obrázku 4.6.

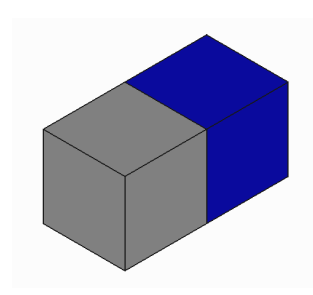
Definice 30. Výsledek *hloubkové konkatence* (ve směru osy x) těles O_1 a O_2 , značíme $O_1 \oplus_x O_2$, je zobrazen na obrázku 4.7.



Obrázek 4.5: Těleso $O_1 \oplus_z O_2$



Obrázek 4.6: Těleso $O_1 \oplus_y O_2$



Obrázek 4.7: Těleso $O_1 \oplus_x O_2$

S každou dimenzí přibývá jeden druh konkatence navíc, a to konkrétně konkatence v prostoru nové dimenze. Opět je zde logický vztah mezi rozměry spojených těles a výsledného tělesa, stejně jako u konkatence obrazů. Je očekávatelné, že operace *rotace*, kterou jsme si představili u dvoudimenzionálních řetězců, také získá více podob.

Pro tři dimenze existují logicky tři druhy rotace, těleso tedy provede otočení ve směru jedné z os prostoru. To samozřejmě ovlivní podobu tělesa i v ostatních dimenzích. Matematická definice této operace by byla zbytečně složitá, proto budeme demonstrovat rotace na grafickém příkladu s klasickou hrací kostkou. Mějme těleso (hrací kostku) na obrázku 4.8. Výsledky rotací jsou zachyceny na obrázcích 4.9, 4.10 a 4.11.



Obrázek 4.8: Těleso K_1



Obrázek 4.9: Těleso $K_2 = K_1^{R^1}$



Obrázek 4.10: Těleso $K_3 = K_1^{R^2}$



Obrázek 4.11: Těleso $K_4 = K_1^{R^3}$

4.1.3 Třídídimensionální jazyk

Zde představujeme jazyky obsahující třídídimensionální řetězce, tedy tělesa. Stejně jako u dvoudídimensionálních jazyků zde neexistuje klasická Chomského hierarchie. V rámci jazyků se ale můžeme bavit například o těch regulárních, jelikož sestavení regulárních výrazů pro tři dimenze není nerealizovatelné.

Jak bylo zmíněno v úvodu kapitoly, třídídimensionální formální jazyky jsou prozatím neprozkoumanou teorií. V práci [32] bylo dokázáno, že lze nalézt hierarchii i pro dvoudídimensionální jazyky, ale za použití speciálních automatů a velmi odlišnou od Chomského hierarchie. Není tudíž vyloučené, že takový důkaz bude nalezen i pro třídídimensionální jazyky.

Pak by tato teorie byla velmi zajímavou i pro popis reálných těles, kdy by byla nalezena unifikovaná formální teorie popisující nám známé objekty pomocí formálních postupů. Není složité představit si vznik automatů analyzujících objekty z hlediska jejich materiálu, odolnosti atd.

Definice 31. *Třídídimensionální jazyk nad abecedou Σ je podmnožinou Σ^{***} .*

Připojíme navíc k jazykům odpovídající tři druhy konkatenací představené v 4.1.2. Definice jsou obdobou jejich dvoudídimensionálních verzí, proto je zde neuvádíme. Značíme klasicky $L_1 \oplus_z L_2$, $L_1 \oplus_y L_2$ a $L_1 \oplus_x L_2$. Obdobně pak pro všechny druhy rotace, značíme $L_1^{R^1}$, $L_1^{R^2}$ a $L_1^{R^3}$. Následují definice uzávěrů jazyků pro všechny tři dimenze.

Definice 32. Necht L je jazyk. *Výškový uzávěr* jazyka L , značen $L^{*\oplus_z}$, je definován jako $L^{*\oplus_z} = \bigcup_{i \geq 0} L^{i \oplus_z}$, kde $L^{0 \oplus_z} = \mu$, $L^{1 \oplus_z} = L$, $L^{n \oplus_z} = L \oplus_z L^{(n-1) \oplus_z}$.

Definice 33. Necht L je jazyk. *Šířkový uzávěr* jazyka L , značen $L^{*\oplus y}$, je definován jako $L^{*\oplus y} = \bigcup_{i \geq 0} L^{i \oplus y}$, kde $L^{0 \oplus y} = \mu$, $L^{1 \oplus y} = L$, $L^{n \oplus y} = L \oplus_y L^{(n-1) \oplus y}$.

Definice 34. Necht L je jazyk. *Hlubkový uzávěr* jazyka L , značen $L^{*\oplus x}$, je definován jako $L^{*\oplus x} = \bigcup_{i \geq 0} L^{i \oplus x}$, kde $L^{0 \oplus x} = \mu$, $L^{1 \oplus x} = L$, $L^{n \oplus x} = L \oplus_x L^{(n-1) \oplus x}$.

V této sekci byly nově představeny operace pro jazyky ve třech dimenzích. Definice byly vyvozeny z jejich obdob pro dvě dimenze, jedná se ovšem o originální definice.

Jak lze pozorovat, s počtem dimenzí se až geometricky navyšuje suma operací, které lze definovat pro daný prostor jazyků.

To značí i zvýšení počtů prvků n -tice pro odpovídající *regulární výrazy*. S nárůstem počtu dimenzí tak narůstá i složitost těchto výrazů a mocnost skupiny jazyků jimi definovaných.

Následující sekce popisují generování uvedených jazyků pomocí gramatik a automatů.

4.2 Gramatiky

V této sekci uvádíme některé zástupce třídídimensionálních gramatik (také *spatial grammars*), mimo jiné ve vztahu k dvoudídimensionálním gramatikám pro demonstraci postupu mezi dimenzemi.

Třídídimensionální gramatiky jsou stále převážně novým tématem pro teoretickou informatiku a zabývá se jimi velmi málo odborníků. I přesto byla vícekrát (například v práci [12]) přednesena myšlenka navrhuující použití interaktivních třídídimensionálních gramatik v modelářských systémech.

Téma dvoudídimensionálních gramatik pokládá základy, z nichž lze vytvořit teorii pro gramatiky a jazyky třídídimensionální. Velmi pokročilý postup, používající teorii týkající se tzv. UAG, tedy *univerzálních gramatik polí* původně navržených pro generování dvoudídimensionálních jazyků, je zmíněn v práci [48]. Obdoba tohoto postupu používající *izometrických gramatik polí* (IAG) je zmíněna v práci [14]. Rozsáhlá teorie použitá v těchto článcích je ale nad rámec rozsahu této práce.

Gramatiky, označované jako *spatial grammars*, jsou převážně teoretickým pojmem, ovšem existuje několik jejich implementací, jejichž přehled poskytuje článek [12]. Zmíněná práce se zabývá implementací interpretu prostorových gramatik, který by svou jednoduchostí byl používán grafickými návrháři jako podpůrný nástroj v CAD aplikacích. Nasazení gramatik v praxi by v mnoha případech přinášelo intuitivní a rychlý způsob návrhu modelářských komponent, ale tomu brání složitější teorie pro jejich popsání. Z práce byla převzata definice prostorové gramatiky, viz 4.2.1.

Zajímavou myšlenkou, pro změnu z oblasti neurologie a lingvistiky, je považování různých obdob znakové řeči za třídídimensionální jazyky a pravidel tvoření vět v těchto jazycích za jim příslušné gramatiky. Tento přístup byl přednesen v práci [19] a i přes zřejmou autorskou neznalost teorie formálních jazyků přednáší myšlenky spadající do této kategorie. Pokud bychom jednotlivé znaky (gesta) znakové řeči zakreslili jako třídídimensionální modely, pak lze z těchto symbolů skládat jazyky a analyzovat tato gesta prostorovým automatem.

V článku [25], kde jsou nasazeny prostorové gramatiky v prostředí grafických procesorů, také poukazuje na praktický přínos implementace těchto gramatik. Zde jsou využity pro jednoduché předzpracování a generování virtuálních těles, použitelné v herním prostředí. Pravidla gramatik jsou méně náročná na paměť a výpočetní čas, přináší tedy velký přínos z hlediska úspory. Dovolujeme si poznamenat, že podobný přístup, používající principu

funkcionálního programování, byl již použit, konkrétně v počítačové hře *No Man's Sky*, kde je generován herní vesmír pomocí funkcionálních výroků.

Předchozí myšlenka byla navíc v práci [13], pojednávající o tzv. *motion grammars*, obohacena o aspekt pohybu těles. Je zde navrženo, aby pro jedno virtuální těleso bylo pomocí příslušné gramatiky možno navrhnout jeho pohyb ve virtuálním světě. Byla by pak například simultánně vytvořena pravidla pro pohyb virtuálních protivníků v počítačové hře.

4.2.1 Spatial grammars

Pojem *spatial grammars*, neboli prostorové gramatiky, v sobě zastřešuje všechny druhy gramatik, které definují jazyky skládající se z tvarů, například tedy i řetězcové gramatiky, množinové gramatiky, grafové gramatiky atd.

Zde je uveden jeden univerzální zástupce *tvarových (shape) gramatik* (upravená definice převzata z [12]), které také spadají do gramatik prostorových. Poznamenejme zde, že se jedná o přístup, který se liší od předchozích přístupů. Proto je zde uveden jen jako existující zástupce prostorových gramatik, jehož specializace na manipulaci s třídimesionálními řetězci mu neumožňuje jeho zobecnění pro n-dimensionální případ.

Definice 35. *Tvarová gramatika* G je definována jako čtveřice $G = (S, L, R, I)$, kde:

S je konečná množina tvarů,

L je konečná množina značek,

R je konečná množina pravidel,

I je počáteční tvar, kde I je podtělesem označeného tělesa $(S, L)^0$.

Pak existuje množina označených tvarů, obsahující prázdný označený tvar, značená $(S, L)^0$, nazývaná též *slovník*. Přepisovací pravidla jsou definována ve tvaru $A \rightarrow B$, kde A a B jsou podtvary slovníku.

Pro použití pravidla na daný tvar (zde pro demonstraci nazveme C) nejdříve musí být v daném tvaru detekován podtvar A . Pro nalezení jsou používány Euklidovské transformace. Ty označíme jako funkci τ . Při nalezení je pak aplikováno vybrané pravidlo a výsledek je následující: $C' = (C \setminus \tau(A)) \cup \tau(B)$.

Zde tedy vidíme odlišnost vůči klasickým gramatikám. Nabízí se sice možnost v gramatice G dosadit namísto množiny prostorových těles množinu n-dimensionálních řetězců, tento přístup pak ale není příliš intuitivní ani názorný.

4.2.2 Třífázová gramatika

Zde demonstrujeme upravenou verzi původně dvoudimensionální gramatiky, představené v sekci 2.3.1.

Úpravou vznikla obsáhlejší n-tice gramatiky, princip byl ale zachován, přičemž se počet fází generování výsledného řetězce zvýšil na tři. Počet členů dané n-tice bude geometricky růst s přibývajícimi dimenzemi, definice se tedy pro vyšší počet dimenzí stane neúnosně obsáhlou, ovšem intuitivní postup gramatiky je zachován v libovolném prostoru.

Definice 36. *Třídimensionální hierarchická pravá lineární gramatika (3RLHG)* je definována jako n -tice $G = (V_h, V_v, V_s, \Sigma_I, \Sigma_J, \Sigma, S, R_h, R_v, R_s)$, kde:

- V_h je konečná množina *horizontálních proměnných*,
- V_v je konečná množina *vertikálních proměnných*,
- V_s je konečná množina *hloubkových proměnných*,
- $\Sigma_I \subseteq V_v$ je konečná množina *přechodných symbolů prvního typu*,
- $\Sigma_J \subseteq V_s$ je konečná množina *přechodných symbolů druhého typu*,
- Σ je konečná množina *terminálů*,
- $S \in V_h$ je *počáteční symbol*,
- R_h je konečná množina *horizontálních pravidel* ve tvaru $V \rightarrow AV'$ nebo $V \rightarrow A$, kde $V, V' \in V_h$ a $A \in \Sigma_I$,
- R_v je konečná množina *vertikálních pravidel* ve tvaru $W \rightarrow BW'$ nebo $W \rightarrow B$, kde $W, W' \in V_v$ a $B \in \Sigma_J$,
- R_s je konečná množina *hloubkových pravidel* ve tvaru $X \rightarrow aX'$ nebo $X \rightarrow a$, kde $X, X' \in V_s$ a $a \in \Sigma$.

Postup generování těles je pak následující: Gramatika $G_h = (V_h, \Sigma_I, S, R_h)$ generující řetězce, která je součástí gramatiky G , z počátečního symbolu derivacemi vytvoří jazyk řetězců L_1 . Prvky tohoto jazyka jsou pak vstupem pro gramatiku $G_v = (V_v, \Sigma_J, S_1 \in L_1, R_v)$, tato dvoudimensionální gramatika vytvoří pomocí svých pravidel z řetězců jazyka L_1 obrazy, z nichž se skládá dvoudimensionální jazyk L_2 . Konečně, obrazy obsažené v jazyce L_2 jsou vstupy gramatiky $G_s = (V_s, \Sigma, S_2 \in L_2, R_s)$, která z obrazů generuje odpovídající tělesa.

Mezistupně generování těles gramatikou tedy obsahují výstupy všech příbuzných gramatik pro nižší dimenze. Tento proces je obecný a bude totožný pro jakoukoliv sumu dimenzí. V každé fázi generování je na výstupu o dimenzi složitější útvar než na vstupu. Dá se tvrdit, že počáteční symbol S je jakýmsi primitivem bez dimenze, ze kterého jsou pak postupně generovány složitější tvary.

Pro lepší představu fungování hierarchické gramatiky ve třech dimenzích následuje jednoduchý příklad.

Příklad 8. Nechť máme abecedu $\Sigma_1 = \{\boxed{B}, \boxed{R}, \boxed{G}\}$ obsahující třídimensionální symboly značící barvy, kde \boxed{B} je modrá, \boxed{R} je červená a \boxed{G} je zelená.

Nad touto abecedou sestrojme 3RLHG gramatiku $G_1 = (V_h, V_v, V_s, \Sigma_I, \Sigma_J, \Sigma_1, S, R_h, R_v, R_s)$, kde:

- $V_h = \{S, A, B, B', C\}$,
- $V_v = \{A, B, X, X', Y, Z, W, V\}$,
- $V_s = \{X, Z, J, I, K, L\}$,
- $\Sigma_I = \{A, B\}$,
- $\Sigma_J = \{X, Z\}$,
- $R_h = \{S \rightarrow AB', B' \rightarrow BC, C \rightarrow A\}$,
- $R_v = \{A \rightarrow XY, Y \rightarrow ZX', X' \rightarrow X, B \rightarrow ZW, W \rightarrow XV, V \rightarrow Z\}$,
- $R_s = \{X \rightarrow \boxed{B}J, J \rightarrow \boxed{R}I, I \rightarrow \boxed{G}, Z \rightarrow \boxed{R}K, K \rightarrow \boxed{G}L, L \rightarrow \boxed{B}\}$.

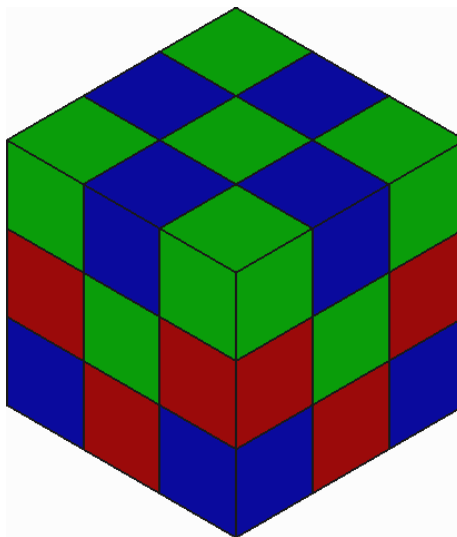
Demonstrujme si tedy generování tělesa $C_1 \in L(G_1)$. Nejdříve v první fázi získáme řetězec znaků náležejících do Σ_I z počátečního symbolu S .

$$S \rightarrow AB' \rightarrow ABC \rightarrow ABA$$

Ve druhé fázi získáme dvourozměrný řetězec (obraz) z řetězce získaného v první fázi.

$$ABA \rightarrow \begin{array}{ccc} X & Z & X \\ Y & W & Y \end{array} \rightarrow \begin{array}{ccc} X & Z & X \\ Z & X & Z \\ X' & V & X' \end{array} \rightarrow \begin{array}{ccc} X & Z & X \\ Z & X & Z \\ X & Z & X \end{array}$$

Nakonec ve třetí fázi získáme ze vzniklého obrazu těleso C_1 vyobrazené na obrázku 4.12.



Obrázek 4.12: Těleso C_1

4.2.3 Bezkontextová třídídimensionální gramatika

Jak bylo zmíněno v sekci 2.3.2, gramatika uvedená v práci [33] nabízí možnost rozšíření do tří dimenzí bez nutnosti rozšíření n-tice, která ji definuje. Pro třídídimensionální přístup ovšem přibývá nutných omezení, která zaručí, že výsledný obraz bude kvádrového tvaru, pro více dimenzí by tedy tato omezení představovala problém při definování jazyků.

Proto uvádíme pouze příklad definice této gramatiky, sestavení dodatečných omezení pro gramatiku je nad rámec této práce a může být výzvou pro autora původní gramatiky.

Definice 37. *Třídídimensionální bezkontextová gramatika* G je čtveřice (V_N, V_T, S_0, P) , kde:

V_N je konečná množina *neterminálů*,

V_T je konečná množina *terminálů*,

$S_0 \in V_n$ je *počáteční symbol*,

P je konečná množina *přechodových pravidel* ve tvaru $N \rightarrow W$, kde $N \in V_N$ a $W \in (V_N \cup V_T)^{***} \setminus \{\mu\}$.

Z definice lze vidět minimální změnu oproti dvoudídimensionálnímu případu, jako obecná definice pro n-dídimensionální bezkontextovou gramatiku se tedy tento typ jeví jako ideální kandidát.

V této sekci byli představeni zástupci gramatik pro třídimensionální jazyky a řetězce. Byla provedena úprava gramatik představených v sekci 2.3 pro demonstraci postupu k obecně n-dimensionálním gramatikám.

Jako nejintuitivnější přístup se jeví použití vícefázové gramatiky, která v každé fázi zvyšuje dimenzi výstupu. I přes zjevnou nevýhodu zvyšování počtu členů n-tice definující gramatiku je zachováno stejné chování gramatiky v každé fázi.

Naopak standardní gramatika bez více fází uvedená v definicích 16 a 37 zachovává v libovolném prostoru velikost čtveřice, ale tato skutečnost je vyvážena nepříznivým nárůstem podmínek pro generování jazyka, které vyžadují složité matematické konstrukce. Následuje sekce hledající ideální zástupce pro třídimensionální automaty, prohledávající prostor.

4.3 Třídimensionální automaty

V této sekci je uvedeno několik zástupců automatů, které jsou schopné analyzovat třídimensionální tělesa. Opět zmiňujeme myšlenku, že takové automaty jsou v principu součástí CAD modelářských systémů, i přes skutečnost, že tak nejsou označovány.

Téma třídimensionálních automatů není ani zdaleka tak rozšířené jako téma automatů dvoudimensionálních, proto uvádíme především nově vyvinuté automaty vzniklé v rámci této práce. To ale neznamená, že téma těchto automatů není vůbec publikováno, mezi významnější práce patří například [24], [39], [47] a [40].

Mezi zástupci je zahrnut obecný Turingův stroj pro třídimensionální pásku jako základní prvek pro výstavbu jiných automatů. Ve třech dimenzích zabírají velkou podmnožinu automatů také automaty celulární, jejichž zástupce jsme již analyzovali v sekci 3.2. Dále jsou uvedeny automaty vzniklé úpravou automatů z předchozí kapitoly, opět pro porovnání a postup k obecně n-dimensionálním automatům.

4.3.1 Třídimensionální Turingův stroj

Zde uvádíme definici a popis fungování Turingova stroje, představeného v práci [41]. Jedná se o klasický Turingův stroj obohacený o tzv. *ukládací pásku*, tedy mechanismus externí paměti stroje pro zápis.

V definici jsou kvůli srozumitelnosti ponechány původní anglické názvy pohybů po obou páskách. Definice byla převzata a upravena z výše uvedené práce.

Definice 38. *Třídimensionální Turingův stroj (3-TM) je šestice $M = (Q, q_0, F, \Sigma, \Gamma, \delta)$, kde:*

Q je konečná množina stavů,

$q_0 \in Q$ je počáteční stav,

$F \subseteq Q$ je množina konečných stavů,

Σ je konečná vstupní abeceda (kde $\# \notin \Sigma$ je hraniční symbol),

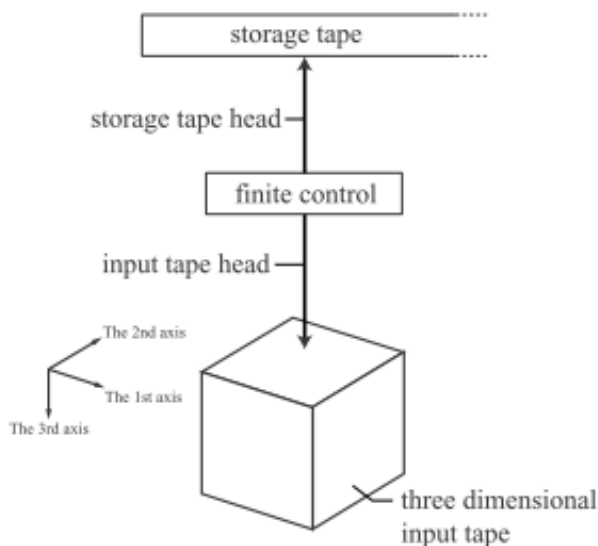
Γ je konečná abeceda ukládací pásky ($B \in \Gamma$ je prázdný symbol) a

$\delta \subseteq (Q \times (\Sigma \cup \{\#\}) \times \Gamma) \times (Q \times (\Gamma \setminus \{B\}) \times \{east, west, south, north, up, down, nomove\} \times \{right, left, nomove\})$ je relace dalšího kroku.

Krok výpočtu stroje se tedy podle relace δ skládá ze čtení symbolů z obou pásek a použití pravidla s korespondujícím stavem stroje, kdy stroj vykonává změnu stavu, zápis symbolu na ukládací pásku a pohyby po obou páskách.

V pásce reprezentující třídimensionální těleso stroj může pohybovat čtecí hlavou šesti směry či zůstat na daném místě a v ukládací pásce se může hlava stroje pohybovat ve dvou směrech.

Stroj je schopný ze vstupu pouze číst, veškeré zápisy jsou prováděny na ukládací pásce. Ilustrace stroje je na obrázku 4.13.



Obrázek 4.13: Třídimensionální Turingův stroj (3-TM) (zdroj: [41])

Síla automatu je dále diskutována v příslušném článku, kde je navržena i úprava na *poznávací (marker) Turingův stroj*, který rozšiřuje sílu klasického Turingova stroje díky možnému vícenásobnému označení některých symbolů v tělesu.

Další modifikací automatu je jeho úprava na *alternující Turingův stroj*, jehož definice obsahuje navíc tzv. *univerzální stav*. Dále jsou v článku diskutovány uzávěrové vlastnosti těchto a ostatních modifikací automatu.

Tento Turingův stroj představuje nejjednodušší model třídimensionálního automatu, přičemž další představené automaty budou obsahovat některé jeho rysy. Je to také důkaz jednoduchosti uzpůsobení Turingova stroje pro jakýkoliv počet dimenzí, což je pro téma této práce důležitá vlastnost.

4.3.2 Šestisměrný konečný automat

Tento automat je úpravou předchozího Turingova stroje, stejně tak jako zasazením automatu ze sekce 3.1 do třídimensionálního prostoru. Analyzovanou entitou je těleso, v němž automatem nejsou prováděny úpravy a automat se v tělesu pohybuje za použití šesti směrů.

Šestisměrný automat proti představenému Turingovu stroji nepřináší žádný nový druh pohybu v tělesu, ovšem lze jej upravit na 26-směrný, jak bude diskutováno dále.

Šest směrů navigace automatu je odvozeno od čtyř směrů v dvoudimensionálním prostoru, takže můžeme sledovat jednoduchou lineární závislost mezi počtem dimenzí a počtem směrů, jimiž se může automat pohybovat.

Definice 39. *Šestisměrný automat*, označován jako $6 - FA$, je sedmice $\Lambda = (\Sigma, Q, \Delta, q_0, q_a, q_r, \delta)$, kde:

Σ je vstupní abeceda,
 Q je konečná množina stavů,
 $\Delta = \{E, W, N, S, U, D\}$ je množina směrů,
 $q_0 \in Q$ je počáteční stav,
 $q_a, q_r \in Q$ jsou akceptující a odmítající stavy,
 $\delta : Q \setminus \{q_a, q_r\} \times (\Sigma \cup \{\#\}) \rightarrow Q \times \Delta$ je přechodová funkce.

Z podoby definice plyne, že úpravy pro změnu dimensionalitu automaty jsou minimální. Je pouze nutné uzpůsobit stroj pro pohyb v řetězcích daného prostoru. Automat tak splňuje požadavky pro vytvoření univerzálního konceptu pro libovolnou dimensionalitu prostoru, kdy při jistých úpravách takovýto automat může vymezovat obecné n -dimensionální regulární jazyky.

Tento koncept je ovšem podmíněn jistou úpravou hierarchie jazyků oproti klasické Chomského hierarchii, kdy, jak bylo zmíněno v předchozí kapitole, nelze vyvozovat pro tuto hierarchii pravidla podobnosti (či náležitosti do stejné množiny) pro jazyky napříč prostory s různým počtem dimenzí. Automat schopný analyzovat jazyky jisté úrovně Chomského hierarchie tedy nemusí být schopen tento úkol provést i pro jiný počet dimenzí, a to i za předpokladu, že disponuje množinou směrů postačujících pro analýzu těchto jazyků.

Logicky lze předpokládat, že automat definuje menší skupinu jazyků než Turingův stroj ze sekce 4.3.1. Nedisponuje totiž pamětí (ani zásobníkovou), což se negativně projevuje na jeho síle. Ovšem z hlediska přehlednosti automat logicky disponuje mnohem jednodušší přechodovou funkcí.

Nabízí se ovšem úprava množiny Δ tak, aby byl automat schopný pohybovat se po 26-okolí, což znamená možný přechod do 8-okolí symbolu ve dvou dimenzích nebo do 8-okolí symbolů nad či pod aktuálním symbolem ve třetí dimenzi plus do těchto symbolů samotných. Množina směrů je pak následující: $\Delta = \{E, SE, NE, W, WE, WN, N, S, U, U-E, U-SE, U-NE, U-W, U-WE, U-WN, U-N, U-S, D, D-E, D-SE, D-NE, D-W, D-WE, D-WN, D-N, D-S\}$

4.3.3 Třídimensionální celulární teselační automat

Jak bylo zmíněno v úvodu sekce, podstatnou část již vytvořené teorie pro třídimensionální automaty tvoří automaty celulární. Význam mají především v simulaci chování virtuálního třídimensionálního prostoru v reakci na podněty a při tvorbě umělé inteligence pro virtuální světy.

Také se těchto automatů používá pro simulaci zátěže modelu tělesa a odhalení slabých částí materiálu, tedy například pro tvorbu nosných konstrukcí. Automaty především umožňují analyzovat tělesa z nově vytvářených slitin a rozhodovat tak o vhodnosti jejich použití pro daný účel.

Zde uvádíme jako zástupce modifikaci automatu představeného v sekci 3.2, kde se jedná o speciální typ celulárního automatu. Třída celulárních automatů obsahuje mnoho takovýchto modifikací, představených například ve článcích [5], [22] a [49]. Následuje vytvořená definice.

Definice 40. *Třídídimensionální celulární teselační automat*, značíme $3-CTA$, je definován jako $M = (\Sigma, Q, I, F, \delta)$, kde:

Σ je vstupní abeceda,
 Q je konečná množina stavů,
 $I \subseteq Q$ je množina počátečních stavů,
 $F \subseteq Q$ je množina konečných stavů,
 $\delta : Q \times Q \times Q \times \Sigma \rightarrow Q$ je přechodová funkce.

Každé buňce (symbolu) v analyzovaném tělese (náležícího do Σ^{***}) je přiřazen stav náležící do množiny I . Při chodu automatu jsou pak stavy všech buněk měněny dle funkce δ . Parametry ovlivňujícími výběr následujícího stavu buňky na pozici (i, j, k) jsou obsah buňky, tedy symbol na dané pozici v tělese, a stavy tří buněk. Konkrétně se jedná o buňky na pozicích $(i + 1, j, k)$, $(i, j + 1, k)$ a $(i, j, k + 1)$, tedy tři sousední buňky, každá přiléhající k analyzované buňce ze směru jiné dimenze.

V této sekci byly demonstrovány vyvinuté automaty jako zástupci třídídimensionálních automatů, převážně vyvinutých na základě automatů pracujících s jiným počtem dimenzí. Úprava definic byla přitom, jak bylo demonstrováno, bez větších obtíží proveditelná pro jakýkoliv zvolený automat.

To pro tuto práci znamená, že vývoj obecně multidídimensionálních automatů může být elegantně proveditelný a to i navzdory faktu, že se jedná o pokročilé téma v oboru teoretické informatiky.

V následujících kapitolách se tedy budeme takovým vývojem zabývat a demonstrovat zvolené zástupce oproštěné od jakékoliv závislosti na zvoleném počtu dimenzí analyzovaného řetězce a jazyka těchto řetězců. Před touto kapitolou ovšem demonstrujme některé vybrané případy nasazení třídídimensionálních automatů v praxi.

4.4 Praktické využití automatů

Využití zde představených automatů a další třídídimensionálních automatů bylo výše zmíněno v úvodu sekce. Současné vědecké články se převážně zaměřují na analýzu těles vytvořených z nově vyvinutých slitin kovů pro simulaci jejich zátěže a zjištění jejich odolnosti.

Pokud upustíme od klasické definice, třídídimensionální automaty jsou prakticky využívány v CAD modelačních programech, kdy se s jejich pomocí provádí kontrola modelu z pohledu více kritérií. My zde uvádíme využití přímo třídídimensionálních automatů se znalostí jejich principů.

Analýza toku paketů

V sekci o využití dvoudídimensionálních automatů (3.6) bylo uvedeno využití dvoudídimensionálního automatu pro analýzu dvoudídimensionální reprezentace datových paketů.

Pokud tento koncept rozšíříme o dimenzi času a budeme tedy analyzovat více paketů, získáme třídídimensionální těleso tvořené prostorově konkatenovanými pakety, kdy pak lze provádět analýzu tohoto tělesa pomocí zvoleného třídídimensionálního automatu.

Využití v IDS

Některé vlastnosti třídimensionálních automatů vyzývají k jejich využití v IDS, tedy *Intrusion detection systems* (viz [29]), což jsou zařízení a mechanismy, které v počítačové síti umožňují detekovat bezpečnostní incidenty a případně pak provádět protiopatření.

Stejně jako v předchozím zmíněném využití by se jednalo o analýzu paketů, ovšem na více abstraktní úrovni. Zmíněné systémy využívají pokročilé filtry a vyhledávací mechanismy pro porovnávání posloupností událostí ve sledované síti se šablonami pro známé bezpečnostní hrozby.

Logicky se tak jedná o předzpracování datového toku v dlouhodobém měřítku a analýzu získaných údajů pomocí vybraného mechanismu. Ten hledá hrozby pomocí databáze signatur, čehož je schopen i třídimensionální automat při využití automatické úpravy konfigurace podle signatury hledaného typu bezpečnostního incidentu.

Virtuální realita

Uměle vytvořené světy jsou v dnešním světě populárním tématem. Lze se domnívat, že v budoucnu bude umožněno uživatelům nástrojů využívajících *augmented reality* vyvíjet do těchto virtuálních světů tělesa, která budou jeho součástí.

Takové objekty ovšem bude nutné analyzovat z hlediska schopnosti existovat v daných podmínkách a z ohledu jejich reakce na podněty ve virtuálním světě. Opět se tedy dostáváme k analýze třídimensionálních virtuálních těles z hlediska jejich strukturálních vlastností a ideálními kandidáty pro tento úkol jsou třídimensionální automaty.

Jelikož je pojem virtuální reality v současné době stále více známý a zařízení zprostředkující tuto realitu jsou postupně dostupnější a rozšířenější, lze se domnívat, že má toto pole nasazení třídimensionálních automatů velký potenciál.

V této kapitole byla představena obecná teorie třídimensionálních jazyků. Byl ustanoven termín *těleso* jako zástupce řetězců pro tři dimenze. Dále byly diskutovány změny klasických operací známých pro nižší dimenze, kdy byl učiněn závěr, že počet těchto operací a jejich druhů podstatně narůstá se zvyšujícím se počtem dimenzí. Například ve třech dimenzích existují tři druhy konkatenace a více druhů rotace, naopak pro klasické řetězce není ani operace rotace použita.

Dále byly uvedeny jazyky těchto řetězců, tedy množiny těles, kdy jazyk může být například množinou modrých krychlí, pokud poznatky zasadíme do reálného světa. Jazyky lze, jak už je pravidlem, ustanovit pomocí gramatik, které je vystihují a definují tělesa v nich obsažená.

Tyto gramatiky byly také popsány, konkrétně se jednalo o tři druhy gramatik známých či vyvinutých autory. Na vyvinutých bylo demonstrováno přesunutí teorie známé pro dvoudimensionální svět do tří dimenzí a ustanoven závěr, že nejvíce vhodnou gramatikou pro obecně n dimenzí by byla vícefázová gramatika, jejíž definice ovšem potřebuje úpravu, aby příslušná n -tice nebyla ovlivněna počtem dimenzí.

Snadnější transformací do tří dimenzí je pak přeměna dvoudimensionálních automatů na třídimensionální. Byl uveden zástupce klasických Turingových strojů (TS) a bylo demonstrováno, že koncept TS je obecně nezávislý na dimenzi a dokáže pracovat v libovolném prostoru při minimálních úpravách. Poté byli představeni zástupci automatů upravených autory pro tři dimenze.

Díky poznatkům zde uvedeným následuje stěžejní kapitola zasazující získané znalosti do obecně multidimensionálních prostorů.

Kapitola 5

Multidimensionální jazyky

V této kapitole autoři představují obecně *n-dimensionální jazyky*, tedy teorii formálních jazyků nezávislou na počtu dimenzí. Spolu s jazyky jsou představeny multidimensionální řetězce a operace pro práci s těmito řetězci, stejně, jako tomu bylo v předchozích kapitolách. I u těchto jazyků je definováno několik operací a příslušných uzávěrů.

Dále jsou definovány některé autory vyvinuté gramatiky jako příklad prostředku k definici uvedených jazyků. Tyto gramatiky jsou uvedeny bez hierarchie, která není pro multidimensionální jazyky definována.

Představené pojmy jsou vystavěny na postupu mezi dimenzemi, který byl podrobně popisován předchozími kapitolami. Díky tomu je možné sestavit obecné definice pro řetězce, jejich vlastnosti a operace s nimi.

Teorie multidimensionálních jazyků nebyla dle autorů v současné době nikde hlouběji prezentována, především ne v souvislosti s jazyky s fixní dimensionalitou. Proto se jedná o hlavní přínos této práce poli teoretické informatiky. Prezentované pojmy jsou definovány jednoduše a popisně, jelikož podrobnější vědecká analýza multidimensionálních jazyků je nad rámec této práce a především obsáhlým tématem.

Jak bylo demonstrováno v předchozích kapitolách, pole vícedimensionálních jazyků je totiž velmi zajímavým tématem, nabízejícím velké množství konceptů automatů či gramatik k vývoji a analýze. Také praktické využití těchto vyvinutých mechanismů je široké, což bylo dokázáno uvedenými příklady.

5.1 Řetězce jako buňky

Jako první pojem v multidimensionálních jazycích ustanovme termín *buňka*. Buňka je multidimensionální řetězec a jako buňku si lze představit i *obraz* či *těleso*. Řetězec je složen ze symbolů zvolené abecedy a má jeden nebo více rozměrů.

Definice 41. *Multidimensionální řetězec* (neboli *buňka*) s je konečné uskupení symbolů abecedy Σ takové, že $s \in \Sigma^{*n}$. Pro n pak platí $n \in \mathbb{N}, n \geq 1$

Už jen z definice 41 lze vyvodit některé vlastnosti řetězce pro obecný počet dimenzí. Předně, buňka není posloupnost symbolů (na rozdíl od klasického řetězce), ale jejich uskupení, jelikož pro více dimenzí ztrácí termín posloupnost objektivitu.

Pak je nutné poznamenat neznámou hodnotu proměnné n , která určuje dimensionalitu pro množinu všech řetězců nad zvolenou abecedou. Ta je stěžejní pro zvolené téma, jelikož tak je buňka obecným řetězcem v jakémkoliv dimenzi.

Důležitou otázkou je pak reprezentace buňky, kdy nelze využít ani prostorového zobrazení, jako tomu bylo u těles. Velmi složitá by pak pro více dimenzí byla i metoda zachycení dvoudimensiálních vrstev, jako tomu bylo v příkladu na obrázku 4.2. Proto se nabízí jediná varianta, a to zachycení jednotlivých symbolů v buňce pomocí jejich souřadnic, jako bylo demonstrováno u tělesa v příkladu 7.

Je zřejmé, že tento způsob navigace v řetězcích je pak nutno uplatnit i u příslušných automatů, více v 6.1.

5.1.1 Velikost buňky

Jelikož nelze předpokládat počet dimenzí vybraného řetězce, je pak problematickou otázkou určení jeho rozměrů (délky pro jednodimensionální řetězce). Tento údaj je totiž přímo závislý na počtu dimenzí ovlivňujícím množstvím údajů, které velikost buňky reprezentují.

Klasicky pak lze používat funkce $\ell_1 \dots \ell_n$, ale při neznámém počtu rozměrů není jasné, kolik druhů funkce ℓ je možné na vybranou buňku použít. Při získání počtu dimenzí to ale není problém. Pro neznámou dimensionalitu buněk pak zavádíme funkci Φ , definovanou v 42. Pro těleso o_1 v příkladu 7 pak tedy $\Phi(o_1) = (3, 4, 4)$ a pro obraz A v tabulce 2.1 pak $\Phi(A) = (5, 4)$. Je tedy vidět univerzálnost funkce Φ pro jakýkoliv typ buňky.

Definice 42. *Rozměrová funkce* (značíme Φ), kde $\Phi : \Sigma^{*n} \rightarrow \mathbb{N}^n$, navrácí pro danou buňku (řetězec) její n-tici rozměrů.

5.1.2 Prázdna buňka

Stejně jako v předchozích kapitolách je nutné definovat prázdný řetězec pro dané dimenze. V tomto případě se ovšem jedná o problém, jelikož, jak bylo zmíněno, případ musí být obecný pro jakýkoliv počet dimenzí. Zavedme tedy termín *prázdna buňka*, která je určena definicí 43.

Definice 43. *Prázdna buňka* (značíme ξ) je takový mutidimensionální řetězec, pro který platí, že $\Phi(\xi) = (a_0, a_1, \dots, a_n) \Rightarrow \forall i, 0 \leq i \leq n : a_i = 0$.

Pokud ale chce zachovat původní symboly *prázdneho obrazu*, *prázdneho tělesa* a *prázdneho řetězce*, nastává problém.

Zde zavádíme termín *třída prázdných řetězců* Ξ , kdy ekvivalent prázdného řetězce pro jakýkoliv počet dimenzí je členem této třídy. Tedy platí: $\{\varepsilon, \lambda, \mu, \xi\} \subset \Xi$. Poznamenejme také, že vždy pro konkrétní počet dimenzí bude platit: $\varepsilon = \xi$, $\lambda = \xi$, $\mu = \xi$ a podobně pro další prostory.

5.1.3 Omezení pro buňky

Je zřejmé, že nemůže existovat buňka s nulovou délkou v jedné či více dimenzích, nikoliv však ve všech. Tedy pokud máme buňku C_1 a platí $\Phi(C_1) = (a_0, a_1, \dots, a_n)$, pak platí následující: $\exists i, 0 \leq i \leq n : a_i > 0 \Rightarrow \nexists j, 0 \leq j \leq n : a_j = 0$.

Zároveň uveďme dosud používané omezení o „obdélníkovém“ tvaru řetězců, tedy v každé dvojici dimenzí buňky bude průmět buňky tvaru obdélníku. Bude tak zachována jednotná velikost pro vrstvy buňky a buňka nebude obsahovat žádné „díry“.

Dále je nutné poznamenat vliv neurčitého počtu dimenzí na pojem *podřetězce*. Ten nazveme v mutidimensionálních jazycích *podbuňka*. Matematická definice tohoto pojmu je

nedostatečně výstižná a komplikovaná, proto není uvedena. Podbuňka je klasickým podřetězcem dané buňky v klasickém smyslu.

Zde ale nastává problém s dimensionalitou podbuňky. Pokud zachováme v podbuňce stejný počet dimenzí, jde o triviální případ. Můžeme ale uvážit, že k dané buňce může existovat podbuňka s menším počtem dimenzí, která se ale v jistém průmětu buňky nachází, stejně tak, jako zvolený obraz by mohl být nalezen ve vrstvě příslušného tělesa.

Tedy zavedme omezení, že multidimensionální podbuňka může disponovat stejným nebo nižším počtem dimenzí, než má zvolená buňka.

5.2 Multidimensionální operace

Zde diskutujeme změny definic operací nad řetězci, aby vyhovovaly obecně pro multidimensionální řetězce. Nejdříve poukážme na fakt, který byl dokázán v předchozích kapitolách, a to, že mezi počtem dimenzí a počtem operací nad řetězci pro tyto dimenze platí přímá úměrnost.

To lze demonstrovat například na konkatenci, která má pouze jedinou podobu pro klasické řetězce a hned tři druhy pro třidimensionální řetězce. Z toho plyne, že obecně pro buňku o x dimenzích existuje x druhů konkatence. Zajímavým případem je pak také operace rotace, která pro jednodimensionální řetězce prakticky neexistuje. Tyto příklady jsou diskutovány dále u definic jednotlivých operací.

5.2.1 Operace konkatence

Výše bylo uvedeno, že počet konkatencí je pro určitý počet dimenzí vždy totožný právě s tímto počtem dimenzí. To nám umožňuje vyvodit jednoduchá omezení pro tuto operaci a uvést tak její následující definici.

Definice 44. Pro buňky c_1 a c_2 náležící do Σ^{*n} a operaci konkatence $\odot_i, i \leq n$ platí, že v průmětu i -té dimenze $c_1 \odot_i c_2 = c_1 c_2$.

Uvedme omezení, že pro konkatenci \odot_i musí konkaténované buňky mít stejné veškeré rozměry kromě rozměru s indexem i . Dále pro libovolné n a i platí následující: $(c_1 \in \Sigma^{*n} \wedge c_2 \in \Xi \wedge c_1 \odot_i c_2 = c_3) \Rightarrow c_3 = c_1$.

Nejvíce komplikovaným prvkem se pro operaci zdá zvolení průmětu konkrétní dimenze. Jedná se ovšem o pouhé určení dimenze, v níž proběhne konkatence a ve které se výsledná buňka bude rozměrově lišit od původních buněk. Tak tomu je i pro klasické řetězce a obrazy.

Obecně lze totiž při multidimensionálním pojetí jednoduchých řetězců například z řetězců $p_1 = „abcabcabc“$ a $p_2 = „xyzxyzxyz“$ vytvořit pomocí konkatence v druhé dimenzi (\odot_2) obraz v tabulce 5.1. Jedná se totiž o řádkovou konkatenci, představenou v sekci 2.2.2. Neurčený počet dimenzí nám tak umožňuje pomocí operací vytvářet buňky s více dimenzemi, než mají buňky na vstupu operace.

Fakticky jsou totiž řetězce p_1 a p_2 obrazy s rozměry $(1, 9)$. Uvedené řetězce (obrazy) jsou samozřejmě vždy i buňkami. Lze si pak povšimnout, že vzniklý obraz v tabulce 5.1 lze dále konkaténovat, a to s řetězci, obrazy i tělesy a kterýmikoliv buňkami, které budou splňovat výše uvedené omezení rozměrů.

a	b	c	a	b	c	a	b	c
x	y	z	x	y	z	x	y	z

Tabulka 5.1: Obraz $X_1 = p_1 \odot_2 p_2$

Obecný je i vliv konkatenace na rozměry výsledné buňky. Vezměme libovolné buňky c_a a c_b o rozměrech $|c_a| = (a_1, \dots, a_n)$ a $|c_b| = (b_1, \dots, b_m)$ a operaci \odot_i , kdy tyto buňky splňují pro danou operaci výše uvedené omezení rozměrů. Pak platí $|c_a \odot_i c_b| = (a_1, a_2, \dots, (a_i + b_i), a_{i+1}, \dots)$. To zpětně platí i pro řetězce, obrazy či tělesa.

5.2.2 Operace rotace

Závislost počtu proveditelných rotací buňky v konkrétním počtu dimenzí není tak přímá, jako tomu bylo u operace konkatenace. Například pro jednodimensionální řetězce není operace rotace vůbec uvedena. Poté je pro obrazy v sekci 2.2.2 zaveden jeden druh rotace a v sekci 4.1.2 jsou uvedeny tři druhy rotace dle os.

V závislosti na předchozí sekci ale můžeme opět zmínit fakt, že řetězec o délce n je při multidimensionálním vnímání i obraz s velikostí $(1, n)$. To znamená, že rotace je proveditelná i pro řetězec, ovšem výsledek bude vždy obrazem, pokud je délka řetězce větší než 1. Pro demonstraci je výsledek provedení dvoudimensionální rotace na řetězci $p_3 = abcabc$ zobrazen v tabulce 5.2.

a
b
c
a
b
c

Tabulka 5.2: Obraz $X_2 = p_3^R$

Z multidimensionálního pohledu pak můžeme pomocí rotace R^3 ze sekce 4.1.2 získat z řetězce i obrazu těleso. Obecně tedy pro každou buňku s jakýmkoliv počtem rozměrů existuje nekonečně mnoho druhů rotací, z nichž každá je prováděna podle normály plochy jednoho z průmětů dimenzí. Definujme tedy následovně.

Definice 45. Necht c_3 je buňka. Pak $c^{R^{ij}}$ je výsledek *multidimensionální rotace* podle normály průmětné roviny dimenzí i a j .

Vliv operace rotace na rozměry buněk je pak takový, že n -tice rozměrů výsledné buňky je permutací rozměrů původní buňky. Při změně dimensionalit pak platí, že n -tice rozměrů původní buňky je doplněna hodnotami 1 na počet prvků výsledné n -tice a poté je provedena permutace.

5.2.3 Další operace

Pro n -dimensionální buňku lze dále definovat i operaci transpozice. Ta, stejně jako operace rotace, bude probíhat nad zvoleným průmětem dvou dimenzí a bude provádět permutaci n -tice velikostí buňky.

Operace projekčního mapování (viz sekce 2.2.2) je naopak ponechána beze změny pro jakýkoliv počet dimenzí, neboť pouze nahrazuje každý symbol buňky jiným dle dané relace.

Pro zajímavost zde uvedme operace *roll-up* a *drill-down*, které jsou používány v oboru informačních systémů a dolování dat při práci nad *n-dimensionálními kostkami*. Definice a demonstrace těchto operací je možné nalézt v [30]. Tyto operace by při jisté úpravě agregačních postupů bylo možné použít i pro *n-dimensionální* buňky.

Je jasné, že pro větší počet dimenzí je možné nalézt i vyšší počet operací nad buňkami, tak, jak tomu bylo u dvoudimensionálních řetězců ve srovnání s jednodimensionálními. Problémem je však velká abstraktnost takových operací, proto bylo uvedeno jen několik zástupců těchto operací.

5.3 Multidimensionální jazyky

V této sekci se budeme zabývat multidimensionálními jazyky, jejich vlastnostmi a uzávěry. V předchozích sekcích, které demonstrovaly multidimensionální operace, byla demonstrována nutná univerzálnost těchto operací. Stejně univerzální pak musí být i zde představené jazyky. Nejprve tedy uvedme formální definici takového jazyka.

Definice 46. *Multidimensionální jazyk* L je množina multidimensionálních řetězců neboli buněk. Je tedy podmnožinou množiny Σ^{*n} .

Z definice 46 vyplývá, že prvky multidimensionálního jazyka jsou buňky, tedy i jednodimensionální řetězce, obrazy a tělesa. Navíc tyto uvedení zástupci buněk mohou být prvky jediného jazyka. Zástupcem multidimensionálního jazyka pak může být například množina $L_m = \{C(\text{obraz } z \text{ tabulky } 2.3), p_1(\text{řetězec ze sekce } 5.2.1), \xi\}$.

To znamená, že jazyky mohou být velmi variabilní v oblasti dimensionalit. Jelikož může daný jazyk obsahovat nekonečně mnoho buněk s různou dimensionalitou, je náročné postihnout takový jazyk prostředkem, jako je formální gramatika či automat. Tyto komplikace jsou náplní sekce 5.4 a kapitoly 6.

Zde uvedme, že hierarchie jazyků pro obecně n dimenzí není zavedena a je velmi nepravděpodobné, že klasický tvar Chomského hierarchie bude mít pro multidimensionální jazyky obdobu. Nalezení vyhovující hierarchie a prostředků, které budou prvky této hierarchie vystihovat, jako je tomu například u regulárních jazyků a konečného automatu, je proto zajímavým předmětem pro další výzkum.

5.3.1 Operace nad jazyky

V sekci 5.2 je uvedeno několik operací, z nichž každá má až nekonečně mnoho druhů. Tyto operace nad buňkami pak mohou být uplatněny nad jazyky buněk, a to stejně jako pro dvoudimensionální jazyky. Proto uvedme pouze definici multidimensionální konkatence jazyků (viz 47), ostatní operace pak budou definovány podobně.

Definice 47. Nechtě L_a, L_b jsou dvoudimensionální jazyky nad abecedou Σ , pak *multidimensionální konkatence* pro i -tou dimenzi L_a a L_b (značeno $L_a \odot_i L_b$) je definována následovně: $L_a \ominus L_b = \{p \odot_i q \mid p \in L_a \text{ a } q \in L_b\}$.

Fakt, že jazyky mohou obsahovat více druhů buněk (s různou dimensionalitou), pak musí být kompenzován omezením platného tvaru buňky uvedeného v sekci 5.1.3 a omezeními jednotlivých operací, která jsou uvedena u jejich definic.

5.3.2 Uzávěry jazyků

Jelikož není definována hierarchie multidimensionálních jazyků, je irrelevantní určit pro jednotlivé třídy jejich uzávěrové vlastnosti. Nicméně můžeme definovat uzávěry jazyků tak, jak tomu bylo v předchozích kapitolách pro demonstraci zápisu těchto uzávěrů.

Stejně jako v sekci definující operace nad jazyky uvedeme pouze jednoho zástupce uzávěrů, kdy ostatní budou definovány podobně.

Definice 48. Necht L je multidimensionální jazyk. *Multidimensionální konkatenční uzávěr* jazyka L pro dimenzi i , značen $L^{*\odot^i}$ je definován jako $L^{*\odot^i} = \bigcup_{n \geq 0} L^{n \odot^i}$, kde $L^{0 \odot^i} = \xi$, $L^{1 \odot^i} = L$, $L^{n \odot^i} = L \odot^i L^{(n-1) \odot^i}$.

V této sekci byly definovány multidimensionální jazyky obsahující různé typy buněk. Dále byly doplněny příkladové definice uzávěrů a operací nad touto množinou. Znovu podotýkáme, že nebyla ustanovena žádná definitivní hierarchie nad obecně multidimensionálními jazyky vzhledem k velké náročnosti tématu.

V další sekci jsou ale uvedeny některé gramatiky generující jazyky tohoto typu, které umožňují vytvořit alespoň některé třídy jazyků bez uvedení hierarchie.

5.4 Multidimensionální gramatiky

V této sekci jsou představeni zástupci obecně n -dimensionálních gramatik spolu s problémy, které jsou spojeny s jejich definicemi.

Nejdříve je nutné poznamenat, že jazyk generovaný multidimensionální gramatikou je podle definice 46 množinou buněk s různou dimensionalitou. To činí problém pro jeho generování pomocí hierarchické gramatiky, představené například v 2.3.1, jelikož v jakékoli dimenzi může mít vygenerovaná buňka konečnou dimensionalitu. Tento problém tedy musí být brán v potaz.

Další důležitá omezení uvedená v 5.1.3 jsou pro gramatiky také závazná. Buňky tedy musí mít při libovolném počtu dimenzí pravidelný tvar, což činí problém například pro gramatiku jednofázového typu. Naopak představené *pure* gramatiky tyto problémy řeší elegantně, proto lze zvážit jejich použití.

5.4.1 Multidimensionální hierarchická gramatika

Jedná se o nejlépe popsatelnou formu gramatiky pro multidimensionální jazyky. Její formy už byly představeny v této práci v sekcích 2.3.1 a 4.2.2. Tyto gramatiky nejdříve generují jednoduchý řetězec složený z přechodových znaků, z něhož jsou dále generovány vícedimensionální buňky dle zadaného počtu dimenzí.

Dříve představené gramatiky tak generovaly pouze buňky o dané dimensionalitě a dle toho byla ovlivněna i velikost n -tice, která definovala tyto gramatiky. Pro další použití je tedy třeba upravit definici gramatiky tak, aby výsledná n -tice měla pevný počet prvků a byla schopná generovat buňky libovolné dimensionalit. Taková gramatika je uvedena v definici 49.

Definice 49. *Multidimensionální hierarchická gramatika (M-DHG)* je definována jako pětice $G = (V, \Sigma_I, \Sigma, S, R)$ kde:

V je množina proměnných, obsahující množiny V_1, \dots, V_n ,

Σ_I je množina přechodných symbolů, obsahující množiny $\Sigma_1, \dots, \Sigma_n$, kdy platí $\Sigma_i \subseteq V_i$ pro libovolné $1 \leq i \leq n$,

Σ je množina terminálů,

$S \in V_1$ je počáteční symbol,

R je množina pravidel ve tvaru $V \rightarrow AV'$, či $V \rightarrow A$, kde $V, V' \in \bigcup_{i=1}^n V_i$ a pro dané i

$$A \in \Sigma \cup \bigcup_{j=i}^n \Sigma_j.$$

Kvůli omezení tvaru výsledné buňky pak uvedme *relaci stejné třídy* nad množinou R , kdy platí, že pravidla R_1 a R_2 jsou ve stejné třídě, pokud jsou obě stejného tvaru, tedy z definice 49 tvaru $V \rightarrow AV'$, nebo $V \rightarrow A$, a zároveň platí, že zmíněná V a V' patří pro obě pravidla do stejné množiny V_i a stejně tak pro A , které patří u obou pravidel buď do množiny Σ , či totožné množiny Σ_i .

V každé fázi generování řetězce se pak mohou paralelně použít pouze pravidla, která jsou ve stejné třídě. Tím je zamezeno nepravidelné generování buňky a dodržena nezávislost gramatiky na počtu dimenzí. Výsledná gramatika tak může generovat multidimensionální jazyk obsahující buňky různé dimensionalit.

5.4.2 Bezkontextová multidimensionální gramatika

Zde uvádíme rozšířenou verzi definice třídimensionální gramatiky uvedené v sekci 4.2.3. Opět se rovná pouze o úpravu gramatiky a komplikovaná omezení pro generování buněk korektního tvaru s různou dimensionalitou nejsou uvedena. Tato obecná omezení mohou být opět výzvou pro další výzkum, jelikož tento druh gramatiky může vymezovat třídu bezkontextových multidimensionálních jazyků.

Dříve bylo uvedeno, že hierarchie pro multidimensionální jazyky není ustanovena a jedná se o velmi komplexní téma, které se nijak nemůže opírat o teorie Chomského hierarchie jazyků kvůli jistým modifikacím. Zde uvedený druh gramatiky, definovaný v 50, by ale mohl vymezovat jednu z úrovní hierarchie, která by pro multidimensionální jazyky mohla být v budoucnu vytvořena.

Definice 50. *Multidimensionální bezkontextová gramatika* G je čtveřice $G = (V_N, V_T, S_0, P)$, kde:

V_N je konečná množina *neterminálů*,

V_T je konečná množina *terminálů*,

$S_0 \in V_n$ je *počáteční symbol*,

P je konečná množina *přechodových pravidel* ve tvaru $N \rightarrow W$, kde $N \in V_N$ a $W \in \bigcup_{i=1}^n (V_N \cup V_T)^* \setminus \Xi$.

V této sekci byli představeni dva zástupci obecně n -dimensionálních gramatik, schopní generovat multidimensionální jazyky tak, jak jsou definovány v definici 46. Jako velmi zajímavý případ se jeví *multidimensionální hierarchická gramatika*, jejíž koncept je jednoduše pochopitelný a definice 49 ji zcela popisuje. Daná omezení pro buňky jsou u obou gramatik

dodržena a první gramatika disponuje navíc definovanou relací na její množině pravidel, která umožňuje tato omezení splňovat.

Tato kapitola představila čtenářům multidimensionální pohled na formální jazyky. Byl ustanoven pojem *buňka* jako zástupce řetězců pro neurčený počet dimenzí. Dále byla ustanovena množina prázdných řetězců pro libovolnou dimensionalitu a operace umožňující určení velikosti buňky. Také byla uvedena nutná omezení pro tvar a vlastnosti buněk.

Následně byly nad těmito multidimensionálními řetězci definovány operace reflektující absenci určitosti počtu dimenzí. Bylo ustanoveno, že pro buňky existuje mnoho druhů operací a jejich typů, například *multidimensionální konkatence* a *rotace*. Zde bylo třeba uvědomit si fakt, že i klasický řetězec může být v multidimensionálním chápání například tělesem a lze s ním tak operovat.

Jako množiny buněk různých velikostí a dimensionalit byly ustanoveny jazyky, k nimž byly uvedeny příklady definic jazykových operací a uzávěrů. V poslední sekci kapitoly pak byly představeny gramatiky, které umožňovaly definovat takovéto jazyky. Jako další možnost popsání multidimensionálních jazyků představujeme v následující kapitole *multidimensionální automaty*.

Kapitola 6

Multidimensionální automaty

V této kapitole se zabýváme dalším stěžejním tématem této práce, kterým jsou multidimensionální automaty. Jedná se o stroje umožňující analyzovat buňky a schopné dle výsledného stavu určit jejich náležitost do daného multidimensionálního jazyka.

Koncept multidimensionálních automatů je dalším autory vyvinutým tématem představeným v této práci. Definice uvedené v této kapitole jsou tedy vytvořeny autory práce a proto není uváděn jejich zdroj.

V kapitole jsou postupně rozebrány problémy definice automatů pro multidimensionální buňky, z nichž některé byly již zmíněny v předchozí kapitole v rámci generování jazyků pomocí multidimensionálních gramatik. Dále je představeno několik zástupců vyvinutých multidimensionálních automatů, z nichž někteří budou implementováni aplikací v kapitole 7. Na konci kapitoly je zmíněno několik praktických případů, kde by mohly být multidimensionální automaty využity.

6.1 Pravidla pro automaty

Zde diskutujeme nutná opatření pro daný typ automatů. Multidimensionální automaty pracují obecně s buňkami, což znamená, že není předem určena dimensionalita buňky na vstupu. To je první velkou komplikací při definici automatů. Podle své definice je ale buňka vždy konečné uskupení symbolů, takže ji lze analyzovat.

Dalším problémem je tvar pravidel, kdy nelze používat písmenné označení, které bylo využíváno například u automatů v sekcích 3.1 a 4.3.2. Jelikož existuje obecně nekonečný počet dimenzí, není praktické označovat pro každou dimenzi několik směrů pohybu pomocí speciálního značení.

Problém různé dimensionalitě buněk na vstupu automatů se týká především možnosti automatu přepadnout mimo řetězec. Pokud by totiž byl na vstupu klasický řetězec a automat použil pravidlo vedoucí na pohyb ve vyšší dimenzi, došlo by k vystoupení automatu z těla řetězce.

Tomu je nutné předejít a vhodným se zdá princip, který byl použit v sekci 3.4. Jedná se o vyznačení hranic obrazu pomocí speciálních symbolů. Speciální přístup vyžadují „rohy“ hranic, kdy je nutné zabránit více druhům pohybu než u klasického symbolu hranice.

Zavedme tedy stejný princip u buněk na vstupu libovolného automatu uvedeného v této sekci. Každá buňka libovolné dimensionalitě bude ve všech dimenzích ohraničena speciálními symboly a v rozích těchto hranic dalšími speciálními symboly. Každý zde uvedený

automat pak bude disponovat prostředky (pravidly) k zachycení těchto symbolů pod hlavou automatu a učinění opatření zabraňujících přepadnutí hlavy automatu mimo řetězec.

Pro tvary pravidel, přesněji určení směru, které přechodové pravidlo udává, pak budeme používat souřadnicový systém, uvedený v sekci 5.1.

6.2 Multidimensionální Turingův stroj

Jako první ze všech automatů definujeme pro multidimensionální jazyky uzpůsobený tvar Turingova stroje. Jak bylo ukázáno v sekci 4.3.1, tento model je jednoduše přizpůsobitelný a Turingův stroj jako koncept je jedním ze základních prvků pro definici odnože jazyků.

Také je pomocí tohoto stroje možné definovat třídu jazyků tak, jak je tomu u klasických formálních jazyků ve formě *kontextových jazyků*. Lze tedy říci, že zde uvedený Turingův stroj vymezuje třídu *multidimensionálních kontextových jazyků*.

Zde uvedený Turingův stroj, definovaný v 51, tedy musí pracovat nad buňkou libovolné dimensionalit, vždy tedy disponuje mechanismy, uvedenými v sekci 6.1. Zároveň, jelikož se autorům zdá tento koncept jako vhodný, automat disponuje pomocnou jednodimensionální nekonečnou páskou pro zápis a čtení, stejně jako v definici 38. Obdoba tohoto automatu byla stručně nastíněna v práci [23], ovšem zde se jedná o nově vyvinutou definici.

Definice 51. *Multidimensionální Turingův stroj (M - TM)* je sedmice $M = (Q, q_0, F, \Sigma, \Gamma, W, \delta)$, kde:

Q je konečná množina *stavů*,

$q_0 \in Q$ je *počáteční stav*,

$F \subseteq Q$ je množina *konečných stavů*,

Σ je konečná *vstupní abeceda* (kde $\boxtimes \cap \Sigma = \emptyset$ je *množina hraničních symbolů*),

Γ je konečná *abeceda ukládací pásky* ($B \in \Gamma$ je prázdný symbol),

W je *množina směrů*, kdy $W \subseteq \{-1, 0, 1\}^n$ a $\forall w = (w_1, w_2, \dots, w_n) \in W$ platí $1 \leq i \leq n$:
 $\exists w_i = \pm 1 \wedge (\forall j, 1 \leq j \leq n, j \neq i : w_j = 0)$,

$\delta \subseteq (Q \times (\Sigma \cup \boxtimes) \times \Gamma) \times (Q \times (\Gamma \setminus \{B\}) \times (W \cup \{\text{nomove}\}) \times \{\text{right}, \text{left}, \text{nomove}\})$ je *relace dalšího kroku*.

Z definice 51 lze zaznamenat přítomnost pravidel ve tvaru určeném v sekci 6.1, tvary těchto pravidel pak názorně prezentujeme v příkladu 9. Dále Turingův stroj už ze samé definice disponuje *hraničními symboly*, pravidla pro multidimensionální automaty jsou tedy splněna.

Stroj M - TM pak má na vstupu zadanou buňku obklopenou symboly z množiny \boxtimes , v níž se pohybuje pomocí ortogonálních pohybů. Pohybuje se tak stejným způsobem jako klasický Turingův stroj a stroj definovaný v sekci 4.3.1.

Při chodu stroje je pak vhodné vybraným způsobem omezit prohledávání prostoru buňky, aby bylo možné korektní zastavení automatu. Toho je dosaženo například konečnou množinou pravidel přechodů, kdy jsou zadanými pravidly možnosti pohybu stroje omezeny pouze na některé dimenze.

Příklad 9. Zde demonstrujeme tvar pravidel pro stroj M - TM . Pokud by například bylo dostačující disponovat pouze pravidly pro navigaci ve čtyřech dimenzích, množina W by obsahovala následující prvky:

- $w_1 = (1, 0, 0, 0)$

- $w_2 = (-1, 0, 0, 0)$
- $w_3 = (0, 1, 0, 0)$
- $w_4 = (0, -1, 0, 0)$
- $w_5 = (0, 0, 1, 0)$
- $w_6 = (0, 0, -1, 0)$
- $w_7 = (0, 0, 0, 1)$
- $w_8 = (0, 0, 0, -1)$

Samozřejmě relace δ navíc disponuje směrem *nomove*, kdy příslušný vektor by byl následující: $nomove = (0, 0, 0, 0)$. Lze pozorovat, že pro každou dimenzi jsou vždy použity dva směry navigace plus jeden speciální směr určující žádný pohyb.

Tím je splněna analogie s klasickým Turingovým strojem pracujícím v jedné dimenzi. Pro neurčitý počet dimenzí je pak množina W samozřejmě nekonečná.

Zde byl představen první zástupce multidimensionálních automatů, a sice *multidimensionální Turingův stroj*, jehož definice ukázala jednoduchost přizpůsobení klasického konceptu Turingova stroje multidimensionálnímu pojetí. Byla představena množina platných směrů navigace v multidimensionálním tělese, která bude používána i dalšími automaty spolu s omezeními pro její prvky.

Dále bylo uvedeno, že jako Turingův stroj může tento automat vymezovat třídu multidimensionálních jazyků, konkrétně tzv. *třídu kontextových multidimensionálních jazyků*. Jedná se o další zajímavou třídu jazyků spolu s třídou zmíněnou u gramatiky definované v sekci 5.4.2.

Automat představený v následující sekci je vlastně specializovaným případem zde popsaného stroje.

6.3 Vícesměrný multidimensionální konečný automat

Dalším představitelem automatů pracujících nad buňkami je *vícecestný konečný automat*. Tento automat také disponuje množinou směrů, stejně jako stroj definovaný v předchozí sekci. Jedná se o obdobu automatů představených v sekcích 3.1 a 4.3.2.

U těchto automatů byl vždy počet směrů omezen na dvojnásobek počtu dimenzí. Zde ale nedisponujeme informací o počtu dimenzí buněk na vstupu automatu, dokonce tento údaj musí být variabilní, tedy automat disponuje totožnou nekonečnou množinou směrových vektorů jako Turingův stroj v definici 51.

Jelikož se ale jedná o konečný automat, neexistuje zde dodatečný pohyb pro setrvání hlavy automatu na stejném místě a zároveň automat nedisponuje páskou či jakoukoliv možností zápisu. Konfigurace automatu se tak skládá pouze z pozice hlavy automatu a stavu automatu.

I když nebyla zavedena hierarchie multidimensionálních jazyků, lze bez nutného dokazování tvrdit, že $\mathcal{L}(M - MWFA) \subset \mathcal{L}(M - TM)$, jelikož zde uvedený automat je obdobou uvedeného Turingova stroje bez pomocných mechanismů, jako je páska pro zápis.

Definice 52. *Vícsměrný multidimensionální automat*, označován jako $M - MWFA$, je sedmice $M = (\Sigma, Q, W, q_0, q_a, q_r, \delta)$, kde:

- Σ je vstupní abeceda,
- Q je konečná množina stavů,
- W je množina směrů, totožná s W v definici 51,
- $q_0 \in Q$ je počáteční stav,
- $q_a, q_r \in Q$ jsou akceptující a odmítající stavy,
- $\delta : Q \setminus \{q_a, q_r\} \times (\Sigma \cup \boxtimes \text{ (taktéž z definice 51) }) \rightarrow Q \times W$ je přechodová funkce.

Automat, uvedený v definici 52, bude dále v příkladu 10 demonstrován, jelikož se jedná o základní tvar konečného automatu pro multidimensionální jazyky.

Příklad 10. Mějme *vícsměrný multidimensionální automat* $M_1 = (\Sigma, Q, W, q_0, q_a, q_r, \delta)$, kde:

$$\Sigma = \{a, b, c\},$$

$$Q = \{q_0, q_a, q_r, B, C, D, E, F, E_1, C_1, F_1\}$$

Také platí následující:

$$\begin{aligned} \delta(q_0, a) &= (D, (1, 0, 0, 0)), & \delta(B, c) &= (q_a, (-1, 0, 0, 0)), \\ \delta(D, a) &= (C, (0, 0, 1, 0)), & \delta(F, a) &= (E_1, (0, -1, 0, 0)), \\ \delta(C, b) &= (E, (0, 1, 0, 0)), & \delta(E_1, b) &= (C_1, (0, 0, -1, 0)), \\ \delta(E, a) &= (F, (0, 0, 0, 1)), & \delta(C_1, a) &= (F_1, (-1, 0, 0, 0)), \\ \delta(B, a) &= (q_a, (-1, 0, 0, 0)), & \delta(F_1, a) &= (q_a, (0, 0, 0, -1)), \\ \delta(B, b) &= (q_a, (-1, 0, 0, 0)), & & \end{aligned}$$

Pravidla daná funkcí δ mohou být případně doplněna dodatečnými pravidly, ovšem jsou implicitně doplněna pravidly, která zabezpečují automat proti přepadnutí hlavy, jak je nutné dle sekce 6.1.

a) Vložení 4-rozměrné buňky na vstup automatu

Zde demonstrujeme chod automatu M_1 na buňce C_1 o rozměrech $(2, 2, 2, 2)$, zadané následovně:

$$\begin{array}{cccc} C_1(1, 1, 1, 1) = a, & C_1(2, 1, 1, 1) = a, & C_1(1, 1, 1, 2) = a, & C_1(2, 1, 1, 2) = a, \\ C_1(1, 2, 1, 1) = b, & C_1(2, 2, 1, 1) = c, & C_1(1, 2, 1, 2) = c, & C_1(2, 2, 1, 2) = a, \\ C_1(1, 1, 2, 1) = b, & C_1(2, 1, 2, 1) = b, & C_1(1, 1, 2, 2) = a, & C_1(2, 1, 2, 2) = b, \\ C_1(1, 2, 2, 1) = c, & C_1(2, 2, 2, 1) = a, & C_1(1, 2, 2, 2) = c, & C_1(2, 2, 2, 2) = a, \end{array}$$

Konfigurace automatu M_1 (uvedeny ve formě: *(stav automatu, pozice hlavy)*) při analýze buňky C_1 jsou pak následující:

1. $(q_1, (1, 1, 1, 1))$
2. $(D, (2, 1, 1, 1))$
3. $(C, (2, 1, 2, 1))$
4. $(E, (2, 2, 2, 1))$
5. $(F, (2, 2, 2, 2))$
6. $(E_1, (2, 1, 2, 2))$
7. $(C_1, (2, 1, 1, 2))$
8. $(F_1, (1, 1, 1, 2))$
9. $(q_a, (1, 1, 1, 1))$

Automat tedy bez problémů analyzuje zadanou 4-rozměrnou buňku C_1 a ukončí svůj chod přechodem do *přijímajícího stavu* q_a .

b) Vložení 3-rozměrné buňky (tělesa) na vstup automatu

Nyní bude na vstup automatu M_1 vložena buňka (těleso) C_2 s rozměry $(2, 2, 2)$ zadaná následovně:

$$\begin{array}{ll} C_2(1, 1, 1) = a, & C_2(2, 1, 1) = a, \\ C_2(1, 2, 1) = b, & C_2(2, 2, 1) = c, \\ C_2(1, 1, 2) = b, & C_2(2, 1, 2) = b, \\ C_2(1, 2, 2) = c, & C_2(2, 2, 2) = a, \end{array}$$

Lze si povšimnout, že buňka C_1 mohla vzniknout pomocí konkatenace \odot_4 (ze sekce 5.2.1) buňky C_2 s další buňkou.

Zároveň podotýkáme, že buňku C_2 tak lze vnímat jako buňku 4-rozměrnou o rozměrech $(2, 2, 2, 1)$ a jako takovou ji bude vnímat i automat M_1 . Definice symbolů v buňce C_2 bude ve 4-rozměrech totožná pouze s dodatečným údajem 1 pro čtvrtou dimenzi.

Výše uvedená pravidla by pro buňku C_2 vedla k přepadnutí hlavy či zaseknutí automatu. Předpokládáme ale existenci pravidel, která tomu zabrání. Konkrétně zde předpokládáme pravidla, která při výskytu konkrétního hraničního symbolu pod hlavou automatu změni stav automatu na B (*boundary*) a vrátí hlavu automatu do předchozí pozice.

Pak jsou konfigurace automatu tyto:

- | | |
|--------------------------|--|
| 1. $(q_1, (1, 1, 1, 1))$ | 5. $(F, (2, 2, 2, 2))$ – hraniční symbol |
| 2. $(D, (2, 1, 1, 1))$ | 6. $(B, (2, 2, 2, 1))$ |
| 3. $(C, (2, 1, 2, 1))$ | 7. $(q_a, (1, 2, 2, 1))$ |
| 4. $(E, (2, 2, 2, 1))$ | |

V tomto příkladu byl demonstrován chod jednoho konkrétního automatu na dvou buňkách s rozdílnou dimensionalitou.

V této sekci byl popsán jednoduchý multidimensionální konečný automat. Bylo poukázáno na hierarchii jazyků tohoto automatu a Turingova stroje představeného v předchozí sekci. Postup automatu byl také demonstrován na jednoduchém příkladu.

Tento automat disponuje dvěma směry pohybu v každé použité dimenzi a na rozdíl od Turingova stroje postrádá možnost zápisu informace.

Dodržení pravidel ze sekce 6.1 je možné pomocí implicitního doplnění množiny pravidel o pravidla zabráňující přepadnutí hlavy automaty mimo buňku.

Dimensionalita vstupu není nijak omezena, jelikož pomoc směrů v množině W může automat postupovat v libovolné dimenzi za předpokladu, že není vykonán přechod na hraniční symboly.

6.4 Multidimensionální celulární automaty

Zde uvádíme teorii týkající se celulárních multidimensionálních automatů. Definice automatu nebyla uvedena kvůli dále zmíněným problémům.

V sekci 3.2 bylo představeno využití celulárního automatu pro analýzu obrazů. Je tedy logické, že tohoto postupu by bylo možné využít i pro analýzu multidimensionálních buněk.

Problém ovšem nastává u neurčité dimenze dané buňky. Pokud je náš cíl tímto automatem definovat nějaký multidimensionální jazyk, je nutné, aby automat umožňoval analyzovat buňky různé dimensionalitě tak, jak je to uvedeno v definici 46.

To však u celulárních automatů činí problém, jelikož postupují dle prvků okolí daného prvku. Toto okolí je dáno při definici zvoleného automatu a nelze je modifikovat dle potřeby.

Jelikož je řešení tohoto problému velmi náročné, není zde uvedena definice takového celulárního automatu. Autoři se domnívají, že při využití *hraničních symbolů* by mohla být konstrukce takového automatu realizovatelná, ovšem její náročnost přesahuje rámec a rozsah této práce.

Na rozdíl od ostatních multidimensionálních automatů jsou multidimensionální celulární automaty více diskutovaným tématem. Poznatky z vědeckých článků, zabývajících se tímto tématem, jsou ale často nesouvisející s formálními jazyky, takže nelze využít principů uvedených v těchto článcích. Významnější články z oboru simulací na téma multidimensionálních celulárních automatů jsou například tyto: [6] a [3].

V prvním ze článků například probíhá konstrukce multidimensionálního celulárního automatu tak, že je provedena operace *slicing*, tedy projekce všech průmětů dimenzí na několik dvoudimensionálních polí. K automatu je pak přistupováno jako ke klasickému celulárnímu automatu.

Tento postup se ale jeví jako příliš komplikovaný pro použití ve formálních jazycích. Navíc se u celulárních automatů standardně nepoužívá omezení vstupního simulačního systému na „obdélníkové“ tvary, což může vést k dalším komplikacím při snaze použít tyto automaty pro analýzu multidimensionálních řetězců.

6.5 Multidimensionální zapomínající automat

V sekci 3.3 bylo uvedeno, že koncept zapomínajícího automatu by byl dobře použitelný i pro multidimensionální jazyky. Jelikož definice tohoto automatu je závislá na definici Turingova stroje, který už je zde definován, je zřejmé, že použití upraveného tvaru tohoto automatu nad buňkami je možné.

Zapomínající automaty se v sekci 3.3 a práci [33] ukázaly jako silný druh automatů, v uvedené práci je dokonce proveden důkaz o následujícím vztahu: $\mathcal{L}(4-NFA) \subset \mathcal{L}(DFA)$. Nelze přímo vyvodit, zda platí i vztah $\mathcal{L}(M-FOA) \subset \mathcal{L}(M-MWFA)$, ovšem předchozím dokázaným vztahem je demonstrována síla zapomínajících automatů.

Pro definici multidimensionálního zapomínajícího automatu je třeba definice multidimensionálního Turingova stroje. Ta je uvedena v sekci 6.2, ovšem tento automat funguje na bázi Turingova stroje bez pomocné pásky, umožňujícího zapsat přímo do analyzované buňky. Příslušný Turingův stroj je tedy definován v 53 (definice byla vytvořena autory) a na základě této definice je definován zapomínající automat v definici 54, odvozené z definice v sekci 3.3.

Definice 53. *Multidimensionální Turingův stroj bez pomocné pásky* je sedmice $(Q, \Sigma, \Sigma_0, q_0, W, \delta, Q_F)$, kde:

- Q je konečná množina stavů,
- Σ je abeceda pásky,
- $\Sigma_0 \subset \Sigma$ je vstupní abeceda,
- $q_0 \in Q$ je počáteční stav,
- W je množina směrů z definice 51,
- $\delta : \Sigma \times Q \rightarrow 2^{\Sigma \times Q \times W}$ je přechodová funkce a
- $Q_F \subseteq Q$ je množina koncových stavů.

Vždy předpokládáme existenci množiny hraničních symbolů \boxtimes pro značení okrajů multidimensionální pásky.

Podotýkáme, že Turingův stroj z definice 53 je platným multidimensionálním automatem a nejsou porušena žádná pravidla, uvedená v 6.1. Stroj implicitně disponuje mechanismy zabraňujícími přepadnutí hlavy z těla buňky, stejně jako předchozí uvedené automaty.

Definice 54. *Multidimensionální zapomínající automat (M-FOA)* je multidimensionální Turingův stroj $(Q, \Sigma, \Sigma_0, q_0, W, \delta, Q_F)$, kde platí: $\Sigma = \Sigma_0 \cup \boxtimes \cup \{@\}$, $@ \notin \Sigma_0$ je tzv. *symbol vymazání*. Pokud pro funkci δ platí $\delta(a, q) = (a', p, d)$, pak $a = a'$, nebo $a' = @$.

V článku [33] jsou demonstrovány komplexní postupy, jak lze pomocí vymazávání symbolů pásky ukládat informace o pásce, což naznačuje možnou platnost výše uvedeného stavu $\mathcal{L}(M - FOA) \subset \mathcal{L}(M - MWFA)$.

Multidimensionální zapomínající automat totiž disponuje všemi směry pohybu jako *víceměrný konečný automat* a navíc umožňuje ukládání informací, což je mechanismus obecně zvyšující sílu automatu.

V této sekci byl představen modifikovaný zapomínající automat pracující nad buňkami s libovolnou dimensionalitou. Bylo demonstrováno vystavění automatu nad multidimensionálním Turingovým strojem a dále byla vysvětlena síla tohoto typu automatu a jeho potenciální větší vyjadřovací síla než síla klasických automatů.

6.6 Využití multidimensionálních automatů

Stejně jako v předchozích kapitolách i pro multidimensionální automaty v této sekci uvádíme několik potenciálních případů užití. Tím demonstrujeme možný přínos multidimensionálních automatů v praxi, kde se může uplatnit jejich schopnost analyzovat entitu bez nutnosti uvedení počtu dimenzí. Podotýkáme, že díky této vlastnosti jsou tyto automaty schopné nasazení ve všech případech užití, které byly demonstrovány v předchozích kapitolách.

OLAP systémy

Databázové systémy využívající technologii OLAP (*Online Analytical Processing*), sloužící k analýze dat a získávání znalostí, často využívají multidimensionální model k reprezentaci dat (viz [46]).

Tento model dat má samozřejmě proměnlivou dimensionalitu dle složitosti zobrazovaných dat. Často se tato dimensionalita mění díky použití operací *roll-up* a *drill-down*, které již byly zmíněny v sekci 5.2.3.

Existuje několik mechanismů sloužících k analýze těchto *n-dimensionálních kostek*, ale mezi tyto mechanismy by s příslušnými úpravami mohl být zahrnut i multidimensionální automat libovolného druhu. Ten by sloužil například k analýze vývoje či predikci příštích hodnot. Vzhledem k možnosti navigace po multidimensionální buňce může automat porovnávat hodnoty a měnit stavy na základě vývoje hodnot.

Identifikace obličeje

Ve více vědeckých článcích z pole biometrie se objevuje pojem „*face space*“ (například [4]). Jedná se o multidimensionální prostor s body reprezentujícím tváře lidí. Pozice bodu v prostoru je určena různými metrikami odpovídající tváře, jako je vzdálenost očí od sebe, od kořene nosu, výška úst atd.

Takový multidimensionální prostor, jehož počet dimenzí může být také libovolný (množina metrik může být zvolena různě), může být analyzován pomocí zde uvedených automatů. Využití takového systému pak může být uplatněno v přístupových terminálech a jiných formách biometrického zabezpečení. Nemalé využití mají podobné systémy i v oblasti kriminalistiky a dohledových systémů.

Virtuální realita

Už u třídimensionálních automatů bylo zmíněno využití těchto automatů pro virtuální realitu. Nicméně u tří dimenzí bylo prováděno jen prozkoumání virtuálního prostoru. Pro vyšší počet, konkrétně čtyři dimenze, pak můžeme modelovat virtuální realitu, která je v rámci času dynamická a nabízí lepší simulační dojem.

Pokud takovou realitu doplníme navíc například scénáři chování uživatele, kdy volba uživatele ovlivní vývoj prostředí v budoucnu, a takovýchto voleb může být více, dostáváme multidimensionální systém. Ten logicky může být analyzován a prohledáván multidimensionálními automaty kvůli hledání chyb a konfliktů jednotlivých scénářů.

V této kapitole byl představen koncept multidimensionálních automatů. Byla uvedena omezení a tvary pravidel pro zástupce těchto automatů, zaručující jejich schopnost analyzovat buňku s libovolnou dimensionalitou.

Na základě těchto pravidel byl vytvořen multidimensionální Turingův stroj, tvořící základní automat této kategorie, který disponoval množinou směrů pohybu v libovolné buňce. Od tohoto stroje byly dále odvozeny další automaty. Celkem byly zmíněny a z větší části definovány 4 druhy multidimensionálních automatů. Jedná se o automaty vyvinuté autory z již zmíněných automatů pro pevný počet dimenzí.

Následně bylo uvedeno několik scénářů z praxe, kde by mohly vyvinuté automaty a jim podobné mechanismy najít uplatnění. Jedná se o využití v databázových systémech pro analýzu dat, biometrické systémy a tvorbu virtuální reality, takže o aktuálně populární témata.

Kapitola 7

Simulátor multidimensionálních automatů

Náplní této kapitoly je představení vyvinuté aplikace, která slouží jako demonstrační prostředek fungování některých multidimensionálních automatů, uvedených v předchozí kapitole.

Jedná se o jednoduchou aplikaci s grafickým uživatelským rozhraním, která může být využita pro testování vyvinutých multidimensionálních automatů či prověření možnosti jejich využití ve vybrané oblasti. Aplikace umožňuje tyto testované automaty ukládat jako soubory a opětovně je načítat pro snadnou editaci.

Vstupem demonstrováných automatů jsou pak buňky, tak, jak byly představeny v 5.1, uložené v dále popsaném formátu jako textové soubory. Takové buňky mohou mít různou dimensionalitu, jak je uvedené v jejich formální definici.

Implementované automaty tedy opravdu pracují nad buňkami s libovolnou dimensionalitou a jejich postup je uživateli jednoduše demonstrován přes grafické uživatelské rozhraní. Jakákoliv manipulace a události v aplikaci jsou zároveň přehledně zapisovány na standardní výstup.

7.1 O aplikaci

Tato aplikace byla implementována v programovacím jazyku *Java* verze 8, pro grafické prvky aplikace pak byla použita populární knihovna *swing*. Pro vývoj bylo zvoleno prostředí *Eclipse* (<http://www.eclipse.org/>).

Aplikace byla vyvíjena s důrazem na jednoduchost používání bez komplikovaných prvků či nepotřebných rozšíření, proto hlavní okno aplikace umožňuje zároveň volbu a editaci automatu a výstup simulace chodu definovaného automatu.

Veškeré třídy a důležité funkce projektu byly dokumentovány pomocí *Javadoc* komentářů, což umožňuje generování přehledné dokumentace pro celý projekt. Při implementaci byly použity pouze volně dostupné knihovny.

7.1.1 Struktura projektu

Součástí projektu jsou jak třídy implementující chování simulovaných automatů, tak i třídy implementující dále představené části rozhraní aplikace. V této sekci jsou tyto třídy stručně představeny.

Třídy reprezentující části uživatelského rozhraní a implementující dodatečné funkce pro toto rozhraní jsou následující: *MainWindow*, *AddRuleWindow*, *CustomTimer*, *ErrorManager*, *StateEditationWindow* a *AddStateWindow*.

Třída MainWindow

Tato třída reprezentuje hlavní okno aplikace a ovládá chod celé aplikace. Skrze toto okno je přístupováno k nahrávání a editaci automatu, zvolení analyzované buňky a slouží i k ovládní simulace.

Třída AddRuleWindow

Tato třída implementuje funkci okna pro vytváření nových pravidel zvoleného automatu. Disponuje zároveň kontrolou uživatelských vstupů pro signalizaci chybného zadání části pravidla.

Třída CustomTimer

Pro přehlednější demonstraci postupu automatu jsou kroky simulace prováděny v časových odstupech zvolených uživatelem. Tato třída pak implementuje toto čekání na další krok v samostatném vláknu, aby mohla být neustále prováděna interakce s hlavním oknem.

Třída ErrorManager

Třída je implementována pomocí návrhového vzoru *Singleton*. Jedná se o třídu, pomocí které jsou vytvářeny dialogy hlásící uživateli problémy, jako je například nekorektní vstup ze souboru.

Třídy StateEditationWindow a AddStateWindow

Jedná se o třídy poskytující rozhraní pro přidávání a obecnou editaci množin stavů a koncových stavů daného automatu. Opět disponují kontrolou uživatelských vstupů pro jednoduchou korekci zadaných hodnot.

Třídy reprezentující implementované automaty jsou následující: *Automaton*, *Cell*, *MultiTS* a *MultiwayAutomaton*.

Třída Automaton

Jedná se abstraktní třídu poskytující rozhraní pro manipulaci s jednotlivými druhy automatu stejným způsobem. Obsahuje funkce implementující načítání automatů ze souborů a jejich následné ukládání a deklarace funkcí implementovaných dále v potomcích třídy, které poskytují provádění kroků chodů automatu a manipulaci s rozdílnými tvary pravidel automatu.

Třída Cell

Tato třída představuje v aplikaci načtenou buňku. Získání obsahu buňky probíhá při načtení zvoleného souboru. Tato třída pak provádí kontrolu takového vstupu a při simulaci poskytuje fungujícímu automatu symboly na zadaných pozicích a signalizuje, zda zadané pohyby automatu nevedou mimo tělo buňky.

Třída MultiTS

Tato třída je potomkem třídy *Automaton* a představuje multidimensionální Turingův stroj. Implementuje tak abstraktní funkce zmíněné výše spolu se speciálními pohyby v ukládací pásce.

Třída MultiwayAutomaton

Tato třída je potomkem třídy *Automaton* a představuje vícesměrný multidimensionální automat. Opět implementuje zděděné abstraktní funkce a provádí pohyby pouze po zadané buňce.

7.1.2 Metriky projektu

Obecné metriky pro zdrojové kódy aplikace jsou zobrazeny v tabulce 7.1.

programovací jazyk	Java 8
počet souborů	10
počet funkcí	145
velikost projektu	99,4 kB

Tabulka 7.1: Informace o projektu

7.2 Rozhraní aplikace

V této sekci jsou popsány jednotlivé části uživatelského rozhraní společně s jejich ukázkami. V předchozí sekci bylo uvedeno, že kvůli zamýšlené jednoduchosti aplikace jsou všechny důležité prvky zobrazeny v rámci hlavního okna aplikace.

Mezi takové prvky patří: výběr druhu automatu, množiny stavů a pravidel definujících automat, panel pro nahrání analyzované buňky, tlačítka pro ovládání simulace a výstupy představující aktuální konfiguraci a historii předchozích konfigurací.

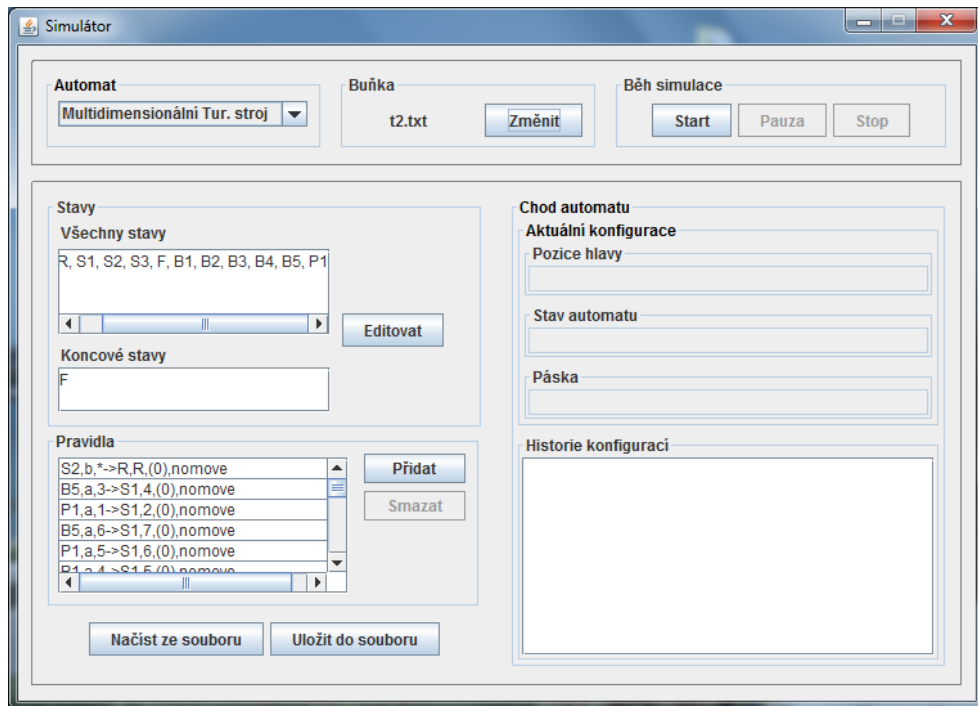
Před editací i simulací musí být zvolen typ automatu, bez této akce jsou veškeré ovládací prvky okna neaktivní, aby nebylo manipulováno s neexistující entitou, která by byla definována a simulována. Po zvolení automatu následuje editace automatu, načtení souboru s buňkou a spuštění simulace.

Ukázka hlavního okna aplikace při editaci automatu je pak na obrázku 7.1.

7.2.1 Editace automatu

Po zvolení automatu je možné tento automat definovat prostřednictvím hlavního okna a dalších pomocných oken. Jednou z možností je načtení automatu ze souboru pomocí příslušných tlačítek v hlavní obrazovce. Platné formáty textových souborů obsahujících definice automatů jsou uvedeny dále v sekci 7.3 u každého typu automatu.

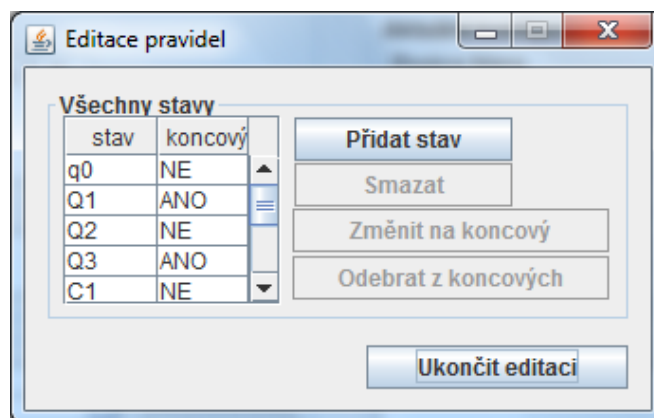
Jakákoliv manipulace s množinami stavů a koncových stavů automatu je prováděna přes okno reprezentované již představenou třídou *StateEditationWindow*. Okno je otevřeno příslušným tlačítkem hlavního okna. Ukázka okna je na obrázku 7.2. Je třeba poznamenat, že z definice 52 vyplývá, že u vícesměrného automatu nebude prováděna manipulace s koncovými stavy.



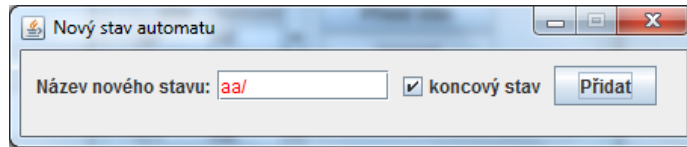
Obrázek 7.1: Hlavní obrazovka aplikace

Jak lze vidět na obrázku 7.2, je mimo jiné umožněno i mazání stavů, tato možnost je ale dostupná pouze v případech, kdy stav není použit v žádném pravidlu či nepatří mezi stavy spadající do definice automatu (například počáteční stav q_0).

Pro přidání nového stavu pak slouží okno reprezentované třídou *AddStateWindow*, zobrazené na obrázku 7.3. Uživatelem zadaný název stavu je při potvrzení prověřen a při nesprávném zadání je text označen červeně, aby byla umožněna jeho úprava. Možnost vytvořit koncový stav je dostupná pouze při editaci multidimensionálního Turingova stroje.



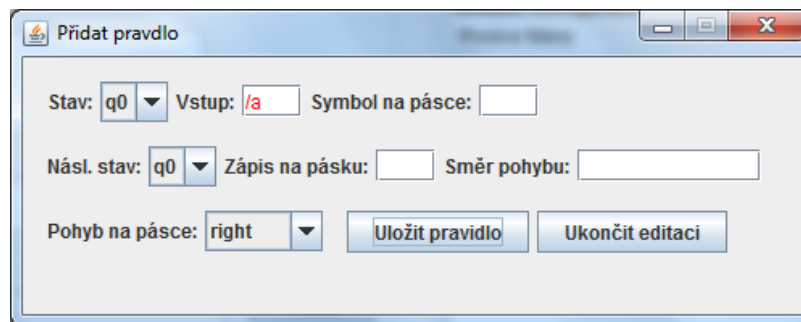
Obrázek 7.2: Okno pro editaci množin stavů



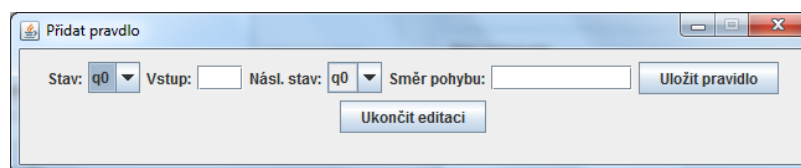
Obrázek 7.3: Okno přidání stavu

Pro zobrazení pravidel slouží v hlavním okně tabulka pravidel, nacházející se vlevo dole na obrázku 7.1. Při obsáhlé množině pravidel či jejich dlouhých tvarech jsou pro tuto tabulku zobrazeny posuvníky, aby nedocházelo k narušení vzhledu okna. Přímo hlavní obrazovka pak umožňuje mazání pravidel. Při zvolení pravidla v tabulce je možné toto pravidlo smazat pomocí příslušného tlačítka, které je zvolením pravidla aktivováno.

Pro přidání nového pravidla slouží okno reprezentované třídou *AddRuleWindow*. Vzhled okna se samozřejmě liší podle typu upravovaného automatu, jelikož pravidla jednotlivých typů automatů nabývají velmi odlišných tvarů. Ukázka tohoto okna při editaci pravidel pro multidimensionální Turingův stroj je na obrázku 7.4 a pro vícesměrný automat na obrázku 7.5.



Obrázek 7.4: Editace pravidel multidimensionálního TS



Obrázek 7.5: Editace pravidel vícesměrného automatu

Na obrázcích lze vidět, že výběr současného a následujícího stavu probíhá pomocí nabídky, stavy tedy musí být již vytvořeny pomocí dříve zmíněného postupu. Ostatní textové vstupy jsou při žádosti o uložení pravidla opět kontrolovány a nevyhovující tvary jsou označeny červeně pro jejich možnou úpravu, jak je zobrazeno na obrázku 7.4.

Při používání výše demonstrovaných editačních oken je interakce s hlavním oknem znemožněna, aby nedošlo k nekonzistentním situacím. Aktivace hlavního okna probíhá ihned po uzavření těchto oken. Automaty vytvořené pomocí výše zmíněných postupů pak lze ulo-

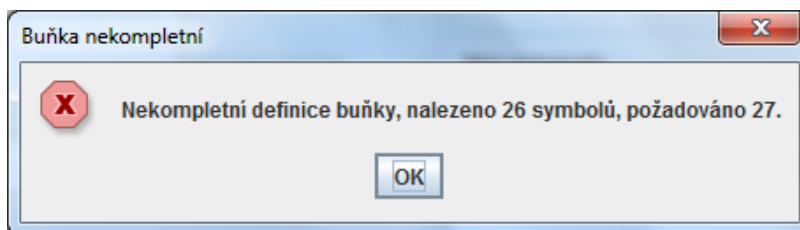
žit do vybraného souboru pomocí daného tlačítka na hlavní obrazovce. Tyto automaty pak samozřejmě lze pomocí druhého tlačítka opět načíst a použít.

7.2.2 Ovládání simulace

Po ukončení editace automatu je možné zahájit simulaci. Ovládání simulace je umožněno pomocí tlačítek hlavního okna v pravém horním rohu na obrázku 7.1. Spuštění simulace je ovšem podmíněno zvolením buňky, na níž bude chod automatu demonstrován.

Tlačítko a případný název souboru obsahujícího zvolenou buňku se nacházejí v prostřední části horní lišty hlavního okna. Načtení buňky probíhá přes klasický dialog zvolení souboru, zprostředkovaný standardní komponentou *JFileChooser*. Soubor s buňkou obsahuje definici jejích jednotlivých symbolů ve formátu: `<index>,<index>,... :«symbol»`. Tyto definice symbolů jsou odděleny středníky, bílé znaky v souboru nemají žádný význam. Symbol pak je písmeno či číslice.

Při načtení buňky se může vyskytnout několik problémů. Buňka musí být v souboru definovaná kompletně, nelze tedy, aby některé její symboly byly nedefinované. Dále musí být dodržena jednotná informace o dimensionalitě jednotlivých symbolů. Pokud je tedy první symbol definován se sedmi indexy, jedná se o buňku v sedmi dimenzích a každý další symbol musí být definován stejně. Problémy s načtením buňky jsou pak ohlašovány skrz dialogy vytvářené třídou *ErrorManager*, ukázka je zobrazena na obrázku 7.6.



Obrázek 7.6: Chybové hlášení

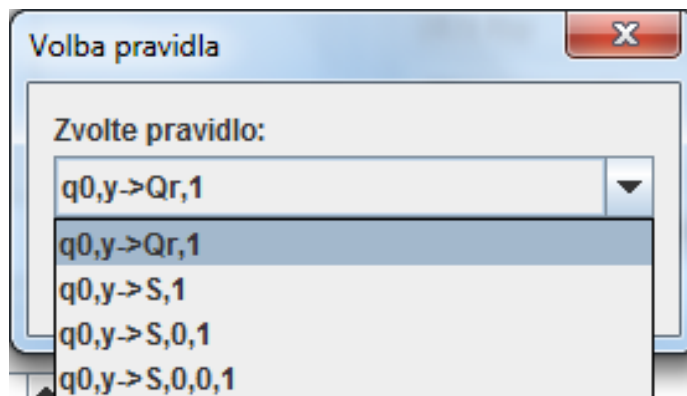
Po načtení buňky může být spuštěna simulace. Při nové simulaci je uživatel vyzván, aby zadal rychlost, s jakou budou prováděny kroky simulace. Tyto časové rozestupy slouží k přehlednější demonstraci chodu automatu. Simulace může být za chodu pozastavena nebo úplně zrušena.

Během kroků simulace se zobrazuje aktuální konfigurace automatu v textových polích v pravé části hlavní obrazovky. Zde také dochází k jediné změně vzhledu okna při změně druhu automatu, jelikož součástí konfigurace Turingova stroje je na rozdíl od multidimenzionálního automatu i stav ukládací pásky.

Spolu s aktuální konfigurací je vypisována i historie předchozích konfigurací ve spodním textovém okně pro možnost kontroly postupu automatu.

Jelikož je žádoucí demonstrovat především nedeterministické automaty, je nutné řešit volbu pravidla pro další krok při výskytu více použitelných pravidel. Tato volba je ponechána na uživateli pomocí okna, zobrazeného na obrázku 7.7.

Ukončení simulace je možné stiskem daného tlačítka, simulace ovšem skončí automaticky při absenci pravidel pro danou konfiguraci automatu či v případě vícesměrného automatu při dosažení koncového stavu. Pro Turingův stroj se uživateli při dosažení koncového stavu vyskytne volba pokračovat dále v simulaci. Simulace Turingova stroje může ovšem



Obrázek 7.7: Výběr pravidla

také skončit přepadnutím hlavy ukládací pásky. Tyto informace jsou uživateli opět podány prostřednictvím dialogů.

Zde popsané postupy lze vyzkoušet pomocí přiložených demonstračních příkladů. Tyto příklady jsou popsány v příloze B. Jedná se celkem o šest příkladů demonstrujících postup obou automatů, jsou dále editovatelné a existuje pro ně přesný popis postupu.

7.3 Implementované automaty

Součástí aplikace jsou třídy, zastupující dále zmíněné dva typy automatů. Aplikace je ovšem navržena za tím účelem, aby bylo přidání dalších typů automatu co nejsnadnější díky jednotnému rozhraní, představovaném abstraktní třídou *Automaton*.

Ve výsledné aplikaci jsou tedy implementovány automaty z definic 52 a 51 s určitými obměnami, které jsou uvedeny u jednotlivých automatů. Předně není přidána podpora pro implicitní doplnění hraničních symbolů buňky. Tato změna je provedena z toho důvodu, že uživatel může případné znační hranice dodat pomocí zvoleného symbolu a volba pravidel (jak je zanesena na obrázku 7.7) nepřináší zbytečné nejasnosti a pohyby, které vedou ke zbytečnému zaseknutí automatu. Jakákoliv omezení touto změnou pak můžou být odstraněna doplněním množiny pravidel nedeterministického automatu.

Pro ukládání automatů jako souborů, které bylo popisováno v sekci 7.2.1, je používán standardní tvar, který je následující (symboly <> jsou zde použity jako vymezení znaky a v souborech nejsou skutečně obsaženy):

```
<stav automatu>,<stav automatu>,...
<koncový stav automatu>,<koncový stav automatu>,...
<pravidlo 1>
<pravidlo 2>
```

Tabulka 7.2: Formát souboru definujícího automat

V pravidlech, povinně oddělených novým řádkem, mohou být pro vstupy použity zástupné symboly * pro zjednodušení definic pravidel, konkrétně například pro přechody do nepřijímajících stavů.

Podmínkou pro správnou činnost automatu je obsažení stavu q_0 mezi stavy automatu, jelikož se dle obou zmíněných definic jedná o počáteční stav automatů a při začátku simulace je oběma automatům tento stav přiřazen.

7.3.1 Multidimensionální Turingův stroj

Tento typ automatu je implementován pomocí třídy *MultiTS*. Ta splňuje všechny body definice 51 a umožňuje editaci obou množin stavů.

Dále lze libovolně zapisovat na pásku, kde byl oproti definici zvolen jako prázdný znak symbol #. Tato změna přináší lepší čitelnost stavu pásky v grafickém rozhraní aplikace. Prázdný znak pásky lze navíc i zapisovat, což je nepodstatná úprava, která je opět provedena z hlediska přehlednosti demonstrace chodu automatu.

Editace pravidel byla již také demonstrována obrázkem 7.4. Lze provádět libovolné pohyby na pásce a aplikace podporuje scénář nekonvenčního zastavení automatu kvůli přepadnutí hlavy přes levý okraj pásky. Jako konfigurace pásky je vždy uvedeno jen několik symbolů pásky okolo hlavy stroje s vyznačeným symbolem pod hlavou (pomocí symbolů $\langle \rangle$).

Pravidla automatu se skládají z levé strany obsahující: aktuální stav, symbol na vstupu a symbol na pásce a pravé strany obsahující: nový stav automatu, zápis na pásku, pohyb v buňce, pohyb na ukládací pásce. Formát pravidla v souboru je pak následující:

```
<stav automatu>,<symbol na vstupu>,<symbol na pásce>-><nový stav  
automatu>,<zápis na pásku>,<vektor pohybu>,<pohyb na pásce>
```

7.3.2 Vícesměrný multidimensionální automat

Automat je implementován třídou *MutiwayAutomaton*. Při tvorbě byla dodržena definice 52. Na rozdíl od multidimensionálního Turingova stroje je znemožněna manipulace s koncovými stavy v okně na obrázku 7.2. Ty jsou totiž dle dané definice fixní a obsahují pouze dva stavy, a to *akceptující* stav Q_a a *odmítající* stav Q_r .

Kvůli speciální množině koncových stavů je pak i při načítání automatu ze souboru ověřeno, zda jsou obsaženy tyto stavy a jsou jedinými koncovými stavy automatu. Jelikož automat nedisponuje páskou, není ani při simulaci tato část konfigurace zobrazena. Při přechodu automatu do koncového stavu je výjimečně povolen pohyb mimo symboly buňky z důvodu zjednodušení definice pravidel.

Pravidla automatu se skládají z levé strany obsahující: aktuální stav a symbol na vstupu a pravé strany obsahující: nový stav automatu a pohyb v buňce. Formát pravidla v souboru je pak následující:

```
<stav automatu>,<symbol na vstupu>-><nový stav automatu>,<vektor pohybu>
```

V této kapitole byla představena a demonstrována aplikace vyvinutá v rámci této práce jako praktická část. Jednalo se o simulátor některých multidimensionálních automatů představených v předchozí kapitole.

Aplikace byla vyvinuta v programovacím jazyku Java a pomocí uživatelského grafického prostředí umožňuje snadnou definici a demonstraci multidimensionálních automatů, konkrétně multidimensionálního Turingova stroje a vícesměrného multidimensionálního automatu.

V kapitole byly uvedeny souhrnné informace o aplikaci včetně popisu hlavních 9 tříd podílejících se na její funkčnosti. Jedná se o třídy reprezentující jak implementované automaty, tak jejich podpůrné mechanismy a grafické prostředí.

Byly popsány jednotlivé komponenty grafického prostředí, jako je hlavní obrazovka a dále editační okna pro definici automatů, spolu s postupem, jak pomocí těchto prvků vytvořit daný automat. Součástí aplikace jsou i uvedené mechanismy zamezující načtení nesprávně formulovaných vstupů.

Dále v kapitole následuje popis ovládání simulace chodu definovaného či načteného automatu nad načtenou multidimensionální buňkou. Dodatečně jsou pak popsány automaty implementované v rámci aplikace. Jsou uvedeny změny, které bylo nutno provést pro funkčnost aplikace, a formát, ve kterém jsou tyto automaty reprezentovány v souborech, které lze pomocí aplikace vytvářet a načítat.

Výsledná aplikace funguje jako jednoduchý a názorný simulátor zmíněných multidimensionálních automatů a poskytuje tak možnost lepšího pochopení jejich mechanismů. To může vést k využití těchto automatů v praxi, kde by tyto automaty mohly nalézt široké uplatnění, jak již bylo dokázáno v předchozí kapitole.

Kapitola 8

Závěr

V této práci byly představeni zástupci vícedimensionálních jazyků a automatů přijímajících tyto jazyky. Jedná se o jazyky dvoudimensionální, třídimensionální, a především jazyky multidimensionální bez konkrétního údaje o počtu dimenzí. Tyto jazyky byly představeny jako množiny nadřazené všem jazykům s konkrétní dimensionalitou, jelikož jsou jejich nadmnožinou.

Na začátku práce byla analyzována oblast dvoudimensionálních jazyků. Touto teorií se již zabývalo několik vědeckých prací, včetně předchozí práce autorů, a tvoří základ pro definování dalších zde představených druhů jazyků. Základním prvkem těchto jazyků je pak dvoudimensionální matice symbolů neboli obraz. Tyto obrazy byly definovány spolu s novými druhy operací nad těmito obrazy. V práci byl později učiněn poznatek, že nové druhy operací se objevují při každém zvýšení počtu dimenzí.

Po představení těchto obrazů a operací byly definovány jejich jazyky a uzávěry těchto jazyků. Dále byli pro případnou definici těchto jazyků uvedeni zajímaví zástupci dvoudimensionálních gramatik. Pro dvoudimensionální jazyky existuje zde také představený koncept regulárních výrazů.

V následující kapitole byli pro tyto jazyky vybráni zástupci automatů, které dovolují analýzu obrazů. Jednalo se celkem o pět zástupců automatů. Bylo poukázáno na možnou podobnost s celulárními automaty a mezi zástupci byl uveden i druh automatu, nasazený prakticky v rámci předchozí práce. Pro podporu tvrzení, že tyto automaty mohou být efektivně využity v praxi, bylo uvedeno několik příkladů takových využití.

V další kapitole byly dříve definované pojmy přepracovány pro definici nového konceptu třídimensionálních jazyků. Byl definován pojem těleso, což je zástupce řetězce pro tři dimenze. Dále byly opět definovány nové typy operací nad těmito tělesy. Bylo poznamenáno, že koncept těchto třídimensionálních operací je prakticky používán v modelovacích programech, i když tak nejsou nazývány. Poté byly definovány samotné třídimensionální jazyky obsahující tyto tělesa.

Pro tyto jazyky těles byly definovány gramatiky utvořené na základu již definovaných dvoudimensionálních gramatik. Dále byly uvedeny některé příklady tzv. *spacial grammars*, které nepřímo souvisejí s těmito gramatikami. Koncept regulárních výrazů pro tuto dimensionalitu nebyl definován. Naopak byli definováni tři zástupci třídimensionálních automatů, umožňujících analýzu těles. Pro tento druh automatů bylo opět několik oblastí, v nichž by mohly tyto automaty nalézt praktické uplatnění.

Na základě pojmů definovaných v předchozích kapitolách byl nakonec představen termín multidimensionálních jazyků. Obdobou řetězce pro takové jazyky je pak buňka, definovaná jako řetězec s libovolnou dimensionalitou. To znamená, že jakýkoliv řetězec, obraz či těleso

je ve své podstatě zároveň buňkou, a to platí i pro řetězce s vyšší dimensionalitou. Pro tyto obecně n -dimensionální řetězce byly definovány operace, které dle práce nabývají nekonečně mnoho obdob vzhledem k nekonečné množině možných dimensionalit.

Multidimensionální jazyky pak jsou množinou buněk, což znamená, že mohou obsahovat řetězce, obrazy i jakékoliv jiné entity zastupující řetězce s libovolnou dimensionalitou. Z toho plyne již představený fakt, že multidimensionální jazyky jsou nadřazeny všem jazykům s konkrétní dimensionalitou. Pro tyto jazyky byly uvedeny dvě multidimensionální gramatiky, uvedené v rámci práce, umožňující generovat buňky o libovolné dimensionalitě.

Pro analýzu takových buněk byli dále uvedeni čtyři zástupci multidimensionálních automatů, schopných se pomocí představených pohybových vektorů pohybovat po buňkách v jakékoliv dimenzi. Především byl uveden zástupce Turingových strojů pro tento n -dimensionální prostor a základní zástupce automatů, jímž je vícesměrný multidimensionální automat. Opět bylo pro představené automaty uvedeno několik praktických užití.

Dva zmínění zástupci multidimensionálních automatů pak byli implementováni pomocí aplikace vyvinuté v rámci práce a popsané v poslední kapitole. Jedná se o jednoduchý simulátor demonstrující funkci těchto automatů na multidimensionálních buňkách. V práci je představeno rozhraní této aplikace spolu s dalšími funkčními komponentami. Aplikace umožňuje definici zvolených automatů podle uživatelských požadavků a simulaci jejich chodu na zadané buňce. Definice a výstupy simulace jsou uživateli zpřístupněny přes grafické rozhraní, umožňující jednoduchou práci a lepší pochopení principů automatů.

Teorie multidimensionálních automatů a jejich jazyků byla stručně prezentována v článku na konferenci *EXCEL@FIT 2017*. Respondenty byla přijata jako zajímavá část teoretické informatiky zasluhující další rozšíření. Tato rozšíření a další práce na pojmech z multidimensionální oblasti formálních jazyků budou součástí budoucích vědeckých publikací. Současně může být pro tyto jazyky vyvinuta jistá obdoba Chomského hierarchie, která by sjednotila poznatky o závislostech tříd jazyků s libovolnou dimensionalitou.

Literatura

- [1] Blum, M.; Hewitt, C.: Automata on a 2-dimensional tape. In *Switching and Automata Theory, 1967. SWAT 1967. IEEE Conference Record of the Eighth Annual Symposium on*, IEEE Publishing, 1967, s. 155–160.
URL <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5397208>
- [2] Čadík, M.; Vašíček, J.; Hradiš, M.; aj.: Camera Elevation Estimation from a Single Mountain Landscape Photograph. In *British Machine Vision Conference 2015*, The British Machine Vision Association and Society for Pattern Recognition, 2015, ISBN 1-901725-53-7, s. 1–12.
URL http://www.fit.vutbr.cz/research/view_pub.php?id=10930
- [3] Capobianco, S.: Multidimensional cellular automata and generalization of Fekete’s lemma. *Discrete Mathematics and Theoretical Computer Science (DMTCS)*, ročník 10, č. 3, 2008: s. 95 – 104, ISSN 13658050.
URL <http://search.ebscohost.com.ezproxy.lib.vutbr.cz/login.aspx?direct=true&db=a9h&AN=41020759&lang=cs&site=ehost-live>
- [4] Catz, O.; Kampf, M.; Nachson, I.; aj.: From theory to implementation: Building a multidimensional space for face recognition. *Acta Psychologica*, ročník 131, č. 2, 2009: s. 143 – 152, ISSN 0001-6918, doi:<http://dx.doi.org/10.1016/j.actpsy.2009.03.010>.
URL <http://www.sciencedirect.com/science/article/pii/S0001691809000419>
- [5] Chen, S.; Guillemot, G.; Gandin, C.-A.: Three-dimensional cellular automaton-finite element modeling of solidification grain structures for arc-welding processes. *Acta Materialia*, ročník 115, 2016: s. 448 – 467, ISSN 1359-6454, doi:<http://dx.doi.org/10.1016/j.actamat.2016.05.011>.
URL <http://www.sciencedirect.com/science/article/pii/S1359645416303391>
- [6] Dennunzio, A.; Formenti, E.; Weiss, M.: Multidimensional cellular automata: closing property, quasi-expansivity, and (un)decidability issues. *Theoretical Computer Science*, ročník 516, 2014: s. 40 – 59, ISSN 0304-3975, doi:<http://dx.doi.org/10.1016/j.tcs.2013.11.005>.
URL <http://www.sciencedirect.com/science/article/pii/S0304397513008232>
- [7] Dong, J.; Jin, W.: Comparison of Two-Way Two-Dimensional Finite Automata and Three-Way Two-Dimensional Finite Automata. In *Computer Science & Service System (CSSS), 2012 International Conference on*, IEEE, 2012, ISBN 9781467307215, s. 1904–1906.
URL <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6394793>

- [8] Giammarresi, D.: Two-Dimensional Languages and Recognizable Functions. In *Developments in Language Theory*, 1993, s. 290–301.
- [9] Giammarresi, D.; Restivo, A.: *Two-Dimensional Languages*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, ISBN 978-3-642-59126-6, s. 215–267, doi:10.1007/978-3-642-59126-6_4.
URL http://dx.doi.org/10.1007/978-3-642-59126-6_4
- [10] Hirakawa, H.; Inoue, K.; Ito, A.: Three-way two-dimensional alternating finite automata with rotated inputs. *Information Sciences*, ročník 176, č. 11, 2006: s. 1546 – 1569, ISSN 0020-0255, doi:http://dx.doi.org/10.1016/j.ins.2005.05.006.
URL <http://www.sciencedirect.com/science/article/pii/S0020025505001805>
- [11] Hladík, Z.: Dvoudimensionální konečné automaty a jejich aplikace. 2014.
- [12] Hoisl, F.; Shea, K.: An interactive, visual approach to developing and applying parametric three-dimensional spatial grammars. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing : AI EDAM*, ročník 25, č. 4, 11 2011: s. 333–356.
URL <http://search.proquest.com/docview/897572438?accountid=17115>
- [13] Hyun, K.; Lee, K.; Lee, J.: Motion Grammars for Character Animation. *Computer Graphics Forum*, ročník 35, č. 2, 2016: s. 103–113, ISSN 1467-8659, doi:10.1111/cgf.12815.
URL <http://dx.doi.org/10.1111/cgf.12815>
- [14] Imai, K.; Matsuda, Y.; Iwamoto, C.; aj.: A Three-Dimensional Uniquely Parsable Array Grammar that Generates and Parses Cubes. *Electronic Notes in Theoretical Computer Science*, ročník 46, 2001: s. 339 – 354, ISSN 1571-0661, doi:http://dx.doi.org/10.1016/S1571-0661(04)80996-4.
URL <http://www.sciencedirect.com/science/article/pii/S1571066104809964>
- [15] Inoue, K.; Nakamura, A.: Some properties of two-dimensional on-line tessellation acceptors. *Information Sciences*, ročník 13, č. 2, 1977: s. 95 – 121, ISSN 0020-0255, doi:http://dx.doi.org/10.1016/0020-0255(77)90023-8.
URL <http://www.sciencedirect.com/science/article/pii/0020025577900238>
- [16] Inoue, K.; Nakamura, A.: Two-dimensional multipass on-line tessellation acceptors. *Information and Control*, ročník vol. 41, č. issue 3, 1979: s. 305–323.
URL <http://linkinghub.elsevier.com/retrieve/pii/S0019995879906041>
- [17] Inoue, K.; Takanami, I.: *A survey of two-dimensional automata theory*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989, ISBN 978-3-540-48203-1, s. 72–91, doi:10.1007/BFb0015929.
URL <http://dx.doi.org/10.1007/BFb0015929>
- [18] Jančar, P.; Mráz, F.; Plátek, M.: Forgetting automata and context-free languages. *Acta Informatica*, ročník 33, č. 5, 1996: s. 409–420, ISSN 1432-0525, doi:10.1007/s002360050050.
URL <http://dx.doi.org/10.1007/s002360050050>
- [19] Jednoróg, K.; Łukasz Bola; Mostowski, P.; aj.: Three-dimensional grammar in the brain: Dissociating the neural correlates of natural sign language and manually coded

- spoken language. *Neuropsychologia*, ročník 71, 2015: s. 191 – 200, ISSN 0028-3932, doi:<http://dx.doi.org/10.1016/j.neuropsychologia.2015.03.031>.
URL <http://www.sciencedirect.com/science/article/pii/S0028393215001463>
- [20] Jiříčka, P.; Král, J.: Deterministic forgetting planar automata are more powerful than nondeterministic finite-state planar automata. *Developments in Language Theory: Foundations, Applications, and Perspectives: Aachen, Germany, 6-9 July 1999*, 2000: str. 71.
- [21] Křivka, Z.; Martín-Vide, C.; Meduna, A.; aj.: *A Variant of Pure Two-Dimensional Context-Free Grammars Generating Picture Languages*. Cham: Springer International Publishing, 2014, ISBN 978-3-319-07148-0, s. 123–133, doi:10.1007/978-3-319-07148-0_12.
URL http://dx.doi.org/10.1007/978-3-319-07148-0_12
- [22] Lei, Y.; Chi, Z.; Jun, H.; aj.: A three-dimensional cellular automata evacuation model with dynamic variation of the exit width. *Journal of Applied Physics*, ročník 115, č. 22, 2014: s. 224905–1 – 224905–10, ISSN 00218979.
URL <http://search.ebscohost.com.ezproxy.lib.vutbr.cz/login.aspx?direct=true&db=a9h&AN=96578629&lang=cs&site=ehost-live>
- [23] Loui, M. C.: Simulations among multidimensional turing machines. *Theoretical Computer Science*, ročník 21, č. 2, 1982: s. 145 – 161, ISSN 0304-3975, doi:[http://dx.doi.org/10.1016/0304-3975\(89\)90081-9](http://dx.doi.org/10.1016/0304-3975(89)90081-9).
URL <http://www.sciencedirect.com/science/article/pii/0304397589900819>
- [24] Makino, T.; Okabe, H.; Taniguchi, S.; aj.: Three-dimensional multiinkdot automata. *Artificial Life and Robotics*, ročník 9, č. 2, 2005: s. 99–101, ISSN 1614-7456, doi:10.1007/s10015-004-0327-3.
URL <http://dx.doi.org/10.1007/s10015-004-0327-3>
- [25] Marvie, J.-E.; Buron, C.; Gautron, P.; aj.: GPU Shape Grammars. *Computer Graphics Forum*, ročník 31, č. 7, 2012: s. 2087–2095, ISSN 1467-8659, doi:10.1111/j.1467-8659.2012.03201.x.
URL <http://dx.doi.org/10.1111/j.1467-8659.2012.03201.x>
- [26] Matz, O.: *Regular expressions and context-free grammars for picture languages*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, ISBN 978-3-540-68342-1, s. 283–294, doi:10.1007/BFb0023466.
URL <http://dx.doi.org/10.1007/BFb0023466>
- [27] Maurer, H.; Salomaa, A.; Wood, D.: Pure grammars. *Information and Control*, ročník 44, č. 1, 1980: s. 47 – 72, ISSN 0019-9958, doi:[http://dx.doi.org/10.1016/S0019-9958\(80\)90131-X](http://dx.doi.org/10.1016/S0019-9958(80)90131-X).
URL <http://www.sciencedirect.com/science/article/pii/S001999588090131X>
- [28] Nivat, M.; Saoudi, A.; Dare, V. R.: Parallel generation of finite images. *International Journal of Pattern Recognition and Artificial Intelligence*, ročník 03, č. 03n04, 1989: s. 279–294, doi:10.1142/S0218001489000243.
URL <http://www.worldscientific.com/doi/abs/10.1142/S0218001489000243>

- [29] Pietro, R. D.; Mancini, L. V.: *Intrusion detection systems*. New York: Springer, první vydání, c2008, ISBN 978-0-387-77265-3.
- [30] Ponniah, P.: *Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals*. Wiley, 2001, ISBN 9780471412540.
URL <https://books.google.cz/books?id=3dFQAAAAMAAJ>
- [31] Průša, D.; Hlaváč, V.: Mathematical formulae recognition using 2d grammars. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, ročník 2, IEEE, 2007, s. 849–853.
- [32] Průša, D.; Mráz, F.: Two-dimensional Sgraffito Automata. In *Proceedings of the 16th International Conference on Developments in Language Theory, DLT'12*, Berlin, Heidelberg: Springer-Verlag, 2012, ISBN 978-3-642-31652-4, s. 251–262, doi:10.1007/978-3-642-31653-1_23.
URL http://dx.doi.org/10.1007/978-3-642-31653-1_23
- [33] Průša, D.: Two-dimensional context-free grammars. *ITAT*, ročník 2001, 2001: s. 27–40.
URL <ftp://cmp.felk.cvut.cz/pub/cmp/articles/prusa/Prusa-ITAT-2001.pdf>
- [34] Průša, D.; Mráz, F.; Otto, F.: Comparing two-dimensional one-marker automata to sgraffito automata. In *International Conference on Implementation and Application of Automata*, Springer Berlin Heidelberg, 2013, s. 268–279.
URL http://link.springer.com/chapter/10.1007/978-3-642-39274-0_24
- [35] Průša, D.; Mráz, F.; Otto, F.: *New Results on Deterministic Sgraffito Automata*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, ISBN 978-3-642-38771-5, s. 409–419, doi:10.1007/978-3-642-38771-5_36.
URL http://dx.doi.org/10.1007/978-3-642-38771-5_36
- [36] Rosenfeld, A.: Some notes on finite-state picture languages. *Information and Control*, ročník 31, č. 2, 1976: s. 177 – 184, ISSN 0019-9958, doi:http://dx.doi.org/10.1016/S0019-9958(76)80006-X.
URL <http://www.sciencedirect.com/science/article/pii/S001999587680006X>
- [37] Rosenfeld, A.; Rheinboldt, W.: *Picture Languages: Formal Models for Picture Recognition*. Elsevier Science, 2014, ISBN 9781483268552.
URL https://books.google.cz/books?id=CL_iBQAAQBAJ
- [38] Rosenfeld, A.; Strong, J. P.: A Grammar for Maps. In *Computer and Information Sciences–1969, {SEN} Report Series Software Engineering*, ročník 2, editace J. T. TOU, Elsevier, 1971, s. 227 – 239, doi:http://dx.doi.org/10.1016/B978-0-12-696202-4.50020-6.
URL <http://www.sciencedirect.com/science/article/pii/B9780126962024500206>
- [39] Sakamoto, M.; Fukuda, M.; Okatani, S.; aj.: Path-bounded three-dimensional finite automata. *Artificial Life and Robotics*, ročník 13, č. 1, 2008: s. 54–57, ISSN 1614-7456, doi:10.1007/s10015-008-0521-9.
URL <http://dx.doi.org/10.1007/s10015-008-0521-9>

- [40] Sakamoto, M.; Ito, T.; Furutani, H.; aj.: Remarks on the recognizability of topological components by three-dimensional automata. *Artificial Life and Robotics*, ročník 13, č. 1, 2008: s. 368–372, ISSN 1614-7456, doi:10.1007/s10015-008-0501-0. URL <http://dx.doi.org/10.1007/s10015-008-0501-0>
- [41] Sakamoto, M.; Tomozoe, N.; Furutani, H.; aj.: A survey of automata on three-dimensional input tapes. *science*, ročník 1, 2008: str. 14.
- [42] Siromoney, G.; Siromoney, R.; Krithivasan, K.: Picture languages with array rewriting rules. *Information and Control*, ročník 22, č. 5, 1973: s. 447 – 470, ISSN 0019-9958, doi:http://dx.doi.org/10.1016/S0019-9958(73)90573-1. URL <http://www.sciencedirect.com/science/article/pii/S0019995873905731>
- [43] Siromoney, R.: On equal matrix languages. *Information and Control*, ročník 14, č. 2, 1969: s. 135–151, ISSN 00199958, doi:10.1016/S0019-9958(69)90050-3. URL <http://www.sciencedirect.com/science/article/pii/S0019995869900503>
- [44] Siromoney, R.; Krithivasan, K.: Parallel context-free languages. *Information and Control*, ročník 24, č. 2, 1974: s. 155 – 162, ISSN 0019-9958, doi:http://dx.doi.org/10.1016/S0019-9958(74)80054-9. URL <http://www.sciencedirect.com/science/article/pii/S0019995874800549>
- [45] Subramanian, K. G.; Nagar, A. K.; Geethalakshmi, M.: *Pure 2D Picture Grammars (P2DPG) and P2DPG with Regular Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, ISBN 978-3-540-78275-9, s. 330–341, doi:10.1007/978-3-540-78275-9_29. URL http://dx.doi.org/10.1007/978-3-540-78275-9_29
- [46] Thomsen, E.: *OLAP Solutions: Building Multidimensional Information Systems*. Wiley Computer Publishing, Wiley, 2002, ISBN 9780471400301. URL <https://books.google.cz/books?id=vZIIYeVH0cwC>
- [47] Uchida, Y.; Sakamoto, M.; Okatani, S.; aj.: Hierarchies based on the number of cooperating systems of three-dimensional finite automata. *Artificial Life and Robotics*, ročník 14, č. 3, 2009: str. 425, ISSN 1614-7456, doi:10.1007/s10015-009-0702-1. URL <http://dx.doi.org/10.1007/s10015-009-0702-1>
- [48] Wang, P.: Three-dimensional sequential/parallel universal array grammars for polyhedral object pattern analysis. *International Journal of Pattern Recognition and Artificial Intelligence*, ročník 08, č. 02, 1994: s. 563–576, doi:10.1142/S0218001494000292, <http://www.worldscientific.com/doi/pdf/10.1142/S0218001494000292>.
- [49] Zhang, X.; Zhao, J.; Jiang, H.; aj.: A three-dimensional cellular automaton model for dendritic growth in multi-component alloys. *Acta Materialia*, ročník 60, č. 5, 2012: s. 2249 – 2257, ISSN 1359-6454, doi:http://dx.doi.org/10.1016/j.actamat.2011.12.045. URL <http://www.sciencedirect.com/science/article/pii/S1359645411009074>

Přílohy

Příloha A

Obsah CD

V této příloze jsou popsány soubory obsažené na disku odevzdaném spolu s touto prací. V kořenovém adresáři disku se vyskytují následující složky a soubory:

project složka obsahující soubory příslušející k vyvinuté aplikaci

tex složka obsahující soubory pro znovuvytvoření této práce, především zdrojové kódy ve formátu \LaTeX

thesis.pdf soubor ve formátu PDF obsahující tuto práci

Složka project

Složka obsahuje následující soubory a podsložky související s vyvinutou aplikací:

src složka obsahující zdrojové kódy aplikace v jazyce Java

doc složka obsahující vygenerovanou *Javadoc* dokumentaci ve formátu HTML (soubor `index.html`)

jar složka obsahující spustitelnou formu aplikace ve formátu JAR

examples složka obsahující podsložky s příklady popsány v příloze **B**

.project konfigurační soubor projektu pro *Eclipse* (pro lepší analýzu zdrojových kódů)

manual.pdf stručný návod k ovládní aplikace ve formátu PDF

Složka examples

Složka obsahuje několik podsložek odpovídajících příkladům z přílohy **B**. Tyto složky mají jednotnou strukturu, jež je následující:

Každá složka s příkladem obsahuje soubor s příponou `txt` obsahující definici automatu popsaného v zadání příkladu. Dále obsahuje soubor `zadani.pdf` obsahující samostatné zadání daného příkladu, totožné se zadáním v následující příloze. Složka navíc obsahuje podsložku `cells`, kde jsou obsaženy soubory s definicemi buněk uvedených v zadání příkladu, určených k načtení do aplikace pro simulaci.

Příloha B

Příklady pro simulátor

V této příloze jsou uvedena zadání jednotlivých příkladů. Tyto příklady slouží k jednoduché demonstraci vyvinuté aplikace. Příklady jsou umístěny do podsložek složky `examples`, jak je popsáno v příloze [A](#).

B.1 Postup při testování příkladu

Ve složce odpovídající simulovanému příkladu se nachází soubor s příponou `txt`, obsahující daný automat. Ten je nutno načíst pomocí tlačítka *Načíst ze souboru* v hlavním okně aplikace po zvolení typu, který odpovídá typu automatu u zvoleného příkladu.

Po načtení souboru s automatem může probíhat dodatečná úprava množin stavů a pravidel automatu. Po dokončení těchto úprav je nutné načíst soubor s multidimensionální buňkou. Tyto soubory se nacházejí v podsložce `cells` daného příkladu.

Po načtení buňky a automatu může být spuštěna simulace, při výskytu více použitelných pravidel bude uživatel požádán o zvolení pravidla. Simulace bude dokončena podle použitých pravidel a buňky budou v závislosti na těchto pravidlech přijaty či odmítnuty.

B.2 Příklady - Vícesměrný multidimensionální automat

B.2.1 Příklad 1

Popis

Tento příklad demonstruje hledání řetězce *aa* v buňce. Pokud buňka obsahuje podřetězec *aa* na jedné z prvních tří os (x,y,z), přičemž řetězec začíná v průsečíku těchto os, je buňka přijata. V opačném případě buňka není přijata.

Automat postupuje po jednotlivých osách a hledá výskyt řetězce *aa*. Při jeho nálezů přechází do akceptujícího stavu Q_a .

Abeceda

a,b

Buňky:

- řetězec s : *ab*,

- obraz $P1$: $\begin{matrix} a & a \\ a & a \end{matrix}$,
- obraz $P2$: $\begin{matrix} a & b \\ b & a \end{matrix}$,
- těleso $O1$ obsahující řetězec aa na třetí ose,
- těleso $O2$ neobsahující řetězec aa

Řetězec s , obraz $P2$ a těleso $O2$ nebudou přijaty.

Stavy automatu:

$q_0, A_1, A_2, A_3, Q_a, Q_r$

Pravidla automatu:

Množina pravidel automatu je následující:

- | | |
|--------------------------------------|--------------------------------------|
| • $q_0, a \rightarrow A_1, 1,$ | • $A_2, a \rightarrow Q_a, 0, 1,$ |
| • $q_0, b \rightarrow Q_r, 1,$ | • $A_3, a \rightarrow Q_a, 0, 0, 1,$ |
| • $q_0, a \rightarrow A_2, 0, 1,$ | • $A_1, b \rightarrow Q_r, -1,$ |
| • $q_0, a \rightarrow A_3, 0, 0, 1,$ | • $A_2, b \rightarrow Q_r, 0, -1,$ |
| • $A_1, a \rightarrow Q_a, 1,$ | • $A_3, b \rightarrow Q_r, 0, 0, -1$ |

B.2.2 Příklad 2

Popis

Tento příklad demonstruje automat přijímající buňky, které ve svých prvních pěti dimenzích obsahují symbol x .

Pětidimensionální prostor je nedeterministicky prohledáván, při nález symbolu x je buňka přijata přechodem do akceptujícího stavu Q_a .

Abeceda

x, y, z

Buňky:

- řetězec $s1$: x ,
- řetězec $s2$: y ,
- obraz $P3$: $\begin{matrix} y & z \\ z & x \end{matrix}$,
- obraz $P4$: $\begin{matrix} y & y \\ y & y \end{matrix}$,
- těleso $O2$ obsahující x ,
- těleso $O3$ neobsahující x ,
- 4-dimensionální buňka $C1$ obsahující x ,
- 4-dimensionální buňka $C2$ neobsahující x

Neбудou přijaty buňky $s2$, $P4$, $O3$ a $C2$.

Stavy automatu:

q_0, S, Q_a, Q_r

Pravidla automatu:

Množina pravidel automatu je následující:

- $q_0, x \rightarrow Q_a, 1,$
- $q_0, * \rightarrow Q_r, 1,$
- $q_0, * \rightarrow S, 1,$
- $q_0, * \rightarrow S, 0, 1,$
- $q_0, * \rightarrow S, 0, 0, 1,$
- $q_0, * \rightarrow S, 0, 0, 0, 1,$
- $q_0, * \rightarrow S, 0, 0, 0, 0, 1,$
- $S, x \rightarrow Q_a, -1,$
- $S, * \rightarrow Q_r, -1,$
- $S, * \rightarrow S, 1,$
- $S, * \rightarrow S, -1,$
- $S, * \rightarrow S, 0, 1,$
- $S, * \rightarrow S, 0, -1,$
- $S, * \rightarrow S, 0, 0, 1,$
- $S, * \rightarrow S, 0, 0, -1,$
- $S, * \rightarrow S, 0, 0, 0, 1,$
- $S, * \rightarrow S, 0, 0, 0, -1,$
- $S, * \rightarrow S, 0, 0, 0, 0, 1,$
- $S, * \rightarrow S, 0, 0, 0, 0, -1$

B.2.3 Příklad 3

Popis

Tento příklad demonstruje automat přijímající řetězce, obrazy nebo tělesa obsahující pouze symboly a . Zároveň musí být každý rozměr (pod)buňky v prvních třech dimenzích rovný 3. Automat také přijímá vícedimensionální buňky, které tyto tělesa obsahují na počátečních souřadnicích.

Kvůli absenci pomocného mechanismu, jako je pomocná páska u multidimensionálního Turingova stroje, je postup automatu signalizován pouze jeho stavy, ve stavech R^* je prováděn postup v buňce směrem vpravo, naopak ve stavech L^* je postupováno směrem vlevo. Tak je prozkoumán celý prostor podmiňující přijetí buňky, přičemž při nález symbolu b či c je proveden přechod do zamítajícího stavu Q_r .

Abeceda

a,b,c

Buňky:

- řetězec s_3 : aaa ,
- řetězec s_4 : aba ,
- obraz P_5 : $\begin{matrix} a & a \\ a & a \end{matrix}$,
 $\begin{matrix} a & a & a \end{matrix}$
- obraz P_6 : $\begin{matrix} a & a & a \\ a & a & a \end{matrix}$,
- těleso O_4 obsahující pouze a o rozměrech $(3, 3, 4)$,
- těleso O_5 obsahující pouze a o rozměrech $(3, 3, 3)$

Přijat nebude pouze řetězec s_4 .

Stavy automatu:

$q_0, L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9, L_{10}, L_{11}, L_{12},$
 $R_1, R_2, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11}, R_{12}, R_{13}, R_{14}, R_{15}, Q_a, Q_r$

Pravidla automatu:

Množina pravidel automatu je následující:

- $q_0, a \rightarrow R_1, 1,$
- $q_0, * \rightarrow Q_r, 1,$
- $R_1, a \rightarrow R_2, 1,$
- $R_1, * \rightarrow Q_r, 1,$
- $R_2, a \rightarrow Q_a, 1,$
- $R_2, * \rightarrow Q_r, 1,$
- $R_2, a \rightarrow L_2, 0, 1,$
- $L_1, a \rightarrow L_2, -1,$
- $L_1, * \rightarrow Q_r, 1,$
- $L_2, a \rightarrow L_3, -1,$
- $L_2, * \rightarrow Q_r, 1,$
- $L_3, a \rightarrow R_4, 0, 1,$
- $L_3, * \rightarrow Q_r, 1,$
- $R_4, a \rightarrow R_5, 1,$
- $R_4, * \rightarrow Q_r, 1,$
- $R_5, a \rightarrow R_6, 1,$
- $R_5, * \rightarrow Q_r, 1,$
- $R_6, a \rightarrow Q_a, 1,$
- $R_6, * \rightarrow Q_r, 1,$
- $R_6, a \rightarrow L_4, 0, 0, 1,$
- $L_4, a \rightarrow L_5, -1,$
- $L_4, * \rightarrow Q_r, 1,$
- $L_5, a \rightarrow L_6, -1,$
- $L_5, * \rightarrow Q_r, 1,$
- $L_6, a \rightarrow R_7, 0, -1,$
- $L_6, * \rightarrow Q_r, 1,$
- $R_7, a \rightarrow R_8, 1,$
- $R_7, * \rightarrow Q_r, 1,$
- $R_8, a \rightarrow R_9, 1,$
- $R_8, * \rightarrow Q_r, 1,$
- $R_9, a \rightarrow L_7, 0, -1,$
- $R_9, * \rightarrow Q_r, 1,$
- $L_7, a \rightarrow L_8, -1,$
- $L_7, * \rightarrow Q_r, 1,$
- $L_8, a \rightarrow L_9, -1,$
- $L_8, * \rightarrow Q_r, 1,$
- $L_9, a \rightarrow R_{10}, 0, 0, 1,$
- $L_9, * \rightarrow Q_r, 1,$
- $R_{10}, a \rightarrow R_{11}, 1,$
- $R_{10}, * \rightarrow Q_r, 1,$
- $R_{11}, a \rightarrow R_{12}, 1,$
- $R_{11}, * \rightarrow Q_r, 1,$
- $R_{12}, a \rightarrow L_{10}, 0, 1,$
- $R_{12}, * \rightarrow Q_r, 1,$
- $L_{10}, a \rightarrow L_{11}, -1,$
- $L_{10}, * \rightarrow Q_r, 1,$
- $L_{11}, a \rightarrow L_{12}, -1,$
- $L_{11}, * \rightarrow Q_r, 1,$
- $L_{12}, a \rightarrow R_{13}, 0, 1,$
- $L_{12}, * \rightarrow Q_r, 1,$
- $R_{13}, a \rightarrow R_{14}, 1,$
- $R_{13}, * \rightarrow Q_r, 1,$
- $R_{14}, a \rightarrow R_{15}, 1,$
- $R_{14}, * \rightarrow Q_r, 1,$
- $R_{15}, a \rightarrow Q_a, 1,$
- $R_{15}, * \rightarrow Q_r, 1,$

B.3 Příklady - Multidimensionální Turingův stroj

B.3.1 Příklad 1

Popis

Tento automat přijme řetězec ab , obraz $\begin{matrix} a & b \\ a & b \end{matrix}$, těleso $T = \begin{matrix} a & b \\ a & b \end{matrix} \oplus_3 \begin{matrix} a & b \\ a & b \end{matrix}$ a buňky, které tyto entity obsahují na počátečních souřadnicích.

Postup automatu začíná standardně od pozice 1,1, odkud pokračuje k pozici 2,1, ověří výskyt řetězce ab a případně dále postupuje do dalších dimenzí. Pro tento postup automat provádí návratové pohyby v buňce, které signalizuje zápisem symbolu r na pomocnou pásku.

Přijetí buňky je pak signalizováno přechodem do koncového stavu F , stav R funguje jako pomocný a signalizuje nepřijetí buňky.

Abeceda

a,b

Buňky:

- řetězec $s1:ab$,
- řetězec $s2:aa$,
- obraz $p1: \begin{matrix} a & b \\ a & a \end{matrix}$,

- obraz $p2$: $\begin{matrix} a & b \\ a & b \end{matrix}$,
- těleso T ,
- těleso $T1 = \begin{matrix} a & a \\ a & a \end{matrix} \odot_3 \begin{matrix} a & b \\ a & b \end{matrix}$

Přijaty nebudou buňky $s2$ a $T1$, ostatní buňky přijaty budou.

Stavy automatu:

$q0, Q1, Q2, Q3, Q4, R, F$

Konečné stavy:

F

Pravidla automatu:

Množina pravidel automatu je následující:

- | | |
|---|--|
| • $q0, a, \# \rightarrow Q1, a, 1, nomove,$ | • $Q2, a, r \rightarrow F, \#, 0, nomove,$ |
| • $q0, b, \# \rightarrow R, \#, 0, nomove,$ | • $Q2, a, r \rightarrow Q3, a, 0, 0, 1, nomove,$ |
| • $Q1, b, a \rightarrow Q1, r, -1, nomove,$ | • $Q3, a, a \rightarrow Q3, b, 1, nomove,$ |
| • $Q1, b, a \rightarrow F, \#, 0, nomove,$ | • $Q3, b, a \rightarrow R, \#, 0, nomove,$ |
| • $Q1, a, a \rightarrow R, \#, 0, nomove,$ | • $Q3, a, b \rightarrow R, \#, 0, nomove,$ |
| • $Q1, a, r \rightarrow Q2, a, 0, 1, nomove,$ | • $Q3, b, b \rightarrow Q4, b, 0, -1, nomove,$ |
| • $Q2, a, a \rightarrow Q2, b, 1, nomove,$ | • $Q4, b, b \rightarrow Q4, a, -1, nomove,$ |
| • $Q2, b, a \rightarrow R, \#, 0, nomove,$ | • $Q4, a, b \rightarrow R, \#, 0, nomove,$ |
| • $Q2, b, b \rightarrow Q2, r, -1, nomove,$ | • $Q4, b, a \rightarrow R, \#, 0, nomove,$ |
| • $Q2, a, b \rightarrow R, \#, 0, nomove,$ | • $Q4, a, a \rightarrow F, \#, 0, nomove$ |

B.3.2 Příklad 2

Popis

Automat v tomto příkladu přijímá tělesa a vícedimensionální buňky obsahující posloupnost symbolů 1,2,3,4,5 na hlavní diagonále prostoru vymezeného osami x, y a z.

Automat postupně pomocí stavů $Q1$, $Q2$ a $Q3$ provádí kroky ve třech dimenzích zkoumaného prostoru tak, aby vždy ve stavu $Q3$ mohl uložit aktuální symbol na pásku a provést posun na pásce vpravo. Při výskytu symbolu 5 je provedena kontrola obsahu pomocné pásky pomocí stavů $C1$, $C2$, $C3$, $C4$. V případě nalezení této posloupnosti je proveden přechod do koncového stavu F , což signalizuje přijetí buňky.

Abeceda

1,2,3,4,5

Buňky:

- těleso $T2$ o rozměrech (5,5,5) obsahující příslušnou posloupnost,
- těleso $T3$ o rozměrech (5,5,5) neobsahující příslušnou posloupnost,

- těleso T_4 o rozměrech (5,5,4) obsahující část příslušné posloupnosti,
- obraz p_3 pro demonstraci nepřijímání obrazů

Buňka T_2 bude automatem přijata, ostatní buňky automatem přijaty nebudou.

Stavy automatu:

$q_0, Q_1, Q_2, Q_3, C_1, C_2, C_3, C_4, F, R$

Konečné stavy:

F

Pravidla automatu:

Množina pravidel automatu je následující:

- $q_0, 1, \# \rightarrow Q_1, 1, 1, \text{nomove},$
- $q_0, *, \# \rightarrow R, \#, 0, \text{nomove},$
- $Q_1, *, \# \rightarrow Q_2, \#, 0, 1, \text{nomove},$
- $Q_2, *, \# \rightarrow Q_3, \#, 0, 0, 1, \text{nomove},$
- $Q_3, 2, \# \rightarrow Q_1, 2, 1, \text{right},$
- $Q_3, 3, \# \rightarrow Q_1, 3, 1, \text{right},$
- $Q_3, 4, \# \rightarrow Q_1, 4, 1, \text{right},$
- $Q_3, 5, \# \rightarrow C_4, \#, 0, \text{left},$
- $C_4, 5, 4 \rightarrow C_3, \#, 0, \text{left},$
- $C_3, 5, 3 \rightarrow C_2, \#, 0, \text{left},$
- $C_2, 5, 2 \rightarrow C_1, \#, 0, \text{left},$
- $C_1, 5, 1 \rightarrow F, \#, 0, \text{nomove}$

B.3.3 Příklad 3

Popis

Tento automat přijímá (stejně jako automat v příkladu B.2.3 pro MWA) řetězce, obrazy nebo tělesa obsahující pouze symboly a . Zároveň musí být každý rozměr této (pod)buňky v prvních třech dimenzích rovný 3. Automat opět přijímá i vícedimensionální buňky, které tyto tělesa obsahují na počátečních souřadnicích.

Automat využívá pásku pro ukládání indexu aktuálního řádku buňky a zbývajících návratových pohybů prováděných ve stavech B_1, B_2, B_3, B_4 a B_5 . Přijetí buňky je pak opět značeno přechodem do koncového stavu F . Při libovolném výskytu symbolu b je proveden přechod do stavu R , který signalizuje nepřijetí buňky.

Abeceda

a,b

Buňky:

- řetězec s_3 : aaa ,
- řetězec s_4 : aba ,
- obraz P_5 : $\begin{matrix} a & a \\ a & a \end{matrix}$,
- obraz P_6 : $\begin{matrix} a & a & a \\ a & a & a \end{matrix}$,
- těleso O_4 obsahující pouze a o rozměrech (3, 3, 4),

- těleso $O5$ obsahující pouze a o rozměrech $(3, 3, 3)$

Nebude přijat pouze řetězec $s4$.

Stavy automatu:

$q0, R, S1, S2, S3, F, B1, B2, B3, B4, B5, P1$

Konečné stavy:

F

Pravidla automatu:

Množina pravidel automatu je následující:

- $q0, b, * \rightarrow R, \#, 0, nomove,$
- $q0, a, \# \rightarrow S1, 1, 0, nomove,$
- $S1, a, 1 \rightarrow S2, 1, 1, nomove,$
- $S1, a, 2 \rightarrow S2, 2, 1, nomove,$
- $S1, a, 3 \rightarrow S2, 3, 1, nomove,$
- $S1, a, 4 \rightarrow S2, 4, 1, nomove,$
- $S1, a, 5 \rightarrow S2, 5, 1, nomove,$
- $S1, a, 6 \rightarrow S2, 6, 1, nomove,$
- $S1, a, 7 \rightarrow S2, 7, 1, nomove,$
- $S1, a, 8 \rightarrow S2, 8, 1, nomove,$
- $S1, a, 9 \rightarrow S2, 9, 1, nomove,$
- $S1, b, * \rightarrow R, \#, 0, nomove,$
- $S2, b, * \rightarrow R, \#, 0, nomove,$
- $S2, a, 1 \rightarrow S3, 1, 1, nomove,$
- $S2, a, 2 \rightarrow S3, 2, 1, nomove,$
- $S2, a, 3 \rightarrow S3, 3, 1, nomove,$
- $S2, a, 4 \rightarrow S3, 4, 1, nomove,$
- $S2, a, 5 \rightarrow S3, 5, 1, nomove,$
- $S2, a, 6 \rightarrow S3, 6, 1, nomove,$
- $S2, a, 7 \rightarrow S3, 7, 1, nomove,$
- $S2, a, 8 \rightarrow S3, 8, 1, nomove,$
- $S2, a, 9 \rightarrow S3, 9, 1, nomove,$
- $S3, b, * \rightarrow R, \#, 0, nomove,$
- $S3, a, 1 \rightarrow F, \#, 0, nomove,$
- $S3, a, 1 \rightarrow B1, 1, 0, right,$
- $S3, a, 2 \rightarrow B1, 2, 0, right,$
- $S3, a, 3 \rightarrow F, \#, 0, nomove,$
- $S3, a, 3 \rightarrow B3, 3, 0, 0, 1, right,$
- $S3, a, 4 \rightarrow B1, 4, 0, right,$
- $S3, a, 5 \rightarrow B1, 5, 0, right,$
- $S3, a, 6 \rightarrow B3, 6, 0, 0, 1, right,$
- $S3, a, 7 \rightarrow B1, 7, 0, right,$
- $S3, a, 8 \rightarrow B1, 8, 0, right,$
- $S3, a, 9 \rightarrow F, \#, 0, nomove,$
- $B1, a, \# \rightarrow B2, 2, -1, nomove,$
- $B2, a, 2 \rightarrow B2, 1, -1, nomove,$
- $B2, a, 1 \rightarrow P1, \#, 0, 1, left,$
- $P1, a, 1 \rightarrow S1, 2, 0, nomove,$
- $P1, a, 2 \rightarrow S1, 3, 0, nomove,$
- $P1, a, 4 \rightarrow S1, 5, 0, nomove,$
- $P1, a, 5 \rightarrow S1, 6, 0, nomove,$
- $P1, a, 7 \rightarrow S1, 8, 0, nomove,$
- $P1, a, 8 \rightarrow S1, 9, 0, nomove,$
- $P1, b, * \rightarrow R, \#, 0, nomove,$
- $B3, a, \# \rightarrow B4, 4, 0, nomove,$
- $B3, b, * \rightarrow R, \#, 0, nomove,$
- $B4, b, * \rightarrow R, \#, 0, nomove,$
- $B4, a, 4 \rightarrow B4, 3, 0, -1, nomove,$
- $B4, a, 3 \rightarrow B4, 2, 0, -1, nomove,$
- $B4, a, 2 \rightarrow B4, 1, -1, nomove,$
- $B4, a, 1 \rightarrow B4, 0, -1, nomove,$
- $B4, a, 0 \rightarrow B5, \#, 0, left,$
- $B5, a, 3 \rightarrow S1, 4, 0, nomove,$
- $B5, a, 6 \rightarrow S1, 7, 0, nomove$