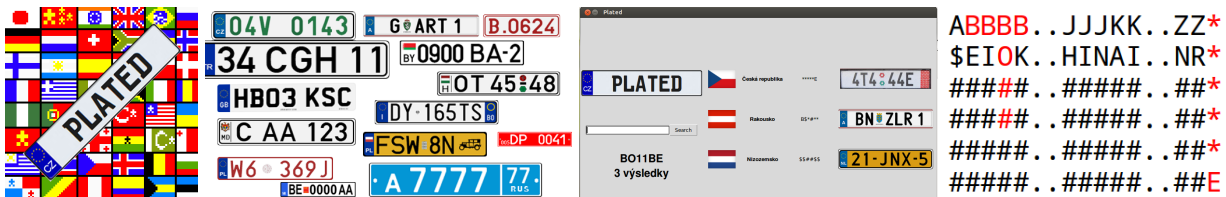


Identifikace státní příslušnosti vozidel pomocí dvoudimenzionálních automatů

Zdeněk Hladík*



Abstrakt

Cílem této práce je aplikace umožňující zjistit státní příslušnost automobilu na základě tvaru jeho registrační značky (dále RZ). Aplikace využívá dvoudimenzionálních konečných automatů a současně vytvořenou obsáhlou databázi povolených tvarů RZ evropských států. V současné době aplikace podporuje 27 států a cca 9 000 příslušných pravidel. Využití této aplikace je např. pro mýtné brány, bezpečnostní záznamy nebo statistiky provozu. Jedná se o demonstrační příklad využití dvoudimenzionálních automatů a operací s 2-D řetězci v praxi pro komerční využití.

Klíčová slova: finite automata — twodimensional languages — car nationality — licence plates

Příložené materiály: [Demonstrační video](#)

* xhladi17@stud.fit.vutbr.cz, Faculty of Information Technology, Brno University of Technology

1. Představení

Tato práce využívá teorii dvoudimenzionálních jazyků pro praktický účel. Se státními poznávacími značkami se setkáváme každý den, přičemž například na českých dálnicích je výrazný podíl zahraničních automobilů.

Pro finanční akce, jakými jsou výběr mýta a pokuty, nebo pro pouhou statistiku provozu je nutné detekovat státní příslušnost automobilu. Ta je často určena štítkem či nálepkou se zkratkou státu. Tyto prvky však nejsou vždy povinné či viditelné průmyslovými kamerami (často i pouhým okem), na rozdíl od formátů RZ jednotlivých zemí a primárních znaků na RZ. Je proto vhodné vytvořit aplikaci, která na základě zadaného obsahu RZ zobrazí názvy států a vzory RZ formátů, kterým zadaný tvar odpovídá.

Pokud se zaměříme na aplikaci, neexistuje žádný veřejně dostupný program pro jakoukoliv platformu, který by dovozoval určit státní příslušnost automobilu pouhým zadáním RZ. Existují online databáze tvarů RZ (např. [World License Plates](#)), jejichž prohledávání

je neefektivní, a interně vyvinuté aplikace, které jsou často neaktualizovatelné a nedokonalé.

Teorie dvoudimenzionálních automatů už byla zmíněna v mnoha pracích, např. [1], avšak žádný významný případ praktického využití není autorovi znám, přičemž případné využití 2-D teorie v případě rozeznávání grafických prvků nebo biometrického zabezpečení (rozeznávání obličeje, otisků) je nevědomé. Především však nebylo nikdy příslušné teorie užito v oblasti rozeznávání příslušnosti textu do speciálních tabulek pravidel jazyků, jako je tomu v této práci.

Aplikace používá 2-D pole obsahující pravidla náležící každému státu, která jsou při zadání řetězce analyzována na případný výskyt pravidla, které by pokrývalo zadaný řetězec. Je k tomu použito jednoduchého automatu, speciálně vyvinutého v rámci aplikace.

Aplikace je vhodná pro jakýkoliv ze zmíněných scénářů použití, je vyvinuta multiplatformně (systémy Windows a UNIX) s databází, jejíž obsah je možno upravovat či doplňovat o nová pravidla a vzory. Obsah

databáze je jednoduše pochopitelný, formátovaný tak, aby ho mohl upravovat i koncový uživatel bez znalosti vývoje softwaru.

2. Formální jazyky a automaty

Pro rychlé představení teorie z oboru formálních jazyků je předpokládáno, že čtenář je obeznámen se základními pojmy, jako je abeceda, konkatenace, jazyk a princip fungování konečného automatu.

Stručně lze popsat následovně: *abeceda* je množina jistého počtu znaků, které lze spojovat (*konkaténovat*) za účelem vytváření slov, neboli *řetězců*. *Jazyk* je pak množina těchto slov a pokud lze tuto množinu popsat pomocí konečného automatu, jde o jazyk *regulární*. *Konečný automat* je stroj přijímající vstup a měnící svůj stav na základě tohoto vstupu a svého současného stavu. Konečné automaty se proto mohou použít pro analýzu vstupního slova a indikaci stavem automatu, zda vstup náleží do příslušného jazyka. Pro formální definice a detailnější popis výše uvedených pojmů odkazujeme na [2].

2.1 Řetězce ve 2-D

Abeceda jako taková je opět pouze množina znaků (ty mohou ale reprezentovat i barvu či vlastnost materiálu), ve většině případů značená jako Σ . Ovšem 2-D řetězce vzniklé spojením znaků této abecedy mají formu obrazů. Rozšířená teorie a definice příslušných pojmů jsou zpracovány například v [3].

Definice 2.1. Dvoudimenzionální řetězec (*nebo* obraz) nad abecedou Σ je *dvoudimenzionální obdélníkové pole skládající se z elementů abecedy* Σ .

Tedy ve vyšší formě abstrakce jde i o klasický obraz, který se může jako *ASCII art* skládat z písmen nebo z barevných prvků. Každý takovýto obraz p má pak dva rozměry, výšku a šířku, tedy: $\ell_1(p)$ a $\ell_2(p)$. Dvojice $(\ell_1(p), \ell_2(p))$ je pak nazývána velikost obrazu p . Obraz o velikosti $(0, 0)$ je značen λ a obrazy o velikosti $(m, 0)$ nebo $(0, m)$, kde $m \neq 0$ nejsou uvažovány. Pokud se pak odkazujeme na jeden element na příslušných souřadnicích m a n v obrazu p , značíme tento element jako $p(m, n)$.

2.2 Operace ve 2-D

Pro dvoudimenzionální řetězce pak existují speciální operace: *sloupcová* a *řádková konkatenace*. Za účelem demonstrace definujeme dva čtvercové obrazy $P3$ a $P4$ nad abecedou $\Sigma = \{0, 1\}$. Výsledky těchto operací pak jsou značeny jako $P3 \odot P4$ a $P3 \ominus P4$. Zásadním pravidlem je, že lze sloupcově konkaténovat pouze obrazy

1	0
0	1

Obraz P3

1	1
0	1

Obraz P4

1	0
0	1
1	1
0	1

$P3 \ominus P4$

1	0	1	1
0	1	0	1

$P3 \odot P4$

se stejnou výškou, obdobně to platí i pro řádkovou konkatenaci. Formální definice těchto operací viz [3].

Navíc pro 2-D řetězce existuje operace *rotace*. Název operace sám o sobě vyjadřuje její výsledek, proto neuvádíme formální definici či příklad použití. Jde o otočení obrazu o 90 stupňů ve směru hodinových ručiček, značeno například pro obraz $P3$ jako $P3^R$.

Definice 2.2. Dvoudimenzionální jazyk nad abecedou Σ je *podmnožinou* Σ^{**} , kde Σ^{**} značí množinu všech obrazů nad abecedou Σ .

Pokud lze takový obraz popsat pomocí 2-D konečného automatu, jde opět o *regulární* 2-D jazyk. Například pro abecedu $\Sigma = \{0, 1\}$, jazyk $L1$ odpovídající pravidlu $L1 = \{p \mid p(i, 1) = p(i, \ell_2(p)), i = 1, \dots, \ell_1(p)\}$ je jazykem obsahujícím jen obrazy, jejichž první sloupec je totožný s posledním. Obsahuje pak například obrazy $P5$ a $P6$.

1	0	1	1
1	1	0	1

Obraz P5

1	1
0	0

Obraz P6

2.3 2-D konečný automat

Původní práce ([1]) představená v roce 1967 zavedla první typ 2-D automatu použitelného pro rozeznávání 2-D řetězců. Jde o jednoduchý stroj pohybující se podle zadaných pravidel na 2-D pásce analyzující její

obsah prostřednictvím změn svých stavů. Nejlépe představitelný 2-D automat je však *čtyřsměrný automat* představený v [3].

Definice 2.3. 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” stavy;

$\delta : Q \setminus \{q_a, q_r\} \times \Sigma \rightarrow Q \times \Delta$ je přechodová funkce

Automat pak začíná analyzovat obraz od pozice $(1, 1)$ se stavem q_0 , pohybuje se po obrazu ve směrech z množiny Δ po obraze, mění svůj stav podle δ . Funkce δ rozhoduje o směru přechodu a změně stavu automatu tak, že při přechodu do stavu q_a je určeno, že přijímaný obraz spadá do příslušného jazyka, naopak přechod do stavu q_r značí, že obraz do jazyka nepatří.

Existuje mnoho obdob 2-D konečných automatů, ze zajímavějších uvádíme automat spojený s pravděpodobnostní logikou (viz [4]), automat kombinovaný s maticemi náležitosti (viz [5]) a online teselační dvoucestný automat (viz [6]). Konečné automaty lze ve dvou dimenzích kombinovat také s *celulárními automaty*, umožňujícími pracovat s celým vstupem najednou bez nutnosti přechodů mezi symboly obsaženými v obraze.

3. Registrační značky

V dnešní době je v každém státě zaveden nejméně jeden formát pro značení vozidel pro jejich jednoznačnou registraci a identifikaci. V každém státě však platí odlišná pravidla pro tvar tohoto značení (v České republice a dále označeno jako RZ).

3.1 Analyzované státy

Pro účely aplikace vyvíjené v rámci této práce jsou zásadní pouze státy geograficky blízké a státy, jejichž vozidla se význačně podílejí na českém provozu.

Tyto státy jsou následující: Česká republika, Slovenská republika, Německo, Polsko, Rakousko, Belgie, Bulharsko, Estonsko, Francie, Chorvatsko, Itálie, Litva, Lotyšsko, Maďarsko, Nizozemsko, Rumunsko, Řecko, Slovinsko, Velká Británie, Španělsko, Bělorusko, Bosna a Hercegovina, Makedonie, Rusko, Moldavsko a Ukrajina.

3.2 Pravidla značení

V každém jmenovaném státě se používá nejméně jeden formát standardního značení a jeden formát značení

pro speciální vozy, jako jsou vozy armády, ministerstev a diplomatů. Největším podílem do sestavené databáze přispělo Německo, které je specifikováno celkem 3 002 pravidly.

Pro příklad složitosti, Ministerstvo dopravy ČR povoluje nejméně 12 textových tvarů RZ (viz [Vyhláška o registraci vozidel 243/2001 Sb.](#)), zahrnujících formát z roku 1960 (např. ZR 00 11), formát z roku 1967 (např. ZRI 00 11) a formát z roku 2011 (1A1 1111). Ze zbylých formátů lze uvést *RZ pro konzulární mise, veteránské nebo závodní vozy, vozy pro osoby s imunitou a vozy pro export* (viz obr.), z nichž některé jsou naprosto neznámé i pro občany tohoto státu.



Obrázek 1. Exportní značka (zdroj: www.platesmania.com)

3.3 Databáze

Pro každý povolený počet znaků na RZ v každém státě je vytvořen soubor pravidel, který vznikl pomocí operací řádkové konkatenace a rotace z kapitoly 2.2.

Každé pravidlo (jako zdroje pravidel byly použity vyhlášky příslušných států o registraci a identifikaci vozidel) je reprezentováno ve formě řetězce, který může obsahovat zástupné znaky, značící například číslice nebo písmena speciální abecedy (např. 12 písmen azbuky v Ukrajině) a zároveň obsahuje znaky standardní abecedy, pro něž je nutné najít přímou shodu.

Tyto řetězce jsou pak řádkově konkatenovány s řetězci stejné délky pro stejný stát a následně je použito rotace, proto jsou v souboru pravidla uspořádána ve sloupcích, kde první řádek je konkatenace prvních znaků pravidel. Rozměry tabulky pravidel pak jsou (počet pravidel, délka tvaru).

Podoba tabulky pro pravidla $L\$*#\$, $LA\$#\$, $LB#\$$ a $LZ#\$$ je znázorněna na obrázku 2.

```

LLLL
$ABZ
*$###
##$$

```

Obrázek 2. Tabulka pravidel

4. Aplikace

Řešená aplikace je naprogramována v jazyce Python 2.x a zahrnuje GUI pro komunikaci s běžným uživatelem. Model použití je jednoduchý: uživatel zadá tvar RZ a

program ho vyhledá ve své databázi, přičemž některým tvarům může odpovídat i více pravidel pro více států (např. shodný formát Itálie a Slovenska), výsledky jsou seřazeny podle četnosti výskytu vozidel příslušného státu v ČR (Ize editovat) a zobrazí uživateli výsledky ve formě názvu státu, příslušného pravidla a obrázku znázorňujícího, jak odpovídající tvar vypadá graficky.

4.1 Automat

Vyvinutý automat se pohybuje po tabulce pravidel třemi směry (je vynechán směr nahoru), tedy jde o třícestný automat, využívající heuristických množin. Kvůli vstupům odlišným od klasického automatu dvou-dimensionálního ale vyvinutý automat není pravým dvoudimensionálním automatem.

Pro pochopení automatu jako dvoudimensionálního je třeba uvažovat vytvoření jazyka L na základě vstupního řetězce s . Tento jazyk pak obsahuje pouze obrazy obsahující sloupec, který reprezentuje pravidlo pokrývající zadaný řetězec. Automat pak provádí rozhodnutí, zda jednotlivé tabulky pravidel náleží do jazyka L .

Definice 4.1. *Vyvinutý konečný automat je osmice $L = \{\Sigma, s, P, \Delta, \mathfrak{R}, T, Q, \delta\}$, kde:*

Σ je vstupní abeceda;

$s \in \Sigma^*$ je vstupní řetězec;

$P \in \Sigma^{**}$ je obraz (tabulka) pravidel;

$\Delta = \{D, R, L\}$ je množina směrů;

\mathfrak{R} je množina "odpovídajících" indexů;

$T \subseteq \mathfrak{R}$ je množina "neodpovídajících" indexů;

Q množina stavů, kde $Q = \{q | q \sim P(m, n)\}$;

$\delta \in Q \times \mathfrak{R} \times s(i) \rightarrow Q \times (\mathfrak{R} \setminus T) \times \Delta$ je přechodová funkce

Neexistuje žádná struktura reprezentující konečný automat pro každý stát, jelikož takováto datová struktura by byla obsáhlá a náročná na výpočetní paměť při práci, navíc by nebylo možné ji jednoduše editovat pro potřeby změny databáze. Algoritmus hledání je jednotný pro všechny státy, jednoduchý na pochopení, využívající prvky 2-D konečného automatu a teorie množin pro zjednodušení výpočtu.

Počáteční pozice automatu je (1,1), řetězec s je zadaná RZ a tabulka pravidel T je soubor vyhledaný v databázi. Algoritmus 1 stručně popisuje algoritmus aplikace představující pohyb automatu po tabulce.

4.2 Heuristiky

Množin a podmnožin je využito pro možnost odstranění směru nahoru mezi pohyby automatu. Pokud by automat hledal postupně po sloupcích pravidlo odpovídající zadanému tvaru, musel by se při každém neúspěchu navracet na první řádek automatu. Nyní

Algoritmus 1: Hledání sloupce v tabulce

```

Data: tabulka T, řetězec s
r = první řádek tabulky T;
for písmeno i v r do
    // pohyb ve směru R
    if i == s[0] then
        | přidej index i do množiny a
    end
end
i=0;
r=další řádek tabulky T;
while r není prázdný do
    i++;
    for index j v a do
        // pohyb ve směru R
        if r[j] == s[i] then
            | přidej index j do množiny a1;
        end
    end
    a = a1;
    if a je prázdná then
        | break;
    end
    i++;
    // pohyb ve směru D a L
    r=další řádek tabulky T;
end
if a není prázdná then
    | nalezen odpovídající sloupec;
end

```

při zjištění, že pravidlo v daném indexu neodpovídá vzoru, pouze odstraní pravidlo z množiny a pokračuje s jiným kandidátem, což v nejlepším případě ušetří až cca 80 % výpočetního času.

Algoritmus pracuje v jednom čase pouze s jedním řádkem souboru, což je při obsáhlých souborech velmi příznivé z hlediska výpočetní paměti.

4.3 Práce s databází

Algoritmus je proveden pro každý stát s pravidly stejné délky, jako je zadaná RZ. Databáze pak pro zobrazení výsledků obsahuje soubory spojující pravidla s odpovídajícími obrázky, které jsou zobrazeny uživateli, viz obrázek 4.

Aplikace umožňuje editaci databáze pro případy změn pravidel pro tvary RZ (např. plánovaná změna značení řeckých automobilů) nebo zahrnutí dalších států do databáze, například při použití programu v jiném státě.

5. Použití aplikace

Program a příslušná databáze jsou uloženy v pevné adresářové struktuře, je ale možné použít odkazů pro oddělení databáze od spuštění aplikace.

5.1 Hledání

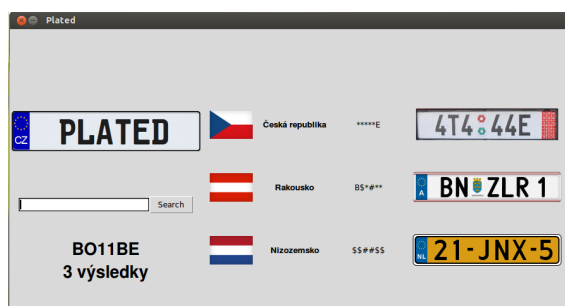
Po spuštění je uživatel vyzván k zadání RZ. Pro demonstraci volíme RZ: BO11BE.

Po zadání jsou v databázi nalezeny všechny státy s odpovídající délkou pravidel (6), tedy například Česká republika, Německo, Belgie. Je použito algoritmu 1 pro nalezení odpovídajícího pravidla. Například pro ČR:

```
ABBBB..JJJKK..ZZ*
$EIOK..HINAI..NR*
#####..####*
#####..####*
#####..####*
#####..####*
#####..###E
```

Obrázek 3. Demonstrace algoritmu vyhledávání pravidla

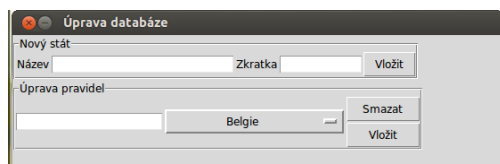
Při nálezku pravidla je využito spojovacího souboru CZ . imgs pro nalezení obrázku vzoru RZ. Na obrázku 4 je zobrazen výsledek.



Obrázek 4. Výsledek vyhledávání

5.2 Editace

Pro editaci je možné využít standardních editorů, při neznalosti aplikace je však vhodné použít vyvinutý editor databáze (viz obr 5). Lze upravovat státy či jednotlivá pravidla bez rizika ztráty integrity databáze. Lze vložit nové obrázky se vzory pravidel ve formátu PNG.



Obrázek 5. Editor databáze

Pro demonstraci slouží video, jehož odkaz je v sekci Abstrakt.

6. Závěr

Tato práce představila teorii dvoudimensionálních automatů a jazyků pro originální využití, jako je klasi-

fikace státních poznávacích značek. Výsledkem práce je aplikace umožňující zjistit státní příslušnost auta se zadanou RZ.

Výsledná aplikace je schopná rozeznávat RZ z celkem 27 evropských států s 9 000 pravidel a podávat výsledky v jednoduchém grafickém prostředí. Je použito několika heuristických vylešnění, díky nimž aplikace pracuje svižně a multiplatformně.

V komerční sféře má tato aplikace široké využití a v současné době probíhá konzultace s firmou provádějící audit mýtného o nasazení principů aplikace do praxe. Aplikace byla testována koncovými uživateli a shledána jako velmi užitečná.

Přínosem práce pro výzumnou sféru je demonstrace principu fungování 2-D konečných automatů na praktickém příkladu bez nutnosti pochopení obsáhlé matematické teorie, která za ním stojí.

Nadále by mohla práce vést ostatní výzkumníky k hledání nových oborů, kde lze tuto teorii využít, jako je například rozeznávání obrazu a bezpečnostní prvky na bázi obrazu.

V budoucnosti je plánováno rozšiřování databáze aplikace, sofistikovanější GUI pro praktické použití aplikace a spolupráce s dalšími firmami, které projeví zájem o nasazení aplikace v praxi.

Poděkování

Rád bych poděkoval vedoucímu této práce prof. Alexanderu Medunovi za jeho čas a pomoc.

Literatura

- [1] M. Blum and C. Hewitt. Automata on a 2-dimensional tape. *8th Annual Symposium on Switching and Automata Theory (SWAT 1967)*, pages 155–160, 1967.
- [2] Alexander Meduna. *Automata and languages*. Springer, London, 2000.
- [3] Grzegorz Rozenberg and Arto Salomaa. *Handbook of formal languages*, pages 215–267. Springer Verlag, Berlin, 1997.
- [4] T Okazaki. A note on two-dimensional probabilistic finite automata. *Information Sciences*, vol. 110(3-4):303–314, 1998.
- [5] Yongming Li. Finite automata theory with membership values in lattices. *Information Sciences*, vol. 181(issue 5):1003–1017, 2011.
- [6] Katsushi Inoue and Akira Nakamura. Two-dimensional multipass on-line tessellation acceptors. *Information and Control*, vol. 41(issue 3):305–323, 1979.