

Vysoké učení technické v Brně
Fakulta strojního inženýrství
Ústav automatizace a informatiky

Strukturální metody identifikace objektů pro řízení průmyslového robotu

Dizertační práce

Vypracoval: Ing. Martin Minařík
Školitel: doc. RNDr. Ing. Jiří Šťastný, CSc.

Brno červenec 2009

Klíčová slova: *strukturální metody, rozpoznávání objektů, syntaktická analýza, deformační gramatika, invariance vůči rotaci, diferenciální primitiva, Earlyho parser, LR(k) parser, Tomita parser, LRE(k) parser, invariance vůči změně měřítka*

Keywords: *structural method, pattern recognition, syntactic analysis, deformation grammar, rotation invariance, differential primitives, Earley parser, LR(k) parser, Tomita parser, LRE(k) parser, invariance towards change of scale*

Obsah

1 ÚVOD 5
2 CÍLE DIZERTAČNÍ PRÁCE 6
3 ROZPOZNÁVÁNÍ OBJEKTŮ POMOCÍ STRUKTURÁLNÍCH METOD 7
3.1 Earlyho Parser 7
4 ROZPOZNÁVÁNÍ DEFORMOVANÝCH OBJEKTŮ 8
4.1 Metody pro zjištění vzdáleností mezi atributovými popisy obrazů 8
5 SYNTAKTICKÁ ANALÝZA S OPRAVOU CHYB 9
5.1 Konstrukce rozšířené deformační gramatiky 9
5.2 Modifikovaný Earlyho algoritmus 10
5.3 Efektivní implementace Earleyho algoritmu 10
5.4 Zpracování e-pravidel 11
5.5 Předpověď predikce a redukce 12
5.6 Využití předpovědi pro zefektivnění analýzy nejednoznačné gramatiky pro rozpoznávání deformovaných objektů 13
5.7 Optimalizace gramatiky pro konkrétní vstupní slovo 14
5.8 LR parsery 14
5.9 Tomita Parser a jeho modifikace 15
5.10 Modifikovaný hybridní LRE(k) algoritmus 15
5.11 Vyhodnoceni popisovaných parserů 15
6 ZAJIŠTĚNÍ INVARIANCÍ U METOD SYNTAKTICKÉ ANALÝZY 16
6.1 Zajištění invariance vůči volbě počátečního bodu popisu 16
6.2 Zajištění invariance vůči rotaci 17
6.3 Invariance vůči změně měřítka 18
6.4 Zpracování vnitřních hran objektu 18
7 VÝSLEDKY 19
7.1 Výsledky úspěšnosti rozpoznávání objektů 20
7.2 Výsledky optimalizací Earlyho parseru 20
8 ZÁVĚR A ZHODNOCENÍ VÝSLEDKŮ DEFORMAČNÍCH GRAMATIK 22
9 LITERATURA 23
CURRICULUM VITAE 25
ABSTRAKT 26
ABSTRACT 27

1 ÚVOD

Důležitou oblastí současných informačních technologií je počítačové vidění (*computer vision*), které se v poslední době prudce rozvíjí. Jeho úkolem je rozvíjení metod pro rozpoznání objektů [26], které nacházejí praktické uplatnění v celé řadě úloh, například diagnostické povahy; ze všech možných případů uveďme například průmyslovou robotiku, kde se obvykle pro zpracování obrazu používají metody „Momenty objektu“, „Fourierovy deskriptory“, „Ramena objektu“ a podobně. Vyšší nároky, kladené na metody rozpoznávání obrazu, často i požadavky pro zpracování v reálném čase (řízení průmyslových robotů), vyžadují použití rychlých metod při zachování vysokého procenta úspěšnosti identifikace jednotlivých objektů.

Metody pro rozpoznávání objektů lze rozdělit podle použitého popisu a způsobu vyhodnocování popisu objektů do dvou základních skupin:

- příznakové metody
- strukturální (syntaktické) metody

První skupina popisuje objekty pomocí číselných charakteristik. Číselný vektor popisující objekt se nazývá *příznakový* vektor a metody využívající jej jsou tudíž známé pod názvem *příznakové* metody. Pokud jsou důležitým nositelem informací o objektech jejich strukturální charakteristiky, tyto metody nejsou vhodné. Transformací úlohy do příznakového prostoru dochází ke ztrátě strukturálních charakteristik a je obtížné či téměř nemožné je získat zpět. Proto v takových případech je vhodnější popsat objekty metodami druhé skupiny pomocí elementárních popisných vlastností, tak zvaných *primitiv* a relací mezi nimi. [29].

Metody druhé skupiny, u nichž je každý objekt popsán relační strukturou, jsou známé pod názvem *strukturální* metody. Pokud omezíme popisné relace na jedinou, relaci *„následuje za“*, přechází struktura na posloupnost neboli řetězec symbolů. Každý symbol řetězce odpovídá jednomu popisnému primitivu z množiny primitiv, použitých k popisu. Tuto množinu lze chápat jako abecedu formálního jazyka a řetězce jako slova tohoto formálního jazyka. Úlohu rozpoznávání lze pak popsat jako problém rozhodnout, zda slovo generované danou gramatikou odpovídá řetězci, popisujícímu daný objekt. Zatímco u příznakových metod rozpoznávání je využíván kvantitativní popis předmětů číselnými parametry (*příznakovým vektorem*), u strukturálních metod má vstupní popis kvalitativní charakter odrážející strukturu objektu. Strukturální metody tedy dávají na základě hlubšího teoretického rozboru velmi dobré předpoklady pro úspěšnou identifikaci objektů [18].

Průběh zpracování a rozpoznávání digitalizovaného obrazu lze rozdělit do několika základních kroků:

- *předzpracování obrazu*
- *segmentace obrazu*
- *popis objektů*
- *rozpoznávání objektů*

Při hodnocení algoritmů počítačového rozpoznání obrazu se bere v úvahu jejich úspěšnost klasifikace, podíl nerozpoznaných a špatně klasifikovaných objektů. Důležité jsou i jejich

časové a paměťové požadavky, které dle aplikačního nasazení mohou být klíčové (například řízení průmyslových robotů).

2 CÍLE DIZERTAČNÍ PRÁCE

U klasických strukturálních metod je chybné rozpoznání objektu způsobeno převážně vlivem šumu nebo obrazových poruch ve vstupním obraze. Ačkoliv vhodně použité metody preprocesingu (více v [13], [28], [30], [32]) mohou významně zlepšit vlastnosti obrazu a tedy míru úspěšně rozpoznaných objektů, stále se díky nadměrnému šumu nebo poruchám obrazu může vyskytnout zkreslení snímaného tvaru objektů. Klasické metody syntaktické analýzy poté obvykle selžou. Existují dva základní druhy deformace obrazu:

- deformace, které můžeme předpokládat
- náhodné deformace

Mezi deformace, které lze předpokládat, patří deformace rohů objektu nebo deformace krátkých přímých úseků hran objektu. Tyto deformace je vhodné do návrhu analyzátoru zahrnout.

Pro klasifikaci objektů, jejichž tvar je náhodně deformován, je možné použít například metody pro stanovení vzdálenosti mezi atributovými popisy obrazů nebo deformační gramatiky. V dalším textu jsou objekty, jejichž tvar obrazu je náhodně deformován, nazývány z důvodů jednoduššího popisu jako „*deformované objekty*“.

Tato práce je primárně zaměřena na výzkum možností využití strukturálních metod pro rozpoznání náhodně deformovaných objektů se zaměřením na míru úspěšnosti klasifikace. Důležitou součástí této práce je nejen důkladný teoretický rozbor dané problematiky včetně návrhu příslušných algoritmů, ale i jejich implementace v testovacím prostředí, která simulačními experimenty prokáže jejich použitelnost v praxi. Konkrétní cíle práce jsou:

- Analyzovat možnosti rozpoznávání deformovaných objektů pomocí strukturálních metod, konkrétně deformačních gramatik.
- Optimalizovat návrh deformační gramatiky – prozkoumat možnosti návrhu deformační gramatiky s cílem dosažení maximální efektivity analýzy.
- Analyzovat možnosti automatického vytváření gramatiky popisující rozpoznávaný objekt a navrhnout vhodný způsob.
- Analyzovat použitelné parsery a vybrat vhodný typ. Modifikovat jej k umožnění analýzy deformační gramatiky a dále k dosažení dostatečné efektivity analýzy.
- Vhodným způsobem zajistit veškeré potřebné invariance.
- Vytvořit komplexní testovací prostředí obsahující všechny potřebné funkce, které zajistí objektivní zhodnocení analyzované metody.

3 ROZPOZNÁVÁNÍ OBJEKTŮ POMOCÍ STRUKTURÁLNÍCH METOD

Rozpoznávání předmětů (klasifikace) spočívá v zařazování předmětů do tříd. *Třída* je množina objektů, jejichž atributy mají z hlediska klasifikace společné rysy. *Předmět* je fyzikální objekt, který v počítačovém vidění představuje nejčastěji určitou oblast segmentovaného obrazu [29].

Samotnou činnost klasifikace vykonává *klasifikátor*. Klasifikátor nerozhoduje o třídě předmětu podle předmětu skutečného, nýbrž podle jeho vjemu. Tento vjem se nazývá *obraz* (předmětu). Pro jednoduchost pod pojmem *objekt* je v textu míněn segmentovaný (více v [7]) obraz jednoho předmětu s detekovanými hranami (více v [27]).

U syntaktického přístupu je obraz reprezentován řetězcem, tvořeným sledem primitiv obrazu a polohových vztahů mezi nimi, tedy vstupní popis má kvalitativní charakter odrážející strukturu objektu [29]. Máme-li objekt popsán hranicí, která je uzavřená, můžeme snadno nalézt primitiva. Těmito primitivy je pak definován tvar objektu. Množina slov, popisujících obrazy jedné třídy, tvoří jazyk této třídy. Jazyk, pokud je konečný, může být definován množinou jeho slov. Jednotlivá slova jazyka jsou generována takzvanými přepisovacími pravidly gramatiky.

Jeden ze způsobů, jak rozpoznat strukturu daného neznámého obrazu spočívá v porovnání jeho strukturální reprezentace ve formě řetězce s reprezentacemi vzorových obrazů jednotlivých tříd. Takový způsob je nezbytný například v úlohách, kdy počet tréninkových vzorů je nedostatečný pro odvození gramatik, nebo když každý obraz může být považován za prototyp třídy obrazů. Tyto případy lze též řešit pomocí algoritmů pro zjištění vzdálenosti mezi atributovými popisy obrazů. Tento způsob je jednoduchý a rychlý. Je-li však třeba pro rozpoznání úplný popis obrazu, je nezbytná syntaktická analýza.

Úkolem syntaktického rozpoznávání obrazu je určit, zda analyzovaný obraz reprezentovaný slovem odpovídá slovu dané gramatiky, tedy zda gramatika může tento obraz generovat. Obraz je reprezentován řetězcem jazyka, který je generován danou gramatikou. Gramatiky lze rozdělit podle různých hledisek do více typů, jedno z nejpoužívanějších rozdělení je takzvaná Chomského hierarchie gramatik, podrobněji v [9].

Existují dva základní způsoby syntaktické analýzy (podrobněji například v [31], [4]). Jedná se o analýzu shora-dolů (*top down*) a analýzu zdola-nahoru (*bottom up*).

3.1 EARLYHO PARSER

Obecně analýza prohledáváním do šířky je značně neefektivní a prohledávání je potřeba nějakým způsobem omezit. Bylo nalezeno množství metod k realizaci toho omezení prohledávání do šířky, povětšinou na úkor obecnosti gramatiky, jež jsou tyto metody schopny zpracovat. Jednu z nich, která omezí větvení při prohledávání do šířky do rozumných mezí při zachování plné obecnosti zpracované gramatiky, objevil v roce 1970 Jay Earley.

Earlyho parser (*Earley Parser,* podrobnější popis například v [20], [24], [31], [4]) je dynamický algoritmus provádějící syntaktickou analýzu shora dolů. Algoritmus je pojmenován po svém objeviteli – Jay Earley. Earlyho parsery jsou zvláště zajímavé, protože dokáží zpracovávat bezkontextové gramatiky. Jsou zvláště efektivní, pokud jsou pravidla

analyzované gramatiky zapsána pomocí *levé rekurze* [19]. Modifikace tohoto algoritmu byla použita v řešeném simulačním prostředí.

Earlyho parser je obvykle zařazován do kategorie parserů provádějící analýzu shora dolů (*top-down*), ale ve skutečnosti tento parser dosahuje svých výborných výsledků též omezeným prohledáváním do šířky s rozpoznáváním zdola nahoru (*bottom-up*). Dále při uvážení jeho hlavních vlastností, schopnosti zpracovat levou rekurzi a požadavků na speciální opatření pro zpracování e-pravidel někdo preferuje jeho zařazení jako shora dolů (*bottom-up*) metoda. Lze též najít určitou podobnost mezi Earlyho algoritmem a CYK algoritmem, podrobněji například v [25].

Breath-first, bottom-up parsery, mezi které lze Earlyho parser zařadit, jsou oblíbené díky svým vlastnostem, pracují on-line, dokáží zpracovat levou rekurzi a lze je upravit i ke zpracování e-pravidel.

4 ROZPOZNÁVÁNÍ DEFORMOVANÝCH OBJEKTŮ

Z předchozího textu je patrné, že rozpoznávání nedeformovaných objektů pomocí strukturálních metod je bezproblémové, nabízí výbornou rychlost a stoprocentní úspěšnost klasifikace, avšak u náhodně deformovaných objektů je téměř nemožné.

4.1 METODY PRO ZJIŠTĚNÍ VZDÁLENOSTÍ MEZI ATRIBUTOVÝMI POPISY OBRAZŮ

Tyto metody nepatří přímo mezi strukturální metody, ale pouze využívají strukturálního popisu objektů ve formě řetězce. Vzdálenost mezi řetězci udává míru podobnosti dvou řetězců, z nichž jeden odpovídá analyzovanému objektu a druhý obrazu třídy a lze ji využít pro rozpoznávání objektů. Pro stanovení této mezi řetězcové vzdálenosti lze použít nejrůznější metody ke stanovení vzdálenosti mezi atributovými popisy obrazů [29]. Nejprve je nutno určit reprezentaci objektu pomocí řetězce, například pomocí *Freemanových řetězcových kódů* (*chain codes*), podrobněji v [8], které se používají k popisu hranic objektů. Hranice je určena počátečním bodem a posloupností symbolů, odpovídajících úsečkám jednotkové délky v několika předem stanovených směrech. V této práci se používá osm směrů po 45°, takzvané osmi – okolí.

Mezi metody pro zjištění vzdáleností patří například Hammingova vzdálenost $H_d(s,t)$ [10], [21], Levenshteinova vzdálenost $L_d(s,t)$[12], Damerauova vzdálenost $D_d(s,t)$ [10], [9] Jaccardova vzdálenost J_d [10], Minkowského vzdálenost $M_d(s,t,power)$ [3] a metoda Needleman-Wunsch [10]. Výsledky těchto a dalších dostupných metod jsou uvedeny například v [29].

Informovanost těchto metod není nejvyšší, jedná se pouze o „nějakou“ metriku, proto tyto metody mohou produkovat nerozpoznané a chybně rozpoznané objekty. Pokud jsou parametry těchto metod pro zjištění vzdáleností mezi atributovými popisy obrazů správně nastaveny, tyto metody nabízejí relativně dobrou míru úspěšně identifikovaných objektů při dosažení výborné rychlosti klasifikace. Avšak chybné rozpoznání objektu může nastat.

5 SYNTAKTICKÁ ANALÝZA S OPRAVOU CHYB

Předchozí kapitola pojednávající o metodách využívajících strukturálního popisu objektu k jejich rozpoznávání ukázala, že je možné využít strukturálních metod i pro rozpoznávání náhodně deformovaných objektů, ovšem úspěšnost popsaných metod není vždy nejvyšší. Snahou této práce je tuto úspěšnost co nejvíce zvýšit, případně zcela zamezit chybnému rozpoznání objektu a tedy najít metodu vhodnou i pro „přesné“ rozpoznávání náhodně deformovaných objektů. Dosáhneme toho důsledným využitím všech dostupných znalostí o cílech klasifikace, tedy vlastní gramatiky popisující objekt. Oproti metodám popsaným v předchozí kapitole je tedy tento přístup informovanější, více v [16], [17].

Pokud budeme provádět klasickou syntaktickou analýzu řetězce popisujícího náhodně deformovaný objekt nějaké třídy, patrně nebude do dané třídy klasifikován, nebo jen zcela „náhodou“. Řešením je rozšíření původní gramatiky o chybová – deformační pravidla zahrnující všechny možné náhodné deformace objektu. Původní gramatika je regulární nebo bezkontextová, rozšířená deformační gramatika je vždy bezkontextová a navíc víceznačná, tedy její syntaktická analýza bude složitější. Úloha se poté změní na úlohu nalezení nedeformovaného řetězce, jehož vzdálenost je od analyzovaného minimální [31].

5.1 KONSTRUKCE ROZŠÍŘENÉ DEFORMAČNÍ GRAMATIKY

Rozšířená deformační gramatika má za úkol spolehlivě generovat všechny možné deformace objektů, jež mohou nastat.

Nejprve se vytvoří terminální a neterminální symboly rozšířené gramatiky. Rozšířená gramatika má být schopna generovat stejné objekty jako gramatika původní, její množina terminálu tedy obsahuje stejné symboly jako množina terminálních symbolů původní gramatiky a též někdy může obsahovat navíc nějaké terminály popisující deformace.

Množina neterminálních symbolů rozšířené gramatiky oproti původní obsahuje navíc deformační neterminály, což jsou neterminály generující všechny možné deformace původních terminálů. Dále se do rozšířené gramatiky vkládají pravidla popisující deformace původních terminálů v pravidlech původní gramatiky, více v [31]. Nakonec se vytvoří deformační pravidla rozšířené gramatiky popisující:

- počáteční symbol nové rozšířené gramatiky
- deformace počátečního symbolu
- přepis nově přidaných neterminálů na terminály
- náhodné deformace terminálů
- přepis na prázdný symbol (e-pravidla)
- všechny možné náhodné deformace hran objektu

S rozšířenou gramatikou pracuje *Syntaktický analyzátor s opravou chyb* [31], který vyhledává takovou deformaci vstupního řetězce, která je spojena s nejmenším součtem vah chybových (deformačních) pravidel. Rozšířená deformační gramatika je víceznačná gramatika.

Problém je tedy najít vhodnou, dostatečně rychlou metodu, schopnou analyzovat víceznačnou gramatiku a akumulovat váhy deformačních pravidel. Parserů splňujících tato kritéria je více, v této práci byly analyzovány následující tři metody:

- Modifikovaný Earlyho parser
- Modifikovaný Tomita parser
- Modifikovaný hybridní LRE(k) algoritmus (hybrid mezi Earlyho parserem a LR(k) parserem)

V navrženém testovacím prostředí byl pro syntaktickou analýzu s opravou chyb použit modifikovaný Earlyho algoritmus, který oproti původní variantě navíc akumuluje příslušné váhy pravidel, použité při derivaci deformovaného řetězce podle rozšířené deformační gramatiky.

Deformační gramatika sestrojená způsobem uvedeným v předchozí kapitole je maximálně univerzální, obsahuje deformace původních pravidel, všechny možné deformace každého terminálu v jakýkoliv jiný terminál nebo kombinaci terminálů, deformace počátečního symbolu, deformace terminálu na prázdný řetězec. Výhoda takto navržené deformační gramatiky je její univerzální použitelnost pro obsáhlou skupinu problémů. Ovšem při rozpoznávání deformovaných objektů si obvykle vystačíme s jednodušší gramatikou, která ačkoliv je méně obecná, poskytuje díky menšímu počtu pravidel efektivnější (rychlejší) rozpoznávání objektů. Pro rozpoznávání deformovaných objektů jsou pravidla popisující deformace počátečního symbolu zbytečná. Též e-pravidla se při některých typech zápisu gramatiky popisující objekt nemusí používat a též dle použitého parseru a způsobu, jakým parser zpracovává e-pravidla, mohou být též zbytečná.

5.2 MODIFIKOVANÝ EARLYHO ALGORITMUS

Modifikace Earlyho parseru k zajištění korektní analýzy deformačních gramatik spočívá v akumulaci vah pravidel v průběhu analýzy v operaci kompletace, podrobněji v [31].

Jestliže je položka $[A \to \alpha\bullet, i, x]$ v I_j a položka $[B \to \beta \bullet A\gamma, k, y]$ v I_i, pak:

- Existuje-li již položka ve tvaru $[B \to \beta A \bullet \gamma, k, z]$ v I_j , pak pokud $\boldsymbol{x + y < z}$ nahradíme u této položky hodnotu **z** hodnotou $\boldsymbol{x + y}$.
- Neexistuje-li, přidáme novou položku $[B \to \beta A \bullet \gamma, k, x + y]$.

5.3 EFEKTIVNÍ IMPLEMENTACE EARLEYHO ALGORITMU

Popis Earlyho algoritmu je uveden v celé řadě publikací a článků…ovšem snad vždy jsou popsány pouze jeho hlavní principy, jeho podrobný popis a případné poznámky, jak ho efektivně implementovat, zcela chybí. Tato práce se též podrobně zabývá jeho efektivní implementací s uvážením jeho dalšího použití pro analýzu nejednoznačné deformační gramatiky.

Pokud se podíváme na analýzu nějakého řetězce pomocí prohledávání do šířky, uvidíme, že velké množství provedených operací (redukce) jsou zcela zbytečné. Earley si povšimnul společné vlastnosti těchto zbytečných redukcí (které jsou nekompatibilní s přístupem shora dolů), že je nemožné derivovat je z počátečního symbolu. Proto navrhnul metodu, která redukce omezí pouze na ty, které se mohou derivovat z počátečního symbolu. Výsledný parser má časovou složitost n^3 pro vstupní řetězec délky n, oproti C^n u prohledávání do šířky.

Při analýze nejednoznačné gramatiky se počet generovaných Earlyho stavů s rostoucí délkou řetězce významně zvětšuje. Jeden z hlavních důvodů nižší efektivity analýzy

deformační gramatiky je, že algoritmus tráví značné množství času prohledáváním jednotlivých úrovní seznamu stavů. Proto je vlastní implementace toho seznamu stavů pro celkovou rychlost algoritmu zcela klíčová. Například datová struktura TList z Delphi (indexovaný lineární spojový seznam) je svojí rychlosti nevyhovující, výrazně lepší je dynamické pole.

Dále je výhodný oddělený přístup k jednotlivým úrovním seznamu z důvodu efektivnějšího prohledávání seznamu stavů. Poznamenejme, že je nutno ponechávat i předchozí úrovně seznamu přístupné během celé analýzy, protože se neustále používají.

Konstrukce jedné úrovně seznamu stavů probíhá ve třech fázích. Dvě odpovídají neomezenému algoritmu, kde byly nazývány „*shift*" a „*reduce*", zde jsou nazývany „*scan*" a „*complete*". Třetí operace je nová a je vztažená k *top-down* komponentě algoritmu, nazývá se „*predictor*".

Earlyho stavy v seznamu stavů se dají rozdělit do dvou druhů, první jsou již zkompletované stavy (znak „*" Earlyho tečkové notace je až na konci stavu) a druhý, nezkompletované stavy (znak „*" Earlyho tečkové notace není na konci stavu). Predikce prohledává nezkompletované stavy, kompletace prohledává již zkompletované stavy a scanování prohledává obecně oba druhy stavů. Prohledávání seznamu stavů při provádění některé z operací můžeme tedy dále značně zefektivnit rozdělením každé úrovně seznamu stavu do dvou částí, jedna pro již zkompletované stavy a druhá pro ještě nezkompletované stavy. Výsledkem každé operace je nějaký nový stav, který se správně zařadí do příslušného seznamu (zkompletovaných / nezkompletovaných) stavů.

Tedy Scanner, Completer a Predictor zpracovávají čtyři seznamy stavů pro každý znak vstupního řetězce, kde jsou pro vstupní symbol Q_p čtyři seznamy stavů:

- $Itemset_{p-1}$ – seznam, který obsahuje pouze stavy dostupné před zpracováním Q_p
- $Completed_p$ – seznam stavů, které byly zkompletovány díky Q_p
- $Active_p$ – seznam, který obsahuje nezkompletované stavy po průchodu Q_p
- $Predicted_p$ – seznam nově předpovězených stavů

Seznamy c$ompleted_p$, $active_p$ a $predicted_p$ dohromady formují $itemset_p$. Na počátku $itemset_{p-1}$ je naplněn (jako výsledek zpracování Q_{p-1}) a ostatní seznamy jsou prázdné. Konstrukce setu $itemset_0$ je speciální. Pokud *completed* set pro poslední symbol vstupního řetězce obsahuje položku *(S -> ... *, 0),* tedy položku obsahující celý vstupní řetězec a redukující ji na počáteční symbol, byla analýza úspěšná a vstupní řetězec lze vygenerovat danou gramatikou.

5.4 ZPRACOVÁNÍ E-PRAVIDEL

Podobně jako velké množství parserů, Earlyho parser nedokáže zvládnout e-pravidla bez speciálních opatření. Problém nastává během predikce položek tvaru *A->*..., p+1* jako následek výskytu **A* v položce v seznamu $active_p$ nebo $predicted_p$, parser může narazit na prázdnou predikci *A->*, p+1*. Toto znamená, že neterminál *A* byl zkompletován právě před symbolem *p+1* a tato kompletace by měla být přidána do seznamu $completed_p$, který doposud obsahoval položky *p* nejvýše. Toto znamená nutnost spustit Completer znovu. Tímto ale obtíže nekončí. Pokud Completer spustíme znovu, zachytí nově přidané položky, které jsou úrovně *p + 1*. Completer bude tedy konzultovat $itemset_p$, který je nekompletní,

protože se stále přidávají položky do jeho částí $active_p$ a $predicted_p$. Pokud se v seznamu najde výskyt **A,* přidá se místo toho kopie *A** a takovéto položky mohou vyžadovat další predikce (pokud za * následuje další neterminál), část z nich mohou být již zkompletované položky patřící do $completed_p$ a znamenající opět více práce pro Completer. Další položky mohou mít úroveň nižší než *p*, což přináší zpět vzdálenější položky, které mohou či nemusejí být zkompletovány… Nejjednodušší a obvykle zcela postačující cesta, jak tyto obtíže vyřešit, je nechat běžet Completer a Prediktor, dokud je co přidávat. Jediná nevýhoda tohoto řešení je zvýšení časové složitosti analýzy.

Další přístup byl navržen Earleym [6], [5], který navrhl, aby se v Completeru tečka posouvala přes neterminální symbol a následně došlo k vyhledání, zda se do S_i přidají nějaké položky. Z důvodu efektivnosti je nutné kolekci sledovaných neterminálů ukládat do datové struktury s rychlým přístupem. Tento způsob je relativně komplikovaný a nepříliš efektivní, podrobněji v [11].

Patrně nejlepší způsob zahrnuje jednoduchou modifikaci Prediktoru založenou na myšlence nulovatelnosti. Neterminál A je nulovatelný, pokud $A \rightarrow e$ (z A lze derivovat e), terminální symboly pochopitelně nulovatelné nejsou. Prediktor tedy může být popsán následovně:

- Pro každý stav v S(k) tvaru $(X \rightarrow \alpha \bullet Y\ \beta, j)$ (kde j je původní úroveň stavu), přidej $(Y \rightarrow \bullet\ \gamma, k)$ do S(k) pro každé pravidlo s Y na levé straně. Pokud je Y nulovatelné, též přidej $(X \rightarrow \alpha\ Y \bullet \beta, j)$ do S(k).

Tedy tečka se přesune přes neterminál, pokud neterminál může derivovat prázdný symbol a tímto může „zmizet“, podrobněji v [11]. Seznam nulovatelných neterminálů je zvláště u deformačních gramatik velice jednoduchý, neboť se jedná o všechny neterminály.

Každý z popisovaných způsobů zajištění zpracování e-pravidel vede ke zvýšení časové a paměťové složitosti algoritmu. Proto pokud je možné, je výhodné se jejich zpracování vyhnout. U deformačních gramatik je zvláště výhodný způsob založený na myšlence nulovatelnosti. E-pravidla dokonce ani nemusí být v deformační gramatice vytvořena, stačí pouze, pokud jsou e-pravidla požadována, během operace Prediktoru přidávat stavy dle výše uvedeného předpisu.

5.5 PŘEDPOVĚĎ PREDIKCE A REDUKCE

Pokud se více zaměříme na činnost Prediktoru, zjistíme, že některé předpovězené položky jsou bez dalšího užitku. Pokud by se nám podařilo snížit počet těchto zbytečných předpovědí, dosáhli bychom zvýšení časové i paměťové efektivity analýzy. Jedním ze způsobů, jak efektivitu zlepšit, je vzít v úvahu následující symbol vstupního řetězce, pro což lze efektivně využít FIRST set [22, strana 27], [4]. Bouckaert, Pirotte a Snelling analyzovali různé varianty Earleyho parseru pro dva rozdílné režimy předpovědi [14] a ukázali, že předpověď Prediktoru redukuje počet položek o 20-50% nebo dokonce více na některých praktických gramatikách.

Druhým typem předpovědi je předpověď redukce, která redukuje počet kompletovaných položek pomocí množiny symbolů, jež mohou následovat za položkou. Efektivita této předpovědi se nedá jednoduše stanovit. Earley doporučuje předpověď redukce, ale nebere v úvahu obtížnost konstrukce a udržení množin předpovědi. Bouckaert, Pirotte and Snelling

[14] definitivně zavrhují předpověď redukce na základě toho, že může jednoduše zdvojnásobit počet položek, které je nutno uchovávat, ale počítají například $E \rightarrow *F\ [+\text{-}],1$ jako dvě položky. Závěrem lze říci, že zisk z předpovědi redukce není velký a její implementační náklady projevující se ve zvýšení časové a paměťové složitosti algoritmu, jež je významná, nevyváží úsporu vyplývající z redukce počtu kompletovaných položek. Například velmi dobře navržený Earley/CYK parser od Graham, Harrison a Ruzzo [23] nezahrnuje předpověď redukce.

5.6 VYUŽITÍ PŘEDPOVĚDI PRO ZEFEKTIVNĚNÍ ANALÝZY NEJEDNOZNAČNÉ GRAMATIKY PRO ROZPOZNÁVÁNÍ DEFORMOVANÝCH OBJEKTŮ

Předpověď predikce využívající FIRST množiny je u deformačních gramatik nepoužitelná, FIRST množiny vlastně udávají, jaké terminály je možno derivovat z jakých neterminálů. U deformačních gramatik je možné (již z podstaty deformační gramatiky) derivovat z neterminálu jakýkoliv terminál.

Předpověď redukce v popsané podobě je opět u deformačních gramatik nepoužitelná, množina symbolů, které mohou následovat za položkou, by byla vždy stejná a obsahovala by opět všechny terminály.

Ovšem i u deformačních gramatik lze nalézt způsoby, jak zmenšit počet generovaných položek. Pokud se zaměříme na jeden symbol vstupního řetězce, který odpovídá jednomu popisnému primitivu, je zcela zřejmé, že symbol následující za ním nemůže odpovídat primitivu opačnému, tedy například primitivum popisující úsečku vpravo nebude následováno primitivem popisujícím úsečku vlevo. Tento případ při popisu reálných objektů nemůže při vhodné volbě měřítka primitiv nastat a efektivitu analýzy tedy zvýšíme, pokud vyřadíme všechny položky vedoucí k danému opačnému primitivu na dalším místě. Nové položky do seznamu položek přináší operace predikce, tedy toto „filtrování“ bude umístěno v operaci predikce, kterou tedy nyní můžeme definovat následovně:

- Predikce: Pro každý stav v SListP(k) tvaru $(X \rightarrow \alpha \bullet Y\ \beta, j)$ (kde j je původní úroveň stavu), přidej $(Y \rightarrow \bullet\ \gamma, k)$ do *SListP(k)* pro každé pravidlo $Y \rightarrow \alpha$, kde α neobsahuje *NSign*, kde *NSign* je „opakem“ primitivy, kterou vyjadřuje aktuální vstupní symbol.

Operace S*can* pro *n-tý* symbol vstupního řetězce prochází seznam *SListP(n-1)* a hledá vhodné pravidlo tvaru $(X \rightarrow \alpha \bullet a\ \beta, j)$, kde a je onen *n-tý* symbol vstupního řetězce, tedy terminál. V deformační gramatice je pravidel vyhovujících tomuto zápisu více, liší se terminálem, pro který jsou. V *n-té* úrovni seznamu se použijí pouze ta, jejichž část za znakem Earlyho tečkové notace začíná $n+1$ symbolem vstupního řetězce. Je tedy zbytečné v operaci predikce generovat položky, které začínají jiným terminálním symbolem, než je následující symbol ve vstupním řetězci. Finální definice operace predikce je tedy následující:

- Predikce: Pro každý stav v *SListP(k)* tvaru $(X \rightarrow \alpha \bullet Y\ \beta, j)$ (kde j je původní úroveň stavu), přidej $(Y \rightarrow \bullet\ \gamma, k)$ do *SListP(k)* pro každé pravidlo $Y \rightarrow \alpha$, kde α neobsahuje *NSign*, a kde γ je neterminál nebo terminál = *NextSign*; *NSign* je „opakem“ primitivy, kterou vyjadřuje aktuální vstupní symbol, *NextSign* je $k+1$ symbol vstupního řetězce; $k \in 0..n-1$, kde n je délka vstupního řetězce.

5.7 OPTIMALIZACE GRAMATIKY PRO KONKRÉTNÍ VSTUPNÍ SLOVO

Earlyho parser ve své operaci predikce prochází přepisovací systém gramatiky a vyhledává pravidla určitých vlastností a podle nich do seznamu stavů vkládá položky, které se nemusí vždy využít. Při velkém počtu pravidel (typicky deformační gramatiky) významně narůstá režie při neustálém vyhledávání vhodné položky během provádění jednotlivých operací Earlyho parseru. Pokud zmenšíme počet pravidel gramatiky, zrychlí se prohledávání přepisovacího systému v operaci predikce a též se může zmenšit počet zbytečně vložených položek, což opět vede ke zrychlení prohledávání seznamu stavů, ústících v celkové zrychlení algoritmu.

Jedno konkrétní vstupní slovo nemusí obsahovat všechny terminální symboly gramatiky, a tedy se při jeho analýze nemusí využívat celý přepisovací systém gramatiky. Pokud tato pravidla nepotřebná pro analýzu daného vstupního slova z gramatiky odstraníme, dosáhneme z výše uvedených důvodů větší efektivity analýzy.

5.8 LR PARSERY

LR parser je parser pro bezkontextové gramatiky, který čte vstupní řetězec zleva doprava a produkuje nejpravější derivaci. Termín LR(k) parser je též často používán, k označuje počet *look ahead* symbolů, které se během parsovacího procesu využívají. Obvykle je $k = 1$ a je často vynecháváno. Bezkontextová gramatika je nazývána LR(k) gramatikou, pokud pro ni existuje LR(k) parser. LR parsery provádí *bottom-up* analýzu, protože se pokouší dedukovat počáteční symbol ze vstupního řetězce.

Množství programovacích jazyků lze popsat LR(1) gramatikou nebo gramatikou k ní blízkou, a proto jsou LR parsery často použity pro syntaktickou analýzu zdrojového kódu.

LR parsery mají řadu výhod:

- Množství programovacích jazyků může být analyzováno nějakou variantou LR parseru. Jedna důležitá výjimka je C++.
- LR parsery mohou být implementovány velmi efektivně.
- Ze všech parserů, které pracují se vstupem zleva doprava, LR parsery detekují syntaktické chyby (tedy že vstupní řetězec neodpovídá gramatice) tak brzo, jak je to jen možné.

LR parsery je obtížné generovat „ručně“, obvykle jsou generovány nějakým parser generátorem. V závislosti na způsobu generování tabulek parseru (*action table, goto table*) jsou tyto parsery nazývány Simple LR parser (SLR) [4], Look-ahead LR parser (LALR) [2], [4] a Canonical LR [4] parser. Typ parseru určuje množinu gramatik, kterou je schopen zpracovat. Oblíbený Yacc produkuje LALR parsery.

Největším problémem při vytváření parseru je konstrukce jeho tabulek, která je obecně velmi náročná. Tabulky se obvykle vytváří před analýzou nějakého vstupního řetězce, často za pomoci takzvaného parser generátoru. Díky takto předem vytvořeným tabulkám je vlastní analýza velice rychlá. Konstrukce tabulek LR(0) je popsána například v [1], další typy jsou naznačeny například v [4]. Tedy existuje jeden algoritmus pro syntaktickou analýzu, ale více algoritmů pro konstrukce tabulek parseru. Tabulky parseru musí být jednoznačné, tedy jedna

buňka tabulky nemůže obsahovat dvě akce. V případě, že obsahuje, hovoříme o takzvaném konfliktu v tabulce.

Použití parserů z rodiny LR pro analýzu deformační gramatiky je nemožné, protože nejednoznačná deformační gramatika produkuje v tabulkách parseru neřešitelné konflikty, které ani jediná další informace, kterou máme, váhy jednotlivých pravidel nemohou odstranit. Ovšem existují další parsery, které LR parser používají nebo z něj vycházejí a dokáží analyzovat nejednoznačnou deformační gramatiku. Proto byl zde věnován prostor jeho základnímu popisu.

5.9 TOMITA PARSER A JEHO MODIFIKACE

Z předchozí kapitoly je zřejmé, že existují gramatiky, které LR parsery nejsou schopné analyzovat. Ovšem Tomita [15] popsal dobře fungující a velmi efektivní metodu, pomocí které je možné analyzovat i takové gramatiky.

Tomitovu metodu lze jednoduše popsat jako prohledávání do šířky [4] v místech analýzy, které LR parsery nemohou zpracovat kvůli konfliktům. Jeho podrobnější popis lze nalézt například v [4]. Tomita parser je možno použít pro analýzu nejednoznačné deformační gramatiky, tedy umožňuje využít tabulky parseru rodiny LR, které se vyznačují svojí efektivitou. Nejdůležitější modifikací parseru je nutnost akumulovat váhy deformačních pravidel k získání výsledné vzdálenosti analyzovaného řetězce od obrazu dané třídy objektu. Tohoto dosáhneme založením proměnné, jež na začátku analýzy nastavíme na hodnotu 0 a při každé redukci přičteme váhu redukovaného pravidla. Lze též využít váhy jednotlivých pravidel možných v daném okamžiku pro redukci a vybrat pouze pravidlo / pravidla s nejmenší vahou.

5.10 MODIFIKOVANÝ HYBRIDNÍ LRE(K) ALGORITMUS

Další způsob, jak obejít konflikty v parsovacích tabulkách LR(k) parseru pro obecnou bezkontextovou gramatiku, spočívá ve využití podobnosti LR(k) a Earlyho parseru. LR tabulky jsou vlastně předem vypočítané veškeré operace, které Earlyho parser vykonává během své činnosti. Hybridní LRE(k) parser využívá informace z LR tabulek a vyhýbá se tedy výpočtu těchto informaci za běhu parseru. Popis LRE(k) parseru lze nalézt například v [20]. Hybridní LRE parser je schopen analyzovat obecnou bezkontextovou gramatiku a je tedy možné ho využít pro analýzu nejednoznačné deformační gramatiky. Nezbytnou modifikací parseru je nutnost akumulovat váhy deformačních pravidel k získání výsledné vzdálenosti analyzovaného řetězce od obrazu dané třídy objektu. Tohoto dosáhneme způsobem podobným jako u modifikovaného Tomita parseru. Založíme proměnnou, její hodnotu na začátku analýzy nastavíme na hodnotu 0 a při každé redukci přičteme váhu redukovaného pravidla. Připomeňme, že při analýze deformační gramatiky nás zajímá pouze úspěch / neúspěch a hodnota váhovací proměnné.

5.11 VYHODNOCENI POPISOVANÝCH PARSERŮ

Oba popsané algoritmy jsou mírně efektivnější a „šikovnější“ než Earlyho parser použitý v testovacím prostředí, jejich míru efektivity lze přesně stanovit jen obtížně. LRE(k) parser se jeví jako nejefektivnější parser z analyzované trojice parserů. Oproti Earlymu parseru

nemají problém se zpracováním e-pravidel (nutno použít tabulky parseru LR(1) nebo lepší). Do testovacího prostředí tyto algoritmy nebyly zahrnuty z důvodů potřeby vytvoření tabulek parseru k dané gramatice, což je značně náročné a použití nějakého parser generátoru je díky komplikaci v podobě vah pravidel deformační gramatiky téměř vyloučeno.

6 ZAJIŠTĚNÍ INVARIANCÍ U METOD SYNTAKTICKÉ ANALÝZY

Metody syntaktické analýzy jsou invariantní vůči translaci popisovaného objektu, další invariance již ale musíme zajistit dodatečně. Jedná se především o invarianci vůči rotaci a invarianci vůči volbě počátečního bodu popisu objektu.

Invariance vůči volbě počátečního bodu popisu zajistí rozpoznání objektu při jakékoliv volbě bodu hrany objektu jako počátečního bodu popisu. V úvahu připadají následující dva způsoby:

- rotace řetězce popisujícího objekt
- automatická volba bodu počátku popisu

Invariance vůči rotaci způsobí, že metoda zajistí rozpoznání předloženého objektu v jakémkoliv jeho natočení. Invariance vůči rotaci patří k nezbytným vlastnostem téměř každého algoritmu pro rozpoznávání objektů. Syntaktické metody nejsou standardně invariantní vůči rotaci, je tedy nutné tuto invarianci nějakým způsobem zajistit. V úvahu připadají následující dva způsoby:

- využití diferenciálních primitiv pro popis objektů
- rotace objektů

Způsoby (například rotace předkládaného objektu a rotace řetězce popisujícího rozpoznávaný objekt) běžně používané u klasických strukturálních metod jsou použitelné i při rozpoznávání deformovaných objektů pomocí deformačních gramatik, ale zvyšují časovou složitost analýzy, která je významně vyšší než u klasické syntaktické analýzy. Dochází též i ke komplikaci při vyhodnocování výsledků z důvodu velkého počtu výsledných hodnot. Patrně jediná nepoužitelná výjimka zde je normalizace předkládaných řetězců [31], která se z důvodu proměnné délky řetězce popisujícího deformovaný objekt nedá použít.

Je nutné zvolit způsob, který současně umožní jednoduše zajistit potřebné invariance a zároveň způsob, který co nejméně navýší časovou složitost při zachování jednoduchého vyhodnocování výsledků analýzy.

Dále někdy mohou být požadovány invariance vůči změně měřítka a invariance vůči stranovému převrácení.

6.1 ZAJIŠTĚNÍ INVARIANCE VŮČI VOLBĚ POČÁTEČNÍHO BODU POPISU

První způsob zajištění této invariance je pomocí rotace řetězce popisujícího objekt. Rotací řetězce je myšleno přesunutí posledního terminálního symbolu na začátek.

$$abcde \overset{1.rotace}{\rightarrow} eabcd \rightarrow \cdots$$

Jsou analyzovány všechny možnosti volby počátečního bodu a je vybrána ta nejlepší, což u deformačních gramatik je ta s nejmenší váhou. Toto řešení je jednoduché a spolehlivé. U klasických syntaktických metod je zcela postačující, ovšem při analýze deformační gramatiky, jejíž doba analýzy je znatelně vyšší, není nejvhodnější. Potřebný čas je úměrný *n* násobku doby analýzy, kde *n* je délka analyzovaného řetězce, tedy roste exponenciálně s délkou analyzovaného řetězce.

Druhý způsob spočívá, že řetězce popisující objekty jsou k analýze předkládány vždy se stejně zvoleným bodem počátku popisu, například bod objektu, který má maximální x a minimální y souřadnice (tedy bod objektu, který je nejvíce „vpravo nahoře“). Největší výhodou tohoto způsobu je jeho rychlost, protože celková doba analýzy je blízká době analýzy bez zajištění této invariance. Je pouze vyšší o dobu „režie“ algoritmu volby počátečního bodu popisu.

6.2 ZAJIŠTĚNÍ INVARIANCE VŮČI ROTACI

Invarianci vůči rotaci můžeme zajistit použitím diferenciálních primitiv [29] pro popis objektu. Popis objektu vytvořený pomocí diferenciálních primitiv je na rozdíl od „obyčejných“ primitiv (které jsou invariantní pouze vůči translaci) invariantní i vůči rotaci a tedy invariancí vůči rotaci se není třeba dále zabývat. Takto vytvořený popis má ovšem při použití pro rozpoznávání deformovaných objektů pomocí deformačních gramatik dvě nevýhody. Ačkoliv je vzniklý popis objektu pomocí diferenciálních primitiv stejně dlouhý jako popis pomocí „klasických“ primitiv, je ale obvykle jednodušší a pro jeho vytvoření je potřeba menší počet primitiv, například u popisu testovacího objektu „stylizovaný dům“:

- ddfffbbbcccaaccbbcccaaaggg, klasická primitiva, n_p=6
- aadaadaabaabacacabaabaadaa, diferenciální primitiva, n_p = 4

Dále jsou u diferenciálních primitiv vždy vyšší váhy, z důvodu samotného principu diferenciálních primitiv, viz část objektu na obr. 1.



Obr. 1a (vlevo) Nedeformovaná část objektu, obr. 1b (vpravo) deformovaná část objektu

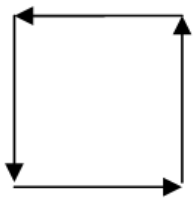
- popis nedeformované části objektu, obr. 1a: *cccc*
- popis deformované části objektu, obr. 1b pomocí klasických primitiv: *cehc*, hodnota vah popisující deformaci w = 2
- popis deformované části objektu, obr. 1b pomocí diferenciálních primitiv: *adcd*, hodnota vah popisující deformaci w = 3

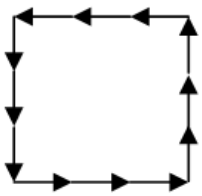
Vyšší hodnota vah je způsobena, že pro popsání deformace dvou primitiv je nezbytné změnit směr popisu třikrát, což ve spojení s jednodušším popisem ve výsledku znamená menší rozdíly ve váhách mezi objekty (větší podobnost popisu objektů) a tedy vyšší pravděpodobnost nesprávné klasifikace objektu. Ovšem tento způsob zvláště pak ve spojení s automatickou volbou počátečního bodu popisu nabízí výbornou rychlost rozpoznávání.

Pro zajištění maximální možné úspěšnosti klasifikace je tedy výhodnější použít způsob jiný, spočívající v rotaci rozpoznávaného objektu. V případě použití osmi - okolí pro popis objektu je možných variant rotace objektu osm, což lze ještě efektivně analyzovat.

6.3 INVARIANCE VŮČI ZMĚNĚ MĚŘÍTKA

Invariance vůči změně měřítka zajistí korektní rozpoznání objektů stejného tvaru, avšak různé velikosti. Tato invariance obvykle nebývá požadována a popis objektů pomocí primitiv obvykle nebere v potaz jakékoliv měřítko primitiv, je vytvořen pouze z celých hran daných směrů viz obr. 2a.





Obr. 2a (vlevo) Popis objektu pomocí primitiv bez zvoleného měřítka a obr. 2b (vpravo) se zvoleným měřítkem

Ovšem u analýzy deformačních gramatik je způsob neberoucí v potaz měřítko primitiv nevhodný. Velikost vlastních deformaci tvaru bývá oproti velikosti objektu nevýznamná. Proto při velmi malé délce slova popisujícího objekty dochází již při aplikaci několika málo deformačních pravidel ke značné deformaci objektu.

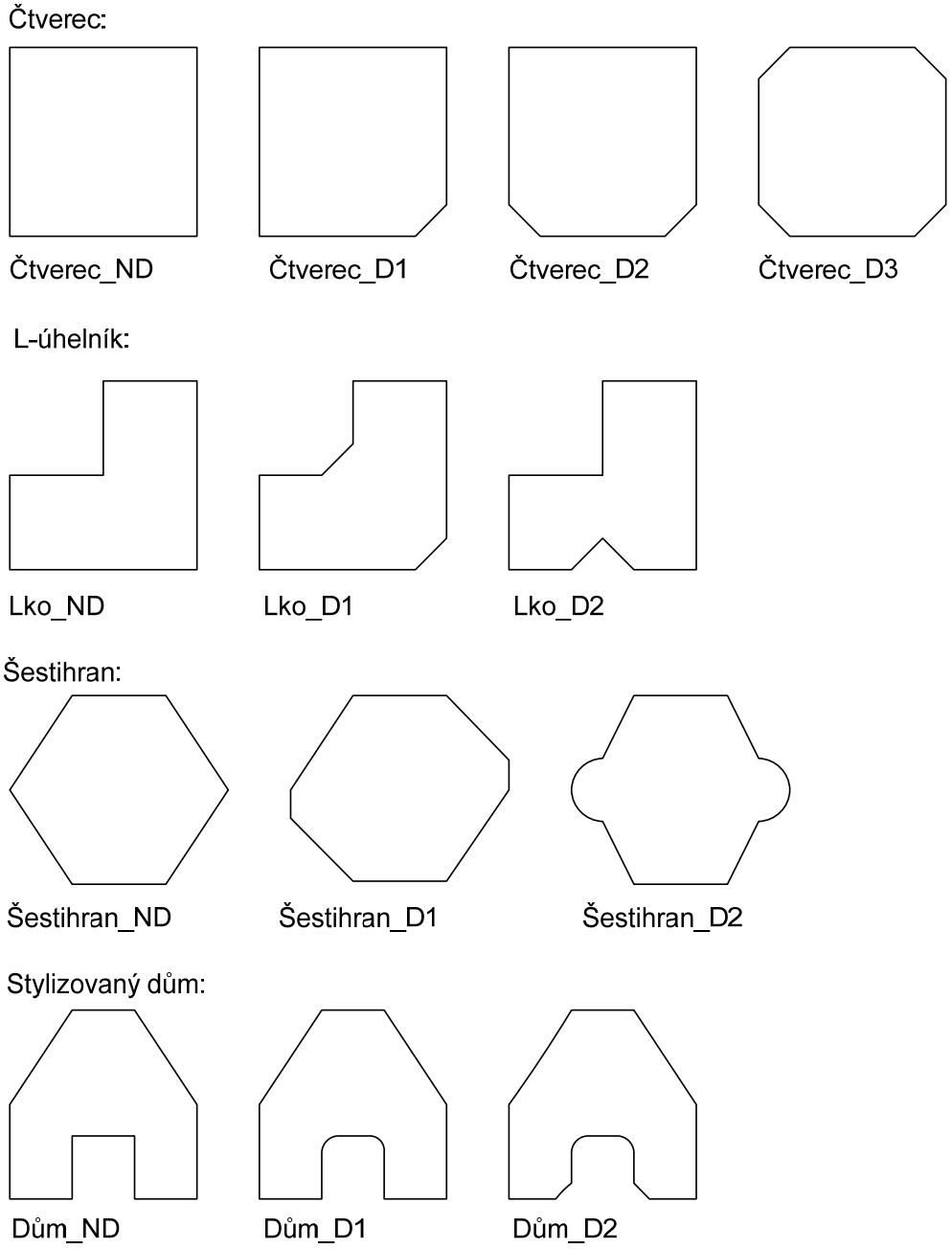
Při použití deformačních gramatik je tedy nutno dbát na výběr vhodné délky slova a měřítka primitiv popisujících rozpoznávaný objekt. Je nezbytné volit slova dostatečné délky vzhledem k velikosti a množství deformací. Při dostatečné délce slov tato metoda funguje korektně, nedochází k chybné identifikaci objektu. Tato metoda je též díky této skutečnosti vhodná pro rozpoznávání objektů v měřítku.

6.4 ZPRACOVÁNÍ VNITŘNÍCH HRAN OBJEKTU

Některé objekty mají též i svoji vnitřní hranu (například matice) a je tedy nezbytné, aby metoda pro jejich rozpoznání dokázala vnitřní hrany zpracovat. U syntaktických metod je tato úprava velice jednoduchá. Stačí do datové struktury popisující objekt vložit odkaz na popis vnitřní hrany, která se dále popíše a analyzuje stejně jako vnější hrana objektu. S popisem objektu se pracuje beze změn, včetně veškerého zajištění invariancí, pracuje se s oběma popisy hran společně. Dochází k úpravě vyhodnocování analýzy, do které je nutné zahrnout zpracování vnitřní hrany. Při analýze deformační gramatiky s vnitřními hranami taktéž nedochází k žádným změnám.

7 VÝSLEDKY

Všechny výsledky popisovaných algoritmů uvedených v této kapitole byly získány na konfiguraci 1.8GHz single core Athlon 64 s 3072MB RAM a operačním systémem Windows Server 2003 32bit. Pro zjištění úspěšnosti klasifikace objektů byla použita sada čtyř testovacích objektů, které se podobají dvourozměrným obrazům reálných předmětů a zároveň jsou vhodné pro generování několika jejich zdeformovaných variant. Záměrně byly zvoleny některé objekty tvarově blízké a především s podobnými deformovanými tvary. Ačkoliv je počet testovacích objektů malý, ke každému testovacímu objektu ovšem přísluší několik jeho deformovaných verzí, tudíž celkový počet objektů k otestování úspěšnosti rozpoznávání je více než dostatečný. Testovací objekty a jejich deformované varianty jsou na obr. 3.



Obr. 3 Testovací objekty

Pro testy vlivu optimalizace na rychlost algoritmu byly použity stejné testovací objekty, pouze o větší délce slova (jiného měřítka primitiv).

7.1 VÝSLEDKY ÚSPĚŠNOSTI ROZPOZNÁVÁNÍ OBJEKTŮ

Pro čtyři gramatiky generující testovací objekty z obr. 3 se postupně analyzují jednotlivé řetězce odpovídající testovacím objektům a jejich deformovaným variantám. Invariance vůči rotaci je v testovacím prostředí zajištěna pomocí rotace objektů a invariance vůči volbě bodu počátku popisu je zajištěna pomocí automatické volby bodu počátku popisu. Délka slov popisujících objekty je volena s přihlédnutím na velikost deformací objektů. Pro porovnání jsou pro objekt „čtverec_ND“ uvedeny výsledky pro analýzu bez i s e-pravidly. Výsledky rozpoznávání jsou uvedeny v tab. 1. Výsledky rychlosti algoritmů jsou pouze orientační, protože implementovaný algoritmus obsahuje množství ladících informací, které dobu analýzy znatelně prodlužují.

Objekt	Řetězec	Délka	E-pravidla	Rozpoznáno	Váha	Čas [s]
Čtverec						
Čtverec_ND	ddddddbbbbbbccccccaaaaaa	24	NE	ANO	0	1.281
Čtverec_ND	ddddddbbbbbbccccccaaaaaa	24	ANO	ANO	0	1.375
Čtverec_D1	ddddddbbbbbbcccccceeaaaaa	24	NE	ANO	2	1.281
Čtverec_D1	ddddddbbbbbbccccceaaaaa	23	ANO	ANO	2	1.390
Čtverec_D2	ddddddbbbbbhhccccceeaaaaa	24	NE	ANO	4	1.515
Čtverec_D2	ddddddbbbbbhccccceaaaaa	22	ANO	ANO	4	1.406
Čtverec_D3	ggddddffbbbbhhccccceeaaaa	24	NE	ANO	8	1.422
Čtverec_D3	ddddfbbbbhccccceaaaag	20	ANO	ANO	10	1.548
L-úhelník						
Lko_ND	dddbbbddddbbbccccccaaaaaa	24	NE	ANO	0	1.235
Lko_D1	dddbbffddbbbccccceeaaaaa	24	NE	ANO	4	1.218
Lko_D2	dddbbbddddbbbccehccaaaaaa	24	NE	ANO	2	1.265
Šestihran						
Šestihran_ND	dddfffhhhccceeeggg	18	NE	ANO	0	1.078
Šestihran_D1	fffbhhccceeeaggddd	18	NE	ANO	2	0.953
Šestihran_D2	fffiihhhccceeeiigggddd	22	NE	ANO	4	1.015
Stylizovaný dům						
Dům_ND	ddfffbbbcccaaccbbcccaaaggg	26	NE	ANO	0	1.469
Dům_D1	dddbbffddbbbccccceeaaaaa	26	NE	ANO	2	1.312
Dům_D1a	ddddddbbbbbbccccceeaaaaa	27	ANO	ANO	2	1.719
Dům_D2	ddfffbbbccceaajcjbbhcccaaaggg	29	NE	ANO	4	1.688

Tab. 1 Výsledky úspěšnosti rozpoznávání objektů

7.2 VÝSLEDKY OPTIMALIZACÍ EARLYHO PARSERU

Rychlost Earlyho parseru použitého pro analýzu deformační gramatiky je v celé řadě aplikací zabývající se rozpoznáváním klíčová. Jeho rychlost je příznivě ovlivněna popsanými optimalizacemi. Protože implementovaný algoritmus obsahuje množství ladících informací, které dobu analýzy znatelně prodlužují, jsou ve výsledcích uváděny počty Earlyho stavů, nejen doby analýzy. Invariance vůči volbě počátečního bodu popisu a invariance vůči rotaci nejsou pro jednoduchost a zkreslení těchto výsledků zahrnuty. Do celkových výsledků optimalizací Earlyho parseru uvedených v tab. 2 se promítá

optimalizace gramatiky pro konkrétní vstupní slovo (0-25% snížení počtu stavů), filtrování předpovědí nemožných stavů podle aktuálního vstupního symbolu v operaci predikce (až 60% snížení počtu stavů), zamezení generování zbytečných stavů v závislosti na následujícím vstupním symbolu (30% snížení počtu stavů).

	Bez optimalizace			S optimalizací			Optimalizace
Objekt	Čas[s]	Stavů C/P	C+P	Čas[s]	Stavů C/P	C+P	počtu stavů
Čtverec	0.282	3101/3020	6121	0.062	1541/850	2391	60.9%
Čtverec D1(w_D=2)	0.291	3101/3020	6121	0.063	1587/864	2451	60.0%
L-úhelník	0.250	3431/3130	6561	0.062	1881/941	2822	56.9%
L-úhelník D1(w_D=4)	0.266	3431/3130	6561	0.078	1897/962	2859	56.4%
Dům	0.453	5461/4428	9889	0.125	2917/1148	4065	58.9%
Dům D1(w_D=8)	0.469	5461/4428	9889	0.078	1762/921	2683	72.8%

Tab. 2 Celkové výsledky

Z tab. 2 je patrno, že jednotlivé optimalizace se navzájem ovlivňují a celkové výsledky jsou proto nižší. Běžně je dosahováno až 60% snížení počtu Earlyho stavů. Nižší počet stavů při analýze příznivě ovlivňuje rychlost analýzy. Dále doba analýzy závisí na délce analyzovaného slova, se kterou roste exponenciálně. Pro zvýraznění výsledků jsou volena velmi dlouhá slova ($n > 64$).

8 ZÁVĚR A ZHODNOCENÍ VÝSLEDKŮ DEFORMAČNÍCH GRAMATIK

Navržená deformační gramatika spolehlivě generuje všechny možné varianty deformace objektu, a tedy umožňuje použít některou z metod syntaktické analýzy i pro náhodné deformace objektů. Deformační gramatiky mají potenciál rozpoznat jakýkoliv deformovaný objekt, záleží pouze na přesnosti jeho popisu. Oproti metodám pro zjištění vzdálenosti mezi atributovými popisy obrazů je výpočtově (tedy i časově) náročnější, její efektivita je závislá na použitém parseru a efektivitě jeho implementace, která je významně složitější než implementace metod pro zjištění vzdálenosti mezi atributovými popisy obrazů, které pouze využívají „nějakou metriku“. Deformační gramatiky využívají veškeré dostupné znalosti o cílech klasifikace, tedy vlastní gramatiku popisující objekt a tím jsou oproti metodám pro zjištění vzdálenosti mezi atributovými popisy obrazů informovanější. Na rozdíl od nich při správném použití neprodukují chybně rozpoznané objekty, což je největší výhoda této metody.

Při použití je pouze nutno dbát na výběr vhodné délky slova popisujícího rozpoznávaný objekt. Deformační gramatiky mají potenciál objekt rozpoznat dokonale, ovšem taková analýza by byla značně pomalá. Proto je vždy potřeba zvolit vhodný kompromis mezi rychlostí a přesností popisu (měřítko primitiv a tedy následně délka řetězce) rozpoznávaných objektů, zajištující korektní rozpoznání objektů. Z těchto důvodů je tato metoda zvláště vhodná pro rozpoznávání objektů v měřítku.

Earlyho parser byl pro analýzu deformační gramatiky zvolen, protože nepotřebuje parsovací tabulky, které je obtížné vytvořit. Patrně nejefektivnější z popsaných parserů je LRE(k) parser, který vykazuje potenciál zrychlit analýzu o více než polovinu.

Velký vliv na celkovou rychlost této metody rozpoznávání mají způsoby zajištění invariancí. Byly testovány dva způsoby zajištění invariancí, první za pomocí diferenciálních primitiv pro zajištění invariance vůči rotaci a rotace řetězce pro zajištění invariance vůči volbě bodu počátku popisu. Druhý způsob kombinoval rotaci objektu pro zajištění invariance vůči rotaci a automatickou volbu bodu počátku popisu pro zajištění invariance vůči volbě bodu počátku popisu.

Z výsledků je patrno, že ačkoliv diferenciální primitiva jsou nejrychlejším způsobem pro zajištění invariance vůči rotaci, mohou někdy nepřímo vést k chybné identifikaci objektu, zapříčiněné jednodušším strukturálním popisem objektu a menšími rozdíly v součtu vah deformačních pravidel. Druhý popsaný způsob rotace objektů je použitelný pouze u jednoduchých primitiv (typicky osmiokolí), u složitějších primitiv s více možnostmi natočení dochází k přílišnému zvýšení časové složitosti výpočtu. Tento způsob též zachovává maximální možnou míru úspěšnosti klasifikace objektů.

Rotace řetězce pro zajištění invariance vůči volbě bodu počátku popisu je velice jednoduchý způsob, který ale u dlouhých řetězců významně zvyšuje celkovou časovou složitost analýzy, protože syntaktická analýza se musí pro rozpoznání jednoho objektu provést tolikrát, kolik znaků je délka řetězce. Z tohoto důvodu je automatická volba bodu počátku popisu, kde stačí provést syntaktickou analýzu řetězce pouze jednou, značně efektivnější a přitom vlastní normalizace řetězce je jednoduchá a rychlá.

Invarianci vůči změně měřítka objektu je možné zajistit pomocí volby měřítka (délky) primitiv a zápisu vlastní gramatiky popisující daný objekt. U rozpoznávání deformovaných objektů pomocí deformačních gramatik je k zajištění správné klasifikace objektů tato vhodná volba měřítka primitiv zcela nezbytná.

Přínos práce spočívá ve využití výsledků při řešení konkrétních aplikací. Nejvhodnější způsob zajištění invariancí záleží na konkrétních požadavcích řešené aplikace, ale obecně nejlépe se jeví kombinace automatické volby bodu počátku popisu pro zajištění invariance vůči volbě bodu počátku popisu a rotace objektu pro zajištění invariance vůči rotaci.

Ze zjištěných výsledků je zcela zřejmé, že použití deformačních gramatik k rozpoznávání náhodně deformovaných objektů je možné.

9 LITERATURA

[1] Aho, Sethi, Ullman, Addison-Wesley: Compilers: Principles, Techniques, and Tools, 1986. ISBN 0-201-10088-6.

[2] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. Compilers: Principles, Techniques, and Tools Addison--Wesley, 1986.

[3] Deza, E.; Deza, M. (2006), Dictionary of Distances, Elsevier, ISBN 0444520872.

[4] Dick Grune, Ceriel J.H. Jacobs: Parsing Techniques, VU University Amsterdam, Amsterdam, The Netherlands. Originally published by Ellis Horwood, Chichester, England, 1990; ISBN-10: 038720248X, ISBN-13: 978-0387202488, dostupné z ftp://ftp.cs.vu.nl/pub/dick/PTAPG_1st_Edition/BookBody.pdf.

[5] Earley, J. (1968) An Efficient Context-Free Parsing Algorithm. PhD Thesis, Carnegie-Mellon University.

[6] Earley, J. (1970) An efficient context-free parsing algorithm. Commun. ACM, 13, 94–102.

[7] Gonzalez, Rafael C. & Woods, Richard E.: Thresholding. In Digital Image Processing, pp. 595–611. Pearson Education. ISBN 81-7808-629-8, 2002.

[8] Hlaváč, V., Šonka, M.: Počítačové vidění. Grada, Praha, 1993.

[9] Hopcroft, J. E., Ullman, J. D.: Formálne jazyky a automaty. Alfa, Bratislava, 1978.

[10] Chapman, S.: String Metrics, University of Sheffield, 2005. dostupné z: http://www.dcs.shef.ac.uk/~sam/stringmetrics.html.

[11] John Aycock, R. Nigel Horspool, „Practical Earley Parsing“, The Computer Journal, Vol. 45, No. 6, 2002, British Computer Society 2002, Pages 620-630.

[12] Levenshtein.net, dostupné z: http://www.levenshtein.net/.

[13] Luong Chi Mai, Introduction to Computer Vision and Image Processing E-book, Department of Pattern Recognition and Knowledge Engineering, Institute of Information Technology, Hanoi, Vietnam, dostupné z:

http://eisc.univalle.edu.co/materias/Procesamiento_De_Imagenes_Digitales/Libro%20de%20Ayuda/Introduction%20to%20Image%20Processing%20and%20Computer%20Vision.pdf.

[14] M. Bouckaert, A. Pirotte, M. Snelling, "Efficient parsing algorithms for general context-free parsers", Inform. Sci., vol. 8, no. 1, p. 1-26, Jan 1975.

[15] Masaru Tomita, Efficient parsing for natural language, Kluwer Academic Publishers, Boston, p. 201, 1986.

[16] Minařík, M., Šťastný, J., Popelka, O.: A Brief Introduction to Recognition of Deformed Objects, International Conference on Soft Computing Applied in Computer and Economic Environment ICSC, Kunovice, 2008, pp.55-64, ISBN 978-80-7314-134-9.

[17] Minařík, M., Šťastný, J.: Recognition Of Randomly Deformed Objects, 14th International Conference on Soft Computing, Mendel 2008, June 18-20, Brno, Czech Republic, pp.275-280, ISBN 978-80-214-3675-6.

[18] Minařík, M.: Strukturální metody identifikace objektů, FSI Junior konference 2006, leden 2007, Brno. s. 78-89. ISBN: 80-214-3364-9.

[19] Moore, Robert C.: Removing Left Recursion from Context-Free Grammars. 6th Applied Natural Language Processing Conference: 249-255. May 2000, dostupné z http://acl.ldc.upenn.edu/A/A00/A00-2033.pdf.

[20] Philippe McLean & R. Nigel Horspool, A Faster Earley Parser, Dept. of Computer Science, University of Victoria, Victoria, BC, Canada V8W 3P6, dostupné z: http://webhome.cs.uvic.ca/~nigelh/Publications/fastEarley.pdf.

[21] Richard W. Hamming. Error Detecting and Error Correcting Codes, Bell System Technical Journal 26(2):147-160, 1950.

[22] Rybička, J.: Úvod do teorie formálních jazyků, učební pomůcka, nakladatelství Konvoj, Brno 1999.

[23] S.L. Graham, M.A. Harrison, W.L. Ruzzo, "An improved context-free recognizer", ACM Trans. Prog. Lang. Syst., vol. 2, no. 3, p. 415-462, July 1980.

[24] Seifert, S.: Ein Earley-Parser fur Zeichenketten Erzeugende Hypergraphgrammatiken, Studienarbeit im Fach Informatik, 2004, dostupné z http://www2.informatik.uni-erlangen.de/Lehre/SA-DA/download/SA-seifert.pdf.

[25] Sipser, Michael: Introduction to the Theory of Computation (1st ed.), IPS, p. 99, ISBN 0-534-94728-X, 1997.

[26] Šťastný, J., Minařík, M. Object Recognition by Means of New Algorithms. ICSC - International Conference on Soft Computing Applied in Computer and Economic Enviroment. Kunovice: Evropský polytechnický institut, Kunovice, 2006. s. 99-104. ISBN: 80-7314-084-5.

[27] Šťastný, J., Škorpil, V.: Analysis of Methods for Edge Detection. International journal Communications, 2003, ISSN 0018-2028.

[28] Šťastný, J., Škorpil, V.: Filtering Methods for Picture Processing, International Conference TELECOMMUNICATIONS AND SIGNAL PROCESSING- TSP 99, Brno, 1999.

[29] Šťastný, J.: Netradiční metody a algoritmy pro rozpoznávání objektů technologické scény, zkrácená verze habilitační práce FSI VUT, Brno, 2005, ISBN 80-214-3117-2.

[30] Šťastný, J.; Minařík, M. A Brief Introduction to Image Pre-Processing for Object Recognition. ICSC - International Conference on Soft Computing Applied in Computer and Economic Enviroment. Kunovice: Evropský polytechnický institut, Kunovice, 2007. s. 55-64. ISBN: 80-7314-084-5.

[31] Železný, M.: Strukturální metody rozpoznávání – přednášky, katedra kybernetiky Fakulty aplikovaných věd, Západočeská univerzita v Plzni, dostupné z: http://artin.zcu.cz/courses/smr/Smr-031021.pdf.

[32] Železný, M.: Zpracování digitalizovaného obrazu – přednášky, katedra kybernetiky Fakulty aplikovaných věd, Západočeská univerzita v Plzni, dostupné z: http://artin.zcu.cz/courses/zdo/ZDO_aktual_060217.pdf.

CURRICULUM VITAE

Základní údaje:

- Jméno, příjmení: ing. Martin Minařík
- Adresa trvalého bydliště: Krausova 11, Brno - Černovice 618 00
- Mobilní telefon: 608 517 464, e – mail: shaadow@volny.cz
- Datum narození : 16.06.1981, věk: 28 let
- Národnost : česká, stav : svobodný

Vzdělání:

- 1987 – 1995 Základní škola Kneslova 28, Brno
- 1995 – 1999 Gymnázium přírodovědného zaměření Taborská 185, Brno, zakončeno maturitou
- 1999 – 2000 VUT FEI Technická 8, Brno, po roce přestup na VUT FSI
- 2000 – 2005 VUT Brno, Fakulta strojního inženýrství, Technická 2, Brno, studijní obor Inženýrská informatika a automatizace, studium řádně zakončeno statní závěrečnou zkouškou, téma diplomové práce „Výrobní informační systém“
- 2005 – doposud VUT Brno, Fakulta strojního inženýrství, Technická 2, Brno, doktorský studijní program, téma dizertační práce „Strukturální metody identifikace objektů pro řízení průmyslového robotu“

Znalost jazyků:

- NJ pasivně, základní znalost
- AJ aktivně, pokročile včetně příslušné odborné terminologie

Pedagogická činnost:

- V rámci doktorského studia výuka předmětů Informatika 1, Zpracování informací, Simulace systémů, Programovací techniky

Vlastnosti:

- schopnost práce v teamu
- schopnost rozvíjet výsledky druhých
- nekonfliktnost
- časová flexibilita
- schopnost a ochota učit se novým věcem

Zájmy:

- hudba
- sport (MTB, vodní sporty, zimní sporty, horolezectví)

V Brně dne 22.08.2009

ABSTRAKT

Tato dizertační práce se zabývá využitím strukturálních metod identifikace objektů pro řízení průmyslového robotu.

Nejprve je popsán současný stav znalostí v dané oblasti, tedy celý proces rozpoznávání objektů pomocí metod klasické syntaktické analýzy. Největší nevýhodou je nemožnost rozpoznávat objekty, jejichž digitalizovaný obraz je nějakým způsobem porušen či zkreslen (díky nadměrnému šumu nebo poruchám obrazu), tedy deformován. Pro rozpoznávání deformovaných objektů jsou dále popsány metody využívající strukturálního popisu pro jejich rozpoznávání – metody pro stanovení vzdáleností mezi atributovými popisy obrazů.

Vlastní jádro celé práce spočívá až v kapitole 5, kde jsou popsány deformační gramatiky, jež jsou schopny popsat všechny možné deformace objektu. Komplikací při jejich analýze je nejednoznačnost deformační gramatiky, která snižuje efektivitu analýzy.

Dále je věnován prostor výběru a modifikaci vhodného parseru schopného efektivně analyzovat deformační gramatiku. Popsány jsou tři typy parseru: modifikovaný Earlyho parser, modifikovaný Tomita parser a modifikovaný hybridní LRE(k) parser. Pro Earlyho parser je popsán efektivní způsob jeho implementace.

Nezbytnou součástí rozpoznávání objektů je zajištění invariancí, čímž se tato práce též detailně zabývá. Závěrem jsou uvedeny výsledky popsaných algoritmů (úspěšnost a rychlost rozpoznávání deformovaných objektů) a je popsáno navržené testovací prostředí a algoritmy v něm implementované. V závěru jsou shrnuty zjištěné možnosti deformačních gramatik a jejich výsledky.

ABSTRACT

This PhD thesis deals with the use of structural methods of objects identification for industrial robots operation.

First, the present state of knowledge in the field is described, i.e. the whole process of objects recognition with the aid of common methods of the syntactic analysis. The main disadvantage of these methods is that is impossible to recognize objects whose digitalized image is corrupted in some ways (due to excessive noise or image disturbances), objects are therefore deformed. Further, other methods for the recognition of deformed objects are described. These methods use structural description of objects for object recognition, i.e. methods which determine the distance between attribute descriptions of images.

The core part of this PhD thesis begins in Chapter 5, where deformation grammars, capable of description of all possible object deformations, are described. The only complication in the analysis is the ambiguity of the deformation grammar, which lowers the effectiveness of the analysis.

Further, PhD thesis deals with the selection and modification of a proper parser, which is able to analyze a deformation grammar effectively. Three parsers are described: the modified Earley parser, the modified Tomita parser and the modified hybrid LRE(k) parser. As for the modified Earley's parser, ways of its effective implementation are described.

One of the necessary parts of the object recognition is providing the invariances, which this PhD thesis covers in detail, too. Finally, the results of described algorithms are mentioned (successfulness and speed of deformed objects recognition) and suggested testing environment and implemented algorithms are described. In conclusion, all determined possibilities of deformation grammars and their results are summarized.