Kapitola 5

Metody rozpoznávání

Metody rozpoznávání představují jednu ze základních oblastí umělé inteligence. Z hlediska historie a svého rozvoje patří k nejstarším a možno říci též nejpropracovanějším. Je tomu tak díky možnosti jejich bezprostředního využití v praxi; bez počítačového zpracování fotografických snímků si dnes umíme jen těžko představit řešení úloh dálkového průzkumu Země, zpracování leteckých a družicových snímků, automatickou identifikaci podle otisků prstů, zpracování snímků z rentgenové či izotopové defektoskopie apod. Také oblast optického snímání dokladů, čtení identifikačních údajů z visaček kupovaného zboží nebo hlasové komunikace s počítačem je řešitelná prakticky jen při využití metod rozpoznávání.

5.1 Základní pojmy

Rozpoznávání je základní složkou *vnímání prostředí*. V tomto skriptu je budeme chápat jako třídění, resp. klasifikaci, tj. zařazování objektů do tříd podle jejich společných nebo podobných vlastností. Na objektech je nejprve třeba definovat hledisko, ze kterého objekt zkoumáme. Protože česká terminologie pro problémovou oblast rozpoznávání není jednotná, vysvětlíme nejprve význam pojmů používaných v dalším textu.

Pod pojmem *objekt* rozumíme tu část reálného světa, kterou zařazujeme do tříd. Pojmem *obraz* pak budeme označovat symbolický popis objektu, jímž budeme objekt zařazovat do příslušné klasifikační třídy. V souvislosti se zpracováním vizuální informace, která vznikne sejmutím množiny trojrozměrných objektů nazývané *scénou*, pak hovoříme o (vizuálním) *snímku*, jehož formalizací (symbolickým popisem) dostaneme *obraz scény*. Představíme-li si vytvořené symbolické popisy reprezentující jednotlivé rozpoznávané objekty jako body v obecně n-rozměrném prostoru, pak tento prostor nazveme *obrazovým prostorem* a části tohoto prostoru představující třídy, do nichž jsou rozpoznávané objekty zařazovány, vzniknou rozkladem obrazového prostoru na podprostory. Počet podprostorů obrazového prostoru odpovídá počtu tříd.

Zařazujeme-li objekty do předem definované (pevné) množiny klasifikačních tříd, hovoříme o *klasifikaci* objektů, zatímco v případě, kdy množina tříd, do nichž objekty zařazujeme, není předem definována, nýbrž k rozkladu obrazového prostoru dochází teprve v průběhu třídění objektů, hovoříme o *rozpoznávání* objektů.

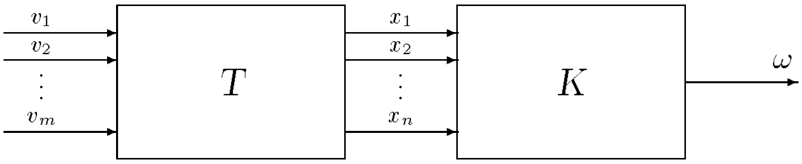
Podle použitého symbolického popisu, postupu jeho vytváření a vyhodnocování rozdělujeme metody rozpoznávání do dvou velkých skupin. Jedna skupina popisuje objekty souborem číselně vyjádřených charakteristických vlastností, obvykle metrické povahy. Vektor číselných hodnot charakterizující objekt se nazývá *vektor příznaků*; jeho složkami jsou jednotlivé *příznaky*. Vektory příznaků leží v *příznakovém prostoru*. *Příznakové metody* jsou velmi obecné a málo závislé na aplikační oblasti. Budeme se jimi zabývat v první části kapitoly.

Příznakové metody není vhodné použít tam, kde nejdůležitějším nositelem informace o třídě jsou strukturní vlastnosti objektů. Převedením úlohy do příznakového prostoru se totiž strukturní charakteristiky ztrácejí a je obtížné je obnovit. V takových úlohách je výhodnější popsat objekty elementárními prvky (složkami) vyjadřujícími strukturu objektu – tzv. *primitivy* – a relacemi mezi nimi. Každý rozpoznávaný objekt je potom popsán relační strukturou, a pokud volba relací odpovídá přirozené skladbě objektu, jsou substruktury popisu objektu popisem jeho významových částí. Úlohu rozpoznávání potom chápeme jako hledání vhodných morfismů mezi strukturou popisující objekt a strukturou příslušející dané klasifikační třídě. Tato skupina metod rozpoznávání, které je věnována druhá část kapitoly, je proto nazývána *metodami strukturními*.

Metody rozpoznávání, které jsou založeny na obou principech, tj. zčásti využívají popis příznakový a zčásti popis strukturní, nazýváme dnes *metodami hybridními* a budeme se jim věnovat v závěru kapitoly.

5.2 Obecná klasifikační úloha

Principiálně lze úlohu klasifikace či rozpoznávání znázornit hrubým blokovým schématem podle obr. 5. 1:



T . . . transformace vstupních charakteristik – vytvoření obrazu
K . . . klasifikátor
$\mathbf{v}$ . . . vektor vstupních charakteristik
$\mathbf{x}$ . . . obraz (symbolický popis) objektu
ω . . . indikátor třídy

Obr. 5. 1: Principiální schéma systému rozpoznávání

Vektor charakteristických veličin, resp. vlastností, které lze na objektu identifikovat, – budeme je nazývat *vstupními charakteristikami* – nechť má dimenzi m a označíme ho

$$\mathbf{v} \;=\; [\,v_1, v_2,\; .\;.\;.\; , v_m]^T \;.$$

Vstupní veličiny charakterizující klasifikovaný či rozpoznávaný objekt jsou podrobeny transformaci T, jíž je vytvořen symbolický popis objektu o dimenzi $n \leq m$ (výsledkem transformace T je obraz objektu), který označíme

$$\mathbf{x} \;=\; [\,x_1, x_2,\; .\;.\;.\; , x_n]^T \;.$$

Obraz objektu $\mathbf{x}$ přivádíme na vstup *klasifikátoru* K, který zařazuje objekty charakterizované symbolickým popisem (obrazem) $\mathbf{x}$ do příslušných klasifikačních tříd na základě definovaného *rozhodovacího pravidla*. Výstupní veličinu klasifikátoru K nazýváme *indikátorem třídy*, do níž je klasifikovaný (rozpoznávaný) objekt klasifikátorem zařazen, a označíme ji ω. Rozhodovací pravidlo pak můžeme obecně definovat jako skalární funkci vektorového argumentu

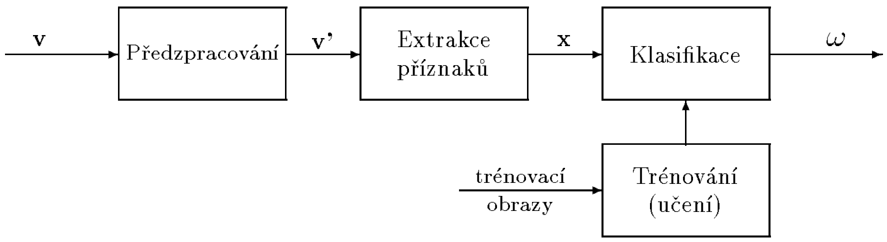
$$\omega \;=\; d\,(\,\mathbf{x}\,) \quad .$$

Pro jednotlivé typy metod rozpoznávání získává výše uvedené obecné rozhodovací pravidlo konkrétnější podobu a jeho odvození bude předmětem následujících odstavců.

5.3 Příznakové metody rozpoznávání

5.3.1 Podstata příznakových metod rozpoznávání

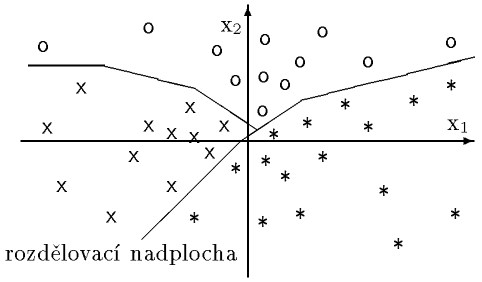
Objekt, který má být aplikací některé metody úspěšně rozpoznán, tj. zařazen do příslušné klasifikační třídy, musí být nejprve vhodným způsobem popsán, tj. musí být vytvořen symbolický popis objektu (obraz) odrážející v sobě podstatné vlastnosti objektu a splňující obecně platné požadavky kladené na *formální popisy*. Příznakové metody proto vytvářejí v prvé fázi zpracování informací o rozpoznávaném objektu ze zjištěných charakteristik objektu (obvykle naměřených hodnot numerické povahy) *vektor příznaků*. Jen v nejjednodušších případech jsou složky vektoru příznaků tvořeny přímo hodnotami měřených veličin; naměřené hodnoty jsou obvykle podrobeny nějaké transformaci, jejímž cílem je *ohodnocení významnosti* příznaků a jejich seřazení podle významnosti, jejich zobrazení do prostoru s předem přesně definovanými vlastnostmi v němž je následně provedena klasifikace a případně také snížení dimenze vektoru příznaků. Základem úspěšné aplikace příznakových metod je volba takového počtu nejvýznamnějších příznaků, který zabezpečí vysokou spolehlivost klasifikace a přitom bude dosaženo co nejnižší výpočetní složitosti metody.



Obr. 5.2: Architektura "příznakového" systému rozpoznávání

Obraz klasifikovaného (rozpoznávaného) objektu je tedy v případě příznakových metod tvořen sloupcovým vektorem příznaků $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$, jehož složkami jsou jednotlivé příznaky. Vektor příznaků je produktem speciálního bloku systému rozpoznávání označeného na obr. 5.2 jako blok "extrakce příznaků", kterým jsou obvykle s využitím značně komplikovaných transformačních algoritmů z vektoru charakteristik objektu $\mathbf{v}$ (zpravidla podrobeného předběžné transformaci nazývané *předzpracováním* vektoru charakteristických veličin) vybrány ty příznaky, které pro následující rozhodovací proces (blok "klasifikace") přinášejí o klasifikovaném objektu nejvíce informace; říkáme, že jsou nejvíce *informativní*. Množinu všech obrazů objektů $\mathbf{x}$ označíme $\mathcal{X}$ a budeme ji nazývat *obrazovým prostorem* nebo též *příznakovým prostorem* dané klasifikační úlohy, protože – jak už bylo zmíněno – obrazy jednotlivých objektů jsou zde tvořeny vektory příznaků.

Při vhodném výběru příznaků je podobnost objektů v každé třídě vyjádřena geometrickou blízkostí jejich obrazů v obrazovém prostoru. Jednotlivým třídám odpovídají shluky obrazů, které lze v dvourozměrném případě oddělit vhodnou křivkou, u vícerozměrných obrazů pak nadplochou nazývanou *rozdělovací nadplocha* – viz obr. 5. 3.



o . . . obrazy třídy X_1
x . . . obrazy třídy X_2
$*$ . . . obrazy třídy X_3

Obr. 5. 3: Geometrická reprezentace dvourozměrného obrazového prostoru

Existuje-li pro určitou klasifikační úlohu taková rozdělovací nadplocha, že v každé z oblastí $X_1, X_2, \ldots, X_R$ reprezentujících jednotlivé klasifikační třídy (R je počet tříd), do nichž jsou objekty zařazovány, leží jako na obr. 5. 3 jen obrazy jediné třídy, říkáme, že jde o úlohu s *oddělitelnými (separovatelnými) množinami obrazů* , jejíž klasifikační třídy jsou navzájem disjunktní, čili pro ně platí

$$\begin{aligned} \mathcal{X} &= X_1 \cup X_2 \cup \ldots \cup X_R \quad \text{a} \\ X_r \cap X_s &= \emptyset , \quad r, s = 1, \ldots, R, \quad r \neq s \quad . \end{aligned}$$

Klasifikačním třídám X_i , $i = 1, \ldots, R$ přiřadíme příslušné indikátory tříd ω_i , které jsou prvky množiny indikátorů tříd označované

$$\Omega = \{\omega_1, \omega_2, \ldots, \omega_R\} \; .$$

Je-li možno všechny třídy navzájem oddělit částmi nadrovin, hovoříme o *lineární oddělitelnosti*. U úloh s oddělitelnými množinami obrazů reprezentuje každý obraz výhradně objekty jen "své" klasifikační třídy. Máme-li informaci o této skutečnosti, nazýváme ji à priorní informací o oddělitelnosti. Intuitivně lze očekávat, že u úloh s oddělitelnými množinami obrazů lze dosáhnout bezchybné klasifikace.

Převážná část praktických úloh rozpoznávání však nemá oddělitelné množiny obrazů. V takovém případě umístění rozdělovací nadplochy v obrazovém prostoru představuje nějaký kompromis. Třídy X_i , $i = 1, 2, ..., R$ nelze vymezit tak, aby v každé z nich byly výlučně jen obrazy ze stejné třídy, a proto část objektů bude v procesu rozpoznávání vždy chybně zařazena. Takovou úlohu chápeme tak, že rozpoznávané objekty jsou do příslušných tříd klasifikovány jen s určitou pravděpodobností $p(\omega_i)$, resp. že zařazení objektu do třídy je s pravděpodobností $1 - p(\omega_i)$ chybné. Rozhodovací pravidlo pak dostává pravděpodobnostní charakter a jelikož při rozhodování využíváme podmíněných pravděpodobností, klasifikátor pracující na tomto principu se nazývá *Bayesovým klasifikátorem* nebo někdy též *klasifikátorem s minimální chybou* , protože pravděpodobnost zařazení objektu do chybné třídy se snažíme minimalizovat [Kotek93].

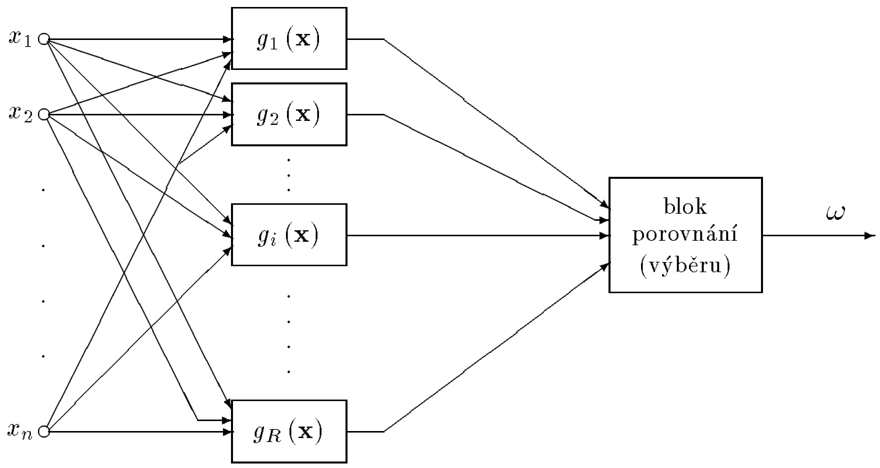
5.3.2 Návrh a nastavení klasifikátoru

Navrhnout klasifikátor příznakové metody znamená určit rozdělovací nadplochy mezi jednotlivými klasifikačními třídami (viz obr. 5.3). K tomu je využíváno *diskriminačních funkcí* $g_1(\mathbf{x})$, $g_2(\mathbf{x})$, . . . , $g_R(\mathbf{x})$, které se vybírají tak aby pro všechny obrazy $\mathbf{x} \in X_r$ platilo např.

$$g_r(\mathbf{x}) > g_s(\mathbf{x}) \; ; \quad s = 1, 2, \ldots, R \; ; \quad s \neq r \quad .$$

Potom je rozdělovací nadplocha mezi sousedními podprostory X_r a X_s určena rovnicí

$$g_r(\mathbf{x}) - g_s(\mathbf{x}) = 0 \quad .$$



Obr. 5.4: Struktura klasifikátoru na bázi diskriminačních funkcí

Pro stanovení rozdělovacích nadploch v obrazovém prostoru se používá např. lineárních nebo nelineárních diskriminačních funkcí, kritéria minimální vzdálenosti, kritéria minimální chyby, parametrických či neparametrických metod odhadů ap. Jejich popis lze nalézt např. v [Kotek90], [Kotek93] a dalších.

Strukturu klasifikátoru navrženého na bázi diskriminačních funkcí ukazuje obr. 5.4. Obraz klasifikovaného objektu $\mathbf{x}$ je "přiváděn" paralelně na vstupy R bloků, jimiž jsou vyčíslovány hodnoty jednotlivých diskriminačních funkcí $g_s(\mathbf{x})$, $s = 1, 2, \ldots, R$. Funkční hodnoty jsou pak porovnávány porovnávacím blokem (pro výše uvedený příklad např. blokem výběru maxima), na jehož výstupu se objeví indikátor ω_r r-té třídy, jejíž diskriminační funkce $g_r(\mathbf{x})$ nabývá pro přivedený obraz $\mathbf{x}$ nejlepší (např. maximální) hodnotu.

Klasifikátor se strukturou podle obr. 5.4 pak zařazuje objekty do příslušných klasifikačních tříd na základě rozhodovacího pravidla, které má tvar skalární funkce dvou vektorových argumentů

$$\omega = d(\mathbf{x}, \mathbf{q}) \quad ,$$

kde vektor $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ je obraz klasifikovaného objektu a vektor $\mathbf{q} = [q_1, q_2, \ldots, q_n]^T$ reprezentuje *parametry nastavení klasifikátoru*. Jeho chování je deterministické, tzn. týž obraz $\mathbf{x}$ bude vždy klasifikován do stejné třídy. Kvalita rozhodování je však závislá na jeho nastavení, resp. na hodnotách složek vektoru $\mathbf{q}$. Uvažujme, že pro konkrétní aplikaci lze nastavit libovolné rozhodovací pravidlo z nějaké množiny pravidel $\mathbf{D}$. Množinu $\mathbf{D}$ uspořádáme parametrem $\mathbf{q}$ a mimo to definujeme *ztrátovou funkci*, jíž "oceníme" případ zařazení klasifikovaného obrazu do nesprávné třídy. Jelikož proces zařazování obrazů do tříd se mnohokrát opakuje, můžeme pro dostatečně mohutnou množinu pokusů vyčíslit *střední ztrátu* $\mathbf{J}(\mathbf{q})$, jejíž velikost závisí na použitém rozhodovacím pravidlu. Protože naší snahou je navrhnout, resp. nastavit, klasifikátor tak, aby produkoval minimum chybných rozhodnutí, budeme hledat takové rozhodovací pravidlo

$$\omega = d(\mathbf{x}, \mathbf{q^*}) \quad ,$$

při kterém střední ztráta nabývá svého minima, tj.

$$\mathbf{J}(\mathbf{q^*}) = min\,\{\mathbf{J}(\mathbf{q})\} \qquad \text{pro všechna} \quad d(\mathbf{x}, \mathbf{q}) \in \mathbf{D} \ .$$

Rozhodovací pravidlo $\omega = d(\mathbf{x}, \mathbf{q^*})$ potom nazveme *optimálním rozhodovacím pravidlem* a $\mathbf{q^*}$ vektorem *optimálního nastavení klasifikátoru*. Stanovení optimálního nastavení klasifikátoru nebývá pro běžné klasifikační úlohy jednoduchou záležitostí. Používá se řada kritérií, jimiž lze spolehlivost klasifikace ocenit, avšak jejich podrobnější popis je opět nad rámec skripta. Lze je pochopitelně nalézt prakticky v každé obsáhlejší publikaci pojednávající o metodách rozpoznávání.

Nastavení klasifikátoru (nalézt skutečně optimální nastavení klasifikátoru se podaří jen u nejjednodušších úloh, v praxi se většinou spokojíme s nastavením *suboptimálním*, s nímž dosáhnene přijatelné spolehlivosti rozpoznávání) docílíme zpravidla jeho *natrénováním*. Trénovací procedura je speciálním typem učení, kdy klasifikátoru postupně předkládáme (i opakovaně) jednotlivé prvky z množiny *trénovacích obrazů* někdy nazývané *vzorovými obrazy* a současně zadáváme, do které klasifikační třídy trénovací obraz přísluší (ke každému vzorovému obrazu udáváme příslušný indikátor třídy). Je-li mohutnost množiny trénovacích obrazů dostatečná a algoritmus učení efektivní, docílíme natrénováním přibližně optimálního nastavení klasifikátoru.

Přesnost (kvalita) nastavení klasifikátoru a velikost trénovací množiny spolu těsně souvisejí. Obecně platí, že čím je trénovací množina větší, tím silnější záruky správného nastavení klasifikátoru lze poskytnout. Známe-li statistické vlastnosti obrazů, umíme potřebnou velikost trénovací množiny odhadnout. Potíž je ale v tom, že v praxi tyto statistické vlastnosti neznáme. Vždyť právě trénovací množina vzorových obrazů chybějící informaci nahrazuje. Teprve dodatečně po zpracování trénovací množiny se návrhář klasifikátoru může dozvědět, jestli je mohutnost trénovací množiny dostatečná či nikoli. Musíme proto připustit, že trénovací množina bude dodatečně rozšiřována, a to i mnohokrát, dokud nebude dosaženo dostatečné přesnosti nastavení klasifikátoru.

Výše uvedené metody nastavení klasifikátoru využívají k určení jeho správného nastavení údaje o správné klasifikaci v podobě množiny trénovacích obrazů, kterou lze chápat jako informaci od učitele. V literatuře se proto označují jako metody využívající *učení s učitelem*, angl. *supervised learning*. Někdy se však stane, že máme k dispozici pouze množinu obrazů, která obsahuje vektory příznaků bez dalších údajů o správné klasifikaci. Naskýtá se proto otázka, zda lze takovou množinu obrazů využít jako trénovací pro správné nastavení klasifikátoru.

Metody využívající k nastavení klasifikátoru množin obrazů bez àpriorní informace o zařazení obrazů do tříd se sournně označují jako metody *učení se bez učitele*, angl. *unsupervised learning*. Jejich příkladem jsou metody *shlukové analýzy*, které umožňují nastavení klasifikátoru nejen bez údajů o správné klasifikaci, ale v krajním případě i bez apriorní znalosti počtu tříd.

Výchozí data pro metody shlukové analýzy tvoří množina obrazů, která obsahuje neklasifikované vektory příznaků. Úkolem metod ja nalézt shluky těchto vektorů v obrazovém prostoru, tj. skupiny vektorů, jejichž prvky jsou si vzájemně "blízké". Množinu výchozích obrazů rozkládáme na co možná "nejkompaktnější" podmnožiny. Úspěšného výsledku však může být dosaženo pouze tehdy, když prvky výchozí (trénovací) množiny tvoří v obrazovém prostoru $\mathcal{X}$ shluky obrazů.

Metody shlukové analýzy jsou založeny na předpokladu, že dva vektory příznaků patří do stejného shluku, jsou-li si v prostoru $\mathcal{X}$ geometricky "blízké". Geometrická vzdálenost mezi dvěma vektory příznaků je samozřejmě dána metrikou definovanou v obrazovém prostoru. Podle způsobu její definice rozlišujeme různé druhy vzdáleností obvykle pojmenované podle jejich autorů. Shluky potom ztotožníme s třídami a dostáváme výsledný rozklad obrazového prostoru na jednotlivé klasifikační třídy. Vektory příznaků uvnitř shluku mají mezi sebou malou vzdálenost – jsou si "podobné"; vektory patřící do různých shluků leží daleko od sebe – jsou si "nepodobné". Některá rozdělení vektorů do shluků pokládáme za lepší než jiná a k jejich ocenění definujeme různé typy *vzdáleností*. Existuje mnoho různých hierarchických i nehierarchických algoritmů shlukové analýzy; jejich využitelnost, resp. efektivnost prakticky vždy závisí na typu úlohy. Při shlukování často využíváme spíše "ocenění" podobnosti než nepodobnosti. Teoretická podstata metod shlukové analýzy však přesahuje rámec skripta a lze se s ní seznámit např. v [Kotek93], [Lukasová88].

5.4 Strukturní metody rozpoznávání

5.4.1 Podstata strukturních metod rozpoznávání

Jak již bylo řečeno v úvodu kapitoly, strukturní metody rozpoznávání jsou založeny na využití nenumerického popisu rozpoznávaných objektů. strukturní obrazy jsou tvořeny souborem nalezených základních popisných elementů – primitiv, jejich vlastnostmi a relacemi mezi nimi. *Primitiva* představují minimální kvalitativní charakteristiky, které lze na rozpoznávaném objektu definovat, a obvykle jsou detekována klasickými příznakovými metodami. *Relace* mezi primitivy mohou být prostorové, funkční ap. Pokud jsou primitiva zvolena tak, že odpovídají podstatným částem objektů, a pokud relace vyjadřují všechny důležité relace mezi primitivy, získaný popis (obraz) objektu vystihuje hlavní strukturní vlastnosti objektu.

Strukturní popisy objektů umožňují řešit podstatně bohatší třídu úloh nežli metody příznakové. Strukturní metody mají navíc některé významné vlastnosti, které proces rozpoznávání v řadě konkrétních případů výrazně zjednodušují. Kromě klasifikace objektů do tříd totiž dovolují také popisovat objekty pomocí jejich částí a vztahů mezi nimi, čili vyjadřovat strukturní vlastnosti objektů.

Na začátku úlohy strukturního rozpoznávání stojí *volba primitiv a relací mezi primitivy*. Neexistuje žádná obecná metoda, která by tento problém řešila, a v nejvetší míře záleží na intuici a zkušenostech řešitele a na jeho àpriorních informacích o úloze. Cílem zpravidla bývá nalezení kompromisu mezi mnohdy protichůdnými požadavky, neboť

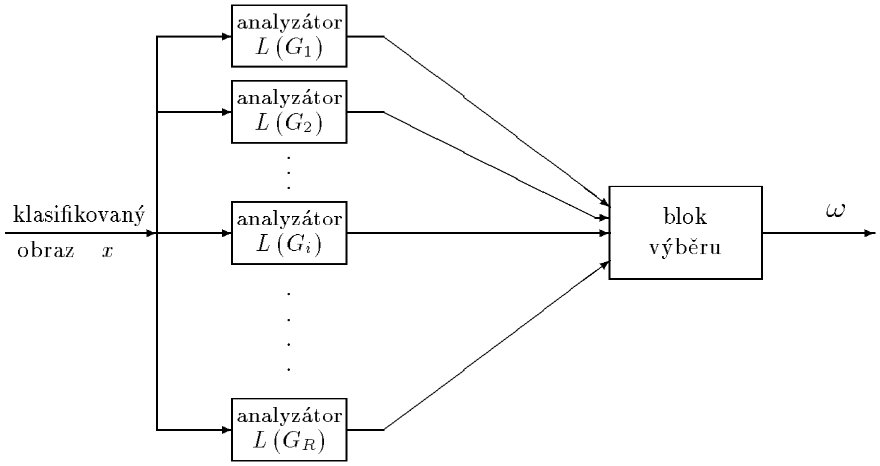
a) počet typů primitiv a relací by měl být co nejmenší,
b) primitiva by měla odpovídat přirozeným a výrazným strukturním elementům popisovaného objektu,
c) určení primitiv a relací ze získaných dat o objektu by mělo být co nejjednodušší.

Pro jednotlivé aplikační oblasti existují některá typická, nejčastěji používaná primitiva a typické relace mezi nimi. Při rozpoznávání tvarů dvourozměrných objektů jsou to např. přímkové a charakteristické křivkové úseky jejich hranic, při rozpoznávání objektů tvaru mnohostěnu na základě plošných snímků jsou to možné typy projekcí jejich vrcholů apod. Zjištění primitiva můžeme chápat jako úspěšné aplikování jistého detekčního postupu, který je současně definičním předpisem tohoto primitiva. Abychom vyhověli požadavku malého počtu typů primitiv, musí být jejich definiční předpisy dosti široké. Tím však ztrácíme část informace, kterou by bylo možné využít při rozpoznávání. Proto bývá někdy detekování primitiva doplněno určením jeho podstatných vlastností, které mohou mít nenumerickou nebo numerickou povahu. V prvním případě je lze chápat jako unární relace doplňující strukturní popis, ve druhém případě je vyjadřujeme číselným vektorem (podobným vektoru příznaků) označovaným jako vektor *sémantické informace* nebo jednodušeji jako *sémantický vektor*.

Základní myšlenka metod strukturního rozpoznávání vychází z předpokladu, že obrazy objektů (vytvořené strukturní popisy) každé třídy jsou si strukturně podobné a že tvoří *jazyk* dané třídy objektů. Úloha zařazování objektů do tříd je tedy převedena na úlohu rozhodování o tom, ke kterému z jazyků jednotlivých tříd popis patří nebo ke kterému z těchto jazyků je strukturně nejblíže (je "nejpodobnější").

Jazyky používané k popisu tříd zpravidla obsahují značný počet slov a nemusejí být konečné. Prosté porovnávání klasifikovaného strukturního obrazu s možnými slovy jazyka je proto neefektivní, ne-li přímo nemožné. Je proto nezbytné zavést některá omezení, která proces analýzy strukturních popisů zjednoduší. Jednou z možností je použití jednoduchých relačních struktur, které nad množinou symbolů použitých pro označení primitiv definují pouze unární relace a výrazně omezenou množinu binárních relací. Unárními relacemi jsou v takové struktuře definovány vlastnosti jednotlivých primitiv a binárními relacemi vztahy mezi primitivy. Zredukujeme-li množinu binárních relací na jedinou relaci úplného ostrého uspořádání, např. x "je vlevo od" y, jsou slovy jazyka popisujícího objekty dané třídy lineární řetězce symbolů reprezentujících primitiva. Pro analýzu takových řetězců pak existuje propracovaný aparát metod syntaktické analýzy používaný v kompilátorech programovacích jazyků.

Jak již bylo řečeno výše, jazyky používané pro popis objektů daných tříd nebývají konečné a je proto třeba nalézt formalismus, kterým lze jazyky popsat, resp. jednoznačně definovat. Takovým formalismem je – podobně jako je tomu u jazyka přirozeného – *gramatika*, která říká, jak konstruovat slova daného jazyka. S definicí gramatiky formálního jazyka se čtenář seznámil již v předmětu Teoretická informatika a proto se jí v dalším nebudeme zabývat.



Obr. 5. 5: Struktura syntaktického klasifikátoru

Klasifikace strukturního popisu neznámého (zařazovaného) objektu je potom transformována na úlohu zařazení neznámého slova (reprezentujícího daný strukturní popis) do třídy, jejíž gramatika toto slovo generuje. Vzhledem k takové transformaci můžeme pak pro zařazování neznámých objektů do klasifikačních tříd použít některé z propracovaných metod *syntaktické analýzy*, která jednoznačně rozhodne, kterou z gramatik je slovo reprezentující rozpoznávaný objekt generováno, a do takové třídy je klasifikovaný objekt zařazen. Struktura klasifikátoru vytvořeného na tomto principu je vyobrazena na obr. 5. 5. V úlohách rozpoznávání ale mnohdy nevystačíme s obyčejnými, tzv. "řetězovými" gramatikami používanými pro definici běžných programovacích jazyků. Často se používají speciální gramatiky, jako např. *pavučinové*, *grafové* nebo *gramatiky polí* (array grammars) určené pro značně úzce vymezené okruhy úloh. Jejich podrobnější specifikace a způsob analýzy takových strukturních popisů však již přesahují rámec předmětu a tím i skripta.

V úlohách rozpoznávání chceme často vymezit klasifikační třídy, resp. jejich popisy (u strukturních metod nejlépe jejich gramatiky) pokud možno automaticky předložením konečné množiny příkladů, popř. i kontrapříkladů několika objektů příslušných ke každé z klasifikačních tříd. Chceme tedy, aby vhodný mechanismus sám nalezl zákonitosti, kterými jsou klasifikační třídy definovány. Hovoříme o *inferenci strukturních popisů* nebo o *inferenci gramatik*, která je analogií procesu učení, který byl uveden u příznakových metod rozpoznávání. Protože dosud známé algoritmy pro inferenci gramatik opět řeší pouze dosti speciální případy, nebudeme se jimi v tomto skriptu dále zabývat.

5.4.2 Vytváření strukturních popisů

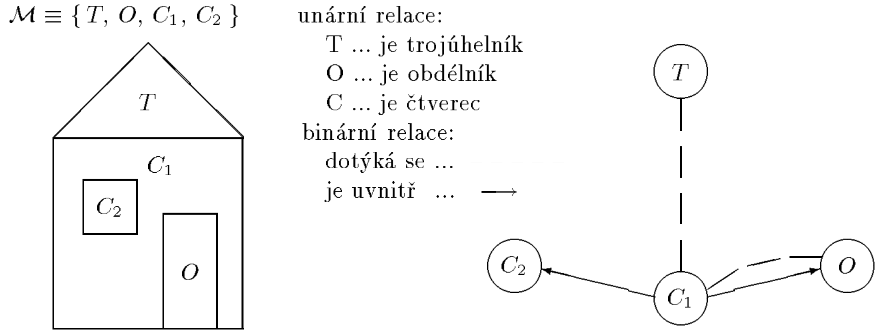
Nechť $\mathcal{M}$ je množina prvků $a_1, a_2, \ldots a_N$; $\mathcal{M} \equiv \{ a_1, a_2, \ldots a_N \}$. Podmnožinu t kartézského součinu $\mathcal{M}^k$, $t \subset \mathcal{M}^k$, nazveme *k–ární relací* na $\mathcal{M}$. Pro $k = 1$ půjde o relaci *unární*, vztahující se k samotnému objektu a vyjadřující vlastnost daného objektu (obdoba jednomístných predikátů). *Binární* relace ($k = 2$), definovaná jako podmnožina množiny všech uspořádaných dvojic $< a_i, a_j >$, $a_i, a_j \in \mathcal{M}$, $i, j \in 1, \ldots, N$, vyjadřuje pak vztah mezi i–tým a j–tým prvkem množiny $\mathcal{M}$.

Označme $\mathcal{N} \equiv \{ q_1, q_2, \ldots q_M \}$ množinu jmen relací $q_1, q_2, \ldots q_M$. *Relační strukturou* $\mathcal{S}$ pak nazveme uspořádanou dvojici $< \mathcal{M}, F >$, $\mathcal{S} = < \mathcal{M}, F >$, kde $\mathcal{M} \equiv \{ a_1, a_2, \ldots a_N \}$ je množina označovaná jako *nosič relační struktury* a F je funkce přiřazující každému jménu relace q_h , $q_h \in \mathcal{N}$, $h = 1, \ldots, M$, k_h-ární relaci na $\mathcal{M}$.

Protože každou k-ární relaci, $k \geq 2$, lze rozložit na množinu nejvýše binárních relací, lze každou relační strukturu $\mathcal{S}$ převést na relační strukturu $\mathcal{S}'$ s nejvýše binárními relacemi. V dalším textu se proto bez ztráty obecnosti omezíme pouze na relační struktury s unárními a binárními relacemi. Vlastností použitých binárních relací (tranzitivita, symetričnost ap.) lze potom využívat ke zjednodušování strukturních popisů. Řadu relací nebo prvků relací není totéž nutné ve strukturní reprezentaci explicitně uvádět, neboť je lze s využitím známých vlastností binárních relací kdykoli odvodit.

Relační struktury s nejvýše binárními relacemi lze znázorňovat grafy. Prvky nosiče relační struktury se znázorňují jako uzly, prvky binárních symetrických relací jako neorientované hrany a prvky binárních nesymetrických relací jako hrany orientované. Prvky unárních relací pak vyjadřují jména a vlastnosti jednotlivých prvků nosiče relační struktury.

Pro použití relačních struktur k popisu rozpoznávaných objektů v úlohách strukturního rozpoznávání pak existuje následující postup:
1. Podle zadaných definičních předpisů nalezneme všechna primitiva a přiřadíme jim prvky nosiče relační struktury.
2. Každý prvek nosiče bude mít vlastnost (unární relaci) označenou jménem odpovídajícího primitiva.
3. Získaná primitiva mohou být doplněna informací o dalších vlastnostech podstatných pro rozpoznávání. Příslušný element nosiče bude prvkem další unární relace, případně bude doplněn číselným (sémantickým) vektorem. Unární relace tak můžeme rozdělit do dvou skupin – v první budou relace určující, které popisují definiční vlastnost primitiva, ve druhé pak relace doplňující, které definovaný prvek nosiče popisují podrobněji.
4. Vztahy mezi detekovanými primitivy vyjádříme binárními relacemi.



Obr. 5. 6: Příklad vytvoření relační struktury

Vytvoření jednoduché relační struktury reprezentující strukturní popis jednoduché scény si ukažme na následujícím příkladě:

Příklad 5. 1: Máme dánu jednoduchou scénu vyobrazenou v levé části obr. 5. 6. Jako primitiva zvolíme elementární útvary, např. obdélníky, resp. čtverce, a trojúhelníky. V obrázku najdeme všechna primitiva a každému přiřadíme prvek nějaké množiny (nosiče relační struktury). Některé z prvků nosiče patří do unární relace "je obdélník", jiné do relace "je trojúhelník", další do "je čtverec". Mezi elementárními obrazci reprezentovanými primitivy budeme dále hledat vztahy "dotýká se" a "je uvnitř", které reprezentujeme binárními relacemi stejného jména. Vytvořenou relační strukturu potom znázorníme grafem nakresleným v pravé části obr. 5.6. Popis můžeme dále ještě doplnit informací o ploše útvarů, např. definujeme unární relace "je malý čtverec", "je velký čtverec", nebo připojíme ke každému prvku nosiče číselný údaj udávající velikost plochy, délku strany ap.

5.4.3 Volba primitiv a relací

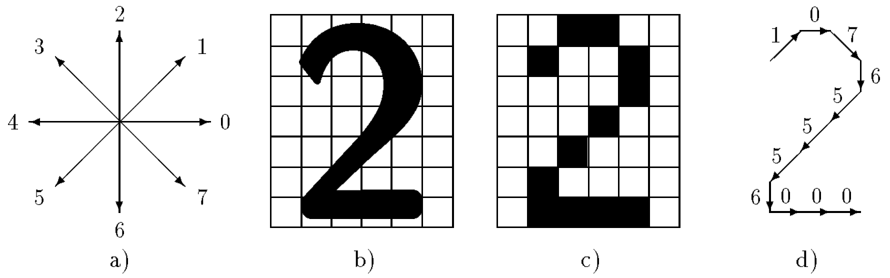
Volba primitiv a relací hraje poměrně významnou roli při tvorbě efektivního klasifikátoru založeného na strukturních metodách rozpoznávání. V rámci jednotlivých aplikačních oblastí se na základě dosavadních zkušeností ustálily jisté problémově orientované "zvyklosti", jak primitiva volit. Některé známé postupy jsou však natolik obecné, že je lze využívat v mnoha aplikačních oblastech a stojí za to, se o nich alespoň v krátkosti zmínit.

Strukturní metody rozpoznávání se nejčastěji uplatňují při klasifikaci časových průběhů určitých snímaných veličin (signálů), při rozpoznávání dvojrozměrných útvarů nebo při analýze scény, tj. při rozpoznávání množiny trojrozměrných objektů a jejich vzájemných vztahů na základě dvourozměrného snímku scény. V každém z uvedených případů se provádí *popis jistého čárového obrazu*. Pro vytvoření strukturního popisu musí být příslušný čárový obraz segmentován, tj. musejí být určeny části, které mohou představovat nebo obsahovat primitiva strukturního popisu. Velmi často je účelné čárový obraz dekomponovat na jednodušší čárové obrazy a teprve ty pak segmentovat. Dekompozice může být i mnohoúrovňová. Segmentaci, vyjadřující rozdělení obrazu na úseky, však nelze zaměňovat s pojmem dekompozice, který vyjadřuje rozklad úlohy na podúlohy (viz odst. 2.6).

Pro popis časových průběhů signálů je typické, že se používá jediné relace, a to relace *zřetězení*, která popisuje bezprostřední následnost časových segmentů signálu. Při ekvidistantní segmentaci časových průběhů (ekvidistantní metody výběru primitiv jsou založeny na segmentaci čárového "obrazu" průběhu signálu do úseků konstantní délky, přičemž typické tvary úseků tvoří primitiva popisu) je signál v každém z časových úseků konstantní délky popsán symbolem z abecedy (průběh signálu v daném časovém úseku lze znázornit některým z definovaných primitiv). Přiřazení symbolu primitiva se často provádí s využitím příznakových metod rozpoznávání na základě lokálních charakteristik signálu v daném časovém úseku. Ekvidistantní segmentace se využívá především u časových průběhů stochastického charakteru s nezanedbatelným zastoupením vyšších frekvencí ve spektru.

Hodnotnějšího popisu, zejména u "pomalejších" časových křivek, lze dosáhnout volbou primitiv, která výstižně zachycují významné tvarové vlastnosti křivek. Primitivem je pak segment křivky, v němž má křivka stejný tvar (přímkový úsek, část paraboly, část kružnice ap.), přičemž časová "délka" segmentu hraje druhořadou roli a je zachycena např. jako sémantická informace (viz odst. 5.4.5). Podrobnější popis této problematiky lze nalézt v kterékoli publikaci zabývající se strukturními metodami rozpoznávání (v češtině např. [Kotek93], [Mařík85]).

Pro vytváření strukturních popisů dvourozměrných útvarů se obvykle využívá ekvidistantních metod výběru primitiv. Jako příklad si uveďme využití tzv. *řetězových kódů* jako jedné z ekvidistantních metod často používaných ke strukturním popisům čárových obrazů bez rozvětvení, např. jednoduchých souvislých křivek, hraničních křivek obrazových segmentů detekovaných v obrazu scény apod. Na obr. 5.7 je znázorněn jednoduchý příklad použití tzv. *Freemanova řetězového kódu* pro prvotní reprezentaci (zakódování) číslice **2**, která může být zobrazena jednoduchou souvislou čarou (bez rozvětvení).



Výsledná reprezentace tvaru číslice **2**: 10765556000

Obr. 5.7: Příklad popisu číslice **2** Freemanovým řetězovým kódem
a) směrová růžice Freemanova kódu
b) "překrytí" kódovaného znaku kódovacím rastrem
c) přibližná reprezentace znaku v kódovacím rastru
d) posloupnost směrů v kódovacím rastru reprezentující znak **2**

Obrázek 5.7a) zobrazuje směrovou růžici Freemanova řetězového kódu použitou pro zakódování směrů detekovaných v kódovacím rastru, kterým musíme reprezentovanou křivku "překrýt", resp. do nějž musíme reprezentovanou křivku zobrazit (jak je uvedeno na obr. 5. 7 b)), obr. 5. 7 c) pak ukazuje přibližnou reprezentaci křivky v kódovacím rastru, kterou získáme "začerněním" těch políček rastru, kterými reprezentovaná křivka prochází, resp. těch, která křivka překrývá z více než 50 % jejich plochy, a konečně obr. 5. 7 d) znázorňuje vytvoření kódové reprezentace, kterou získáme tak, že postupně od zvoleného výchozího bodu křivky (u snímků reprezentujících nejrůznější scény obvykle začínáme bodem, který je nejvíce vlevo nahoře, což je dáno způsobem prohledávání snímku scény) spojujeme "začerněná" políčka s jejich nejbližšími sousedy (máme-li více možností, tj. má-li aktuální políčko více "začerněných" sousedů, vybereme takové, k němuž se dostaneme po nejmenší změně směru postupu) a spojnicím sousedních políček přiřazujeme směrový kód podle označení směrů ve směrové růžici. Výsledný kód reprezentující kódovaný znak pak najdeme v dolní části obrázku.

Syntaktický popis křivky lze dále upravit tak, že místo směrů určovaných absolutně vzhledem ke směrové růžici použijeme k popisu křivky změny směrů měřené relativně vždy vzhledem ke směru předchozího úseku. Jde tedy o jakousi "první derivaci" (první diferenci) původního (absolutního) popisu. Takový popis je výhodnější v tom smyslu, že rovněž plně vyjadřuje tvar reprezentované křivky a navíc je invariantní vůči natočení křivky v kódovacím rastru. Tím je odstraněna nepříjemná vlastnost "absolutního" kódování křivky – závislost popisu na orientaci.

Čárové obrazy lze dále reprezentovat také prostředky jazyka PDL (Picture Description Language) ; jeho popis a použití však příslušejí do předmětu "Zpracování vizuální informace" a tudíž jsou mimo rámec tohoto skripta. Čtenáře zajímajícího se o tuto otázku lze odkázat např. na [Hlaváč93], [Mařík93].

Při rozpoznávání trojrozměrných objektů a analýze konfigurací většího počtu těchto objektů, tj. při tzv. *analýze scény* , vycházíme obvykle ze senzorického obrazu scény – *snímku* scény. Snímek není nic jiného, než dvourozměrný obraz původně trojrozměrné reálné scény, a lze jej chápat jako množinu dvourozměrných reprezentací objektů vyskytujících se na scéně, které lze dále popsat čárovými obrazy. Je však třeba důsledně odlišovat *strukturní popis snímku* od *strukturního popisu scény* . Na úrovni snímku je přístup k volbě primitiv a relací stejný jako u popisu dvourozměrných objektů, kdežto na úrovni scény primitiva určují jednotlivé typy objektů (např. krychle, hranol, koule, jehlan, ...) a relace vyjadřují např. vzájemnou polohu primitiv apod.

Úkolem strukturní analýzy na úrovni snímku je odhalit elementy popisu scény, a to na základě informací o přípustných strukturách reprezentujících objekty scény na úrovni snímku. Jako samostatná primitiva popisu na úrovni snímku se někdy volí nejen prvky (dílčí objekty) podle tvaru, ale i přípustné tvary z hlediska sémantické reprezentace snímku (obvykle jde o projekce reálných situací ve scéně). Definitivní označení primitiva se pak provádí až v průběhu syntaktické analýzy popisu snímku tak, aby bylo dosaženo přípustné struktury snímku scény z hlediska syntaktického i sémantického.

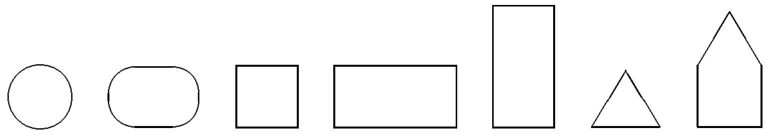
Na závěr odstavce poznamenejme, že strukturní popisy *na úrovni scény* mají přednostní význam pro deskripční popis scény jako celku, např. pro účely rozhodování robota (odkud kam má stanovený předmět přemístit apod.), než pro úlohy rozpoznávání, kde využíváme popisů scény *na úrovni snímku*.

5.4.4 Příklad aplikace strukturních metod rozpoznávání

Použití metod strukturního rozpoznávání si ilustrujme na jednoduchém případě rozpoznávání prostých geometrických objektů vyobrazených v následujícím příkladě:

Příklad 5. 2: Pro vyobrazené rozpoznávané geometrické objekty:

a) zvolte vhodně primitiva,
b) vytvořte příslušné symbolické strukturní popisy,
c) definujte gramatiky generující slova jednotlivých jazyků, která vyjadřují (popisují) Vámi vytvořené strukturní popisy,
d) nakreslete konkrétní strukturu syntaktického klasifikátoru, který bude vyobrazené geometrické objekty klasifikovat do příslušných tříd.



(1) (2) (3) (4) (5) (6) (7)

Řešení:

a) Vzhledem k jednoduchým tvarům geometrických objektů vystačíme s volbou následujících primitiv, která pro zjednodušení dalšího zápisu označíme malými písmeny:

primitivum	—	|	/	\	(	)
symb. označení	a	b	c	d	e	f

b) Na jednotlivých geometrických objektech detekujeme primitiva a formulujeme jejich strukturní popisy takto:

(1) $\mathsf{e\,f}$
(2) $\mathsf{e\,a\,f\,a}$, resp. $\mathsf{e\,a\,a\,f\,a\,a}$ ... $\mathsf{e}(\mathsf{a})^n\,\mathsf{f}(\mathsf{a})^n$
(3) $\mathsf{b\,a\,b\,a}$
(4) $\mathsf{b\,a\,a\,b\,a\,a}$, resp. $\mathsf{b\,a\,a\,a\,b\,a\,a\,a}$ ... $\mathsf{b}(\mathsf{a})^n\,\mathsf{b}(\mathsf{a})^n$
(5) $\mathsf{b\,b\,a\,b\,b\,a}$, resp. $\mathsf{b\,b\,b\,a\,b\,b\,b\,a}$ ... $(\mathsf{b})^n\,\mathsf{a}(\mathsf{b})^n\,\mathsf{a}$
(6) $\mathsf{c\,a\,d}$
(7) $\mathsf{c\,b\,a\,b\,d}$, resp. $\mathsf{c\,b\,b\,a\,b\,b\,d}$ ... $\mathsf{c}(\mathsf{b})^n\,\mathsf{a}(\mathsf{b})^n\,\mathsf{d}$

c) Definujeme gramatiky generující jazyky, do nichž příslušejí slova popisující jednotlivé rozpoznávané objekty:

$$\begin{aligned} G_1 &= < V_{N1}, V_{T1}, S_1, R_1 > \\ V_{N1} &\equiv \{ S_1, X_1, Y_1 \} \\ V_{T1} &\equiv \{ \mathsf{e}, \mathsf{f} \} \\ R_1 &\equiv \{ S_1 \rightarrow X_1 Y_1, \\ &\qquad X_1 \rightarrow \mathsf{e},\ Y_1 \rightarrow \mathsf{f} \} \end{aligned}$$

$$\begin{aligned} G_2 &= < V_{N2}, V_{T2}, S_2, R_2 > \\ V_{N2} &\equiv \{ S_2, X_2, Y_2, Z \} \\ V_{T2} &\equiv \{ \mathsf{a}, \mathsf{e}, \mathsf{f} \} \\ R_2 &\equiv \{ S_2 \rightarrow X_2 Y_2,\ X_2 \rightarrow \mathsf{e} Z, \\ &\qquad Y_2 \rightarrow \mathsf{f} Z,\ Z \rightarrow \mathsf{a} Z \,|\, \mathsf{a} \} \end{aligned}$$

$$\begin{aligned} G_3 &= < V_{N3}, V_{T3}, S_3, R_3 > \\ V_{N3} &\equiv \{ S_3, X_3, Y_3 \} \\ V_{T3} &\equiv \{ \mathsf{a}, \mathsf{b} \} \\ R_3 &\equiv \{ S_3 \rightarrow (X_3 Y_3)^2, \\ &\qquad X_3 \rightarrow \mathsf{b},\ Y_3 \rightarrow \mathsf{a} \} \end{aligned}$$

$$\begin{aligned} G_4 &= < V_{N4}, V_{T4}, S_4, R_4 > \\ V_{N4} &\equiv \{ S_4, X_4, Y_4 \} \\ V_{T4} &\equiv \{ \mathsf{a}, \mathsf{b} \} \\ R_4 &\equiv \{ S_4 \rightarrow (X_4 Y_4)^2, \\ &\qquad X_4 \rightarrow \mathsf{b},\ Y_4 \rightarrow \mathsf{a} Y_4 \,|\, \mathsf{a} \} \end{aligned}$$

$$\begin{aligned} G_5 &= < V_{N5}, V_{T5}, S_5, R_5 > \\ V_{N5} &\equiv \{ S_5, X_5, Y_5 \} \\ V_{T5} &\equiv \{ \mathsf{a}, \mathsf{b} \} \\ R_5 &\equiv \{ S_5 \rightarrow (X_5 Y_5)^2, \\ &\qquad X_5 \rightarrow \mathsf{b} X_4 \,|\, \mathsf{b},\ Y_5 \rightarrow \mathsf{a} \} \end{aligned}$$

$$\begin{aligned} G_6 &= < V_{N6}, V_{T6}, S_6, R_6 > \\ V_{N6} &\equiv \{ S_6, X_6 \} \\ V_{T6} &\equiv \{ \mathsf{a}, \mathsf{c}, \mathsf{d} \} \\ R_6 &\equiv \{ S_6 \rightarrow \mathsf{c} X_6 \mathsf{d}, \\ &\qquad X_6 \rightarrow \mathsf{a} \} \end{aligned}$$

$$\begin{aligned} G_7 &= < V_{N7}, V_{T7}, S_7, R_7 > \\ V_{N7} &\equiv \{ S_7, X_7, Y_7 \} \\ V_{T7} &\equiv \{ \mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d} \} \\ R_7 &\equiv \{ S_7 \rightarrow \mathsf{c} X_1 \mathsf{d}, \\ &\qquad X_7 \rightarrow Y_7 \mathsf{a} Y_7,\ Y_7 \rightarrow \mathsf{b} Y_7 \,|\, \mathsf{b} \} \end{aligned}$$

d) Syntaktický klasifikátor výše uvedených geometrických objektů bude klasifikovat do sedmi klasifikačních tříd, tzn. bude obsahovat sedm bloků, jimiž bude realizována syntaktická analýza sedmi různých jazyků $L(G_1)$ až $L(G_7)$ generovaných gramatikami G_1 až G_7. Bloková struktura klasifikátoru bude odpovídat struktuře znázorněné na obr. 5. 5 pro $R = 7$.

5.4.5 Využití sémantické informace

Nechť $\mathcal{S} = < \mathcal{M}, F >$ je relační struktura. Z hlediska zpracování informace o objektu vystupuje do popředí především definiční vlastnost primitiva, o níž říkáme, že je nositelem syntaktické informace o primitivu, neboť jedině na splnění či nesplnění této vlastnosti závisí, zda vygenerujeme popisný symbol a přiřadíme mu příslušné jméno (unární relace).

Doplňková informace o primitivu a_j má z hlediska syntaktického zpracování charakter *sémantické* informace. Bývá obvykle vyjádřena jako vektor numerických hodnot $\mathbf{y}^j = [\, y_1^j, \ldots, y_L^j \,]^T$ některých měřitelných veličin nebo binárních hodnot, které poskytují informaci o existenci či neexistenci některé vlastnosti. Potom lze definovat *rozšířený prvek nosiče* jako dvojici $m_j = < a_j, \mathbf{y}^j >$, kde $a_j \in \mathcal{M}$. V dalším budeme předpokládat, že nosič $\mathcal{M}$ je tvořen symboly a_j nebo dvojicemi m_j. Protože nemůže dojít k nejasnostem, nebudeme mezi těmito případy rozlišovat.

Obdobně lze přiřadit doplňkovou (sémantickou) informaci i každému prvku každé binární relace ve formě sémantického vektoru. Nechť výše definovaná relační struktura obsahuje celkem M binárních relací K_α, $\alpha = 1, \ldots, M$ a každá binární relace obsahuje K_α prvků (uspořádaných dvojic primitiv). Pro lepší srozumitelnost dalšího textu očíslujeme postupně všechny prvky všech binárních relací, přičemž prvky relací označíme u_k, kde $k = 1, \ldots, K$, $K = \sum_\alpha K_\alpha$. Sémantický vektor příslušný prvku u_k binární relace lze zapsat ve tvaru $\mathbf{v}^k = [\, v_1^k, \ldots, v_J^k]^T$, prvek binární relace lze pak rozšířit na $z_k = < u_k, \mathbf{v}^k >$. Vektory $\mathbf{y}^j$ a $\mathbf{v}^k$ jsou v literatuře obvykle nazývány *vektory atributů*.

Zavedení sémantické informace má velký praktický význam:

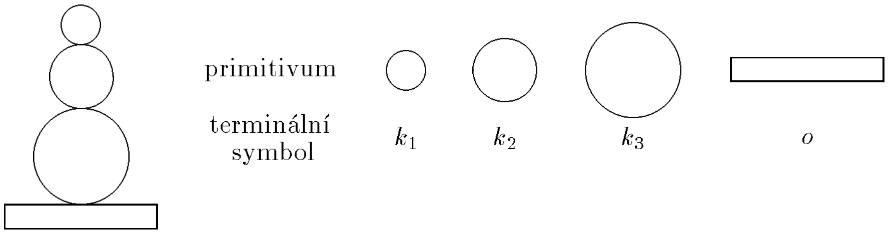
a) Řadu primitiv, které bylo nutno odlišovat z důvodu rozdílnosti některých naměřených numerických parametrů, lze při využití nové definice primitiva považovat za jediné primitivum, přičemž rozdíly se projeví jen v hodnotách složek příslušného sémantického vektoru. Totéž platí i o relacích. Snížení počtu primitiv a relací bez ztráty informace přispívá ke zjednodušení reprezentace jednotlivých tříd.

b) Je možné vytvořit velmi obecné *deformační schéma* (viz odst. 5.4.6 a 5.4.7) zahrnující všechny možnosti ovlivnění strukturního popisu šumem. Toto schéma lze využít k systematickému odstranění poruch strukturního obrazu objektu a poté k jeho (sub)optimální klasifikaci.

c) Prostřednictvím sémantické informace je umožněna významová kontrola syntaktických operací nad daným strukturním popisem. Tak například lze výsledek syntaktické analýzy zkontrolovat z hlediska fyzikální realizovatelnosti, spojit syntaktickou analýzu přímo s výběrem primitiv apod.

d) Sémantické informace je možné využít též k řízení syntaktické analýzy. Např. u atributové gramatiky lze podmínky i příkazy sestavit tak, aby se jednalo výlučně o vztahy mezi složkami vektorů sémantické informace.

Příklad 5. 3: Pro popis obrazce uvedeného na obr. 5. 8 zvolme jako primitiva základní typy geometrických útvarů a jako jedinou relaci zvolme relaci zřetězení.



Obr. 5. 8: Obrazec typu "sněhulák" a volba primitiv

Vyjdeme-li od "hlavy" "sněhuláka", můžeme ho popsat řetězem terminálních symbolů

$$k_1\ k_2\ k_3\ o \quad .$$

Třídu všech "sněhuláků" o třech "koulích" lze popsat např. regulární gramatikou

$$G_{sněh} \ = < V_{N\,sněh}\,,\ V_{T\,sněh}\,,\ S\,,\ R_{sněh} \ > \ ,$$

kde $V_{N\,sněh} \equiv \{\,S\,,\,A\,,\,B\,,\,C\,\}\,,$
$V_{T\,sněh} \equiv \{\,k_1\,,\ k_2\,,\ k_3\,,\ o\,\},$
$R_{sněh} \equiv \{\,S \rightarrow k_1\,A\,,\ A \rightarrow k_2\,B\,,\ B \rightarrow k_3\,C\,,\ C \rightarrow o\,\} \quad .$

Bude-li primitivem označeným jediným symbolem k jakákoli kružnice (reprezentující rovinnou projekci sněhulákovy "koule") s tím, že k primitivu přiřadíme sémantický vektor o jedné složce $\mathbf{y} = [\,poloměr\_koule\,]$, redukuje se popis "sněhuláka" na řetěz

$$k_{(6)}\ k_{(8)}\ k_{(10)}\ o \quad .$$

Třídu "sněhuláků" lze pak reprezentovat atributovou regulární gramatikou

$$G_{a\_sněh} \ = < V_{N\,a\_sněh}\,,\, V_{T\,a\_sněh}\,,\ S\,,\ R_{a\_sněh}\,,\mathcal{C}_{a\_sněh}\,,\mathcal{P}_{a\_sněh}\,,\ \mathbf{v} \ > \ ,$$

kde $V_{N\,a\_sněh} \ \equiv \ \{\,S\,,A\,\},\ V_{T\,a\_sněh} \equiv \{\,k\,,\ o\,\},$
vektor $\mathbf{v}$ obsahuje jediný parametr v_1 označující poloměr koule a množina přepisovacích pravidel bude ve tvaru

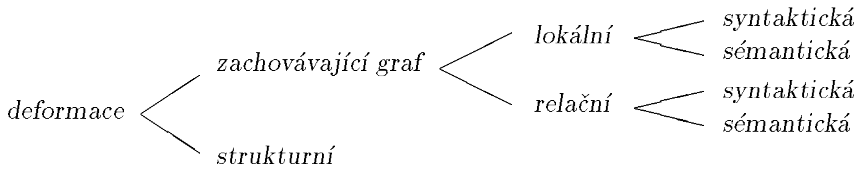
$R_{a\_sn\check{e}h}$:	r_i	$\mathcal{C}_{a\_sn\check{e}h}$	$\mathcal{P}_{a\_sn\check{e}h}$
	$S \rightarrow kA$		$v_1 = 0$
	$A \rightarrow kA$	$par \geq v_1$	$v_1 = par$
	$A \rightarrow o$		

Jak je z příkladu patrné, zavedením sémantické informace bylo možno snížit počet primitiv, neterminálních symbolů i přepisovacích pravidel. Navíc lze použitím atributové gramatiky zajistit větší variabilitu "sněhuláků". Jesliže např. pro zobrazení sněhulákovy "koule" použijeme primitivum "elipsa" místo "kružnice", abychom mohli reprezentovat i "šišaté" sněhuláky, lze potom v rámci sémantických pravidel použít např. rozměry dvou poloos elipsy, resp. jeden délkový rozměr a poměr délek obou poloos atp., a vytvořenými strukturními popisy reprezentovat daleko rozmanitější tvary sněhuláků.

5.4.6 Deformační schéma

V předcházejícím odstavci jsme uvedli, že relační strukturu $\mathcal{S}$ doplněnou sémantickou informací lze popsat $\mathcal{S} = <\mathcal{M}, F>$, kde $\mathcal{M} \equiv \{m_1, \ldots, m_N\}$ je nosič struktury, $m_j = < a_j, \mathbf{y}^j >$ a F je funkce pojmenovávající relace mezi prvky nosiče, přičemž prvky všech binárních relací označíme $z_k = < u_k, \mathbf{v}^k >$, $k = 1, \ldots, K$.

Nechť struktura $\tilde{\mathcal{S}}$ je etalonem třídy a v procesu předzpracování informace je vytvořena relační struktura $\mathcal{S}$, $\mathcal{S} \not\equiv \tilde{\mathcal{S}}$ (připomeňme si, že i řetěz symbolů je relační strukturou, kterou však v souladu s předchozím textem označujeme jako slovo W). Dále předpokládejme, že získaný obraz reálného objektu $\mathcal{S}$ vznikl postupnými deformacemi ideální relační struktury $\tilde{\mathcal{S}}$. Pak je možné vytvořit *obecné deformační schéma* kategorizující jednotlivé typy přípustných deformací:



V prvním dělení je třeba rozlišovat dva typy deformací:

a) *Deformace zachovávající graf struktury* – platí $\mathcal{S} \not\equiv \tilde{\mathcal{S}}$, ale $\mathcal{G} \equiv \tilde{\mathcal{G}}$, kde $\tilde{\mathcal{G}}$ je neoznačený orientovaný graf struktury $\tilde{\mathcal{S}}$. Při tomto typu deformace dochází k chybnému označení jednoho nebo více primitiv či k chybnému přiřazení jména binární relace.

b) *Deformace strukturní* – platí $\mathcal{G} \not\equiv \tilde{\mathcal{G}}$, tzn., že u obrazu $\mathcal{S}$ je vložena či vypuštěna určitá struktura (u řetězových jazyků jde o vložení či vypuštění terminálních symbolů).

Deformace zachovávající graf struktury lze dále dělit na

aa) *deformace lokální*, kdy platí $z_k = \tilde{z}_k$ pro všechna k, ale alespoň pro jedno j platí $m_j \neq \tilde{m}_j$. Při lokálních deformacích nedochází k deformacím binárních relací, avšak vzniká chybné přiřazení prvků nosiče k unárním (definičním) relacím.

ab) *deformace relační*, kdy alespoň pro jedno k platí, že $z_k \neq \tilde{z}_k$. V tomto případě dochází k chybnému zařazení některé dvojice prvků nosiče do binární relace. Deformace relační v sobě velmi často zahrnuje deformaci lokální, přičemž detekce primitiv časově předchází detekci prvků relací.

Deformace lokální dále dělíme na

aaa) *deformace syntaktické*, kdy $a_j \neq \tilde{a}_j$ pro některá j a
aab) *deformace sémantické*, kdy $a_j = \tilde{a}_j$ pro všechna j, avšak pro některá j platí $\mathbf{y}^j \neq \tilde{\mathbf{y}}^j$.

Obecně lze lokální deformaci považovat za dvojstupňovou transformaci $\tilde{m}_j$ na m_j:

$$\tilde{m}_j = <\tilde{a}_j, \tilde{\mathbf{y}}^j> \xrightarrow{T_1} m'_j = <a_j, \mathbf{w}^j> \xrightarrow{T_2} m_j = <a_j, \mathbf{y}^j>,$$

kde symbol T_1 označuje syntaktickou lokální deformaci a T_2 sémantickou lokální deformaci.

Deformace relační lze podobně jako deformace lokální dále dělit na

aba) *deformace syntaktické*, kdy $u_k \neq \tilde{u}_k$ pro některá k a
abb) *deformace sémantické*, kde $u_k = \tilde{u}_k$ pro všechna k, ale pro některá k platí $\mathbf{v}^k \neq \tilde{\mathbf{v}}^k$.

Obecně lze relační deformaci opět považovat za dvojstupňovou transformaci $\tilde{m}_j$ na m_j:

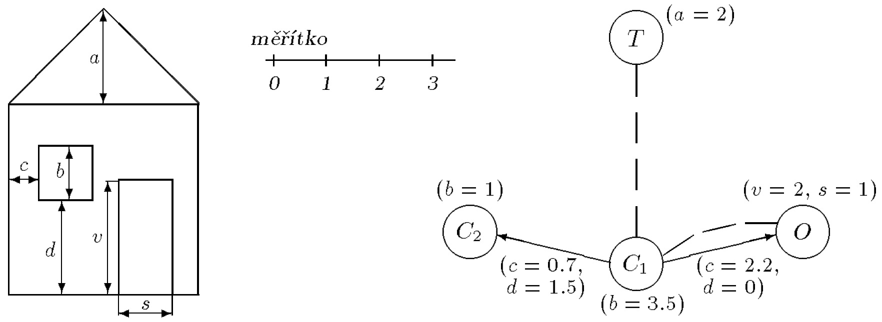
$$\tilde{z}_k = <\tilde{u}_k, \tilde{\mathbf{v}}^k> \xrightarrow{T'_1} z'_k = <u_k, \mathbf{x}^k> \xrightarrow{T'_2} z_k = <u_k, \mathbf{v}^k>,$$

kde symbol T'_1 označuje syntaktickou relační deformaci a T'_2 sémantickou relační deformaci.

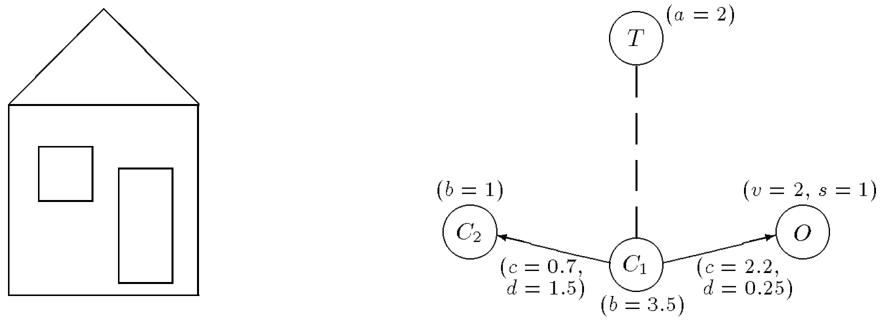
Závěrem je třeba upozornit na fakt, že každá strukturní deformace je ze své podstaty deformací syntaktickou, v jejímž důsledku musí dojít i k deformaci sémantické.

Příklad 5. 4: Na příkladě "domečku" z obr. 5. 6 si ilustrujme jednotlivé typy deformací. Nedeformovaný obraz lze vyjádřit relační strukturou $\mathcal{M}$ z obr. 5. 6, v níž symboly C, T, O vyjadřují jednotlivá primitiva (C ... čtverec, T ... trojúhelník, O ... obdélník), a symboly a, b, c, d, v, s reprezentují odpovídající sémantickou informaci (viz obr. 5. 9a):

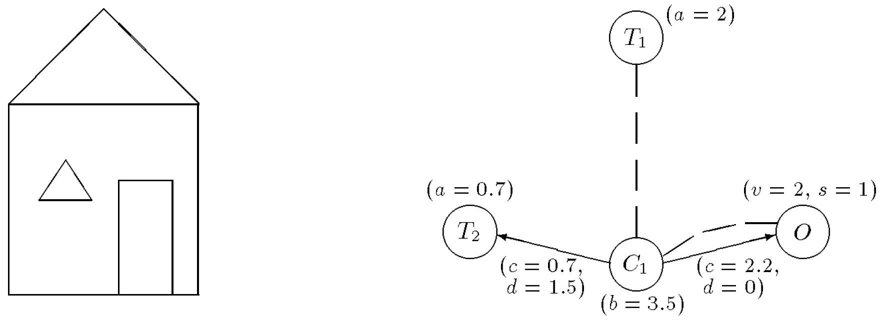
a) nedeformovaný obraz



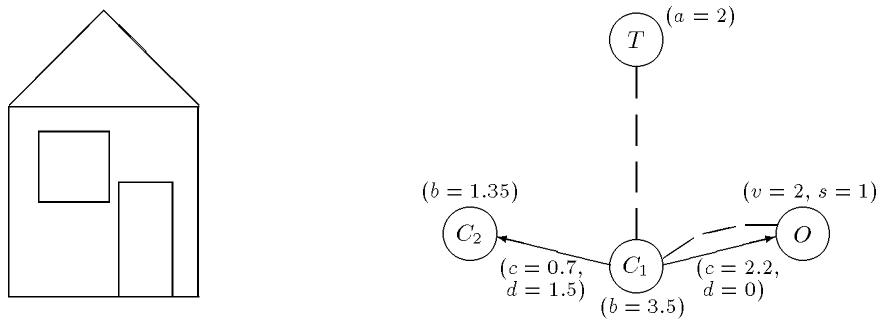
b) strukturální deformace



aaa) lokální syntaktická deformace



aab) lokální sémantická deformace



aba) relační syntaktická deformace

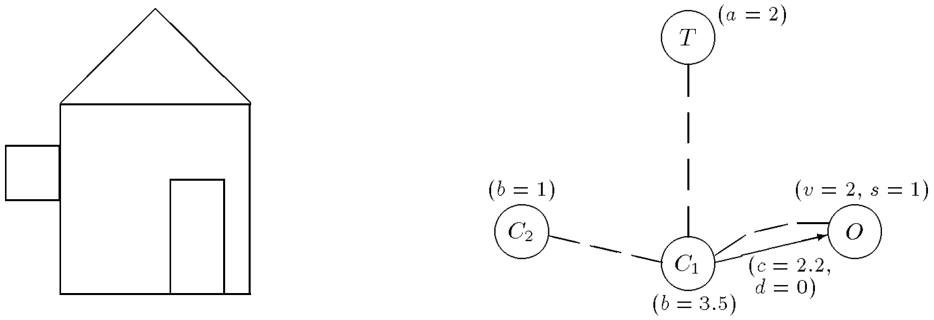
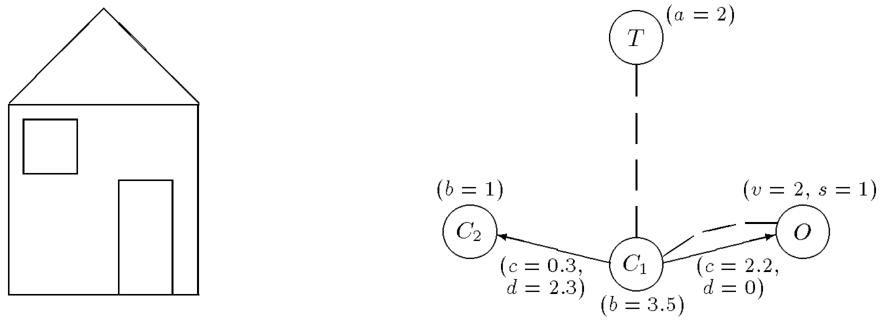


abb) relační sémantická deformace



Obr. 5.9: Příklady možných deformací strukturních popisů

Pro každou konkrétní úlohu strukturního rozpoznávání existuje hranice, kdy již nelze několik relačních či lokálních deformací považovat za deformace zachovávající graf struktury, ale spíše za jednu nebo více deformací strukturních.

U obrazů scény, jako tomu je v případě uvedeném na obr. 5.9, lze takovou deformaci chápat jako případ, kdy došlo k odstranění jisté substruktury z obrazu scény a posléze přidání substruktury jiné.

Pro každou konkrétní úlohu je dále nutné zvolit konkrétní deformační schéma, tj.

– zvolit typy přípustných deformací,
– určit konkrétní přípustné syntaktické deformace,
– určit funkce hustoty pravděpodobnosti pro sémantické vektory.

Každé konkrétní deformační schéma lze popsat tzv. *deformační gramatikou* G^D, která obsahuje

a) jako terminální symboly jména všech primitiv, tj. deformovaných i nedeformovaných, a všechny deformované i nedeformované prvky binárních relací (pokud se jedná o popisy relačními strukturami),
b) substituční pravidla, podle nichž lze přepisovat nedeformované obrazy na obrazy syntakticky deformované,
c) tabulky rozložení hustot pravděpodobnosti $q(\mathbf{y}^j/a_j, \tilde{a}_j)$, $q(\mathbf{v}^k/u_k, \tilde{u}_k)$ pro konkrétní sémantické vektory a pro příslušnou sémantickou deformaci.

Na deformační gramatiku se lze dívat jako na *gramatiku transformační*, schopnou převádět nedeformované strukturní popisy na deformované. Pokud ke každému pravidlu z deformační gramatiky přidáme reálné číslo (váhu), hovoříme o tzv. *váženém deformačním schématu*. Ve speciálním případě, jsou-li váhy rovny pravděpodobnosti užití pravidla, dostáváme *stochastické deformační schéma*.

5.4.7 Použití deformačních schémat ke klasifikaci poškozených popisů

Reprezentace tříd gramatikami

Syntaktický analyzátor, který je na základě konkrétního deformačního schématu schopen provádět nejen analýzu, ale i opravu poškozených (deformovaných) slov, se nazývá *syntaktický analyzátor s opravou chyb (error–correcting parser)*. Takový analyzátor pracuje nejen s primitivy nedeformovaných slov a se substitučními pravidly gramatik G_r, $r = 1, \dots, R$ (R je počet tříd, které byly získány inferencí z nedeformovaných slov), nýbrž i s primitivy a produkčními pravidly obsaženými v deformační gramatice G^D. Dochází tedy k rozšíření všech původních gramatik o gramatiku G^D, čímž vznikají rozšířené gramatiky G_r^E, $r = 1, \dots, R$; $G_r^E = G_r \cup G^D$. Syntaktické analyzátory s opravou chyb vyhledávají k danému slovu W nejbližší (ve smyslu zvolené vzdálenosti) gramatiku G_r.

Uveďme nyní některé možnosti volby vzdálenosti mezi slovem z $L(G_r^E)$ a jazykem $L(G_r)$:

a) *Minimální vzdálenost:* Definice vzdálenosti mezi slovem a jazykem se opírá o definici vzdálenosti slova z $L(G_r^E)$ a slova z $L(G_r)$, která se definuje buď
   - jako nezbytný počet použití pravidel z G^D při transformaci daného slova z $L(G_r)$ na slovo z $L(G_r^E)$, nebo
   - jako součet vah asociovaných k použitým pravidlům z G^D, nebo
   - jako součin příslušných pravděpodobností v případě stochastického deformačního schématu.

   Minimální vzdálenost mezi některým slovem z $L(G_r^E)$ a jazykem $L(G_r)$ se definuje jako minimální vzdálenost mezi některým slovem jazyka $L(G_r)$ a daným slovem z $L(G_r^E)$. Metoda minimální vzdálenosti není sice statisticky optimální, vede však na výpočetně relativně jednoduché algoritmy.

b) *Bayesovská vzdálenost:* Definice Bayesovské vzdálenosti využívá metody statistického testování hypotéz. Za předpokladu konečného počtu n symbolů ve slově W je Bayesovská vzdálenost obrazu $\tilde{W} \in L(G_r)$ a obrazu $W \in L(G_r^E)$ definována vztahem

$$B(\tilde{W}, W) = -\sum_{j=1}^{n} \ln p(a_j/\tilde{a}_j) + \ln q(\mathbf{y}^j/a_j, \tilde{a}_j) - \ln P(\tilde{W}),$$

   kde $p(a_j/\tilde{a}_j)$ značí pravděpodobnost syntaktické deformace $a_j \longrightarrow \tilde{a}_j$,
   $q(\mathbf{y}^j/a_j, \tilde{a}_j)$ je pravděpodobnost sémantické deformace při dané deformaci syntaktické, vypočtená dosazením do příslušné funkce hustoty pravděpodobnosti,
   $P(\tilde{W})$ udává apriorní pravděpodobnost obrazu $\tilde{W} \in L(G_r)$ a je velmi jednoduše určitelná v případě stochastické gramatiky G_r.

   Definice Bayesovské vzdálenosti D jazyka $L(G_r)$ a slova $W \in L(G_r)$ se opírá o předpoklad, že slovo může vzniknout deformacemi několika, řekněme h_r, slov z $L(G_r)$:

$$D(L(G_r), W) = -\ln\left(\sum_{h=1}^{h_r} \prod_{j=1}^{J} p({}_h a_j/{}_h\tilde{a}_j)\, q({}_h\mathbf{y}^j/{}_h a_j/{}_h\tilde{a}_j)\, P({}_h\tilde{W})\right).$$

   Statisticky optimální rozhodnutí o příslušnosti slova W k třídě se opírá též o àpriorní pravděpodobnosti tříd, přičemž chápeme podmíněnou pravděpodobnost jako

$$p(W/r) \;=\; -\exp\; D(L(G_r),\, W) \;\;.$$

Algoritmus statisticky optimálního rozhodnutí je však časově velmi náročný, a proto jsou v praxi používána suboptimální rozhodovací kritéria, např. vzdálenost $D(L(G_r),\, W)$ je aproximována podle vztahu

$$D(L(G_r),\, W) \;\sim\; \min_h\; B(\tilde{W},\, {}_hW) \;\;.$$

Syntaktické analyzátory s opravou chyb pracují ve srovnání s analyzátory jazyků definovaných nerozšířenými gramatikami o poznání "inteligentněji", avšak jsou pochopitelně ve své činnosti podstatně pomalejší, což může být vážnou překážkou v praktických aplikacích.

Závěrem odstavce poznamenejme, že zatím neexistuje efektivně fungující systém rozpoznávání, který by se opíral o využití všech typů deformací. Smyslem použití obecného deformačního schématu není směrovat tvůrce systémů rozpoznávání k akceptování všech možných deformací, nýbrž poskytnout metodický návod k výběru nejvhodnějších typů deformací pro danou aplikaci. I velmi účinné špičkové systémy strukturního rozpoznávání akceptují ve svém konkrétním deformačním schématu jen některé typy deformací, např. lokální syntaktické deformace v kombinaci s deformacemi strukturními nebo oba typy lokálních deformací při současném použití stochastických bezkontextových stromových gramatik apod.

Reprezentace relačními strukturami

Budiž dána relační struktura $\tilde{\mathcal{S}} = <\tilde{\mathcal{M}},\, \tilde{F}>$ jako *etalon třídy*, testovanou (rozpoznávanou) relační strukturu označme $\mathcal{S} = <\mathcal{M},\, F>$. Systém schopný vyhledávat na základě znalosti deformačního schématu z hlediska zvoleného kritéria optimální přiřazení prvků nosiče $\tilde{\mathcal{M}}$ k prvkům nosiče $\mathcal{M}$ se nazývá *systém pro hledání izomorfismu s opravou chyb* (error–correcting isomorphism system). Možností přiřazení prvků nosiče $\mathcal{M}$ s n prvky je $n!$, přičemž uvedený systém musí vybrat takové přiřazení, které je etalonu nejbližší ve smyslu zvolené vzdálenosti.

Obvykle užívané vzdálenosti pro dané g-té přiřazení nosičů struktur pro struktury $\tilde{\mathcal{S}}$ a $\mathcal{S}$ jsou:

a) Při užití stochastického deformačního schématu se obvykle pracuje s *deformační pravděpodobností* definovanou vztahem

$$D_g(\mathcal{S}/\tilde{\mathcal{S}}) \;=\; \prod_{j=1}^{J} p(a_j/\tilde{a}_j)\, q(\mathbf{y}^j/a_j,\tilde{a}_j)\, \prod_{k=1}^{K} p(u_k/\tilde{u}_k)\, q(\mathbf{v}^k/u_k,\tilde{u}_k) \;,$$

v němž jsou vzaty v úvahu jak syntaktické, tak sémantické deformace.

b) Při užití váženého deformačního schématu se zavádějí dvě míry, a to

ba) *součtová vzdálenost* pro syntaktické deformace při g-tém přiřazení definovaná jako

$$W_g\,(\mathcal{S}/\tilde{\mathcal{S}}) \;=\; \sum_{j=1}^{J} w\,(a_j/\tilde{a}_j) \;+\; \sum_{k=1}^{K} w\,(u_k/\tilde{u}_k) \quad ,$$

kde $w\,(a_j/\tilde{a}_j)$, resp. $w\,(u_k/\tilde{u}_k)$ jsou váhy pro příslušnou syntaktickou deformaci,

bb) *kvadratická odchylka* pro sémantické deformace

$$E_g\,(\mathcal{S}/\tilde{\mathcal{S}}) \;=\; \sum_{j=1}^{J} \sum_{n=1}^{N} w\,(a_j)\,(\tilde{y}_n^j \,-\, y_n^j)^2 \;+$$
$$+\; \sum_{k=1}^{K} \sum_{d=1}^{D} w\,(u_k)\,(\tilde{w}_d^k \,-\, v_d^k)^2 \quad ,$$

kde $w\,(a_j)$, resp. $w\,(u_k)$ jsou váhy přiřazené symbolu a_j či elementu binární relace u_k.

Rozpoznávaná relační struktura $\tilde{\mathcal{S}}$ je pak přiřazena k takovému etalonu třídy $\hat{\mathcal{S}}$ a v takovém g-tém přiřazení, pro něž jsou struktury $\tilde{\mathcal{S}}$ a $\hat{\mathcal{S}}$ nejbližší ve smyslu zavedené definice vzdálenosti.

Algoritmus pro hledání izomorfismu s opravou chyb je časově náročnější než algoritmus syntaktického analyzátoru s opravou chyb při stejném počtu primitiv použitých pro vytvoření obrazu objektu, neboť u relačních struktur je zapotřebí navíc prohledávat jednotlivá přiřazení mezi strukturami $\tilde{\mathcal{S}}$ a $\hat{\mathcal{S}}$. V podstatě jde o poměrně zdlouhavé prohledávání stromových struktur reprezentujících odvození popisných struktur. Proto v reálných aplikacích bývají algoritmy pro hledání izomorfismu s opravou chyb doplněny některou z řídicích (prohledávacích) strategií probraných ve druhé kapitole skripta, čímž dojde k podstatnému zrychlení celého algoritmu (snížení jeho algoritmické složitosti), i když někdy za cenu nalezení pouze suboptimálního řešení.

5.5 Hybridní metody rozpoznávání

Strukturní metody rozpoznávání nacházejí v současné době poměrně značné uplatnění v praktických aplikacích díky jejich celkové efektivnosti podpořené výkonností současných výpočetních prostředků. Čím strukturně složitější objekty je nutné rozpoznávat, tím více vyniká efektivita strukturních metod nad metodami příznakovými. Obě skupiny metod si však navzájem nekonkurují, nýbrž se spíše doplňují, protože pro různé typy úloh nebo pro různé úrovně systémů rozpoznávání se lépe hodí jednou ten typ metody, podruhé zase onen. Čím dále většího významu nabývají metody kombinované, nazývané *hybridní*, které využívají výhod obou přístupů. Je to přirozené, neboť na jedné straně každý příznakově popsaný objekt má svoji vnitřní strukturu, kterou je možno alespoň částečně v procesu rozpoznávání využít, a na druhé straně se s ideálními, čistými, šumem nedeformovanými strukturními obrazy, které by bylo možno přiřadit k příslušné klasifikační třídě pouze na základě "čisté" syntaktické analýzy či na základě úplného izomorfismu relačních struktur (rozpoznávané struktury s etalonem třídy), v praxi téměř nesetkáváme.

Přísně vzato, o metodách hybridních můžeme hovořit již např. při použití stochastických gramatik, umožňujících určovat příslušnost obrazu klasifikovaného objektu ke třídě na základě numerické (pravděpodobnostní) informace, při užití sémantické informace apod. Na druhé straně, využití kontextové informace u příznakových metod rozpoznávání má mnohdy charakter syntaktického zpracování.

V poslední době se objevují pokusy vytvořit jednotící pohled na příznakové a strukturní metody rozpoznávání zejména na bázi atributových gramatik. Systémy strukturního rozpoznávání totiž obvykle nejsou používány izolovaně, ale bývají součástí složitějších systémů rozpoznávání. Veškerá vstupní a výstupní informace u nich často mívá čistě numerický charakter. Jejich činnost obecně probíhá ve třech fázích:

1. ve *fázi generativní*, během níž se vytvoří symbolický popis objektu,
2. ve *fázi syntaktické analýzy*, kdy je symbolický popis (obraz) objektu vytvořený v předchozí fázi podroben analýze a eventuální korekci s cílem přiřadit ho k některé z definovaných klasifikačních tříd,
3. ve *fázi sémantické analýzy*, kdy je výsledný popis objektu prověřován z hlediska významového, eventuálně je tento popis doplňován o sémantickou (číselnou) informaci.

Vytváření symbolického popisu objektu obvykle vychází z numerické informace (na objektu naměřených dat, výsledků hierarchicky nižší úrovně příznakového rozpoznávání apod.). Sémantická informace může také sloužit jako numerický výstup systému strukturního rozpoznávání.

Současné systémy rozpoznávání pro řešení složitějších úloh bývají koncipovány jako víceúrovňové, založené na aplikaci jak příznakových, tak i strukturních metod rozpoznávání. Problémem je ale formalizace přechodu od příznakových popisů k popisům strukturním a naopak přechod od metod strukturních k příznakovým na různých úrovních rozpoznávání. Jednou z možností této formalizace se ukazuje použití poměrně obecného aparátu atributových gramatik, kterého lze pro tento účel využít dvojím způsobem:

a) *generativně* (ve fázi generativní), kdy "vstupem" gramatiky je numerický vektor $\mathbf{v}$ a cílem činnosti metody je vygenerování symbolického (strukturního) popisu objektu. Po každém použití gramatického pravidla lze jako další pravidlo použít takové pravidlo $r_i \in R$, které

   i) je přípustné z hlediska gramatického,

   ii) je aktivováno, tj. jsou pro něj splněny podmínky z množiny $\mathcal{C}_i$.

   Pro generativní užití atributových gramatik je typické, že se neprovádějí změny ve vstupních datech, tj. množina procedur $\mathcal{P}$ je prázdná, pokud ovšem nejsou definovány speciální příkazy pro řízení generativního procesu nebo pro paralelní vytváření sémantických popisů.

b) *analyticky* (ve fázi sémantické analýzy), kdy je "vstupem" gramatiky syntaktický popis, včetně sémantických vektorů, a cílem činnosti metody je vygenerování vektoru numerických parametrů, tj. vektoru $\mathbf{v}$ gramatiky.

Přepisovací pravidla atributové gramatiky jsou používána tak, jak je při syntaktické analýze řetězů symbolů obvyklé. Je-li dané pravidlo $r_i \in R$ použito, provede se kontrola splnění podmínek z $\mathcal{C}_i$ a je-li výsledek této kontroly pozitivní, provedou se procedury (vykonají se příkazy) z $\mathcal{P}_i$ měnící (tvořící) některé parametry z $\mathbf{v}$. Nejsou-li podmínky z $\mathcal{C}_i$ splněny, je nutné při syntaktické analýze použít jiného vhodného přepisovacího pravidla. Není-li žádné takové pravidlo již k dispozici a analýza vstupního řetězu symbolů primitiv nebyla dokončena, je daný strukturní popis chybný.

Příklad 5. 5: Nechť je dána atributová gramatika [Kotek93]

$$G_A = < V_N\,, V_T\,, S\,, R\,, \mathcal{C}\,, \mathcal{P}\,, \mathbf{v} >$$

taková, že $V_N \equiv \{A,\, B\}$, $V_T \equiv \{a,\, b\}$, $\mathcal{P} \equiv \emptyset$, $\mathbf{v} = [\,v_1,\, v_2\,]$ a

pravidlo	množina R	množina $\mathcal{C}$
r_1	$S \longrightarrow aA$	$v_1 > 0.5$
r_2	$S \longrightarrow bA$	$v_2 > 0.3$
r_3	$A \longrightarrow aB$	$v_1 + v_2 > 0$
r_4	$A \longrightarrow bB$	$v_1 + v_2 \leq 0.07$
r_5	$B \longrightarrow a$	$v_1 \geq 0\,,\ v_2 \leq 1$
r_6	$B \longrightarrow b$	$v_1 < 0.7\,,\ v_2 > 0$

Použijeme ji generativně ($\mathcal{P} \equiv \emptyset$). Je-li vektor $\mathbf{v} = [\,v_1,\, v_2\,] = [\,0.6,\, -0.7\,]$, pak při tvorbě slova uvedenou gramatikou lze použít pouze pravidla r_1, r_4 a r_5 a vytvořeným slovem je slovo aba. Obdobně pro $\mathbf{v} = [\,-1,\, 1\,]$ dostaneme slovo bbb, pro $\mathbf{v} = [\,-1,\, 1.2\,]$ slovo bab atd. Všimněme si, že pro některé hodnoty parametrů není možné generovat žádné slovo jazyka (např. pro $\mathbf{v} = [\,0,\, 0\,]$), jindy je možné generovat více slov (např. pro $\mathbf{v} = [\,0.6,\, -0.55\,]$ jsou to slova

aaa a aba). Přípustný definiční obor parametrů musí být vždy definován tvůrcem gramatiky.

Vysoká obecnost formalismu atributových gramatik umožňuje jednoduše implementovat mechanismus pro účinné řízení procesu syntaktické analýzy. V takovém případě se zavádějí další parametry do vektoru $\mathbf{v}$, a to parametry bez fyzikální interpretace, sloužící jako "poznámky" či "upozornění" pro řídicí algoritmus vlastní syntaktické analýzy. Těchto parametrů lze ovšem využívat jen v rámci podmínek gramatiky (v rámci množin $\mathcal{C}_i$).

Příklad 5. 6: Rozšiřme u atributové gramatiky z předchozího příkladu vektor $\mathbf{v}$ o parametr v_3 a modifikujme první tři pravidla gramatiky takto [Kotek93]:

pravidlo	množina R	množina $\mathcal{C}$	množina $\mathcal{P}$
r_1	$S \longrightarrow aA$	$v_1 > 0.5$	$v_3 := 1$
r_2	$S \longrightarrow bA$	$v_2 > 0.3$	$v_3 := 0$
r_3	$A \longrightarrow aB$	$v_3 = 0$ a $v_1 + v_2 > 0$	
...	...	...	

Pak při vstupu $v_1 = 0.6$ a $v_2 = -0.55$ je generováno pouze slovo aba, protože použití pravidla r_3 je vyloučeno nastavením parametru v_3 příkazem $\mathcal{P}_1$.

Formalismus atributových gramatik pro potřeby vzájemného přenosu informace mezi systémy příznakového a strukturního rozpoznávání se ukázal být natolik účinným, že řada tvůrců metod rozpoznávání navrhuje zcela integrovat oba přístupy (příznakový a strukturní) jen do jednoho systému metod rozpoznávání založeného na bázi atributových gramatik. Efektivní dekompozice úlohy rozpoznávání by byla zajišťována strukturním popisem objektů, sémantická (numerická) informace by pak umožňovala minimalizovat počty primitiv a relací (podobně jak bylo uvedeno v příkladě 5. 3) a urychlovat tak proces syntaktické analýzy. Dále platí, že problémy zapsané v některém aparátu lze také zapsat prostřednictvím aparátů ostatních, avšak jiná vyjádření mohou být značně těžkopádná a náročná na zpracování. Vždy se proto snažíme použít takový popis, který je pro danou úlohu přirozený a v němž implementace algoritmů rozpoznávání vychází s minimální algoritmickou i paměťovou složitostí.