

BAKALÁŘSKÁ PRÁCE

ÚVOD DO INFERENCE



Pavel Dvořák
10. ledna 2002

Prohlášení

Tímto prohlašuji, že jsem bakalářskou práci vypracoval samostatně, za pomoci mého vedoucího bakalářské práce Mgr. Aleše Horáka a na základě zde uvedené literatury. Touto cestou bych mu rád poděkoval za odborné konzultace při řešení bakalářské práce.

V Brně 10. ledna 2002

.....
Pavel Dvořák

Obsah

1 Úvod	4
1.1 Metody umělé inteligence	4
2 Inferenční mechanismus	4
2.1 Indukce	5
2.2 Dedukce	6
2.2.1 Úvod	6
2.2.2 Dedukce ve výrokové logice	6
2.2.3 Hilbertovský formální systém výrokové logiky	8
2.2.4 Gentzenovský formální systém výrokové logiky	10
2.2.5 Shrnutí	11
2.3 Gramatiky a jazyk	11
2.3.1 Inference gramatik	12
2.4 Zpětné řetězení	13
2.5 Přímé řetězení	13
2.6 Redukce	13
2.6.1 Operátory redukce úloh na množinu podúloh	13
2.6.2 Popis elementárních úloh	14
2.6.3 Důkazy vět	15
2.6.4 Hry	15
2.7 Heuristiky	15
2.7.1 Význam heuristické informace	16
2.7.2 Význam heuristických metod	16
2.8 Analogie	17
2.8.1 Vysvětlování pomocí analogie dynamickou situací poznávání	18
2.9 Nemonotónní usuzování	19
2.10 Defaulty	19
2.11 Abdukce	19
2.12 Generování a testování	20
3 Aplikace inference	20
3.1 Systolické systémy	21
3.1.1 Definice:	21
3.2 Neuronové sítě	21
3.2.1 Specifika neuronových sítí	21
3.2.2 Model neuronu	22
3.2.3 Neuron prvé generace (McCulloch)	22
3.2.4 Podmíněný reflex	23
3.2.5 Perceptron	23
3.2.6 Adaptace perceptronu (Hebb)	24
3.2.7 Vícevrstvé sítě	24

3.3	Počítačové vidění	25
3.3.1	Snímání a digitalizace	26
3.3.2	Předzpracování obrazu	26
3.3.3	Segmentace	26
3.3.4	Popis nalezených objektů	26
3.3.5	Porozumění obsahu	27
3.3.6	Aplikace počítačového vidění	27
3.4	Počítačové zpracování přirozeného jazyka	27
3.4.1	Úvod	27
3.4.2	Automatické indexování textů	29
3.4.3	Automatické indexování pomocí TEZAURU	29
3.4.4	Tezaurus	30
3.4.5	Automatické referování	31
3.4.6	Překlad textů	31
3.5	Úvod do teorie her	32
3.5.1	Definice hry v normálním tvaru	32
3.5.2	Maticové hry	33
3.5.3	Definice smíšeného rozšíření hry	33
3.5.4	Nekonečné hry	34
3.5.5	Hry proti přírodě	34
3.6	Expertní systémy	34
3.6.1	Charakteristické rysy expertních systémů	35
3.6.2	Aplikace expertních systémů	35
4	Splnitelnost a platnost formule	36
4.1	Úvod	36
4.1.1	Algoritmy rozhodování splnitelnosti a platnosti formulí	37
5	Závěr	40

1 Úvod

Předmětem umělé inteligence je studovat a realizovat takové metody a postupy (inference), které aplikovány na řešení problémů dávají výsledky analogické těm, kterých by dosáhl při řešení stejných problémů člověk, přičemž výsledky získané člověkem budou všeobecně považovány za projev jeho inteligence nebo též jeho tvůrčí myšlení.

Umělá inteligence spolupracuje s mnoha vědními disciplínami, jde zejména o matematiku (matematická logika, matematická statistika, teorie automatů a formální jazyky apod.), dále o elektrotechniku, optiku, akustiku, psychologii, lingvistiku atd. Řada vědních disciplín umělé inteligence využívá výpočetní techniku, např. počítačovou grafiku, formální manipulace s výrazy či robotiku.

Kromě dalších oblastí umělé inteligence (porozumění přirozenému jazyku, analýza scény) lze vyvinutých metod (inferencí) použít např. pro automatické vyhodnocování radarových snímků, řízení a regulaci technologických celků (měření teplot, tlaků, koncentrace látek).

1.1 Metody umělé inteligence

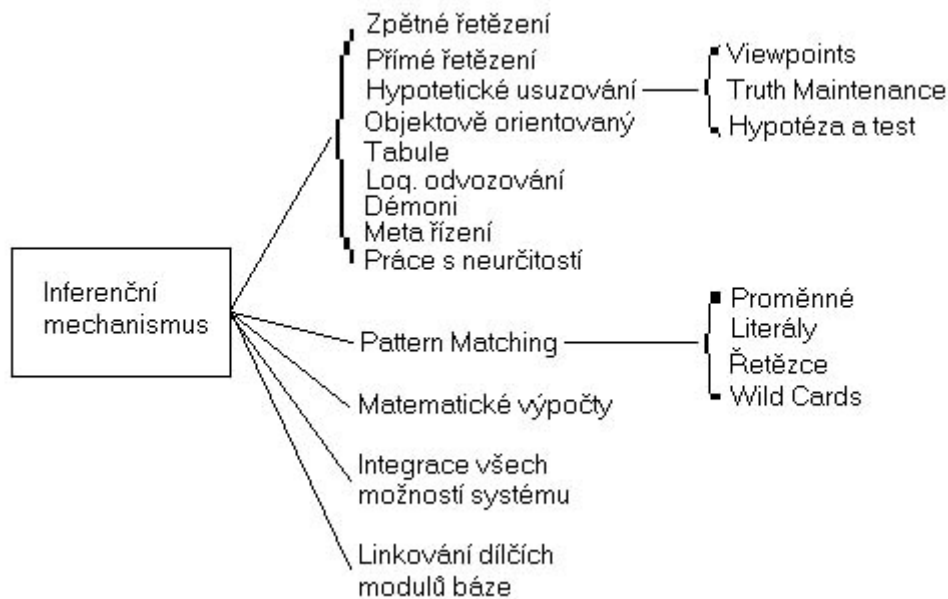
1. *Matematicko-logický* – snaží se použít účinných prostředků pro řešení úloh z oblasti umělé inteligence, které spočívají v aplikaci metod matematiky a logiky. V podstatě jde tedy o vypracování metod srovnatelných ve výsledcích s intelektuální činností lidí, které však nepátrají po principech mozkové činnosti lidí.
2. *psychologický* – vychází ze spolupráce s lékařem a psychology, snaží se simulovat postupy a metody používané člověkem při řešení úloh.

2 Inferenční mechanismus

Giarratano a Riley [16] uvádějí následující přehled typů inference (=odvozování): **dedukce, indukce, abdukce, heuristiky, analogie, defaulty, nemonotonní usuzování, generování a testování.**

Většina uvedených způsobů odvozování našla své uplatnění v expertních systémech. První tři typy (dedukce, indukce, abdukce) vycházejí z výrokové logiky z pojmu implikace.

Jiné možné rozdělení druhů inference [4]:



2.1 Indukce

Co je to **indukce**?

Indukce je zobecnění speciálních případů. Při indukci z opakovaného pozorování, že **A** a **B** se vyskytují současně, odvozujeme, že je mezi nimi implikace. Na indukci (generalizaci z příkladů) je založena většina metod strojového učení; tyto metody lze použít pro automatizované získávání znalostí z dat.

Definice principu matematické indukce:

Buď M množina, která má tyto dvě vlastnosti:

1. $1 \in M$,
2. pro každé přirozené číslo p platí: jestliže $p \in M$, potom je též $p + 1 \in M$.

Potom množina M obsahuje všechna přirozená čísla.

Příklad

Zadání: Dokažte, že číslo $2^{3k} + 3^{4k}$ není pro žádné přirozené číslo k dělitelné číslem 73.

Řešení: pro $k = 1$ věta platí, neboť $2^3 + 3^4 = 89$ a to není dělitelné číslem 73. Pomocná věta (1) je dokázána. Přístupme k důkazu pomocné věty (2). Buď p přirozené číslo a předpokládejme, že $2^{3p} + 3^{4p}$ není dělitelné číslem 73. Vyšetřme číslo $2^{3(p+1)} + 3^{4(p+1)}$. To je rovno

$$2^{3p+3} + 3^{4p+4} = 2^3 * 2^{3p} + 3^4 * 3^{4p} = 8 * 2^{3p} + 81 * 3^{4p} = 8 * (2^{3p} + 3^{4p}) + 73 * 3^{4p}.$$

První z obou sčítanců není podle indukčního předpokladu dělitelný číslem 73 a druhý je. Součet tedy číslem 73 dělitelný není.

Při zavedení dobře uspořádaných množin můžeme rozšířit pojem indukce.

Definice dobře uspořádané množiny:

Množina M se nazývá dobře uspořádaná, jestliže každá její neprázdná podmnožina má nejmenší prvek (M množina, $a \in M$ se nazývá nejmenší prvek, jestliže pro všechna $x \in M$: $a \leq x$).

Definice principu transfinite indukce:

Nechť M je dobře uspořádaná množina, $V(x)$ je vlastnost prvků z M . Dále nechť platí:

1. $V(x)$ platí pro nejmenší prvek v M (a_0 nejmenší prvek v M , $V(a_0) = 1$),
2. z platnosti $V(x)$ pro každé $x \in A(y)$, kde $y \in M$ libovolné, plyne i platnost $V(y)$.
Pak $V(x)$ platí pro každé $x \in M$.

Podrobnější informace lze nalézt v [1, 4, 18].

2.2 Dedukce

2.2.1 Úvod

Výroková logika by jako model lidského vnímání světa a myšlení o něm neměla valnou cenu, kdyby se v ní nedalo vyjádřit, co z čeho vyplývá. Jde o schopnost, která se v případě skutečného lidského myšlení nazývá schopnost dedukce.

2.2.2 Dedukce ve výrokové logice

Logický důsledek

Problém rozhodování, zdali určitá formula A vyplývá z množiny formulí U , se nazývá problém dedukce a je jedním ze základních problémů logiky. Ve výrokové logice hovoříme o formuli A vyplývající z množiny formulí U jako (tauto)logickém důsledku U . Množina formulí U je v tomto pojetí speciální množinou předpokladů (speciálních axiomů), na níž je postavena určitá teorie.

Formule A je (tauto)logickým důsledkem množiny formulí U , platí-li pro všechny modely množiny formulí U , že formule A je v nich splněna (true).

Tvrzení, že formule A je logickým důsledkem dané množiny formulí U obvykle vyjadřujeme $U \models A$.

Formule množiny předpokladů U nemusí být nutně tautologiemi. Je-li však U nesplnitelná množina, pak nemá žádný model a proto jejím logickým důsledkem je libovolná

formule (tj. lze získat formuli A i její negaci $\neg A$). Nesplnitelná množina předpokladů tedy vede ke sporným důsledkům.

Logickým důsledkem prázdné množiny formulí může být pouze tautologie (označuje se $\models A$).

Teorie a její axiomy z hlediska sémantiky

Pro matematické teorie je typické, že vycházejí vždy z nějaké množiny základních tvrzení – axiomů a dále, že platná tvrzení teorie jsou důsledky této zvolené množiny axiomů.

Teorii lze definovat takto:

1. je dána množina U výchozích formulí – speciálních axiomů (předpokladů) teorie,
2. množina $T(U)$ se nazývá teorií vybudovanou na U , je-li každý prvek množiny $T(U)$ formulí, která je logickým důsledkem U , platí-li tedy $T(U) = \{A \mid U \models A\}$.

Je zřejmé, že uvažujeme-li o dedukci, tedy o vyplývání výroků z jiných výroků, je třeba vzít v úvahu to, aby naše závěry byly platné v rámci daného způsobu interpretace.

Předpoklady a závěr dedukce

Problém dedukce bývá zpravidla formulován tak, že z množiny hypotéz $\{H_1, H_2, \dots, H_n\}$ vyplývá závěr Z . Množina hypotéz $U = \{H_1, H_2, \dots, H_n\}$ tedy tvoří speciální axiomy teorie a Z je jejím tautologickým důsledkem ($\{H_1, H_2, \dots, H_n\} \models Z$).

Je-li $U \models Z$, hovoříme o platnosti formule Z ve všech modelech množiny formulí U .

Příklad: (aplikace principu dedukce)

Zadání: Brown, James a Smith jsou podezřelí z daňového podvodu. Svědčili pod přísahou takto:

Brown: Jones je vinen a Smith je nevinen.

Jones: Je-li vinen Brown, pak je vinen i Smith.

Smith: Já jsem nevinen, ale nejméně jeden ze zbývajících je vinen.

Řešení: Označme postupně b, j, s výroky. Výpovědi všech tří podezřelých budou nyní vypadat takto:

Brown: $(\neg j \wedge s)$

Jones: $(\neg b \rightarrow \neg s)$

Smith: $(s \wedge (\neg b \vee \neg j))$.

b	j	s	$\neg j \wedge s$	$(\neg b \rightarrow \neg s)$	$(s \wedge (\neg b \vee \neg j))$
0	0	0	0	1	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	0	1	0
1	0	1	1	1	1
1	1	0	0	1	0
1	1	1	0	1	0

Vinen je zřejmě Jones.

TABULKOVOU metodu lze nahradit REZOLUČNÍ metodou.

Pravidla přirozené dedukce

Již od dob klasické logiky jsou známa některá pravidla, podle nichž probíhá deduktivní uvažování. Pravidla jsou tvaru výchozí dedukce \rightarrow závěr, tj. jestliže

$$A_{11}, \dots, A_{1n} \models B_1, \dots, A_{k1}, \dots, A_{kn} \models B_k,$$

pak platí

$$C_1, \dots, C_n \models D.$$

Sémanticky se tato implikace vyjadřuje ve tvaru zlomku tak, že nad čarou jsou uvedeny výchozí dedukce, pod ní dedukce odvozená z těchto výchozích dedukcí.

$$\frac{A_{11}, \dots, A_{1n} \models B_1, \dots, A_{k1}, \dots, A_{kn} \models B_k}{C_1, \dots, C_n \models D}$$

Pravidla přirozené dedukce se dělí do dvou skupin. První z nich zahrnuje pravidla strukturální (pravidlo oslabení, o permutaci, o redukci), druhou skupinu odvozovacích pravidel tvoří pravidla operační (pravidlo dodání konjunkce, odnětí konjunkce, dodání implikace, odnětí implikace, dodání disjunkce sukcedentu, dodání negace, pravidlo dvojí negace).

2.2.3 Hilbertovský formální systém výrokové logiky

Ještě před zavedením Hilbertovského a Gentzenovského formálního axiomatického systému je nutné zavést pojem formální systém.

Formální (axiomatický) systém, v němž se mají odvozování neboli důkazy vět provádět, musí mít definován:

1. *jazyk* – syntax, tj. množiny symbolů pro označení logických konstant, atomických formulí, symbolů pro výrokové spojky a pomocné symboly (,),
2. *axiomy* – formule představující základní věty tohoto jazyka,

3. *odvozovací pravidla* – pravidla umožňující na základě syntaxe základních vět odvodit nové věty.

Odvozovací pravidla spolu s axiomy jsou tím základem, v němž se různé formální systémy navzájem liší.

Hilbertovský formální systém

Odvozovací pravidlo má hilbertovský formální systém výrokové logiky jedině, a to pravidlo *modus ponens*

$$\frac{\vdash A \quad \vdash A \rightarrow B}{\vdash B}$$

„Z formulí (vět) A a $A \rightarrow B$ odvodí formuli (větu) B .“ Formule B se nazývá bezprostřední důsledek formulí A a $A \rightarrow B$.

Dále obsahuje Hilbertovský formální systém 3 axiomy:

- (1) $A \rightarrow (B \rightarrow A)$
- (2) $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
- (3) $((\neg B) \rightarrow (\neg A)) \rightarrow (A \rightarrow B)$

Věta o dedukci

Jedním z nejužitečnějších pravidel sloužících ke zjednodušení struktury zápisu důkazů je pravidlo dedukce vyjádřené takto:

$$\frac{U \cup \{A\} \vdash B}{U \vdash A \rightarrow B}$$

Korektnost a úplnost

Formální systém je sémanticky korektní, je-li každá v něm dokazatelná formule formulí logicky platnou (tautologie).

Formální systém je sémanticky úplný, existuje-li v něm ke každé formuli, která je logicky platná, její formální důkaz.

Postova věta

Formule dokazatelná v hilbertovském axiomatickém systému výrokové logiky jsou právě tautologie, tj. pro libovolnou formuli A platí: $\vdash A \Leftrightarrow \models A$ (formule A je dokazatelná právě když je pravdivá).

2.2.4 Gentzenovský formální systém výrokové logiky

Pro tablovou metodu zjišťování splnitelnosti výrokové formule je charakteristické vytváření určitých sekvencí podformulí zkoumané v každé z větví sémantického tabla podle pravidel vycházejících z vlastností formačních spojek těchto podformulí. Pravidla, podle nichž se posloupnost sekvencí vytváří, lze převést v pravidla spojování sekvencí podformulí ve výsledné formule. Na tom je založen gentzenovský formální systém výrokové logiky.

Gentzenovský formální systém

Gentzenovský systém disponuje narozdíl od Hilbertovského (tři axiomy a jedno pravidlo) pouze jedním axiomem a řadou odvozovacích pravidel tvaru:

$$\frac{S_1, \dots, S_n}{S}$$

Axiom gentzenovského formálního systému výrokové logiky je množina formulí U obsahující komplementární pár atomických formulí $\{p, \neg p\} \in U$.

Odvozovací pravidla gentzenovského systému jsou dvojího druhu:

a) α -pravidla daná schématem

$$\frac{U_1 \cup \{\alpha_1, \alpha_2\}}{U_1 \cup \{\alpha\}}$$

a tabulkou:

α	α_1	α_2
A	$\neg\neg A$	
$A_1 \vee A_2$	A_1	A_2
$\neg(A_1 \vee A_2)$	$\neg A_1$	$\neg A_2$
$A_1 \rightarrow A_2$	$\neg A_1$	A_2
$A_1 \leftarrow A_2$	A_1	$\neg A_2$
$\neg(A_1 \Leftrightarrow A_2)$	$\neg(A_1 \rightarrow A_2)$	$\neg(A_2 \rightarrow A_1)$

b) β -pravidla daná schématem

$$\frac{U_1 \cup \{\beta_1\} \quad U_2 \cup \{\beta_2\}}{U_1 \cup U_2 \cup \{\beta\}}$$

a tabulkou

β	β_1	β_2
$B_1 \wedge B_2$	B_1	B_2
$\neg(B_1 \vee B_2)$	$\neg B_1$	$\neg B_2$
$\neg(B_1 \rightarrow B_2)$	B_1	$\neg B_2$
$\neg(B_1 \leftarrow B_2)$	$\neg B_1$	B_2
$B_1 \Leftrightarrow B_2$	$B_1 \rightarrow B_2$	$B_2 \rightarrow B_1$

Tomuto systému se říká přirozená dedukce, neboť do značné míry vystihuje způsob přirozeného deduktivního uvažování.

2.2.5 Shrnutí

Dedukce je tedy logické usuzování, při kterém musí závěr plynout z předpokladů. Při dedukci musí být pravdivá (platit) implikace (pravidlo) a předpoklad; z toho můžeme jednoznačně odvodit i pravdivost závěru (modus ponens). Analogicky, pokud budeme předpokládat, že platí implikace a neplatí závěr, můžeme jednoznačně odvodit nepravdivost předpokladu (modus tollens). Dedukce je tedy způsob usuzování, které zachovává pravdu (truth preserving reasoning). Podrobnější informace lze nalézt v [5, 19].

2.3 Gramatiky a jazyk

Gramatika jazyka představuje soubor pravidel, pomocí nichž lze z prvků abecedy (množina všech primitiv = písmena) vytvářet slova (popisy předmětů) náležející do daného jazyka. Pomocí gramatiky lze charakterizovat i nekonečné jazyky.

K popisu formálního jazyka se používá gramatika, což je matematický model generátoru slov tohoto jazyka.

Gramatika je čtveřice pravidel $G = (V_n, V_t, P, S)$, kde V_n , V_t jsou disjunktní, prvky V_n se nazývají neterminální symboly, prvky V_t terminální symboly, S je tzv. axiom gramatiky neboli počáteční symbol a P je konečná neprázdná podmnožina množiny $V^* * V^*$, jejíž prvky se nazývají substituční pravidla, $V = V_n \cup V_t$.

Množina všech slov, která mohou být generována gramatikou G , se nazývá jazyk $L(G)$. Gramatiky, které generují tentýž jazyk, se nazývají ekvivalentní.

Podle tvaru substitučních pravidel se gramatiky dělí na 4 typy:

1. *gramatiky typu 0* – obecné gramatiky, nemají žádná omezení na tvar substitučních pravidel,
2. *gramatiky typu 1* – kontextové gramatiky, jejichž substituční pravidla mají tvar $W_1\alpha W_2 \rightarrow W_1UW_2$ a mohou obsahovat pravidlo $S \rightarrow \varepsilon$ (ε je prázdné slovo), $W_1, W_2, U \in V^*$, $U \neq \varepsilon$, $\alpha \in V_n$ (tz. neterminální symbol α lze nahradit slovem U v kontextu slov W_1 a W_2).
3. *jazyky typu 2* – bezkontextové gramatiky, jejichž substituční pravidla mají tvar $\alpha \rightarrow U$, kde $U \in V^*$, $U \neq \varepsilon$, $\alpha \in V_n$ a mohou obsahovat pravidlo $S \rightarrow \varepsilon$ (neterminální symbol lze nahradit slovem U nezávisle na kontextu, ve kterém je α použit).
4. *jazyky typu 3* – regulární gramatiky, jejichž substituční pravidla mají tvar $\alpha \rightarrow x\beta$ nebo $\alpha \rightarrow x$, kde $\alpha, \beta \in V_n$, $x \in V_t$ a mohou také obsahovat pravidlo $S \rightarrow \varepsilon$.

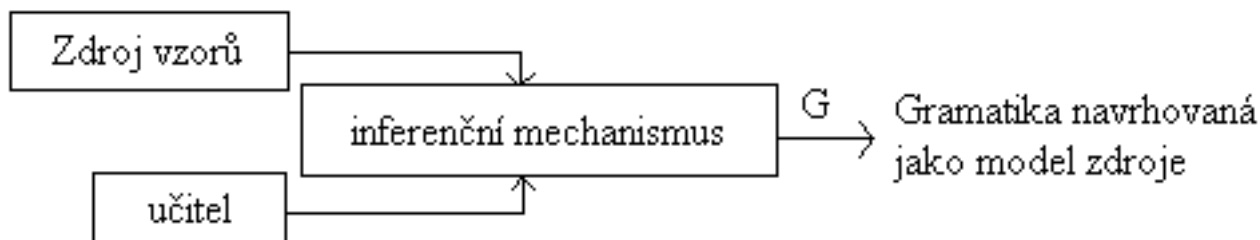
Gramatiky, o nichž jsme hovořily se nazývají nedeterministické. Odpověď na to, zda dané slovo je generováno danou gramatikou či ne, dává postup zvaný syntaktická analýza.

Je-li gramatika regulární (typu 3), je syntaktická analýza jednoduchá. Gramatika se nahradí ekvivalentním konečným nedeterministickým automatem a snadno se zjistí, zda slovo x je automatem přijato či ne.

V případě bezkontextových gramatik G (typu 2) je to pracnější. Existuje způsob realizace zásobníkovým automatem a to shora dolů (od kořene stromu, od počátečního symbolu gramatiky) nebo zdola nahoru.

2.3.1 Inference gramatik

Aby bylo možné modelovat jazyk určité třídy vzorů co nejsprávněji, je snaha vytvořit gramatiku a její pravidla přímo z množiny vzorových obrazů. Tento problém určení gramatiky na základě vzorových vět se nazývá inference gramatik.



Zdroj vzorů generuje řetězy složené z konečného počtu terminálních symbolů. Všechna slova, která mohou být generována zdrojem, jsou obsažena v množině $L(G)$ – v jazyce generovaném gramatikou G , zatímco slova, která nemohou být zdrojem generována, tvoří doplněk množiny $L^C(G)$. Tato informace je vstupem algoritmu inference, jeho úkolem je nalézt neznámou gramatiku G . Slova z $L(G)$ lze snadno získat sledováním výstupu zdroje, avšak prvky $L^C(G)$ musí určit vnější učitel, který má k dispozici další informaci o vlastnostech gramatiky G .

Počet vzorových slov je konečný a to nestačí k jednoznačné definici potenciálně nekonečného jazyka $L(G)$. Jakékoli konečné množině vzorů může být přiřazen nekonečný počet jazyků. Proto zpětně nemůžeme nikdy očekávat jednoznačnou identifikaci gramatiky, která generovala vzorové řetězy. Od inferenčního algoritmu očekáváme, že vytvoří gramatiku, která popíše vzorovou množinu a navíc další slova, která mají v určitém smyslu stejnou strukturu jako vzorová.

Obecně lze metody inference rozdělit do dvou kategorií:

1. ENUMERACE – výběr takové gramatiky z konečné množiny M gramatik, která generuje celou vzorovou množinu nebo alespoň její největší část.
2. INDUKCE – na základě tvaru slov z trénovací množiny se usuzuje, jak by měla vypadat slova podobná slovům z trénovací množiny, a podle toho se stanovují substituční pravidla.

Obecná metoda inference gramatik na základě předloženého seznamu slov jazyka dosud neexistuje, vytvořené metody lze použít pro regulární a bezkontextovou gramatiku a pro

některé další speciální případy. Gramatiky jsou konstruovány na základě heuristik, intuice, zkušenosti a samozřejmě apriorních informací o úloze. Podrobnější informace lze nalézt v [7, 20].

2.4 Zpětné řetězení

Zpětné řetězení (backward chaining) vychází z toho, že máme odvodit nějaký cíl. V bázi znalostí existují pravidla, které mají tento cíl ve svém závěru. Tyto pravidla se tedy pokoušíme aplikovat (za použití dedukce). Abychom zjistili, zda je pravidlo aplikovatelné, musíme vědět, zda platí jeho předpoklad. Pokud je v předpokladu dotaz, lze se na jeho pravdivost zeptat uživatele, pokud je v předpokladu mezilehlý výrok, musíme ho dovodit (podobně jako cíl) z pravidel, které k němu vedou. Celý proces se tak opakuje.

Procházíme tedy bázi znalostí zpětně od cíle k dotazům (od závěrů pravidel k předpokladům). Zpětné řetězení je tak analogické s prohledáváním stavového prostoru do hloubky.

Pořadí, v jakém se vyhodnocují pravidla, je v nejjednodušším případě dáno pořadím podle priority (ta je buď pevně zadána při tvorbě báze, nebo se mění v průběhu konzultace). Podrobněji v [4].

2.5 Přímé řetězení

Při přímém řetězení (forward chaining) vycházíme z faktů, které jsou splněny a pokoušíme se nalézt nějaký závěr, to umožní nalézt další aplikovatelná pravidla a v odvozování lze pokračovat. Přímé řetězení v čisté podobě znamená, že systém se už uživatele na nic neptá; všechny „odpovědi“ musí být zadány před začátkem konzultace. Je zde jistá analogie s prohledáváním stavového prostoru do šířky.

Existují systémy, které umožňují jak zpětné, tak přímé řetězení. Podrobněji v [4].

2.6 Redukce

Předložená úloha je dekomponována na řadu dílčích podúloh (podcílů), které se dále mohou strukturovat v ještě jednodušší úlohy. Celý proces je rekurzivně opakovan až do okamžiku, kdy je celá úloha rozdělena na řešení triviálních elementárních úloh, na jejichž vyřešení má výpočetní systém k dispozici prostředky.

Tyto metody je možné použít na řadu praktických úloh, jako je například dokazování vět, symbolické integrování jako formálních manipulací s výrazy, redukci hledání výhry u řady her na hledání dalšího vhodného tahu, např. u šachů, dámy apod.

2.6.1 Operátory redukce úloh na množinu podúloh

Operátory redukce jsou v konkrétních implementacích navrhovány tak, aby umožnily rozdělení (transformaci) předloženého popisu úlohy na množinu podúloh, a transformují popis dané úlohy na množinu rezultujících popisů úloh. Uvažovaná transformace musí být taková,

že řešení všech rezultujících úloh zaručuje řešení původní úlohy. Je-li rezultující množina jednoprvková, potom jde o záměnu úlohy za úlohu s ní ekvivalentní.

V praktických aplikacích je možné, že existuje více alternativ redukce předložené úlohy na podúlohy. Některé z těchto redukcí se mohou ukázat jako neperspektivní, protože některé z dílčích úloh mohou být neřešitelné.

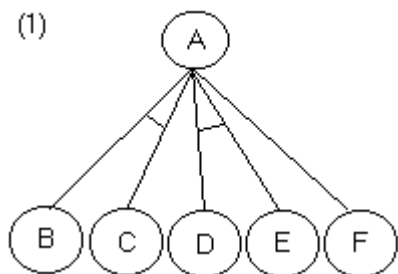
2.6.2 Popis elementárních úloh

Cílem redukce je předložit k řešení takové elementární úlohy, jejichž řešení je zřejmé. Takovými úlohami mohou být úlohy odpovídající jednomu kroku ve stavovém prostoru, případně i úlohy složitější, jejichž řešení je ovšem známo. Úroveň elementárních úloh hraje roli při ukončení procesu redukce nebo i při výběru vhodné alternativy redukce. Do mechanismu redukce je možno zahrnout i proces učení, v němž některé často se opakující řešitelné úlohy lze po úspěšném vyřešení zahrnout mezi elementární úlohy.

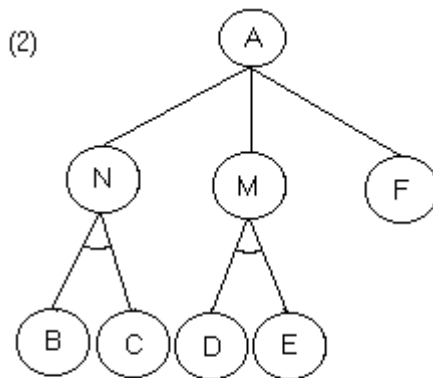
Příklad:

Pro zobrazení řešených problémů redukcí na alternativní množiny rezultujících problémů je užitečné používat grafové struktury.

Uzly M, N, F jsou příklady uzlů typu NEBO (OR-node). Úlohu N lze převést na množinu podúloh B a C, přičemž obě musí být vyřešeny. Uzly B, C jsou příklady uzlu typu A (AND-node) a hrany k nim vedoucí jsou spojeny v transformačním grafu obloučkem. Tyto grafy se též nazývají AND/OR grafy.



A je řešitelná jako řešení podúloh B a C nebo jako řešení podúloh D a E nebo jako úloha F.
převaditelná na ekvivalentní úlohu F.



Zavedeme dodatečné uzly, aby hrany grafu jdoucí k dané alternativě reprezentujících problémů vycházely z takových uzlů. Zde N odpovídá {B,C}, M odpovídá {D,E}.

2.6.3 Důkazy vět

Pro důkazy vět platí následující obecné schéma. Nechť se má dokázat tvrzení S za předpokladu platnosti podmínek T – schématicky S/T . Princip redukce na podproblémy spočívá v tom, že nelze většinou tento důkaz provést přímo. Je nutné zformulovat řadu nových tvrzení X_1, X_2, \dots, X_n tak, že postupně dokážeme:

$$\begin{aligned} & X_1/T \\ & X_2/T, X_1 \\ & \dots \\ & X_n/T, X_1, X_2, \dots, X_{n-1} \\ & S/T, X_1, \dots, X_n \end{aligned}$$

přičemž v každém kroku vlastně dostáváme nové dílčí tvrzení.

2.6.4 Hry

Metoda redukce problémů na podproblémy může být úspěšně použita i pro jisté typy her; speciálně u antagonistických her dvou hráčů, soupeři se střídají v tazích a je jim známo, jaké jsou jejich možnosti v této hře. Půjde nám o hry, v nichž jeden z hráčů vyhraje (a druhý prohraje), případně se hráči shodnou na remíze. Příkladem mohou být hry typu piškvorky, go, dáma nebo šachy. Neuvažujeme zde hry, v nichž se projevuje prvek náhody, např. hry v kostky či karetní hry.

Proces redukce problému je prováděn za účelem toho, abychom v procesu důkazu ukázali, že daná hra může být vyhrána. Při popisu problému je tedy nutno popsat počáteční stav hry (též konfigurace), pro kterou se má dokázat, že ji lze vyhrát. Například u šachů musí být známo rozestavení figurek na šachovnici a to, který soupeř je na tahu. Aplikace operátorů (tj. tahů) ve hře vede k vytvoření struktury, zvané graf hry. Elementárními úlohami jsou pravidla ukončení hry a proces důkazu je realizován do té doby, dokud nejsou nalezeny elementární úlohy. Ve většině her není možné explicitně generovat strom hry vzhledem k jeho rozsahu.

Podrobnější informace v [2, 16].

2.7 Heuristiky

Heuristiky jsou pravidla založená na zkušenosti. Heuristiky (heuristická pravidla) je označení pro znalosti používané v expertních systémech. Jde o znalosti založené na zkušenostech experta, na zobecnění situací, ve kterých se expert rozhodoval.

2.7.1 Význam heuristické informace

Metody slepého prohledávání vedou k nalezení optimálního řešení. Při jejich aplikaci je nutno generovat neúměrně velký počet uzlů, což vede k vysokým nárokům na čas a paměť využitého počítače. V mnoha úlohách lze zformulovat pravidla, která zmenšují rozsah výběru ve stavovém prostoru. Informaci tohoto druhu nazveme *heuristickou informací*. Jednou z možností je zavedení „inteligentního“ operátoru Γ , který bude redukovat počet generovaných uzlů tak, že se omezí pouze na „nejsnadnější“ stavy. Druhý způsob používá heuristickou informaci, kdy je vybírán pro expanzi z právě generovaných uzlů vždy ten nejnadějnější. Příslušnou funkci nazveme ohodnocující funkci.

2.7.2 Význam heuristických metod

1. určují speciální postupy, jimiž je užitečné se řídit při řešení,
2. míří k programování úloh, které bude vykonávat počítač (např. klasifikace dat, konstrukce všech možných kombinací apod.).

Heuristické postupy jsou do značné míry ovlivněny subjektem, jsou nepřenosné a proto se setkáváme s pokusy vymezit standardní vlastnosti heuristických postupů [4, 16, 17]:

1. existuje zásobník různých metod,
2. existuje zásobník zkušeností se získanými řešeními,
3. tyto zkušenosti jsou opatřeny příznaky a identifikátory, respektujícími popřípadě i vztahy k zásobníku řešení úloh,
4. vyskytne se úloha, jejíž řešení není známo ve formě standardního algoritmu
5. pokus o popsání nové úlohy pomocí příznaků a identifikátorů,
6. najde se dohodnutá (minimální) míra vhodných příznaků a následně je možno použít pro řešení nové úlohy těch řešení, která byla vybrána.

Jde vlastně o nalezení náhradního řešení v případě, že přímé řešení není známo, ale pravidla pro hledání a nalezení jsou v zásobě řešení (vlastních i převzatých).

Jiné typy heuristik:

- *výběr po etapách* – při konstrukci výběrového stromu se může stát, že paměť počítače je vyčerpána ještě před nalezením řešení. V tom případě je možné ponechat v paměti pouze jistý výsek tohoto grafu, čímž se paměť uvolní pro další hledání řešení úlohy,
- *ohraničení počtu expandovaných uzlů* – je používán „lépe informovaný“ operátor Γ , který pro další řešení vybírá pouze nejperspektivnější uzly.

Nejznámější heuristický program je „Obecný řešitel problémů“ (GPS) autorů Newella, Shawa a Simona [17]. Tento program byl zkonstruován původně pro dokazování logických teorémů, ale může se použít i pro řešení jiných úkolů, svou strukturou analogických původním (důkazy geometrických vět, transformace geometrických výrazů, hra v šachy apod.). Pro tyto úlohy není možné sestavit efektivní algoritmus vedoucí k řešení. Jediný algoritmus, který můžeme zkonstruovat, je algoritmus pokusů o řešení. Ale použití metody postupného vyzkoušení všech variant se ukazuje prakticky nemožné. Proto je nutné zúžit množství prověřovaných variant, tj. aplikujeme speciální pravidla (heuristiky).

Jediný rozdíl mezi heuristickými programy a algoritmy je rozdílnost v rezultativnosti (algoritmus je vždy zaměřen na získání hledaného výsledku, lze ho dosáhnout – ale ne při libovolných výchozích datech). Systém použitých pravidel v heuristických programech není úplný, tj. heuristické programy

1. nezaručují vyřešení,
2. když řešení dosáhneme, je ho dosaženo s menšími ztrátami času a prostředků.

Jsou to vlastně „neúplné“ algoritmy.

2.8 Analogie

Analogie je odvození závěru na základě podobnosti s jinou situací. Analogie se používá např. při případovém usuzování. Místo aby znalosti měly podobu (obecných) pravidel získaných od experta, jsou tvořeny souborem dříve vyřešených (typických) případů. Případové usuzování lze přirovnat k americkému právu založenému na precedentech, usuzování na základě pravidel je analogické evropskému (kontinentálnímu) pojetí práva. Výhodou je, že na rozdíl od klasických expertních systémů není třeba pracně získávat znalosti (pravidla) od experta, stačí „jen“ získat dostatek reprezentativních případů.

Analogie reprezentuje všechny druhy či typy podobnosti (od geometrické podobnosti až třeba po izomorfismus a homomorfismus).

Analogie je nezbytnou vše pronikající formou chápání, založenou na strukturním izomorfismu naší mnohotvárné zkušenosti se světem. V některých případech tyto izomorfismy překračují hranice kategorií jako když promítáme strukturu z jedné oblasti – zpravidla konkrétnější nebo více názorné na jinou oblast, zpravidla abstraktnější, pojmově neurčitou nebo nefyzické povahy.

Je pozoruhodnou vlastností lidí snadno porozumět novým situacím pomocí analogií se starými situacemi, řešit problémy na základě dříve řešených analogických problémů.

V oblasti AI (umělé inteligence) je zapotřebí pochopit tuto schopnost, aby ji bylo možné vložit do „inteligentních strojů“. Usuzování podle analogie je mocným nástrojem poznání, protože nám umožňuje přenášet poznatky do známé dobře vysvětlené situace na méně známou a dosud nejasnou situaci a to „s dramatickou úsporou myšlení“. Protože usuzování podle analogie je centrem inteligence, měla by se AI pokoušet porozumět tomuto fenoménu a modelovat jej na počítači. To je vědecký cíl AI. Pokud jde o technický cíl vytvářet inteligentní stroje – je usuzování podle analogie potřebné ve většině aplikací AI.

Neschopnost využít minulou zkušenost k řešení problémů v soudobých expertních systémech je jedním ze slabých míst rozsáhlých počítačových expertíz.

Zájem AI je vytvořit počítače s uvažováním „zdravého rozumu“. Jsou proto zkoumány různé formy nededuktivního usuzování, jako jsou non-monotonní logika, bayesiánská pravděpodobnost, induktivní postupy atd.

Vyvozování pomocí analogie se zdá být nejobecnějším druhem závěru a je to možná nejvýznamnější druh. Poskytuje více nebo méně pravděpodobné domněnky, které mohou nebo nemusí být potvrzeny zkušeností a přesnějším vyvozováním.

2.8.1 Vysvětlování pomocí analogie dynamickou situací poznávání

Je to situace, na jejímž počátku je zřetelná nevědomost něčeho, co má být projasněno a vysvětleno. Vysvětlování má jasný směr a plně popsateľný cíl. Cíl je srozumitelný a jasný vysvětlujícímu, zprvu je nejasný a nesrozumitelný tomu, komu je vysvětlování určeno.

Vysvětlující se snaží využít něčeho, co je oběma osobám (vysvětlujícímu a poučovánému) dobře známo a co se má stát prostředkem usnadňujícím pochopení čehosi nesrozumitelného nebo neznámého.

Je veden rozhovor, vysvětlování, popisování a to v neustálé opoře o analogické vztahy ze známého k neznámému. Úspěšnost vysvětlování je v důvěře a spolehnutí založena na předkládaných analogických souvislostech. V takto založeném vysvětlování přichází porozumění z analogie.

Na počátku analogické situace stojí nevědomost něčeho, která má být překonána, objasněna a prosvětlena pomocí předcházejícího známého obsahu. Analogie je metodou poznávání, tedy heuristickou metodou.

Příklad

Volná analogie mezi pojmem reálného čísla a pojmem zobecněné funkce

reálná čísla	zobecněná funkce
racionální čísla	spojité funkce
cauchyovské posloupnosti racionálních čísel	fundamentální posloupnosti spojitých funkcí
ekvivalentní cauchyovské posloupnosti	ekvivalentní fundamentální posloupnosti
reálné číslo je třída ekvivalentních cauchyovských posloupností racionálních čísel	zobecněná funkce je třída ekvivalentních fundamentálních posloupností spojitých funkcí
iracionální čísla	zobecněné funkce, které se se neredukují na spojité funkce

Podrobnější informace v [4, 10].

2.9 Nemonotónní usuzování

Hlavní vlastností *nemonotónního usuzování* je, že předcházející znalosti se mohou revidovat na základě nových poznatků. Je založeno na skutečnosti, že předcházející znalosti mohou přestat platit, dozvíme-li se další informace.

Dedukce v pravém slova smyslu vyžaduje úplnou informaci o zkoumaném problému. Potřebujeme však zpracovávat i neúplné informace a tak musíme být připraveni na to, že naše závěry budou tentativní. V těchto situacích logika nabízí nemonotónní usuzování jako takový druh argumentu (inference), který může být zpochybněn ve světle nově přicházející informace. Vypomáháme si defaultovými pravidly tvaru

$$\frac{\alpha; \beta}{\gamma}$$

kteřá mohou být interpretována takto: jestliže α platí a β může být konzistentně předpokládáno, pak odvod γ .

Formule α se nazývá předpoklad (precondition), β je ospravedlnění (justification) a formule γ se nazývá závěr (consequent) defaultu. Hlavním cílem defaultových logik je vhodným způsobem definovat pojem rozšíření (extenze) teorie s defaulty, což má být analogie pojmu logického uzávěru.

Vývoj nemonotónních inferenčních mechanismů není zdaleka uzavřen a stal se součástí znalostního inženýrství. Podrobněji v [4]

2.10 Defaulty

Pokud nejsou k dispozici speciální znalosti, uvažuje se na základě obecných znalostí. Usuzování za použití defaultů (default reasoning) bývá doplněním usuzování na základě pravidel. Není-li při dané konzultaci aplikovatelné žádné pravidlo, doporučení se odvodí z defaultů (např. můžeme v naší zeměpisné šířce předpokládat, že chřipka je častější choroba než malárie, aniž bychom se ptali pacienta na příznaky).

Podrobnější informace v [4].

2.11 Abdukce

Abdukce je usuzování z pravdivého závěru na předpoklady, které mohly tento závěr způsobit. Při abdukci vycházíme z platnosti implikace a závěru. Z tabulky pravdivostních hodnot pro implikaci je vidět, že předpoklad může být pravdivý nebo nepravdivý. Lze se tedy jen domnívat, že předpoklad může platit. Někdy je abdukce označována za odvozování nejlepšího vysvětlení pro pozorovaná fakta. Abdukce zachovává nepravdu (falsity preserving reasoning); když budeme předpokládat, že platí implikace a neplatí závěr, lze jednoznačně říci, že neplatí předpoklad.

2.12 Generování a testování

Generování a testování – metoda pokusů a omylů. Při generování a testování se opakovaně generuje možné řešení a testuje se, zda vyhovuje všem požadavkům. V případě, že nalezneme vyhovující řešení, cyklus končí. Znalosti jsou v těchto systémech reprezentovány pravidly

IF nastane_situace THEN proved_akci.

V daném okamžiku běhu systému mohou být splněny podmínky více pravidel.

Podrobnější informace můžete nalézt v [2, 16].

3 Aplikace inference

Umělá inteligence nachází stále více uplatnění v aplikačních oblastech. John McCarthy [15] navrhuje tyto aplikační oblasti:

- *hraní her (nalezení herní strategie)* – dnes je možno koupit programy, které hrají na mistrovské úrovni šachy za pár set dolarů. Je v nich nějaká UI, ale proti lidem hrají především díky použití „hrubé síly“ výpočetní techniky – prohlíží stovky tisíc pozic. K překonání mistra světa hrubou silou i dnes známými heuristikami by vyžadovalo prohlížet 200 milionu pozic za sekundu.
- *rozeznávání řeči* – v 90. letech dosáhlo praktických výsledků pro limitované účely. Dnes je možné instruovat počítač řečí, ale většina uživatelů se vrátila ke klávesnici a myši jako k vhodnějším.
- *porozumění přirozenému jazyku* – pouhé uložení jednotlivých slov do počítače nestačí. Ani syntaktická analýza není dostatečná. Počítač musí být seznámen s informacemi o té oblasti, které se text týká, což je dnes možné jen pro určité omezené oblasti.
- *počítačové vidění* – svět je složen z třídímných objektů, ale do lidského oka i do kamery robota vstupuje 2D obraz. Některé systémy pracují v 2D, ale plnohodnotné počítačové vidění vyžaduje 3D informaci, která není jen množinou 2D obrazů. V současnosti jsou jen omezené možnosti, jak reprezentovat 3D scénu, a ty nejsou ani zdaleka tak dobré, jako lidské oko.
- *expertní systémy* – znalostní inženýr zpovídá experta v dané oblasti a snaží se vložit jeho znalosti do systému.
- *neuronové sítě* – šance simulovat některé funkce lidského myšlení.
- *heuristická klasifikace* – jednou z nejpříjemnějších variant expertních systémů je vložit informaci do jedné z fixních kategorií pomocí několika zdrojů informací. Příkladem může být rozhodnutí o poskytnutí úvěru z údajů o klientovi, jeho solventnosti, druhu úvěru apod.
- *systolické systémy*.

3.1 Systolické systémy

3.1.1 Definice:

Systolický systém je multiprocesorový systém, který obsahuje jen několik málo typů poměrně jednoduchých procesorů. Meziprocesorové spoje mají důsledně lokální charakter, směr toku dat a topologie spojů mezi procesy jsou pro celý algoritmus řešení dané úlohy neměnné. V době činnosti systému pracují současně všechny jeho procesory. Každý procesor systému pracuje jen se svou lokální pamětí a komunikuje jen se svými bezprostředními sousedy v síti. Výsledky operací jednotlivých procesorů (tj. dílčí výsledky výpočtu prováděného celým systolickým systémem) se neustále přesouvají se zadanými zpožděními k sousedním procesorům. Všechny procesory daného typu (resp. stejně naprogramované a konfigurované universální procesory) provádějí v době činnosti systému tytéž pro daný typ procesoru pevně určené operace. Výsledky jsou předávány na výstupy procesorů se zpožděními, která jsou také pro odpovídající si výstupy procesorů daného typu totožné. Totožný je i způsob propojení procesorů daného typu s procesory sousedními (s eventuální výjimkou okrajových procesorů).

První systolické systémy byly navrženy pro rychlé a efektivní vykonávání početně náročných matematických operací (násobení, rozklad). Oblastí využití postupně přibývalo, systolické systémy se začaly uplatňovat při třídění a vyhledávání dat, při realizaci front a zásobníků, při kódování a dekódování, při provádění různých transformací atd. Tyto systémy jsou vlastně technickou realizací vysoce efektivních pásových algoritmů a jejich činnost výhodně nahrazuje programovou interpretaci těchto algoritmů. Nejvíce se systolických systémů využívá v systémech pro zpracování obrazu a rozpoznávání, a to především v roli preprocesorů pro úvodní zpracování obrazové informace.

Dále je možné rozlišit jednoúčelové (homogenní a heterogenní) a víceúčelové systolické systémy. Podrobněji v [11, 12].

3.2 Neuronové sítě

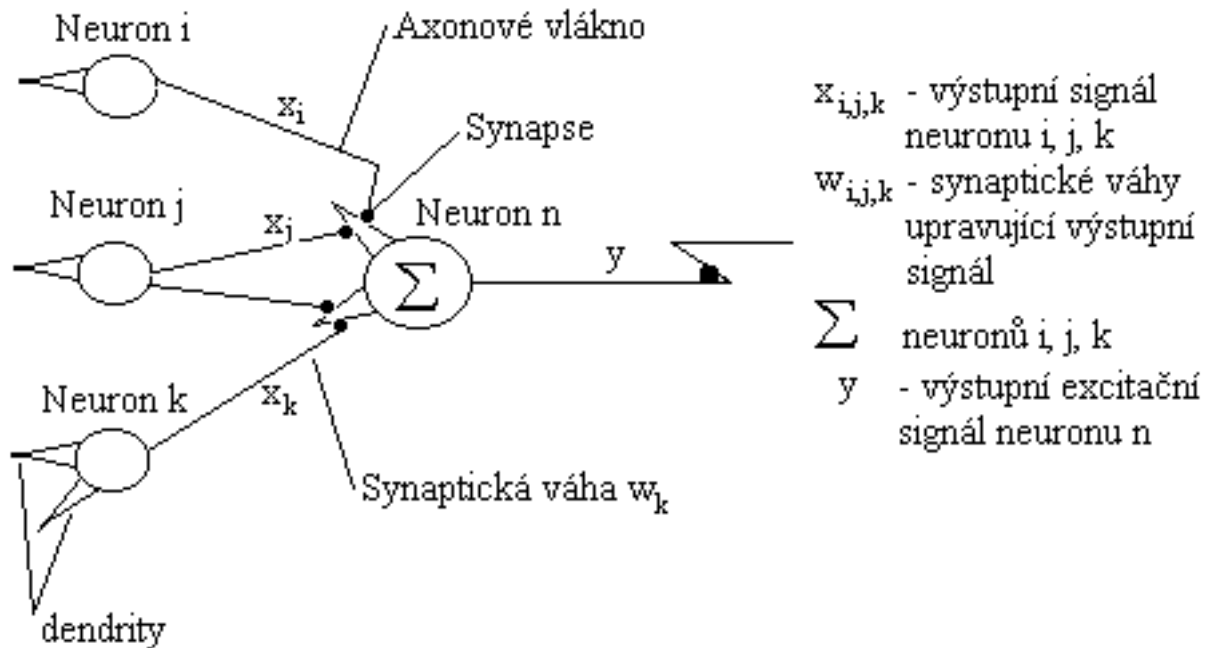
3.2.1 Specifika neuronových sítí

1. *Neuronové sítě jsou biologickými neuronovými sítěmi* – uměle vytvořené neuronové sítě by měly být schopny se chovat stejně nebo alespoň podobně jako jejich biologické vzory. Skýtá se tu šance simulovat některé funkce lidského myšlení.
2. *Neuronové sítě využívají distribuované, paralelní zpracování informace při provádění výpočtů* – ukládání, zpracování a předávání informace probíhá prostřednictvím celé neuronové sítě spíše než pomocí určitých paměťových míst.
3. *Znalosti jsou ukládány především prostřednictvím síly vazeb* mezi jednotlivými neurony. Vazby mezi neurony vedoucí ke „správné odpovědi“ jsou oslabovány pomocí opakované expozice příkladů popisujících problémový prostor.

4. Učení je základní a podstatná vlastnost neuronových sítí.

3.2.2 Model neuronu

Uměle vytvořený neuron je dán svým biologickým vzorem a tvoří jakousi základní „výpočetní jednotku“ složitějšího komplexu – neuronové sítě.



Dendrity – reprezentují místo vstupu signálu do těla neuronu.

Tělo buňky – sčítá signály dané okolními neurony. Takto stanovený vnitřní potenciál vede k excitaci (vybuzení) neuronu.

Axonové vlákno – přináší signál daný stupněm excitace k synapsím.

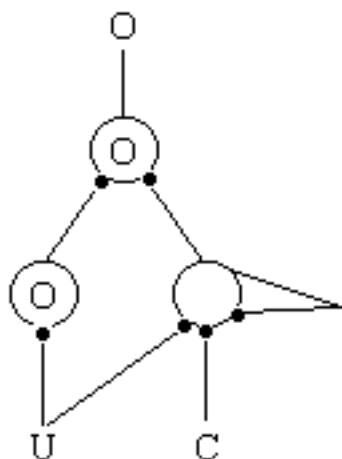
Synapse – tvoří výstupní zařízení neuronů, které signál zesilují či zeslabují a předávají dalším neuronům.

3.2.3 Neuron první generace (McCulloch)

Jedním z prvních modelů neuronu byl navržen McCullochem a jeho zjednodušení spočívalo v zavedení excitačních resp. inhibičních vazeb neuronu. První typ vazby je reprezentován synaptickou vahou rovné hodnotě 1 a druhý pak hodnotou 0. Každý neuron má definovanou svou vnitřní hodnotu prahu, která musí být překonána vnitřním potenciálem neuronu, aby došlo k jeho vybuzení (hodnota výstupního signálu 1). Tento jednoduchý způsob definice umožňuje modelovat různé procesy, např. podmíněný reflex.

3.2.4 Podmíněný reflex

Model se skládá ze třech neuronů s definovanou hodnotou prahů, dvou vstupů (nepodmíněných U-unconditioned a podmíněný C-conditioned), jednoho výstupu (podmíněný reflex O-output) a pouze z excitačních vazeb. Pokud bychom pro přenos signálu neuronové síti zavedli hodiny, pak jednotlivé takty reprezentované tabulkou dokumentují proces vyvolání podmíněného reflexu (takt 6 a 7), kdy vstupní signál je přiveden pouze na vstup C a následuje odezva systému rovna hodnotě 1.



Podmíněný reflex

T	U	C	L	P	O
1	1	0	1	0	0
2	0	0	0	0	1
3	0	1	0	0	0
4	0	0	0	0	0
5	1	1	1	1	0
6	0	1	1	1	1
7	0	0	0	0	1

U,C - vstup systému

L, P - levý a pravý neuron

O - výstupní signál třetího neuronu

3.2.5 Perceptron

Jedním z nejdůležitějších modelů dodnes používaných je tzv. perceptron, jehož potenciál je definovaný jako vážený součet vstupních signálů. Pokud tento vnitřní potenciál neuronu překoná jeho prahovou hodnotu ϑ , dojde k excitaci neuronu na hodnotu 1. V opačném případě je neuron inhibován, což je reprezentováno hodnotou 0.

Matematicky to lze vyjádřit jako funkce signum:

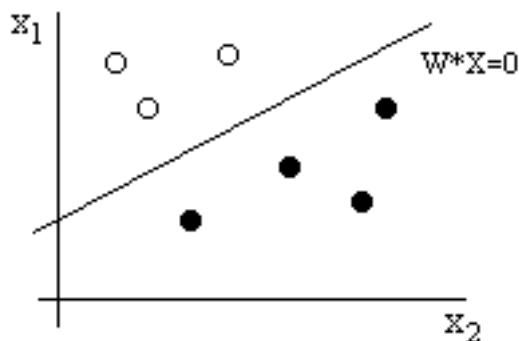
$$y = Sgn\left(\sum_{i=1}^n w_i x_i - \vartheta\right) \dots Sgn(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}$$

Zavedením stálého neuronu na vstupu se stavem excitace $x_0 = 1$ a vazbou k našemu neuronu $w_0 = -\vartheta$ lze předchozí zjednodušit takto:

$$y = Sgn\left(\sum_{i=1}^n w_i x_i\right)$$

Pokud provedeme analýzu výrazu v závorce, získáme rovnici nadroviny (v dvourozměrném případě reprezentovanou přímkou).

$$\left(\sum_{i=1}^n w_i x_i\right) = \bar{W} * \bar{X} = 0$$



Tato rovina rozděluje vstupní prostor na dva poloprostory. Jinými slovy, jsme schopni prostřednictvím perceptronu rozlišit dvě třídy vstupů. Otázkou nyní je, jak stanovit hodnoty vah neuronu, aby byl schopen správně rozpoznávat předložené vstupy. Jedním z nejznámějších principů je adaptace (učení perceptronu rozlišit dvě třídy vstupů perceptronu).

3.2.6 Adaptace perceptronu (Hebb)

1. Inicializace vah a prahu náhodnými malými čísly $w_i(t)$ – váha vstupu i v čase t .
2. Předložení vstupu a požadovaného výstupu z trénovací množiny $(x_0, \dots, s_n) \rightarrow d(t)$.
3. Stanovení skutečné odezvy.

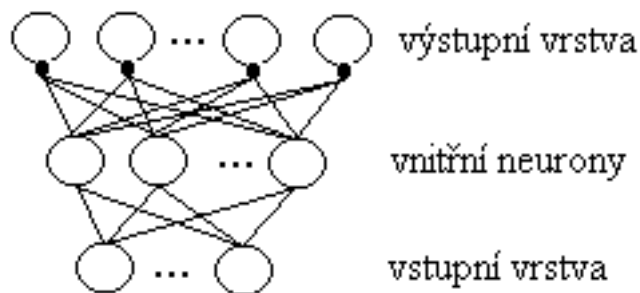
$$y(t) = \text{Sgn}\left(\sum_{i=1}^n w_i(t)x_i(t)\right)$$

4. Adaptace

výstup je správný	$w_i(t + 1) = w_i(t)$
výstup je 0 a měl být 1	$w_i(t + 1) = w_i(t) + x_i(t)$
výstup je 1 a měl být 0	$w_i(t + 1) = w_i(t) - x_i(t)$

3.2.7 Vícevrstvé sítě

Pravděpodobně nejrozšířenější způsob propojení perceptronů jsou tzv. vícevrstvé sítě, jejichž topologie je následující:



Neuronová síť je tvořena minimálně třemi vrstvami neuronů: vstupní, výstupní a alespoň jednou vnitřní vrstvou. Vždy mezi dvěma sousedními vrstvami se pak nacházejí tzv. úplné propojení neuronů.

Podrobnější informace můžete nalézt v [8].

3.3 Počítačové vidění

Počítačové vidění je disciplína, která se snaží technickými prostředky aspoň částečně napodobit lidské vidění. Pro počítačové vidění je typická snaha porozumět obecné trojrozměrné scéně, např. takové, jakou zahlédnete při pohledu z okna do zahrady. Postupy počítačového vidění jsou značně složité, s těžištěm v interpretaci obrazových dat, která jsou nejčastěji reprezentována symbolicky. Jádrem pokročilejších postupů jsou znalostní systémy a techniky umělé inteligence. Této části počítačového vidění se říká vyšší úroveň. Druhou částí je nižší úroveň.

Cílem nižší úrovně je analyzovat vstupní dvojrozměrná obrazová data číselného charakteru a najít kvalitativní symbolickou informaci potřebnou pro vyšší úroveň. Pro nižší úroveň se také používá název zpracování obrazu počítačem.

Předmětem zpracování a případné rozpoznání obrazu je obrazová informace o reálném světě, která do počítače vstupuje nejčastěji televizní kamerou. Počítačové vidění řeší úlohu vytvoření explicitního popisu fyzikálních objektů v obraze.

Ve zpracování obrazu se dá rozlišit několik úrovní. Na nejvyšší úrovni jde o proces porozumění. Představme si např. úlohu automatického vyhodnocení rentgenového snímku srdce. Zpracování je jednat ovlivněno cílem – snahou najít chorobou zasaženou část srdeční stěny včetně možných příčin onemocnění, a na druhé straně souhrnem předběžných znalostí a zkušeností v této problematice.

Postup zpracování a rozpoznávání obrazu se daří rozložit do posloupnosti základních kroků:

- snímání a digitalizace a uložení obrazu v počítači,
- předzpracování,
- segmentace obrazu objekty,

- popis objektů,
- pozorování obsahu obrazu (klasifikace objektů).

Při posuzování algoritmů počítačového vidění se berou v úvahu jejich časové a paměťové požadavky.

3.3.1 Snímání a digitalizace

Při snímání se převádějí vstupní optické veličiny na elektrický signál spojitý v čase i úrovni. Vstupní informací může být jas (z TV kamery, scanneru), intenzita rentgenového záření, ultrazvuk, tepelné záření aj. Snímat se může v jednom nebo více spektrálních pásmech. Pro barevné snímání stačí tři spektrální složky – červená, zelená a modrá.

Digitalizací se převádí vstupní spojitý signál odpovídající monochromatickému obrazu do diskrétního tvaru. Vstupní analogový signál je popsán funkcí $f(i,j)$ dvou proměnných – souřadnic v obraze. Funkční hodnota odpovídá např. jasu. Vstupní signál je vzorkován a kvantován. Výsledkem je matice přirozených čísel popisujících obraz. Jednomu prvku matice se říká obrazový element (picture element – pixel). Z hlediska zpracování obrazu jde o dále nedělitelnou jednotku.

Existují i jiné možnosti reprezentace vstupního obrazu v počítači. Častým případem je popis obrazu koeficienty dvourozměrné Fourierovy transformace. Výhodou je to, že Fourierovu transformaci lze převést okamžitě optickými prostředky již před digitalizací.

3.3.2 Předzpracování obrazu

Cílem je potlačit šum a zkreslení vzniklé při digitalizaci a přenosu dat. Jindy se předzpracování snaží zvýraznit určité rysy obrazu podstatné pro další zpracování. Příkladem může být hledání hran v obraze, tj. obrazových bodů (pixelů) s vysokými hodnotami velikosti gradientu obrazové funkce.

3.3.3 Segmentace

Asi nejtěžší krok postupu zpracování je segmentace, která dovolí v obraze najít objekty. Za objekty se považují ty části obrazu, které nás z hlediska dalšího zpracování zajímají. Při segmentaci se tedy zhusta využívá znalosti interpretace obrazu (sémantika).

3.3.4 Popis nalezených objektů

Lze je popsat buď kvantitativně pomocí souboru číselných charakteristik a/nebo kvalitativně pomocí relací mezi objekty. Způsob popisu objektů je ovlivněn tím, na co se popis bude používat. Za krajně jednoduchý popis objektů lze považovat např. stanovení velikosti (plochy) objektů, tj. počet jemu odpovídajících obrazových bodů v obraze.

3.3.5 Porozumění obsahu

Ve velmi jednoduchém případě můžeme za porozumění považovat klasifikaci objektů v obraze podle jejich velikosti. Jen o málo složitější je klasifikace objektů do několika předem známých tříd, např. na hranaté a kulaté. V obecném případě představuje porozumění interpretaci obrazových dat, o kterých se předem nic nepředpokládá. Porozumění obrazu je potom založeno na znalosti, cílech, tvorbě plánu k jejich dosažení a využití zpětných vazeb mezi různými úrovněmi zpracování.

Příklad

1. *jednoduchý* – jednoduchá automatická analýza transparentního obrazu buněčného preparátu pozorovaného optickým mikroskopem. Nechť je cíl spočítat buňky a roztřídit je podle tvaru buněčných jader na podlouhlé a ostatní. Úloha je z principu dvourozměrná. Předzpracováním se odstraní případný šum a pro segmentaci lze využít skutečnosti, že buněčná jádra jsou mnohem tmavší než zbytek. Nyní zbývá popsat body obrazu, příslušné každému buněčnému jádru, číselnými charakteristikami a podle nich buňky třídit.

2. *složitý systém* – porozumění obrazu používané v autonomních vozidlech. Cílem je vybavit terénní vozidlo pohybující se v blíže neomezené krajině bez řidiče. Poměrně dokonalé experimentální verze již byly vytvořeny. Systém analýzy obrazu je jen jeden z mnoha senzorů, na jejichž základě se řídicí systém vozidla rozhoduje. Jinými zdroji informací jsou např. radar, laserový dálkoměr, informace o absolutní poloze vozidla odvozená z gyroskopu, digitální mapa. Scéna je trojrozměrná a navíc v čase proměnná. Obrazový systém obvykle pracuje s informací ze dvou kamer, aby pomocí techniky stereovidění mohl odhadovat hloubku objektů ve scéně od pozorovatele. Vozidlo si musí vytvářet a průběžně modifikovat vnitřní model okolního světa.

3.3.6 Aplikace počítačového vidění

Jednou z oblastí využití počítačového vidění je dálkový průzkum země (lesnictví, geodézie, meteorologie, archeologie, ...), lékařské aplikace (rozpoznávání rakovinných buněk, texturní analýza myokardu v echokardiografii, ...), technická diagnostika (defektoskopie a defektometrie prozařování) a aplikace ve výrobě (počítadlo lahví, obsluha textilních strojů).

Podrobnější informace můžete nalézt v [7, 8].

3.4 Počítačové zpracování přirozeného jazyka

3.4.1 Úvod

Počítačové zpracování přirozeného jazyka představuje velkou výzvu a perspektivní zaměření výzkumu a vývoje v celé řadě praktických činností člověka s informacemi. Může jít například o:

- *datovába textů* – textové záznamy je třeba třídít, vyhledávat v nich a to pokud možno s ohledem na obsah těchto textů v přirozeném jazyce,
- *překlad textů (či mluveného slova)* – úplná náhrada překladatele počítačem nebo různé úrovně podpory překladatele počítačem,
- *báze znalostí* – úloha automatického učení z textů (případně i z mluvené řeči), neboli automatický převod textových (ústně sdělovaných) informací do formalizované podoby, ve které by se s nimi dalo snadno logicky manipulovat,
- *textové editory* – minimálním požadavkem dnes je automatizovaná korektura aspoň na úrovni pravopisu (překlepů), automatická kontrola gramatické a stylové správnosti napsaných textů. Vrcholnou možnost zřejmě představuje automatický převod mluvené řeči na text,
- *způsob komunikace mezi člověkem a strojem* – např. ovládání operačního systému, získávání informací z databáze, od expertního systému, objednávka, rezervace.

Přirozený jazyk jako systém sloužící ke komunikaci mezi lidmi, se obvykle popisuje dvěma základními „datovými (informačními)“ strukturami: *slovníkem (lexikem) a gramatickou (mluvnicí)*. Slovníkem rozumíme množinu slov, která lze v daném jazyce použít, nebo která v něm mají nějaký význam (lexikální sémantika). Zatímco v teorii formálních jazyků [7, 20] je „gramatika“ pojímána jako uniformní systém pravidel popisujících způsoby, jakými se z „abecedy“ sestavují „slova“, v lingvistice (jazykovědě) se obvykle rozlišuje několik úrovní gramatiky:

1. *úroveň fonetická (fonologická)* – popisuje systém hlásek daného jazyka a obecná pravidla jejich kombinování ve slovech, můžeme hovořit spíše o grafématické (grafické, znakové) úrovni,
2. *úroveň morfologická (morfologicko-lexikální, lexikálně-morfologická)* – popisuje utváření slov a slovních tvarů z určitých základních jednotek nesoucích nějaký význam (tzv. morfů),
3. *úroveň syntaktická* – popisuje způsob skládání slov (slovních tvarů) do frází a vět – jednotek reprezentujících komplexnější myšlenky.

Otázku náročnosti jednotlivých úloh počítačového zpracování přirozeného jazyka bychom tedy mohli formulovat takto: Jakou míru „sémantické hloubky“ – čili jakou míru komplexnosti lexikálního a gramatického rozboru – musí systém zahrnovat? Stačí např. rozlišovat jednotlivá slova, nebo slova a jejich tvary, nebo je nutno rozlišovat i jejich syntaktické funkce, případně něco ještě „hlubšího“?

Základní výzkumně-aplikační disciplína, která se této problematice věnuje, se nazývá aplikovaná lingvistika, počítačová lingvistika nebo počítačová lingvistika. Jako její blízký obor je třeba jmenovat algebraickou lingvistiku, neboli formální lingvistiku – vědu zkoumající přirozený jazyk jako matematický model.

Je třeba si uvědomit, že přirozený jazyk je velmi složitý systém a jeho používání člověkem asi nikdy nebude exaktně popsáno, aby mohlo být plně napodobeno počítačem. V následujících oblastech řekneme známe reálné možnosti a meze automatizace několika základních úloh zpracování přirozeného jazyka – indexování textů, vytvoření tezurů, referování (tvorby abstraktů), překladu textů, učení (extrakce elementárních znalostí) z textů, korektury textů.

3.4.2 Automatické indexování textů

Indexováním nazýváme proces přiřazení selekčních obrazů (výraz nebo množina výrazů určitého selekčního jazyka např. všechna podstatná jména, předem daná podstatná jména, ...) dokumentům nebo dotazům. Následně dochází k zatřídění selekčních obrazů dokumentů do nějaké struktury umožňující vyhledávání požadovaných prvků, resp. porovnávání.

Jako klíčový problém automatického indexování textů, který je nutno vyřešit, pak zbývá určení, která slova textu nejlépe charakterizují jeho celkový obsah. Lingvistické problémy automatického indexování lze rozdělit do těchto okruhů:

- významnost jednotlivých slov (slovních spojení) pro charakterizaci obsahu celého textu,
- tvarosloví (morfologie) přirozeného jazyka,
- synonymie a jí podobné sémantické vztahy mezi slovy a slovními spojeními,
- hynonymie (nejednoznačnost) výrazů přirozeného jazyka.

3.4.3 Automatické indexování pomocí TEZAURU

Jedním z těžko odstranitelných problémů automatického indexování textů výrazy vybranými přímo z textu je fakt, že informační požadavek může být vyjádřen i jinými výrazy přirozeného jazyka. V zásadě lze rozlišit přinejmenším tři typy takových situací:

1. dva různé výrazy mají stejný význam (synonyma, někdy též ekvivalence) – např. klisna = kobyła,
2. dva různé výrazy jsou ve vztahu *obecné: konkrétní* (nadřazené – podřazené), např. zvíře > hospodářské zvíře > kůň > hřebeč ap.,
3. dva různé výrazy mají pouze v širším smyslu něco společného neboli jsou sémanticky asociované – např. kůň – dostih.

Daný problém představuje „koeficient selekční významnosti“. Klasickým nástrojem řešení obou zde uvedených problémů je indexování dokumentů.

3.4.4 Tezaurus

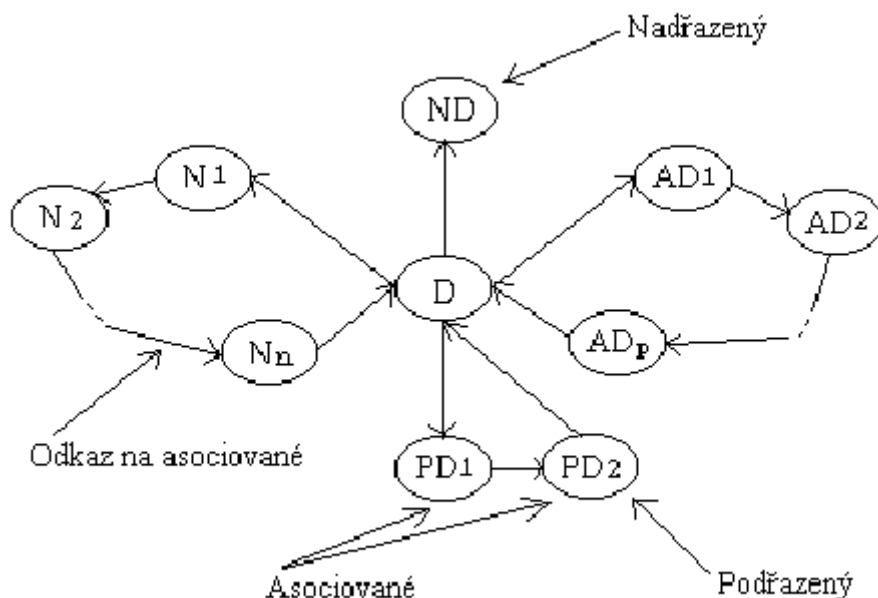
Termínem tezaurus se obecně nazývá slovník obsahující:

1. slovní zásobu (tj. slova a sousloví se specifickým významem) určitého jazyka či několika jazyků,
2. ke každému heslu co nejúplnější seznam odkazů na jiná hesla, která k němu mají sémantický vztah – např. synonyma,
3. případně další informace o heslech – např. historie jejich vzniku, příklady kontextů použití, ...

Tezaury nemusí sloužit pouze pro indexování, ale např. jako pomůcka pro autory textů. Tezaury určené pro indexaci jsou zpravidla omezené na terminologii určité užší odborné oblasti.

Ústředním pojmem je **deskriptor** – z každé skupiny synonymních termínů je vybrán jeden reprezentant, který by měl být používán k popisu (deskripci) skutečného nebo požadovaného obsahu textu. Všechna ostatní synonyma jsou nazvána **nedeskriptory**.

Pouze mezi deskriptory se zachycují vztahy (vazby) – nadřazený, podřazený, asociovaný, ekvivalentní (pouze u více jazyků) deskriptor. Počítačový záznam musí vedle odkazů na ekvivalentní nedeskriptory efektivně poskytovat také odkazy na všechny nadřazené, podřazené a asociované deskriptory.



3.4.5 Automatické referování

V teorii zpracování přirozeného jazyka je obvykle **referát (abstrakt)** uváděn jako jeden z možných výstupů intelektuálního procesu nazývaného informační analýza dokumentů.

Definice referátu dle ČSN:

Referát je zkrácený výklad obsahu dokumentu (nebo jeho části) s hlavními věcnými údaji a závěry, který zdůrazňuje nové poznatky a umožňuje rozhodnout se o účelnosti studia původního dokumentu.

Rozdíl mezi úlohami indexování a referování by mohl být formulován velice jednoduše: může-li být cílem indexování extrahovat (resp. formálně odvodit) z textu vhodný počet slov či sousloví, která nejlépe vystihují o čem text je, pak cílem referování může být extrahovat (resp. formálně odvodit) z textu vhodný počet vět, které nejlépe vystihují, co text přináší nového. Rozlišujeme automatické referování založené na tezauru či založené na měření obsahových souvislostí mezi větami.

3.4.6 Překlad textů

Na celém světě se dnes každodenně překládá asi 150 milionů stran textu. Z toho jen 0,3 % připadá na krásnou literaturu, 35 % tvoří obchodní informace, 21 % průmyslové, 20 % vědecké, 9 % právní, ... Tedy minimálně 85 % informací spadá do "oblasti zájmu". Tyto informace jsou většinou zpracovány počítači (od psaní, šíření, informační analýzu a indexování až po vyhledávání a využívání). Bylo by tedy žádoucí, aby se počítače zapojily do překladu do jiných jazyků.

Strojový překlad 1. generace

Systémy strojového překladu tzv. 1. generace typicky pracovaly způsobem konečného automatu – vytvářely překlad typu „slovo za slovo“. Jednalo se tedy o velmi hrubý překlad a bylo nutnost se nezřídky vracet k originálu při upravování do podoby odpovídající gramatice výstupního jazyka.

Strojový překlad 2. generace

nazýváme systémy, ve kterých jsou nějakým konzistentním způsobem odděleny pracovní fáze řešící:

- analýzu konstrukcí vstupního jazyka a vyjádření jejich významu určitými formalizovanými prostředky,
- vlastní překlad elementárních výrazů vstupujícího jazyka odpovídajícími (v daném kontextu) výrazy vystupujícího jazyka,

- syntézu konstrukcí výstupního jazyka potřebných ke konkrétním vyjádřením daného významu.

Strojový překlad 3. generace

můžeme nazývat ty, které uplatňují některé přístupy umělé inteligence.

Podrobnější informace můžete nalézt v [6].

3.5 Úvod do teorie her

Předpokládáme, že o systému, v němž rozhodujeme, vše podstatné víme, nebo můžeme předem zjistit a že systém nevykazuje vlastní aktivitu, kterou by mohl ovlivnit konečný efekt našich rozhodnutí.

3.5.1 Definice hry v normálním tvaru

Dva hráči, označme je A a B , hrají hru podle těchto pravidel: je dána množina X s prvky x a množina Y s prvky y . x a y budou zpravidla čísla nebo funkce. Dále je dána omezená reálná funkce $E(x, y)$, definována pro všechny dvojice (x, y) , kde $x \in X$, $y \in Y$. Hráč A volí z množiny X prvek x a hráč B volí z množiny Y prvek y , přičemž obě volby jsou na sobě nezávislé. Potom hráči své volby zveřejní a hráč B „zaplatí“ hráči A částku $E(x, y)$. Předpokládá se, že oba hráči znají X, Y a tvar funkce $E(x, y)$.

Množinu X budeme nazývat prostorem strategií hráče A a Y prostorem strategií hráče B . Prvky jednotlivých strategií x , y nazveme pak strategiemi hráčů A a B . Funkci $E(x, y)$ nazveme *výplatní funkci*. Nevylučujeme záporné hodnoty $E(x, y)$, v tom případě hráč A zaplatí hráči B částku $|E(x, y)|$.

V souvislosti s hrami budeme dále vyšetřovat pouze tyto dva případy chování hráče B :

- Hráč B je inteligentní a snaží se minimalizovat výplatní funkci, tj. částku $E(x, y)$, kterou má zaplatit hráči A . V tom případě budeme mluvit o hře hrané podle **minimaxu**.
- Hráč B je lhostejný k výši částky $E(x, y)$, kterou má zaplatit hráči A . V tom případě budeme mluvit o hře hrané **proti přírodě**.

V případě her hraných podle minimaxu mohou logicky nastat čtyři případy:

1. Je-li x' taková optimální strategie hráče A a y' optimální strategie hráče B , je rozumné požadovat, aby platilo $E(x, y') \leq E(x', y') \leq E(x', y)$ pro všechny $x \in X$, $y \in Y$. Levá nerovnost vztahu říká, že jestliže se hráč A odchýlí od své optimální strategie, může na tom nejvýše prodělat, neboť jeho protivník je inteligentní a bude tedy volit svou optimální strategii y' . Pravá nerovnost podobně říká, že odchylka od optimální strategie u hráče B může znamenat nejvyšší ztrátu pro něj. Vidíme také, že je x , y' bodem, ve kterém funkce $E(x, y)$ nabývá maxima pro $x \in X$ a současně minima pro $y \in Y$. Odtud název **minimax**. Číslo $E(x, y)$ se nazývá *cena hry*.

2. Existuje optimální strategie x' hráče A a neexistuje optimální strategie hráče B. Mníme tím, že pro každou volbu strategie hráče B je pro A výhodná volba x' , tj. $E(x, y) \leq E(x', y)$ pro všechna $x \in X, y \in Y$, zatímco pro hráče B neexistuje strategie y' , která by splňovala $E(x', y') \leq E(x', y)$ pro všechna $y \in Y$.
3. Existuje optimální strategie hráče B a neexistuje optimální strategie hráče A.
4. Neexistuje optimální strategie pro žádného z hráčů.

Příklady her z bodů 2, 3 a 4

$$X = \langle 0, 1 \rangle, Y = \langle 0, 1 \rangle, E(x, y) = x + y$$

$$X = \langle 0, 1 \rangle, Y = \langle 0, 1 \rangle, E(x, y) = x + y$$

$$X = \langle 0, 1 \rangle, Y = \langle 0, 1 \rangle, E(x, y) = x + y$$

3.5.2 Maticové hry

Definice maticové hry

Je dána matice čísel

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

Hráč A zvolí řádku, hráč B sloupec, potom oba své volby zveřejní a hráč B zaplatí hráči A tolik, kolik udává číslo na průsečíku řádky zvolené A a sloupce zvoleného B. Je-li číslo záporné, rozumíme tím, že A platí B jeho absolutní hodnotu.

Příklad

$$A = \begin{pmatrix} 5 & 3 & 4 & 4 \\ 3 & 2 & 5 & 7 \\ 7 & 1 & 8 & 6 \end{pmatrix}$$

x' je optimální strategie hráče A a y' je optimální strategie hráče B. Prvek $a_{12} = 3$ má totiž vlastnosti požadované ze vztahu (3). Cena hry je $a_{12} = 3$.

3.5.3 Definice smíšeného rozšíření hry

Budiž dána hra v normálním tvaru s prostory strategií X', Y' , jejichž prvky jsou čísla, a s výplatní funkcí $E'(x', y')$. Smíšené rozšíření této hry je hra v normálním tvaru s prostory strategií X, Y a s výplatní funkcí $E(x, y)$, kde za prostor X se bere nějaká množina pravděpodobnostních rozložení na X' a za prostor Y nějaká množina pravděpodobnostních rozložení na Y' a za výplatní funkci funkce

$$E(x, y) = \int E'(x', y') dx(x') dy(y')$$

Integrál v posledním vztahu se chápe jako Lebesque-Stieltjesův a $x(x')$ značí distribuční funkci rozložení x a podobně, $y(y')$ distribuční funkci rozložení y .

3.5.4 Nekonečné hry

Při maticových hrách měl každý z hráčů konečný počet akcí, pro které se mohl rozhodnout. Hráč A mohl volit jeden z konečného počtu řádků, hráč B jeden z konečného počtu sloupců.

Mnohé rozhodovací situace však vyžadují provést jednu akci z nekonečně mnoha možných – nekonečné hry.

Zatímco každou maticovou hru lze řešit pomocí lineárního programování, u nekonečných her neexistuje žádná univerzální metoda řešení. Situace je podobná jako při řešení diferenciálních rovnic, pro každý typ her je nutné použít jiný způsob řešení, pokud ovšem vůbec nějaký způsob znám je.

Některé z her lze popsat pomocí bodů konečněrozměrných euklidovských prostorů.

3.5.5 Hry proti přírodě

Hlavní otázka je, jak optimálně rozhodovat v situacích, které lze sice modelovat hrou v normálním tvaru, ale kde není rozumné předpokládat, že by hráč jednal tak, aby záměrně snížil hodnotu funkce $E(x, y)$. To odpovídá situacím, kde protějškem hráče A je příroda nebo instituce, které je počínání hráče A lhostejné. Pro rozhodování se nejčastěji používá Bayesův princip:

Nechť X, Y jsou prostory strategií, $E(x, y)$ je výplatní funkce. Známe-li jakou strategii $y' \in Y$ bude proti hráči A příroda používat, je optimální strategií hráče A ta strategie $x' \in X$, pro kterou je

$$E(x, y') = \int E'(x, y') dy'(y')$$

maximální. Ve výrazu značí $y'(y')$ distribuční funkci strategie y' .

Je možné hru dvou osob rozšířit na hru n osob a to buď s nulovým součtem nebo hru n osob s nenulovým součtem. Podrobněji v [9].

3.6 Expertní systémy

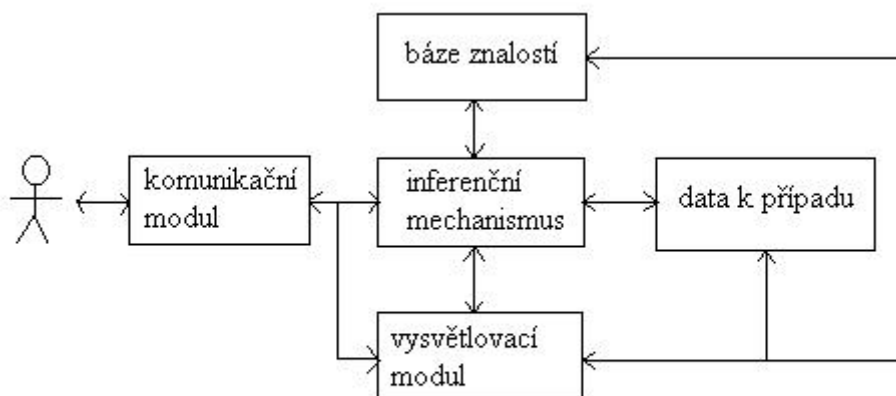
Co je to expertní systém?

Expertní systémy jsou programy pro řešení takových úloh, které jsou všeobecně považovány za obtížné a jejich uspokojivé řešení může provést pouze specialista v daném oboru (expert). Expert se při řešení opírá o svoje znalosti a své vlastní zkušenosti. Expertní systémy jsou založeny na myšlence převzetí znalostí od experta a jejich vhodné reprezentaci tak, aby je mohl využívat program podobným způsobem jako expert a zejména s podobným výsledkem. Expertní systémy jsou tedy praktickou aplikací umělé inteligence.

3.6.1 Charakteristické rysy expertních systémů

1. *oddělení znalostí a mechanismu* pro jejich využití – znalosti experta jsou uloženy v bázi znalostí odděleně od inferenčních mechanismů, jeden inferenční mechanismus může pracovat s různými bázemi znalostí,
2. *neurčitost v bázi znalostí* – v bázi znalostí jsou i nejrůznější heuristiky, které se např. expertovi osvědčily při rozhodování za dlouho dobu praxe,
3. *neurčitost v datech*,
4. *dialogový režim* – expertní systémy jsou nejčastěji konstruovány jako tzv. konzultační systémy. Uživatel komunikuje se systémem způsobem „dotaz systému – odpověď uživatele“,
5. *vysvětlovací činnost* – expertní systém by měl poskytovat vysvětlení svého uvažování,
6. *modularita z transparentnosti báze znalostí* – modularita umožňuje snadnou aktualizaci báze znalostí, transparentnost umožňuje její snadnou čitelnost a srozumitelnost a kontrolu.

Nejpodstatnější charakteristikou je oddělení báze znalostí a inferenčního mechanismu.



Obecné schéma expertního systému

3.6.2 Aplikace expertních systémů

1. Obecné požadavky:

- *problém musí být dostatečně úzký* – musí být možno zakódovat všechny relevantní znalosti, přitom však musí být tak široký, aby expertíza měla smysl,
- *existence expertů v dané oblasti* – není vhodné, když při řešení dané problematiky dochází u expertů k zásadním názorovým rozdílům, dostupnost vhodných dat – významným pomocníkem je možnost používat testovací data s předem známým výsledkem,

- *násobnost zdrojů znalostí a cest odvozování* – i při odstranění některých znalostí či nedostupnosti některých údajů z báze je systém schopen dospět k přijatelným závěrům,
- *strukturovanost problému* – nejlepší je rozdělit bázi znalostí na menší relativně samostatné celky. Tyto celky lze řešit nezávisle a lze dosáhnout dílčí, smysluplné a ověřovatelné výsledky.

2. Některé řešené úlohy:

- úlohy analytické povahy:
 - (a) interpretace dat, vysvětlování empirických údajů,
 - (b) porozumění komplexním signálům – průběžná (on-line) interpretace signálů a dat a určení okamžiku, kdy je nutná intervence,
 - (c) klasifikace,
 - (d) technická a lékařská diagnostika,
- úlohy syntetické povahy:
 - (a) plánování – nalezení posloupnosti akcí k dosažení cíle,
 - (b) technické návrhy – vytvoření konfigurací objektů vyhovujících daným podmínkám,
 - (c) konfigurování počítačů,
 - (d) návrhy terapie v medicíně,
 - (e) automatické programování,
- smíšené metody – zejména aplikace při vyučování.

Při hrubším dělení vystačíme se dvěma skupinami: diagnostické expertní systémy a generativní expertní systémy.

Podrobnější informace o expertních systémech je možno nalézt v bakalářské práci Metody inference od Radima Konečného [14] či v [1, 3, 4, 8].

S expertními systémy je spjata reprezentace znalostí. S podrobnějšími informacemi o reprezentaci znalostí v umělé inteligenci je možné seznámit se v bakalářské práci Úvod do reprezentace znalostí od Jiřího Švece [13] či v [3, 8].

4 Splnitelnost a platnost formule

4.1 Úvod

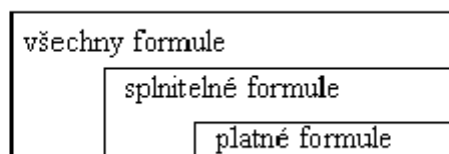
Všechny atomické formule obsahující pouze prvotní proměnné jsou evidentně splnitelné. Atomická formule tvořena logickou konstantou *true* je *tautologie*, zatímco *false* je *kontradikce* neboli nesplnitelná atomická formule.

Kontradikce jsou nesplnitelné neboli nekonzistentní formule.

Platné formule jsou zároveň formulemi splnitelnými neboli konsistentními.

Platnost a nesplnitelnost jsou duální pojmy. Formule A je platná, právě tehdy když formule A je nesplnitelná.

Vztah mezi pojmy splnitelnosti a platnosti formule znázorňuje následující obrázek.



Při rozhodování platnosti nebo splnitelnosti formule A často hovoříme o rozhodovacích algoritmech (rozhodovací procedura, která úspěšně provede a skončí odpovědí ANO, patří-li A do množiny platných (splnitelných) formulí, resp. skončí odpovědí NE, jestliže A do této množiny nepatří.

4.1.1 Algoritmy rozhodování splnitelnosti a platnosti formulí

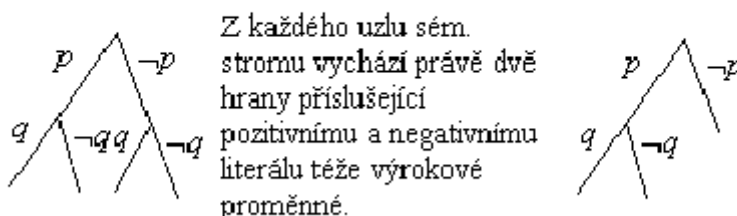
Problém spočívá pouze ve složitosti tohoto rozhodování. Jedním z dosud nevyřešených problémů informatiky je otázka, zda-li existuje rozhodovací procedura splnitelnosti formule.

Rozhodování pomocí sémantického stromu

Vzhledem k tomu, že formuli obsahující k symbolů označujících atomické výroky p_1, \dots, p_k lze interpretovat 2^k způsoby, je třeba uvažovat o efektivnějších procedurách, než je např. u pravdivostních tabulek zkoušení všech variant ohodnocení proměnných atomických formulí.

Řada rozhodovacích algoritmů je založena na pojmu sémantický strom. Každá proměnná p výrokové formule je v něm zastoupena dvojicí literálů, tj. pozitivním literálem proměnné p zastupujícím jeho pravdivostní hodnotu *true* a negativním literálem p zastupujícím jeho pravdivostní hodnotu *false*. Strom složený z hran a uzlů tvoří systém větví procházejících uzly vždy od kořene až po list.

Úplný a neúplný sém. strom se dvěma proměnnými p a q



Úplný sémantický strom zachycuje v případě konečné formule všechna možná ohodnocení jejich proměnných.

Formule je *splnitelná (konsistentní)*, jestliže alespoň jeden list odpovídajícího sémantického stromu nese výslednou hodnotu interpretace true.

Formule je *platná (tautologická)*, jestliže všechny listy jejího úplného sémantického stromu nesou výslednou hodnotu interpretace true.

Rozhodování pomocí tablové metody

Sémantický strom napomáhá přehlednému uspořádání možných ohodnocení atomů dané formule a umožňuje sledování pouze těch větví, kde pravdivostní hodnota koncového listu není určena již dříve během průchodu uvažovanou větví. Podobným nástrojem přehledné prezentace formule je i její formační strom, znázorňující syntax formule.

Jistou modifikací využití formačního stromu je tablová metoda znázornění postupného rozkladu formule až na její literály.

Vhodnou pomůckou pro vytváření sémantického tabla jsou pravidla přepisu formulí na konjunkce literálů vyskytujících se proměnných, které představují sekvence zavěšené na listech sémantického tabla.

α -pravidla:

α	α_1	α_2
$\neg\neg A$	A	
$A_1 \wedge A_2$	A_1	A_2
$\neg(A_1 \vee A_2)$	A_1	$\neg A_2$
$\neg(A_1 \rightarrow A_2)$	A_1	$\neg A_2$
$\neg(A_1 \leftarrow A_2)$	$\neg A_1$	A_2
$A_1 \Leftrightarrow A_2$	$A_1 \rightarrow A_2$	$A_2 \rightarrow A_1$

β -pravidla:

β	β_1	β_2
$B_1 \vee B_2$	B_1	B_2
$\neg(B_1 \wedge B_2)$	$\neg B_1$	$\neg B_2$
$B_1 \rightarrow B_2$	$\neg B_1$	B_2
$B_1 \leftarrow B_2$	B_1	$\neg B_2$
$\neg(B_1 \Leftrightarrow B_2)$	$\neg(B_1 \rightarrow B_2)$	$\neg(B_2 \rightarrow B_1)$

Prakticky jsou postupy rozhodování platnosti, resp. splnitelnosti formule, popsané v tomto a předcházejícím odstavci uplatnitelné pouze pro nepřilíš složité formule. Poněkud lépe je na tom rezoluční metoda.

Robinsonův rezoluční princip

Je-li množina klauzulí nespíitelná, pak opakovaným užitím rezolučního pravidla lze vždy odvodit spor (prázdnou klauzuli), která je nespíitelná.

Popsaný algoritmus zjišťování nespíitelnosti je nedeterministický. Existuje více než jeden způsob postupného výběru dvojic klauzulí pro aplikaci rezolučního principu.

Rezoluční princip poskytuje možnost důkazu, že formule Z je (tauto)logickým důsledkem dané množiny formulí, tedy že platí $\{H_1, H_2, \dots, H_n\} \models Z$. Důkaz tautologického důsledku lze totiž převést na důkaz nespíitelnosti množiny formulí $\{H_1, H_2, \dots, H_n, Z\}$.

Příklad $U = \{p \vee q, p \vee r, \neg q \vee \neg r, \neg p\}$

1.	$p \vee q$	
2.	$p \vee r$	
3.	$\neg p \vee \neg r$	
4.	$\neg p$	
5.	$p \vee \neg r$	použit ř. 1 a 3
6.	q	použit ř. 1 a 4
7.	$p \vee \neg q$	použit ř. 2 a 3
8.	r	použit ř. 2 a 4
9.	p	použit ř. 2 a 5
10.	$\neg r$	použit ř. 3 a 6
11.	$\neg q$	použit ř. 3 a 8
12.	$\neg r$	použit ř. 4 a 5
13.	$\neg q$	použit ř. 4 a 7
14.	false	použit ř. 4 a 9

5 Závěr

Tento projekt vznikl za účelem seznámit čtenáře s úvodem do inference, tedy metodami odvozování. Kladl si za cíl projevit v čtenáři zájem o toto odvětví a nasměřovat ho na další studium této problematiky. Byly zmíněny základní metody inference, u každé metody byly uvedeny základní vlastnosti. Dále byly uvedeny hlavní oblasti, kde se využívají nebo mohou být využity metody odvozování.

Složitější oblasti jsou doplněny obrázky a jednoduchými příklady, které dokreslují podstatu výkladu.

Reference

- [1] Mařík, V. a kol., *Expertní systémy: principy, realizace, využití*, 1984.
- [2] Jiroušek, R., *Metody reprezentace a zpracování znalostí v umělé inteligenci*, 1995.
- [3] Hajíčková, E., Wicha, T., *Reprezentace znalostí v systémech umělé inteligence*, 1984.
- [4] Berka, P. a kol., *Expertní systémy*, 1998.
- [5] Lukasová, A., *Logické základy umělé inteligence – Výroková a predikátová logika*, 1995.
- [6] Strossa, P., *Vybrané kapitoly z počítačového zpracování přirozeného jazyka*, 1999.
- [7] Hlaváč, V., *Počítačové vidění*, 1992.
- [8] Vondrák, I., *Umělá inteligence a neuronové sítě*, 1995.
- [9] Mañas, M., Zelinka, J., *Kapitoly z teorie her a matematického programování*, 1966.
- [10] Stachová, J., *Model a analogie ve vědě, umění a filozofii*, 1994.
- [11] *Sborník přednášek, Metody umělé inteligence a expertní systémy II.*, 1985.
- [12] *Sborník přednášek, Metody umělé inteligence a expertní systémy IV.*, 1989.
- [13] Švec, J., *Úvod do reprezentace znalostí*, bakalářská práce, 2000.
- [14] Konečný, R., *Metody inference*, bakalářská práce, 2000.
- [15] McCarthy, J., *What is Artificial Intelligence*, [whatisai.pdf na http://www-formal.stanford.edu/jmc/index.html](http://www-formal.stanford.edu/jmc/index.html)
- [16] Sedláček, V., *Umělá inteligence*, 1982.
- [17] Müller, J., *Základy systematické heuristiky*, 1975.
- [18] Rosický, J., *Teorie množin*, učební text MU, 1996.
- [19] Štěpánek, P., *Matematická logika*, 1982.
- [20] Křetínský, M., Černá, I., Kučera, A., *Automaty a formální jazyky I.*, učební text FI MU, 1999.