

**ZNALOSTNÉ SYSTÉMY
V OTÁZKACH
A ODPOVEDIACH**

KRISTÍNA MACHOVÁ

Košice
2005

Ing. Kristína Machová, CSc.
Katedra kybernetiky a umelej inteligencie
Fakulta elektrotechniky a informatiky
Technická univerzita v Košiciach
Kristina.Machova@tuke.sk

Publikácia vznikla za podpory VEGA grantu MŠ SR č. 1/1060/04 „Klasifikácia a anotácia dokumentov pre sémantický web“ Vedeckej grantovej agentúry Ministerstva školstva SR.

Lektorovali: doc. Ing. Marián Mach, CSc., FEI TU Košice
doc. Ing. Ján Paralič PhD., FEI TU Košice

© Kristína Machová, Košice 2005

Všetky práva sú vyhradené. Žiadna časť tejto publikácie nesmie byť reprodukovaná, zadaná do informačného systému alebo prednášaná v inej forme či inými prostriedkami bez predchádzajúceho súhlasu autora.

ISBN: 80-8086-018-1

OBSAH

Predslov	1
I. ZÁKLADY ZNALOSTNÝCH SYSTÉMOV	
1. Aká je história vzniku a definícia znalostných systémov?	2
2. Aký bol vývoj a prví predchodcovia znalostných systémov?	3
3. Ako sa líšia vlastnosti znalostných systémov a ľudského riešiteľa? ..	4
4. Aké miesto má človek v tvorbe a používaní znalostných systémov? ..	5
5. Aká je štruktúra znalostných systémov?	6
6. Štruktúra diagnostických znalostných systémov!	8
7. Štruktúra plánovacích znalostných systémov!	9
8. Čo je to produkčný systém?	10
9. Ako je možné produkčný systém formálne definovať?	11
10. Ako funguje dopredné a spätné reťazenie?	12
11. Ako je možné znázorniť inferenčnú sieť?	13
12. Ako je možné inferenčnú sieť prehľadávať?	15
II. SPRACOVANIE NEURČITOSTI	
13. Čo je to neurčitá znalosť?	16
14. Aký je rozdiel medzi neurčitosťou v BZ a v BD?	17
15. Aké druhy neurčitosti poznáme?	18
16. Ako je možné neurčitosť reprezentovať?	19
17. Aké modely práce s neurčitosťou poznáme?	20
18. Ako je definovaný všeobecný extenzionálny model?	21
19. Prečo je Subjektívna Bayes-ovská metóda spracovania neurčitosti subjektívna?	22
20. Ako je v Subjektívne Bayes-ovskej metóde definovaná kombinačná funkcia CTR?	23
21. Ako je v Bayes-ovskej metóde definovaná funkcia GLOB a ostatné kombinačné funkcie?	24
22. Čo predstavuje takzvaný intuitívny model spracovania neurčitosti? ..	25
23. Ako je možné spracovať extrémne hodnoty neurčitosti?	26
24. Čo je to algebraická teória?	27
25. Ako môže vyzeráť funkcia GLOB v algebraickej teórii?	28
26. Ako sú v algebraickej teórii definované ostatné kombinačné funkcie?	30
27. Čo je to Dempster-Shafferova metóda spracovania neurčitosti?	31
28. Ako je definovaný Dempster-Shafferov priestor?	33
29. Ako je definovaná funkcia GLOB v Dempster-Shafferovom modeli spracovania neurčitosti?	34
30. Ako sú definované ostatné funkcie v Dempster-Shafferovom modeli spracovania neurčitosti?	36

31. Na čom je založený fuzzy model spracovania neurčitosti?	37
32. Ako je možné s fuzzy množinami operovať?	38
33. Ako sa realizuje neurčité rozhodovanie vo fuzzy znalostných systémoch?	39
34. Ako môže prebiehať fuzzifikácia v znalostných systémoch?	41
35. Ako môžeme reprezentovať jazykové premenné vo FuzzyClipse?	43
36. Ako môžeme reprezentovať jazykové premenné štandardnými funkciami?	44
37. Ktoré sú základné techniky neurčitého rozhodovania?	45
38. Ako pracuje jednoduché pravidlo FUZZY_GRISP?	46
39. Ako pracuje jednoduché pravidlo FUZZY_FUZZY?	48
40. Ako sa spracovávajú zložené fuzzy pravidlá?	49
41. Ako sa uskutočňuje defuzzifikácia?	50

III. METAZNALOSTI

42. Čo sú to metaznalosti a ako ich delíme?	51
43. Čo sú to strategické metaznalosti?	52
44. Na čo sa používajú podporné znalosti o znalostiach?	53
45. Čo sú to metaznalosti definujúce zmysel pravidiel?	54
46. Môžu metapravidlá detekovať chyby v pravidlách?	55
47. Môžu metapravidlá zdôvodňovať architektúru znalostného systému?	56
48. Ako sa uskutočňuje adaptácia znalostného systému na novú situáciu pomocou metapravidiel?	57

IV. ZÍSKAVANIE ZNALOSTÍ

49. Je možné modelovať možnosti systému pomocou metaznalostí?	58
50. Ako je možné získavať znalosti pre znalostný systém?	59
51. Čo môžeme získať strojovým učením?	60
52. Prečo a ako generujeme rozhodovacie stromy?	61
53. Ako generuje rozhodovacie stromy algoritmus ID3	62
54. Čo je to Kombinačná analýza dát?	64
55. Ako je možné použiť Kombinačnú analýzu dát v znalostných systémoch?	65
56. Ako je definovaný algoritmus Kombinačnej analýzy dát	66
57. Čo predstavujú priame metódy získavania znalostí?	67
58. Aké sú to nepriame metódy získavania znalostí?	70
59. Čo je to Mnohorozmerné škálovanie?	71

60. Ako funguje Johnsonovo mnohorozmerné zhlukovanie	73
61. Ako sa uskutočňuje Priamy prenos znalostí?	74
62. Čo je to Repertoárová sieť?	75
63. Ako sa určuje podobnosť pojmov a konštruktov?	76
64. Čo je to Implikačná analýza?	77
65. Ako sa generujú pravidlá z repertoárovej siete?	79
66. Ako je možné vylepšovať znalosti?	80
V. VYSVETĽOVACÍ MECHANIZMUS	
67. Načo slúži vysvetľovací mechanizmus?	83
68. Ako delíme jednotlivé vysvetlenia?	85
VI. VYHODNOCOVANIE A BEZPEČNOSŤ	
69. Ako sa zabezpečujú znalostné systémy?	86
70. Ako je možné vyhodnocovať znalostné systémy?	87
71. Načo je model používateľa v znalostných systémoch?	90
VII. TABUĽA	
72. V čom spočíva riadenie typu tabuľa v znalostných systémoch?	91
73. Ako funguje riadenie „tabuľa“ vo FEL-EXPERTE?	93
74. Aké sú aplikačné možnosti znalostných systémov?	94
75. Ktoré sú najznámejšie zahraničné znalostné systémy?	95
76. Ktoré sú najznámejšie domáce znalostné systémy?	96
77. Ktoré sú najznámejšie ekonomické aplikácie?	97
78. Referencie	98

Predslov

Znalostné systémy predstavujú klasickú oblasť Umelej inteligencie, ktorá je však aplikačne veľmi úspešná. Predmet znalostné systémy poskytuje študentom vedomostný základ, ktorý potom využijú v mnohých ďalších predmetoch štúdia. Často sa siahajú po metódach znalostných systémov v iných oblastiach Umelej inteligencie, kde sa potom pravidlové znalostné metódy kombinujú s inými metódami za účelom skvalitnenia dosahovaných výsledkov.

Predkladaný učebný text si nečiní nároky na vyčerpávajúci prehľad všetkých tém znalostných systémov, čo by ho robilo príliš rozsiahlym. Podáva základy znalostných systémov v zhutnenej podobe veľmi zrozumiteľným spôsobom. Je určený predovšetkým poslucháčom 4. ročníka Fakulty elektrotechniky a informatiky Technickej univerzity v Košiciach v odbore Umelá inteligencia pre rovnomený predmet Znalostné systémy. Na tento predmet nadväzujú predmety: Projektovanie znalostných systémov, Strojové učenie a napokon Objavovanie znalostí.

Daný učebný text obsahuje nasledovné hlavné okruhy problémov: štruktúra znalostného systému, inferenčný mechanizmus, metódy spracovania neurčitých znalostí, fuzzy znalostné systémy, metaznalosti, spôsoby získavania znalostí (napríklad strojovým učením), modifikácia znalostných báz, zdieľanie znalostných báz, vysvetľovací mechanizmus, bezpečnosť a vyhodnocovanie znalostných systémov.

Moja vďaka patrí doc. Ing. Mariánovi Machovi CSc. a doc. Ing. Jánovi Paraličovi PhD. za starostlivé prečítanie rukopisu a hlavne za podnetné návrhy, ktoré v konečnej verzii textu boli zohľadnené. Taktiež ďakujem Ing. Ladislavovi Dudášovi za pomoc pri kreslení obrázkov a tabuliek.

Tento učebný text vznikol v rámci VEGA projektu 1/1060/04 s názvom „Klasifikácia a anotácia dokumentov pre sémantický web“ Vedeckej grantovej agentúry Ministerstva školstva SR.

V Košiciach 30.9.2005

autor

1. Aká je história vzniku a definícia znalostných systémov?

Znalostné systémy (Popper-Kelemen, 1989), (Mařík a kol., 1997), (Berka, 1994) a (Mikulecký a kol., 1998) sa vyčlenili z oblasti Umelej inteligencie (Csontó, 1990), (Návrat a kol., 2002). Odborníci v oblasti Umelej inteligencie vyvíjali od začiatku snahu vymyslieť programy, ktoré by boli schopné myslieť, teda riešiť úlohy spôsobom, akým by to robil človek, považovaný za rozumného.

V 60-tych rokoch 20. storočia vedci hľadali všeobecné metódy pre riešenie širokých tried úloh. Tieto metódy sa používali v univerzálnych programoch, ako napríklad program GPS – General Problem Solver, ktorý:

- ✓ používa techniku analýzy cieľov a prostriedkov (Means Ends Analysis)
- ✓ snaží sa krok po kroku previesť počiatočný stav riešenia do cieľového stavu (resp. do niektorého z cieľových stavov)
- ✓ riešenie spočíva v hľadaní odlišnosti – diferencie medzi dvojicami stavov: stavom, v ktorom sa riešený problém práve nachádza (hociktorý okrem cieľového) a cieľovým stavom
- ✓ hodnotí relevantnosť operátorov odstraňujúcich diferencie (o koľko posunú riešenie dopredu)
- ✓ testuje aplikovateľnosť operátorov
- ✓ ak relevantný operátor nie je možné bezprostredne aplikovať, potom sa stanovujú nové podciele (ich vyriešenie je predpokladom aplikovateľnosti vyššie uvedeného operátora)
- ✓ proces sa opakuje rekurzívne, kým sa nenájde celá sekvencia operátorov, ktorá úplne odstráni diferenciu, čím vyrieši úlohu, resp. dokáže že úloha nie je riešiteľná

V 70-tych rokoch 20. storočia došli vedci k záveru, že je veľmi ťažké zabezpečiť univerzálnosť programu, a že je potrebné sústrediť sa na:

- ✓ metódy reprezentácie t.j. formulovať problém spôsobom, ktorý uľahčí jeho riešenie
- ✓ metódy vyhľadávania efektívnych spôsobov riešenia, ktoré by nevyžadovali veľa pamäte a času

Napríklad, keď sa dá problém reprezentovať pomocou grafu, alebo petriho siete, potom sa dá efektívne riešiť.

V 80-tych rokoch 20. storočia zhodnotili vedci svoje doterajšie výskumy nasledovne. Efektívnosť riešenia úloh závisí od znalostí viac, ako od formalizmu odvodzovania riešení. Tento poznatok naznačil novú koncepciu „intelektuálneho programu“. Intelektuálny program je potrebné vybaviť množstvom kvalitných špecializovaných znalostí. To viedlo ku rozvoju expertných, resp. znalostných systémov a znalostnému inžinierstvu (technológii zostrojovania znalostných systémov). Z uvedeného vyplýva definícia znalostného systému.

Znalostný systém je súbor špecializovaných počítačových programov a štruktúrovaných údajov, ktoré sú schopné nahradiť špecialistu pri riešení úloh v úzko ohraničenom odbore, prípadne ho prekonať.

2. Aký bol vývoj a prví predchodcovia znalostných systémov?

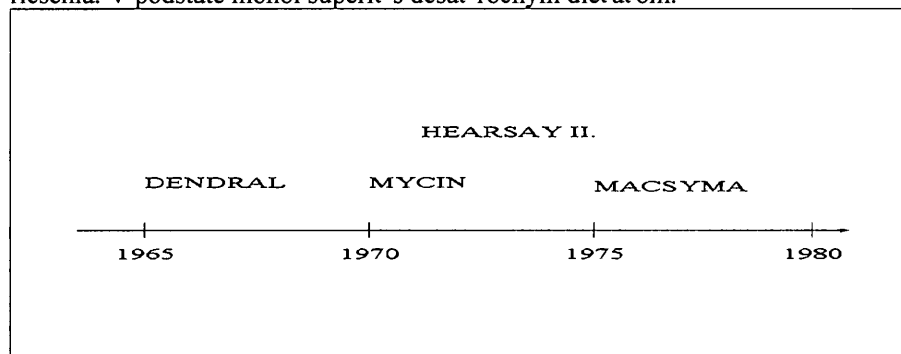
Prvýkrát bola použitá myšlienka aplikácie veľkého objemu špecifických znalostí, vo forme umožňujúcej programu manipuláciu s nimi v projekte DENDRAL, ktorý zohral významnú úlohu v rozvoji znalostných systémov.

DENDRAL (Lindsay, 1980) sa začal vyvíjať v roku 1965. Systém analyzuje údaje hmotového spektrografu a zostavuje pravdepodobné molekulárne štruktúry. Na vstupe má tento systém sumárny vzorec a hmotový spektrograf. Z týchto vstupov odvodí zoznam prípustných štruktúr (môže ísť o tisíce), na ktoré aplikuje niektoré zo svojich znalostí. Tým postupne znižuje počet alternatív, až dostane jediný náležitý štruktúrny vzorec. Analogické postupy je možné využiť aj v iných oblastiach. Napríklad v medicíne na vyhodnocovanie EKG.

MACSYMA (Moses, 1975) vznikla v MIT ako program pre symbolické manipulácie (diferencovanie a integrovanie v symbolickom tvare, zjednodušovanie symbolických výrazov). Používa znalosti vo forme pravidiel o transformáciách jedného symbolického výrazu na iný.

MYCIN (Shortliffe, 1976) sa zameriava na diagnostiku a liečenie infekčných ochorení krvi. Používa znalosti v tvare pravidiel spájajúcich možné diagnózy so zodpovedajúcimi príznakmi. Práca tohto znalostného systému bola porovnateľná s výsledkami expertov v danej oblasti.

HEARSAY II. bol schopný chápať súvislú reč na základe slovníka s tisíckou slov. Bol taktiež schopný riešiť úlohy na rôznej úrovni abstrakcie. Dokázal postupne rozvíjať čiastočné riešenia. V podstate mohol súperiť s desať ročným dieťaťom.



Všetky tieto znalostné systémy predstavovali veľké projekty znamenajúce dlhú prácu, riešené mnohými ľuďmi. Spotrebovali veľa človeko-rokov. Niektoré boli neskôr začlenené do iných systémov. Často boli prvé výsledky publikované po mnohých rokoch, nezriedkavo až desiatich. Obrázok znázorňuje približne časové obdobia vývinu týchto znalostných systémov.

3. Ako sa líšia vlastnosti znalostných systémov a ľudského riešiteľa?

Táto otázka evokuje ďalšie otázky. Načo je dobré vyvíjať software, ktorý by robil niečo, čo vie dobre robiť človek? Bolo by možné ľudských expertov úplne nahradiť znalostnými systémami? Pokúsime sa v ďalšom tieto otázky zodpovedať.

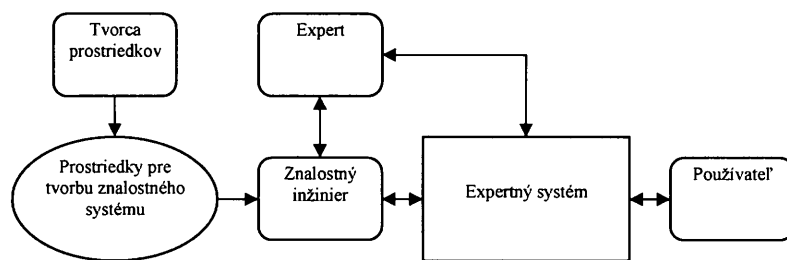
	Ľudská kompetentnosť	Kompetentnosť expertného systému
Trvanlivosť	nie je trvalá	trvalý
Prenos znalostí	ťažký (učenie)	ľahký
Dokumentovateľnosť	zložitá	dobrá
Stabilita	nie je stabilný	stabilný
Cena	drahý	pri vývoji drahý, použitie lacné
Dostupnosť	ťažká	vysoká

V tabuľke je veľmi stručne uvedené, ako sa niektoré vlastnosti (trvanlivosť, prenos znalostí, dokumentovateľnosť, stabilita, cena, dostupnosť), dôležité pre znalostné poradenstvo, premietnu do práce ako ľudského riešiteľa a znalostného (expertného) systému. Z tejto tabuľky sa dozvedáme o nasledovných kompetenciách znalostného systému. Jeho kompetentnosť je **trvalá** na rozdiel od človeka, ktorý ju musí neustálou praxou obnovovať, aby si udržal profesionálnu úroveň a dlhšia prestávka pôsobí nepriaznivo na jeho schopnosti. Je nutné dodať, že praxou si špecialista vedomosti nielen udržiava, ale ich aj prehľbuje a objavuje hlbšie súvislosti. Kompetentnosť znalostného systému (ďalej ZS) je **ľahko odovzdávateľná** jednoduchým kopírovaním. Na druhej strane odovzdávanie znalostí medzi ľuďmi je ťažký, dlhý a drahý proces – učenie. Kompetentnosť ZS je dobre **dokumentovateľná** (báza znalostí) na rozdiel od človeka, ktorého vedomosti dokumentovať je zložitá a časovo náročná. Opäť je potrebné si uvedomiť, že aj v prípade ZS bolo potrebné vedomosti oblasti najprv zdokumentovať. Výkon ZS je **stabilný**, čo sa nedá povedať o človeku, ktorého výkon kolíše pod vplyvom emocionálnych faktorov, stresu a tlaku času. Zatiaľ čo ZS je neustále **dostupný**, človek je často zaneprázdnený a potrebuje dovolenku a odpočinok. S výnimkou vývoja ZS, ktorý je cenovo náročnejší, je používanie ZS **lacné** a rovná sa cene strojového času. Na druhej strane človek expert je spravidla veľmi vysoko cenený.

Na základe uvedených skutočností môžeme konštatovať, že ZS sú potrebné, lebo niekedy pracujú efektívnejšie a teda môžu doplniť prácu experta. Mohli by ho aj úplne nahradiť? Nemôžu, pretože v niektorých oblastiach činnosti ľudská kompetentnosť značne prevyšuje tú umelú. Tak napríklad človek je **tvorivý**, má **schopnosť vyrovnat' sa s neočakávaným zvratom situácie** pomocou predstavivosti a nových prístupov k riešeniu úlohy. Človek je schopný **sa učiť** a tak sa adaptovať na zmenené podmienky a prispôbiť svoje stratégie novým okolnostiam. Človek má **zdravý rozum**, pretože disponuje všeobecnými znalosťami o svete. Je schopný **sebareflexie**, a teda sa nebude vždy znova a znova pokúšať o riešenie problému, ktorý je nad jeho sily (čo by sa u ZS mohlo stať).

4. Aké miesto má človek v tvorbe a používaní znalostných systémov?

Role, ktoré človek hrá v procese tvorby a používania znalostných systémov, je možné pomenovať nasledovne: človek ako expert, ako znalostný inžinier, ako tvorca prostriedkov a napokon ako používateľ. Vzájomné vzťahy týchto úloh sú ilustrované nasledujúcim obrázkom. V tejto symbióze rôznych úloh vystupuje pochopiteľne aj znalostný systém, resp. expertný systém a blok prostriedkov pre tvorbu znalostného systému (PTZS). **Znalostný systém** predstavuje súbor programov, obsahujúci okrem inferenčného mechanizmu a bázy znalostí aj podporné komponenty: editory, ladiace prostriedky, komunikačné prostriedky, grafický vstup/výstup, atď. **PTZS** môže predstavovať prázdny znalostný systém, programovací jazyk alebo knižnica programov.



Jednotlivé role človeka v tvorbe a používaní znalostného systému je možné charakterizovať nasledovne:

Expert je človek, ktorý vďaka vzdelaniu a skúsenostiam je schopný efektívne riešiť problémy úzko vymedzenej oblasti. (Znalostný systém môže modelovať znalosti jedného, alebo viacerých expertov).

Znalostný inžinier má poznatky z oblasti znalostných systémov. (Je to expert na znalostné systémy.) Pýta sa experta na jeho znalosti a organizuje ich. Rieši, akým spôsobom majú byť reprezentované v znalostnom systéme. Je to kombinácia softverového špecialistu a psychológa.

Tvorca prostriedkov vytvára prostriedky pre tvorbu znalostných systémov, čím pripravuje pôdu pre jeho vznik.

Používateľ používa hotový znalostný systém v podstate ako: laik (potrebuje riešenie a jeho vysvetlenie) alebo špecialistu (získa záver menej namáhavou a rýchlejšou cestou ako by to robil sám, alebo je zvedavý na závery iného špecialistu).

5. Aká je štruktúra znalostných systémov?

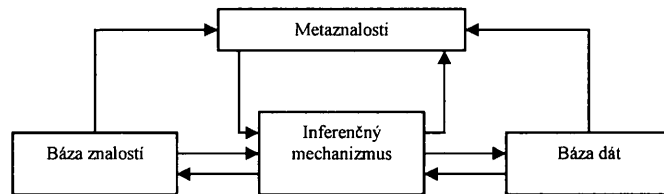
Štruktúra sa odvíja od typov znalostí, ktoré znalostný systém používa. Začnime od človeka. Človek používa niekoľko typov znalostí, ktoré sa líšia hlavne stupňom abstrakcie a šírkou zaberanej oblasti. Typy znalostí používaných človekom sa delia na štyri základné skupiny:

- ✓ **VŠEOBECNÉ ZNALOSTI** o svete majú na rozdiel od ostatných typov vysoký stupeň abstrakcie a mohutnú šírku záberu, lebo sa netýkajú iba úzko vymedzenej oblasti. Sú používané neustále v spojení s ostatnými typmi znalostí.
- ✓ **PREDMETNÉ ZNALOSTI** sú znalosti so stredným stupňom abstrakcie a šírky zamerania. Sú to znalosti úzko vymedzenej predmetnej oblasti (napr. medicínske). Ide o špecifické, profesijné znalosti, vznikajúce štúdiom.
- ✓ **PROBLÉMOVÉ ZNALOSTI** sú znalosti o konkrétnom probléme, bez abstrakcie a šírky zamerania. Sú to znalosti o konkrétnej úlohe (ťažkosti konkrétneho pacienta).
- ✓ **METAZNALOSTI** sú to znalosti o znalostiach, ktoré umožňujú seba-reflexiu (vieme čo vieme).

Nasledovná tabuľka dokumentuje, ktoré zo znalostí človeka používa aj znalostný, resp. expertný systém.

	Človek	Expertný systém
Vedomosti o svete	Všeobecné vedomosti (vysoký stupeň abstrakcie) (veľká šírka záberu)	
	Predmetné vedomosti (vedomosti z odboru) (stredný stupeň abstrakcie) (zúžená šírka záberu)	Predmetné vedomosti (ako všeobecné u človeka) (báza znalostí)
	Konkrétne vedomosti (konkrétny problém) (nízky stupeň abstrakcie) (úzky záber)	Konkrétne vedomosti (údaje o probléme) (báza údajov, báza dát)
Vedomosti o sebe	Metaznalosti (vedomosti o sebe) (sebareflexia) (viem, čo viem)	Metaznalosti

Znalostný, resp. expertný systém nepoužíva všeobecné znalosti, pretože zatiaľ nepoznáme technológie, ktoré by nám umožnili uložiť tak obrovský objem znalostí. Preto najvšeobecnejšie znalosti, ktoré používa znalostný systém sú **predmetné znalosti**. Spravidla sú uložené v báze znalostí (BZ). ZS taktiež používa **problémové znalosti**, ktoré ukladá do bázy dát (BD) a **metaznalosti**, ukladané do rovnomenného bloku. Okrem týchto blokov je súčasťou štruktúry ZS blok IM – **inferenčný mechanizmus**, ktorý umožňuje spoluprácu všetkých typov znalostí za účelom riešenia úloh. Nasledovný obrázok ilustruje súčinnosť všetkých typov znalostí pod taktovkou IM, teda ilustruje základnú štruktúru ZS.



Báza znalostí odpovedá predmetným znalostiam človeka, ktoré používa ako znalosti všeobecného charakteru. Umožňuje riešiť iba určitý druh úloh. Spadajú sem znalosti experta, ktoré sa týkajú konkrétnej oblasti.

Báza dát odpovedá znalostiam človeka o riešenom probléme (problémové znalosti). Zvyčajne sa získavajú od používateľa, ktorý odpovedá na otázky ZS. Používateľ spravidla poskytuje odpovede, ktoré sú mu známe. Môže však poskytnúť odpoveď obsahujúcu hodnotu odčítanú z meracieho prístroja.

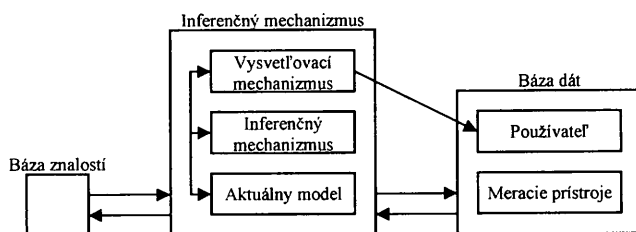
Metaznalosti nemusí nutne ZS obsahovať. Otvplyvňujú činnosť IM, na základe aktuálneho stavu riešenia. S nimi pracuje ZS efektívnejšie. Získavajú sa od experta.

V súvislosti so štruktúrou ZS môže byť prázdny alebo dedikovaný systém. Prázdny znalostný systém je univerzálny. Jeho použitie nezávisí od konkrétnej BZ, ktorá sa dá vymeniť za inú. Je viazaný na formu reprezentácie znalostí, nie na ich obsah. U dedikovaného znalostného systému tvorí báza znalostí a inferenčný mechanizmus jednotný celok. Je schopný riešiť iba problémy jedného druhu, pretože BZ sa nedá vymeniť. Avšak, jeho práca je efektívnejšia ako práca porovnateľného prázdneho ZS.

Z hľadiska charakteru riešených úloh možno ZS rozdeliť do dvoch základných skupín: diagnostické a plánovacie. Podrobnejšia štruktúra znalostného systému sa u týchto dvoch skupín líši.

6. Štruktúra diagnostických znalostných systémov!

Diagnostické ZS riešia problém diagnostikou, t.j. určovaním, ktorá hypotéza z danej konečnej množiny cieľových hypotéz najlepšie korešponduje s dátami o konkrétnom prípade. Riešenie spočíva v postupnom prehodnocovaní dielčích a cieľových hypotéz v rámci expertom pevne daného modelu riešenia. Tomuto spôsobu práce ZS odpovedá štruktúra diagnostického znalostného systému zobrazená na nasledovnom obrázku.



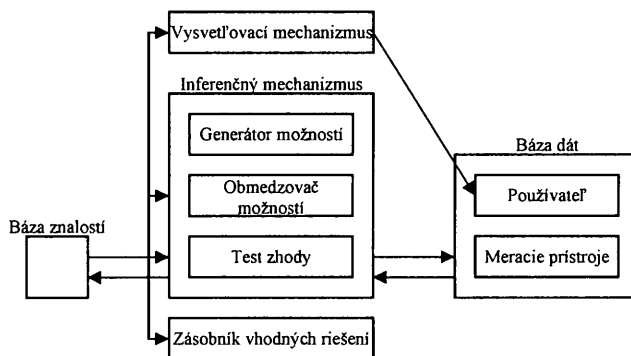
Jadrom ZS je **inferenčný mechanizmus**, resp. riadiaci mechanizmus, ktorý s využitím znalostí z BZ a BD po každej odpovedi používateľa upresňuje aktuálny model konzultovaného prípadu. Jeho povinnosťou je vybrať takú otázku, od zodpovedania ktorej sa očakáva najväčší prínos k upresneniu aktuálneho modelu. Súčasťou inferenčného mechanizmu je aktuálny model a vysvetľovací mechanizmus.

Aktuálny model reprezentuje súčasný stav riešenia úlohy. Predstavuje množinu všetkých momentálne platných poznatkov a faktov o riešenom probléme. Aktuálny model sa môže meniť pridaním/vypustením niečoho nového z BD, alebo odvodením nového poznatku pomocou IM. Aktuálny model môže byť implicitný a prázdny. Implicitný aktuálny model zodpovedá apriórnym predstavám o podobných prípadoch. Tento počiatočný „default“ model sa postupne modifikuje. Jeho veľkosť nenarastá. Je náročnejší na odvádzanie a testovanie. Prázdny aktuálny model je na začiatku prázdny a postupne sa do neho ukladajú informácie, takže neustále rastie.

Úlohou **vysvetľovacieho mechanizmu** je objasniť používateľovi, čo získané riešenie znamená a akým spôsobom bolo odvodené.

7. Štruktúra plánovacích znalostných systémov!

Spôsob práce plánovacieho znalostného systému sa líši od spôsobu práce diagnostického znalostného systému. Plánovací ZS (Tien-Chien Chang, 1990) pozná počiatočný stav a cieľ. Jeho úlohou je nájsť optimálnu cestu z počiatočného stavu k cieľu. Pracuje tak, že generuje všetky možné riešenia a hľadá medzi nimi to optimálne. To však môže spôsobiť kombinatorickú explóziu, ktorej je potrebné sa vyhnúť. Výsledkom činnosti plánovacieho ZS je zoznam prípustných riešení. Spôsobu práce plánovacieho ZS odpovedá aj jeho štruktúra, ktorá je uvedená v nasledujúcom obrázku.



Štruktúrna schéma plánovacieho ZS sa od štruktúrnej schémy predchádzajúceho diagnostického ZS líši tým, že namiesto aktuálneho modelu má zásobník vhodných riešení a IM pozostáva z troch súčastí: generátora možností, obmedzovača možností a testu zhody.

Generátor možností generuje všetky možné riešenia, pričom ovplyvňuje výber prípustných operátorov. **Obmedzovač možností** obmedzuje generatívnu schopnosť použitím znalostí v BZ. Blok s názvom **test zhody** riadi testovanie zhody vygenerovaných riešení s dátami v BD. Príkladom programu generujúceho množstvo riešení a testovaním vyberajúceho tie správne je riešenie hlavolamu, v ktorom je potrebné za písmená dosadiť čísla tak, aby vyhovovali nasledovnej podmienke: SEND + MORE = MONEY. Najjednoduchšie riešenie (test na diferenciu sa vykonáva po vygenerovaní celej postupnosti čísel) je v nasledovnom programe vľavo.

```
generuj([S,E,N,D],[M,O,R,E],[M,O,N,E,Y]):-
    sucet(0,D,E,Y,P1),
    sucet(P1,N,R,E,P2),
    sucet(P2,E,O,N,P3),
    sucet(P3,S,M,O,M),M > 0,S > 0,
    dif([S,E,N,D,M,O,R,Y]).
```

```
generuj([S,E,N,D],[M,O,R,E],[M,O,N,E,Y]):-
    sucet(0,D,E,Y,P1),dif([D,E,Y]),
    sucet(P1,N,R,E,P2),dif([D,E,Y,N,R]),
    sucet(P2,E,O,N,P3),dif([D,E,Y,N,R,O]),
    sucet(P3,S,M,O,M),M > 0,S > 0,
    dif([D,E,Y,N,R,O,S,M]).
```

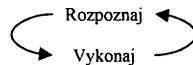
Efektívnejšia je metóda hierarchického usporiadania generovania a testovania, ktorá umožní odmietnuť riešenie, na základe jeho čiastočného popisu. Jedným testom je zamietnutý väčší počet nevyhovujúcich riešení. Príkladom je riešenie hlavolamu send-more-money, ilustrované vyššie uvedeným programom vpravo (test na diferenciu sa vykonáva po vygenerovaní každého ďalšieho čísla).

8. Čo je to produkčný systém?

Produkčný systém je tvorený troma zložkami:

- ✓ súborom produkčných pravidiel
- ✓ bázou dát (BD)
- ✓ interpretom pravidiel.

Produkčné pravidlá majú tvar: $S \rightarrow A$ resp. $P \rightarrow Z$, čo znamená, že ak v báze dát nastala situácia S, potom vykonaj akciu A, resp. ak v báze dát platia predpoklady P, potom platí záver Z. Predpoklady P, resp. situácia S sa niekedy nazývajú „situačnou časťou“ pravidiel. **Báza dát** je prostredím pre beh produkčného systému. Situácia S sa môže vyskytnúť v BD a podobne akcia A sa vykoná nad touto BD. **Interpreter** realizuje produkčné pravidlá. Pracuje v dvoj krokovom cykle:



Interpreter rieši nasledovné úlohy:

- 1) Skontroluje situačné časti všetkých produkčných pravidiel (PP). Nájde pravidlo, ktorého S je splnené na základe údajov v BD.
- 2) Realizuje akciu A vybraného pravidla, teda vlastne pravidlo odpáli.
- 3) Ak nenájde žiadne pravidlo, potom zastane.
- 4) Ak nastane prípad, že je splnená situačná časť viacerých pravidiel, musí rozhodnúť, ktoré použiť. Výber pravidla je dôležitý, pretože môže ovplyvniť výsledok. (Vykonanie akcie jedného z odpáliteľných pravidiel môže zrušiť podmienky pre aplikáciu ostatných odpáliteľných pravidiel.) Stratégia výberu je daná buď usporiadaním PP, alebo sa určuje dynamicky, vyhodnocovaním zadaného kritéria.

Charakteristické vlastnosti produkčného systému sú nasledovné:

- ✓ **Obmedzená možnosť interakcie medzi pravidlami.** Interakcia sa totiž uskutočňuje iba cez BD. Pravidlo je vyvolané iba vtedy, keď je splnená podmienka v jeho situačnej časti. Naopak, ak je splnená situačná časť pravidla, je odpálené bez ohľadu na to, či prispieva k riešeniu alebo ho zahmlieva. Ak majú byť PP aplikované v určitej postupnosti, potom predchodca musí vytvoriť v BD podmienky pre odpálenie svojho nástupcu.
- ✓ **Existencia obmedzení kladených na tvar pravidiel.** Všeobecne neexistujú, avšak v reálnych prípadoch sa prijímajú isté obmedzenia. Pri porovnaní podmienky pravidla so stavom BD sa nesmie BD meniť.
- ✓ **Pravidlá predstavujú elementárne akcie** produkčného systému, ktoré sú najmenším krokom systému z hľadiska vonkajšieho chovania.
- ✓ **Modularita** produkčného systému uľahčuje odstraňovanie závad v chovaní programu. Umožňuje ľahké integrovanie prírastkov nových znalostí. Umožňuje vynechávať pravidlá z produkčného systému. Vynechanie pravidla môže viesť k horšiemu riešeniu, ale nemôže ochromiť schopnosť ZS riešiť problémy (ZS sa bude stále chovať rozumne). Jednotlivé pravidlá sú totiž vyvolávané prostredím (stavom BD), nie v postupnosti dopredu zadanej programátorom. Ak v nejakom stave BD nedochádza k požadovanému chovaniu, stačí pridať PP:
popis situácie nezahody \rightarrow požadovaná akcia.

9. Ako je možné produkčný systém formálne definovať?

Definícia produkčného pravidla závisí od spôsobu jeho chápania. Produkčné pravidlá môžeme chápať dvojako:

- ✓ **Deklaratívne.** Pravidlo deklaruje (definuje) súvislosť medzi jednotlivými fragmentmi znalostí, resp. medzi znalosťami navzájom.
- ✓ **Procedurálne.** Pravidlo definuje spôsob (procedúru), ako odvodiť novú znalosť (záver) pomocou iných znalostí (predpokladov).

Definujme si množinu atomických výrokov $\pi = \{p_1, \dots, p_N\}$. Tieto výroky sa môžu vyskytnúť v pravých stranách produkčných pravidiel, ktorých forma (nad π) bude nasledovná.

$$PP: L_1 \& L_2 \& \dots \& L_n \rightarrow P$$

Ľavá strana PP je tvorená elementárnou konjunkciou literálov L_1, \dots, L_K . Literály sú vlastne atomické výroky, alebo negované atomické výroky. Každý literál sa v konjunkcii vyskytuje najviac raz, či už negovaný alebo nie. Pravá strana PP je tvorená atomickým výrokom p , ktorý sa nevyskytuje v L_1, \dots, L_K . p je teda záver pravidla (consequent) a elementárna konjunkcia predstavuje predpoklady pravidla (antecedenty, evidencie).

Produkčné pravidlo sa niekedy označuje aj ako „if-then“ pravidlo. Niekedy sa pripúšťa, aby predpoklady tvorili disjunkciu. Väčšinou je za disjunkciu považovaná existencia viacerých pravidiel s rovnakým záverom, ale rôznymi predpokladmi.

Fakt, že formálne sa v ľavej časti nejakého PP môžu vyskytnúť tie isté atomické výroky, ktoré sa vyskytujú v pravej časti iného PP (nie toho istého) umožňuje reťazenie pravidiel. Napríklad:

$$\begin{aligned} A \& B &\rightarrow C \\ \text{non}C &\rightarrow D \\ D \& E &\rightarrow F \\ F &\rightarrow G \dots \end{aligned}$$

Reťazenie pravidiel umožňuje vytvárať hierarchickú štruktúru nazývanú **inferenčná sieť** (knowledge pattern). Inferenčná sieť nesmie obsahovať reťaz, ktorá by bola zároveň aj slučkou. Napríklad: $A \& B \rightarrow C$

$$\begin{aligned} \text{non}C &\rightarrow D \\ D &\rightarrow B. \end{aligned}$$

Pre odvodenie nových znalostí pomocou produkčných pravidiel sa najčastejšie používajú dva spôsoby: **1.modus ponens**

ak vieme, že platí tvrdenie	p
a súčasne platí implikácia	$p \rightarrow q$
potom platí tvrdenie	q

2.modus tollens

ak vieme, že neplatí tvrdenie	$\text{non } q$
a súčasne platí implikácia	$p \rightarrow q$
potom neplatí ani tvrdenie	$\text{non } p$

Oba spôsoby majú rovnakú deduktívnu silu. Sú deduktívne ekvivalentné. Avšak, modus tollens sa používa podstatne zriedkavejšie.

10. Ako funguje dopredné a spätné reťazenie?

Reťazenie produkčných pravidiel nastáva, keď záver jedného pravidla obsahuje tú istú znalosť, ako predpoklad ďalšieho. Existujú dva druhy reťazenia: dopredné a spätné.

DOPREDNÉ REŤAZENIE funguje na základe princípu: ak platí predpoklad, potom platí záver. Vykoná sa vždy prvé pravidlo, ktorého situačná časť je splnená. Niekedy sa mu hovorí **priame reťazenie**, alebo reťazenie riadené dátami (**data driven inference**). O tom, ktoré pravidlo sa odpáli, rozhoduje nielen ich poradie, ale aj situácia v báze dát. Jednoduchým príkladom takéhoto reťazenia môže byť nasledovný program v programovacom jazyku Prolog.

```
forward _ chaining :-  
    repeat ,  
    select (Rules ),  
    (empty (Rules ), select _ one (Rules , Rule ), execute (Rule ), fail )
```

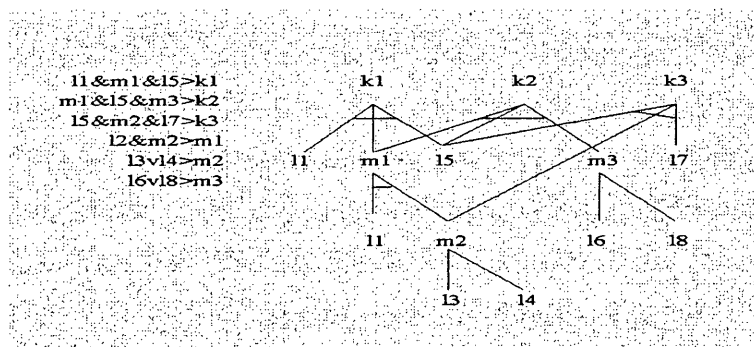
Dopredné reťazenie má svoje nevýhody. Nehodí sa veľmi na dokazovanie konkrétnej hypotézy, pretože popri tom odpáli aj mnoho nepotrebných pravidiel jednoducho preto, že bola náhodou splnená ich situačná časť. Hodí sa skôr na odvodenie všetkých možných výsledkov platných v aktuálnom modeli, teda na základe aktuálneho stavu BD.

SPÄTNÉ REŤAZENIE (Payne-McArtur, 1990) sa zameriava na dokázanie konkrétnej hypotézy. Dekomponuje konkrétny cieľ (dokázanie hypotézy, záveru) na podciele (podmienky v situačnej časti pravidla s daný záverom, resp. hypotézou v pravej časti). Potom sa snaží rekurzívne splniť tieto podciele (podmienky) tak, že hľadá pravidlá, ktoré majú dané podmienky vo svojej pravej časti, teda ako záver. Ide vlastne o prehl'adávanie inferenčnej siete do hĺbky. Jednoduchým príkladom takéhoto reťazenia môže byť nasledovný program v Prologu.

```
backward _ chaining (H) :-  
    select (H , Rules ),  
    select _ one (Rules , Conditions ),  
    exa min e (Conditions ).  
  
exa min e ([First | Others ]): -  
    exa min e (First ),  
    exa min e (Others ).  
  
exa min e (Condition ): -  
    present (Condition )! ;  
    backward _ chaining (Condition )
```

11. Ako je možné znázorniť inferenčnú sieť?

V dôsledku toho že predpoklad a záver pravidla sú rovnakého typu, môže záver nejakého pravidla vystupovať ako dielčí predpoklad iného pravidla. V dôsledku toho tvoria produkčné pravidlá hierarchickú štruktúru, ktorú je možné znázorniť orientovaným grafom, a ktorú nazývame inferenčná sieť. Príklad takejto inferenčnej siete je ilustrovaný nasledovným obrázkom.



Uzly grafu zodpovedajú jednotlivým výrokom. Hrany grafu reprezentujú produkčné pravidlá, pričom jedno PP môže byť vyjadrené jednou, alebo viacerými hranami. Logické funkcie sú chápané v zmysle klasickej boolovskej logiky (AND uzol je v grafe vyznačený dodatočnou čiarkou. OR uzol je bez čiarky.) AND uzol bude splnený, ak budú splnené všetky jeho predpoklady - subciele. OR uzol nebude splnený, ak nebudú splnené všetky jeho predpoklady - subciele. Keby sme sa chceli vyhnúť použitiu OR uzlov, čo je v ZS obvyklé, mohli by sme produkčné pravidlá na vyššie uvedenom obrázku prepísať do nasledovnej formy:

$L1 \& M1 \& L5 \rightarrow K1$
 $M1 \& L5 \& M3 \rightarrow K2$
 $L5 \& M2 \& L7 \rightarrow K3$
 $L6 \rightarrow M3$
 $L7 \rightarrow M3$
 $L3 \rightarrow M2$
 $L4 \rightarrow M2$
 $L2 \& M2 \rightarrow M1$

Uzly inferenčnej siete ako grafu môžeme deliť podľa rôznych hľadísk:

1. Podľa polohy ich delíme na:

- ✓ **Koreňové** (vrcholové), ktorých výroky sa nevyskytujú v predpokladoch žiadneho iného pravidla.
- ✓ **Listové**, ktorých výroky sa nevyskytujú v záveroch žiadneho pravidla.
- ✓ **Medzil'ahlé** sú všetky ostatné uzly.

2. Podľa spôsobu vyšetrovania:

- ✓ **Dotazovateľné.** Vyšetrenie ich platnosti sa realizuje dialógovým spôsobom, položením priamej otázky používateľovi v priebehu konzultácie. Ich platnosť sa môže overovať aj v báze dát.
- ✓ **Nedotazovateľné.** Vyšetrujú sa odvodením, pomocou platnosti svojich subcieľov (podmienok) a pomocou vlastnej funkcie (AND, OR).

3. Podľa cieľovosti:

- ✓ **Cieľové,** ktorých platnosť chceme dokázať, resp. odvodiť.
- ✓ **Necieľové** sú ostatné uzly, ktoré nám pomáhajú dokázať platnosť cieľových uzlov.

Tieto delenia sa navzájom prekrývajú podľa určitých pravidiel, ktoré sú definované v nasledovnej tabuľke.

	cieľový	necieľový	dotazovateľný	nedotazovateľný
koreňový uzol	áno	nie	nie	áno
medziláhly uzol	nie	áno	áno	áno
listový uzol	nie	áno	áno	nie

Medziláhly uzol môže byť aj dotazovateľný nejednoznačným spôsobom. Ak totiž používateľ nevie svojou odpoveďou dokázať medziláhlym uzlom reprezentované tvrdenie, môže byť toto dokázané odvodením, pomocou subcieľov.

Niektoré znalostné systémy umožňujú definovanie cieľovosti a dotazovateľnosti niektorých skupín uzlov tvorcomi bázy znalostí, alebo dokonca používateľovi.

12. Ako je možné inferenčnú sieť prehľadávať?

Inferenčnú sieť je možné prehľadávať rôznymi spôsobmi. V podstate sa kombinuje dopredné/spätne reťazenie s úplným/neúplným prehľadávaním. Z toho vyplývajú štyri možnosti:

- ✓ **Neúplný priamy chod.** Pri tomto chode sa inferenčný mechanizmus pohybuje od necieľového, najčastejšie listového uzlu priamo smerom ku koreňovému, resp. cieľovému uzlu. Pri prechode cez jednotlivé uzly sa tieto neexpandujú, t.j. systém sa nezaujíma o ďalšie hrany, ktoré vedú do aktuálneho uzla. Systém iba zisťuje, platnosť ktorých cieľových uzlov môže ovplyvniť platnosť daného listového – necieľového uzla. Dosiagnuté cieľové uzly sa označujú ako relevantné vzhľadom k zadanému listovému uzlu.
- ✓ **Úplný priamy chod.** Systém postupuje opäť od listových uzlov ku koreňovým so snahou priradovať aktuálnym uzlom pravdivostné hodnoty. Tentoraz ide o „pravú“ inferenciu a systém sa musí zaujímať aj o ostatné hrany vedúce do aktuálneho uzla a preveriť platnosť uzlov, z ktorých tieto hrany vyšli. Inak povedané, ak je to potrebné, systém expanduje uzly po ceste ku koreňu a overuje ich platnosť.
- ✓ **Úplný spätný chod.** Pri tomto chode sa systém snaží dokázať hypotézu spojenú s niektorým koreňovým uzlom. Preto postupuje spätne od zadaného cieľového, resp. koreňového uzla smerom ku listovým uzlom. Pri tomto postupe dokazuje (a teda expanduje, ak je to potrebné) uzly na zvolenej ceste. Nie vždy je nutná expanzia všetkých uzlov. Tak napríklad ďalšia expanzia nemá zmysel, ak: 1) aktuálny uzol už má priradenú pravdivostnú hodnotu (pravdivostná hodnota mu bola priradená pri overovaní iného cieľa), 2) aktuálny uzol je typu AND a jeden z jeho potomkov je nepravdivý, 3) aktuálny uzol je typu OR a jeden z jeho potomkov je pravdivý.
- ✓ **Neúplný spätný chod.** Opäť tu ide o potvrdzovanie hypotézy, pričom sa nedokazuje celá rozvetvená cesta od hypotézy – koreňového uzla ku všetkým relevantným listovým uzlom. V BZ môže byť uložená informácia o tom, ktoré z vetiev stromu vedú k najvierohodnejším alternatívam. Táto informácia umožní systému vyberať práve také vetvy. Uvedený postup je vyložene heuristický a súvisí s problematikou práce s neurčitou informáciou. Pri tomto postupe vychádza systém z koreňového uzla, zisťuje relevantné listové uzly, ktoré môžu ovplyvniť platnosť daného koreňa a vyberie jeden z nich ako alternatívu.

13. Čo je to neurčitá znalosť?

Neurčité znalosti nemajú podobu rigorózných tvrdení. Neurčité znalosti často reprezentujú rôzne tušenia a osvedčené postupy. Neurčité znalosti sú prirodzenou súčasťou riešenia experta a ako také sú neoddeliteľnou súčasťou znalostného systému.

Neurčité znalosti sú potrebné, pretože iba malú časť úloh sa podarilo analyzovať do tej miery, že ich možno uspokojujúco formálne popísať a riešiť. Sú potrebné preto, že iba malá časť vedomostí experta má tvar formálnych matematizovaných teórií, ktoré poskytujú jednoznačné výpočtové postupy vedúce k jednoznačnému riešeniu. Napriek tomu je expert schopný riešiť problémy pomocou znalostí, ktoré sú odvodené z jeho skúseností a z jeho individuálnych mentálnych modelov, t.j. pomocou neurčitých znalostí.

Jednoznačný výpočtový proces predstavuje **algoritmus**, ktorý pracuje s určitými znalosťami. Neformálny úsudkový postup zase predstavuje **heuristika**, ktorá pracuje s neurčitými znalosťami. Charakteristické vlastnosti algoritmu a heuristiky sú prehľadne ilustrované nasledovnou tabuľkou.

Algoritmus	Heuristika
rezultatívnosť	rezultatívnosť
konečnosť	konečnosť
hromadnosť	nie je hromadná
optimálny výsledok	vhodný výsledok
pomalší	rychlejšia
nepraktický	praktická

Algoritmus sa vyznačuje rezultatívnosťou, konečnosťou, hromadnosťou riešenia a garantuje optimálne riešenie. Hromadnosť riešenia znamená, že riešenie sa vždy nájde. Je pomalší a menej efektívny. Heuristika zaručuje iba rezultatívnosť a konečnosť riešenia. Nezaručuje jeho hromadnosť. Negarantuje optimálne riešenie. Poskytuje však vo väčšine prípadov vyhovujúce riešenie. Poskytuje praktickejšie riešenie za kratší čas.

Heuristike ako neformálnemu úsudkovému postupu, možno v určitom zmysle dôverovať. Táto dôvera je založená na existencii dostatočného množstva prípadov, keď sa osvedčila. Heuristika nám prináša riešenie, ktoré sa nedá dokázať, ale dá sa dobre použiť.

Uvažujme, ako by bolo možné zabrániť vstupu teroristov na palubu lietadla. Algoritmické riešenie by spočívalo v osobnej prehliadke každej osoby, ktorá vstúpi na palubu a jej batožiny (vrátane posádky a mechanikov). Heuristické riešenie by spočívalo v osobnej prehliadke iba tých cestujúcich, na ktorých reaguje detektor kovu, a ktorí svojím výzorom alebo správaním vyvolali podozrenie. V tomto prípade algoritmus úplne vylučuje možnosť preniknutia zbrane na palubu lietadla. Na druhej strane však vyžaduje príliš veľa času, je nákladný a nepopulárny.

Ako vyplýva z predošlého, jednou z charakteristických vlastností heuristických znalostí je ich neurčitosť. **Znalosti o neurčitosti** tvoria štvrtý typ ľudských znalostí, popri všeobecných, predmetných a konkrétnych. Táto skupina má osobitné postavenie medzi ostatnými znalosťami. Hovoríme, že má panoramatický charakter, pretože sa týka všetkých ostatných typov znalostí. Znalosti experta nemajú vždy celkom exaktný charakter. Jednotlivé znalosti používa s rôznou dôverou vo výsledok. Miera jeho dôvery je založená práve na použití 4. typu znalostí – znalostí o neurčitosti.

14. Aký je rozdiel medzi neurčitosťou v BZ a v BD?

Existencia neurčitých znalostí v znalostnom systéme je vynútená neurčitosťou vedomostí a faktov, s ktorými znalostný systém pracuje. Táto neurčitosť sa vyskytuje tak v BZ ako aj v BD. Neurčitosť v sa v BZ a v BD vyskytuje z nasledovných konkrétnych dôvodov:

- ✓ **Neurčitosť v BZ** býva zapríčinená tým, že znalosti experta nie sú celkom exaktné. Často sú to domnienky, ktoré sa opierajú o skúsenosti.
- ✓ **Neurčitosť v BD** býva zapríčinená neistými odpoveďami používateľa, subjektívnosťou jeho úsudku, odhadmi nedostupných informácií ako riešením situácie v prípade vysokých nákladov na získanie presných informácií, alebo nepresné a šumom zaťažené dáta.

Pod riešením úloh rozumieme odvodzovanie t.j. inferenciu nových znalostí. Novo odvodené znalosti taktiež vstupujú do inferenčného procesu a slúžia pre odvodenie ďalších znalostí. To, či novo odvodené znalosti budú mať rigorózný alebo neurčitý charakter závisí od charakteru znalostí, z ktorých boli odvodené. Sú možné 4 kombinácie. Tieto kombinácie sú uvedené v nasledovnej tabuľke.

Báza znalostí	Báza dát	Odvodená informácia
určitá	určitá	určitá
určitá	neurčitá	neurčitá
neurčitá	určitá	neurčitá
neurčitá	neurčitá	neurčitá

Z tejto tabuľky vyplýva, že akonáhle boli použité v inferencii nejaké neurčité znalosti či už v BZ alebo v BD, jej výsledkom je neurčitá novo odvodená znalosť.

Tretí prípad v tabuľke (v BZ sú neurčité znalosti a v BD sú určité znalosti a teda výsledkom je neurčitá znalosť) je možné použiť iba pri jednostupňovej inferencii. V tomto prípade odvodené neurčité znalosti nemôžu byť použité v ďalšom odvodzovaní, pretože sa predpokladá, že BD je určitá a teda nie je možné do nej zapísať neurčitý odvodený fakt.

Existujú systémy, ktoré si vedia poradiť aj s týmto problémom. Napríklad EXSYS (Janota, 2000) vie pokračovať v ďalších stupňoch inferenčného procesu aj keď sa vyskytne neurčitá znalosť v BZ a určitá v BD. EXSYS používa váhy. Tento systém odvodenú neurčitú znalosť transformuje na určitú a to prahovaním.

15. Aké druhy neurčitosti poznáme?

Neurčitosť znalostí máva rôzne príčiny. Z toho logicky vyplýva existencia rôznych typov neurčitostí. Jednotlivé typy sa od seba odlišujú aj prejavom navonok, pretože vyžadujú rôzne spôsoby spracovania a manipulácie.

Najčastejšie používané delenie je založené na príčinách vzniku neurčitosti:

- ✓ **Nekompletnosť** (incompleteness) spočíva vo fragmentálnosti ľudských poznatkov (to čo vieme sú iba ostrovčeky poznania). Človek dokáže preklenúť neúplnosť svojich poznatkov pomocou predpokladov o svete. Avšak všeobecné pravidlá (všeobecné znalosti) používané človekom nie sú aplikovateľné vo všetkých prípadoch. Majú výnimky. Preto uvažovanie človeka považujeme za nemonotónne (nemonotónna logika). Keďže sú znalosti chápané ako podmiennečne platné, novo prichodia informácia môže mať za následok ich revíziu. Indikátorom nekompletnosti je výskyt nasledovných lingvistických kvantifikátorov: väčšinou, obvykle, všeobecne, typicky.
- ✓ **Vágnosť** (imprecision) súvisí s procesom získavania znalostí od človeka t.j. so zdieľaním znalostí. To málo čo vieme (nekompletnosť) pri podávaní inému človeku ešte skreslíme (vágnosť). Vágne znalosti sú vyjadrované pomocou slov s vágnym významom: vysoký, nízky, starý. Keďže tieto výrazy nemajú jednoznačný význam, môžu byť rôzne pochopené. Vágnosť môže narastať kombináciou lingvistických kvantifikátorov a vágnych slov: obvykle vysoký, veľmi múdry, príliš bolestivý, temer starý.
- ✓ **Neurčitosť** (uncertainty) je niekedy zúžene chápaná ako jediný reprezentant neurčitosti v ZS. Odráža subjektívny charakter znalostí, resp. „ľudský faktor“ znalostí, teda: používanie heuristik namiesto algoritmov a matematických teórií, odhady a skúsenosti namiesto exaktných znalostí (v živote je bežné zovšeobecňovanie na základe dvoch, troch prípadov), nevedomosť a neznalosť niektorých vecí a súvislostí, chyba merania, povrchnosť chápania, osobná predpojatosť, dôvera experta v svoje znalosti.

Ak predpokladáme, že x je množina predpokladov a y je množina záverov, potom nekompletnosť vieme definovať tvrdením, že k predpokladu neexistuje záver:

$$\exists x \in X : \neg \exists y \in Y$$

resp., nevieme potvrdiť pravidlo, ale vieme potvrdiť jeho všeobecnejšiu variantu:

$$\exists x \in A \subset X : \exists y \in Y$$

Podobne vágnosť môžeme definovať:

$$\exists x \in X : \exists y \in B \subset Y, \text{ čo znamená, že poznáme záver, ale nevieme ho presne špecifikovať.}$$

Napokon neurčitosť môžeme definovať: $\exists x \in X : \exists y \in Y$. To znamená, že vzťah nie je zaručený, ale platí s určitou mierou istoty, s určitou váhou. Teda každé produkčné pravidlo je asociované so stupňom korelácie – váhou w (weight). Táto väčšinou číselná váha určuje silu pravidla, resp. stupeň jeho dôveryhodnosti, stupeň jeho spoľahlivosti, alebo mieru dôvery v jeho platnosť.

$$PP : P \rightarrow Z(w) \quad PP : P \xrightarrow{w} Z$$

16. Ako je možné neurčitost' reprezentovať?

Neurčitost' v BZ (váha produkčného pravidla) môže byť vyjadrená inou formou ako neurčitost' v BD (neurčitost' faktov).

Neurčitost' môže mať formu symbolickú a numerickú. V nasledujúcej tabuľke sú obidve formy stručne charakterizované.

Symbolická reprezentácia neurčitosti	Numerická reprezentácia neurčitosti
slovný popis	číslo
s pôvodom	bez pôvodu
odvádzanie znalosti je problém	odvádzanie znalosti bez problému

Pri **symbolickej** reprezentácii neurčitosti je exaktnosť znalostí kvalifikovaná slovným popisom spolu s presným pôvodom. Napríklad: prší (Paľo odhadol)

padajú kvapky → prší (Jožo skúsil).

Problémové je pri tejto reprezentácii odvádzanie neurčitosti novej znalosti.

Pri **numerickej** reprezentácii neurčitosti je exaktnosť znalostí kvalifikovaná numerickou hodnotou bez pôvodu. Odvádzanie neurčitosti nových znalostí nie je problémom, dá sa vyčíslieť. Napríklad: prší (70%)

padajú kvapky (90%), z toho vyplýva že prší(65%).

Numerická reprezentácia sa ďalej delí podľa:

- ✓ **POČTU POUŽITÝCH HODNÔT na jedno hodnotovú a viac hodnotovú** (typicky dvoj hodnotovú) Pri **dvoj hodnotovej** neurčitosti sa dôvera a nedôvera vyjadrujú zvlášť. Nedôvera je doplnkom ku dôvere: $P(H) + P(\neg H) = 1$. Tieto dve hodnoty sa môžu kombinovať do jednej s názvom činiteľ istoty, alebo do inej dvojice s názvom konfidenčný interval. **Konfidenčný interval** je interval, v ktorom sa môže nachádzať skutočná neurčitost'.
- ✓ **PODĽA ABSOLÚTNOSTI VYJADRENIA** na absolútnu a relatívnu neurčitost'. **Absolútna** neurčitost' sa vyjadruje priamo zvyčajne reálnym číslom z intervalu $\langle 0,1 \rangle$ alebo $\langle -1,1 \rangle$. Hraničné hodnoty tohto intervalu udávajú absolútnu platnosť/neplatnosť znalosti. Často sa hodnota vyberá z N celo číselných hodnôt pre potreby mnoho hodnotových logík. **Relatívna** neurčitost' vyjadruje zmenu absolútnej neurčitosti. Váha pravidla určuje, ako sa zmení neurčitost' záveru v prípade splnenia predpokladov. Patria sem: šanca, činiteľ istoty, miera postačiteľnosti a nevyhnutnosti. Šanca je označovaná ako pravdepodobnostný pomer a je definovaná nasledovne:

$$O(H) = \frac{P(H)}{P(\neg H)} = \frac{P(H)}{1 - P(H)}$$

17. Aké modely práce s neurčitosťou poznáme?

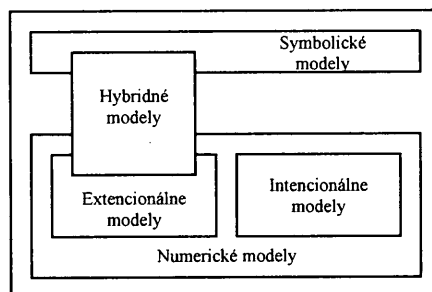
V prípade výskytu neurčitosti v pôvodných znalostiach je potrebné nielen odvodzovať nové znalosti, ale aj odvodzovať ich neurčitosť. Hovoríme o **šírení neurčitosti** cez inferenčnú sieť. Toto šírenie sa deje od predpokladov k záverom podľa určitých pravidiel, ktoré vytvárajú model práce s neurčitou informáciou, resp. kalkul neistého uvažovania.

Numerické modely práce s neurčitou informáciou delíme na:

- ✓ **intenzionálne modely.** Tieto modely riešia celý problém globálne (Petrušová, 1989). Uvažujú o všetkých závislostiach medzi všetkými znalosťami. Úlohou je nájsť najlepšiu distribúciu neurčitosti, ktorá vyhovuje každej čiastočnej - marginálnej znalosti. Odpovede používateľa sú chápané ako marginálne informácie. Vo všeobecnosti nie je možná žiadna konkrétna kombinačná funkcia. Spôsob odvádzania nových váh musí čo najlepšie aproximovať čiastočné - marginálne znalosti, preto sa mení pri každom konkrétnom prípade. Problémom tohto prístupu je obtiažna modifikovateľnosť BZ a dlhé trvanie odvádzania distribúcie neurčitosti. Prednosťou tohto prístupu je, že po odvodení poznáme všetky závislosti a vieme, že riešenie je teoreticky správne.
- ✓ **extenzionálne modely.** Sú založené na princípe lokálnosti t.j. šírení neurčitosti iba v nejakom lokálnom okolí. Taktiež sú založené na princípe extenzionality. Extenzionálne modely predpokladajú existenciu kombinačných funkcií, ktoré sa nemenia od prípadu k prípadu. Výpočet inferenčných váh je realizovaný pevne zvolenou sadou operácií. Výhody extenzionálnych modelov sú nasledujúce: rýchlosť, jednoduchosť, ľahká modifikovateľnosť, ľahká implementovateľnosť. Na druhej strane nevýhodou týchto modelov je, že si nemôžu nárokovať teoretickú správnosť. Teoretická správnosť je totiž podmienená dodržaním určitých obmedzení (napríklad nezávislosť predpokladov), ktoré v praxi často nie sú dodržané. Často nie sú známe ani znalostnému inžinierovi. Teoretická správnosť sa ťažko overuje. Často ani samotní experti nemajú predstavu, čo by malo vyjsť ako správny výsledok.

Princíp extenzionality hovorí, že pravdivostná hodnota zloženého výroku je určená pravdivostnou hodnotou jeho zložiek a nezávisí od platnosti výrokov, ktoré nie sú jeho súčasťou.

Extenzionálne modely práce s neurčitosťou delíme podľa spôsobu reprezentácie neurčitosti na: logické, pravdepodobnostné, fuzzy a hybridné, ktoré spájajú prednosti rôznych modelov. Vzájomný vzťah rôznych modelov je ilustrovaný nasledovným obrázkom.



neurčit
kombin
extenzi
vyňaté
kovatel
vzájom

šírenia
každém
neurčit
na pro
odpove

1)

2)

3)

Neurči
neurčit
priamo
produk
pomoc

4)

5)

Jednot
podmr

V d'al
s neur
Shaffe

18. Ako je definovaný všeobecný extenzionálny model?

Pre symbolické a intenzionálne modely neexistuje všeobecne platný model šírenia neurčitosti. Pre extenzionálne modely existuje model šírenia neurčitosti v tvare sady kombinačných funkcií. V takomto modeli platí princíp modularity, ktorý vyplýva z princípu extenzionality. Princíp modularity znamená, že produkčné pravidlo môže byť pridané, resp. vyňaté z BZ bez uvažovania interakcií s inými produkčnými pravidlami. Táto modifikovateľnosť, podobne ako princíp extenzionality, predpokladá splnenie požiadavky vzájomnej nezávislosti predpokladov (čo v praxi často nie je splnené).

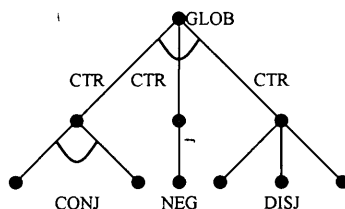
Kombinačné funkcie udávajú predpis pre manipuláciu s neurčitosťou v priebehu jej šírenia inferenčnou sieťou. Inferenčná sieť je zložená z elementárnych typov sietí a ku každému z nich existuje jedna kombinačná funkcia. Extenzionálny model spracováva znalosti neurčité, ale prípadne aj určité. Tieto znalosti sa vyjadrujú AND/OR grafom, prispôbeným na propagáciu neurčitosti. Sem patria tri typy elementárnych informačných sietí a im odpovedajú tri kombinačné funkcie: negácia, konjunkcia a disjunkcia.

- 1) Kombinačná funkcia **negácia** transformuje neurčitosť predpokladu na neurčitosť jeho negácie a je definovaná nasledovne: $N(\neg P1) = f_{NEG}(N(P1))$.
- 2) Kombinačná funkcia **konjunkcia** transformuje neurčitosť predpokladov na neurčitosť ich konjunkcie a je definovaná nasledovne: $N(P1 \wedge P2) = f_{CONJ}(N(P1), N(P2))$.
- 3) Kombinačná funkcia **disjunkcia** transformuje neurčitosť predpokladov na neurčitosť ich disjunkcie a je definovaná nasledovne: $N(P1 \vee P2) = f_{DISJ}(N(P1), N(P2))$.

Neurčitosť v BZ predstavuje prenos znalostí cez neurčité produkčné pravidlá. Poznáme neurčitosť predpokladu $N(P)$ a neurčitosť produkčného pravidla $N(P \rightarrow Z)$. Dostaneme buď priamo neurčitosť záveru $N(Z)$, alebo príspevok k neurčitosti záveru, ak existuje viac produkčných pravidiel s týmto záverom. Neurčitosť záveru, resp. príspevok dostaneme pomocou funkcie CTR a príspevky budeme skladat' pomocou funkcie GLOB.

- 4) Kombinačná **funkcia CTR** je definovaná nasledovne $N(Z) = f_{CTR}(N(P), N(P \rightarrow Z))$. Táto funkcia umožňuje reťaziť neurčité produkčné pravidlá, preto sa niekedy označuje ako sekvenčná kombinácia.
- 5) Kombinačná **funkcia GLOB** skladá príspevky od jednotlivých pravidiel, získaných pomocou CTR. Je definovaná nasledovne $N(Z) = f_{GLOB}(N(P1 \rightarrow Z), N(P2 \rightarrow Z))$. Používa sa v prípade existencie viacerých pravidiel s rovnakým záverom. Niekedy sa preto označuje ako paralelná kombinácia.

Jednotlivé extenzionálne modely používajú ako sadu svojich kombinačných funkcií nejakú podmnožinu množiny {NEG, CONJ, DISJ, CTR, GLOB}. Funkcia DISJ sa málo používa.



V ďalšom sa budeme venovať nasledovným konkrétnym extenzionálnym modelom práce s neurčitou informáciou: Subjektívna Bayes-ovská metóda, Algebraická teória, Dempster-Shafferova metóda, Fuzzy model.

19. Prečo je Subjektívna Bayes-ovská metóda spracovania neurčitosti subjektívna?

Pretože vychádza z teórie pravdepodobnosti chápanej v subjektívnom zmysle, nie vo frekvenčnom zmysle. Zatiaľ čo frekvenčná definícia pravdepodobnosti predstavuje pomer početnosti výskytov sledovaného javu k početnosti výskytov všetkých javov, subjektívna definícia pravdepodobnosti predstavuje pomer odhadu početnosti výskytov sledovaného javu k početnosti výskytov všetkých javov.

Subjektívna Bayes-ovská metóda je jednou z najčastejšie používaných spôsobov práce s neurčitosťou (Berka, 1994). Zohľadňuje tak neurčitosť pravidiel (hrany inferenčnej siete), ako aj apriórnu neurčitosť výrokov (uzlov inferenčnej siete). **Apriórna** neurčitosť je daná „a priori“ pred začatím procesu šírenia neurčitosti. V procese šírenia neurčitosti v inferenčnej sieti sa mení na **aposteriórnu**.

Neurčitosť produkčného pravidla môže byť v Bayes-ovskej metóde vyjadrená dvojakým spôsobom:

- ✓ **absolútne**. Pri absolútnom vyjadrení neurčitosti sa používajú podmienené pravdepodobnosti, a to $P(H/E)$ ako pravdepodobnosť záveru H v prípade splnenia predpokladu E a $P(H/\sim E)$ ako pravdepodobnosť záveru H v prípade nesplnenia predpokladu E.

- ✓ **relatívne**. Relatívne vyjadrenie neurčitosti môže mať formu miery postačiteľnosti a miery postačiteľnosti. Miera postačiteľnosti LS (logical sufficiency) vyjadruje zmenu šance záveru v prípade splnenia predpokladu. Veľká hodnota LS spôsobí veľký nárast šance záveru H. Hovoríme, že predpoklad E je postačujúci k dokázaniu H. Ak predpokladáme, že $O(H)$ je apriórna a $O(H/E)$ aposteriórna šanca, potom je možné LS definovať: $LS = \frac{O(H|E)}{O(H)}$, pričom platí: $O(H) = \frac{P(H)}{1 - P(H)}$. Na druhej strane miera

nevyhnutnosti LN (logical necessity) vyjadruje zmenu šance záveru v prípade nesplnenia predpokladu. Malá hodnota LN spôsobí pokles šance. K dokázaniu záveru H je nutná platnosť predpokladu E. Platí: $LN = \frac{O(H|\sim E)}{O(H)}$.

20. Ako je v Subjektívne Bayes-ovskej metóde definovaná kombinačná funkcia CTR?

Predpokladajme, že v procese inferencie predpoklad E nadobúda nejakú aposteriornu pravdepodobnosť $P(E|E')$. E' označuje relevantné porovnanie súvisiace s konkrétnym prípadom. Na základe konkrétneho pozorovania je možné pravdepodobnosť záveru zložiť z dvoch zložiek: z podpory od splnenia a podpory od nespĺnenia predpokladu, čo je možné vyjadriť nasledovne:

$$P(H|E') = P(H|E|E') + P(H|\neg E|E') = P(H|E, E')P(E|E') + P(H|\neg E, E')P(\neg E|E')$$

Ak poznáme pravdivostnú hodnotu výroku E , potom pozorovanie neprináša ďalšie informácie

$$P(H|E, E') = P(H|E)$$

o hypotéze H a teda platí: $P(H|\neg E, E') = P(H|\neg E)$. To nám umožňuje nasledovné úpravy:

$$P(H|E') = P(H|E)P(E|E') + P(H|\neg E)P(\neg E|E')$$

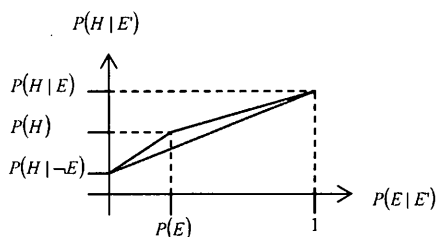
$$P(H|E') = P(H|E)P(E|E') + P(H|\neg E)[1 - P(E|E')]$$

$$P(H|E') = P(H|E)P(E|E') + P(H|\neg E) - P(H|\neg E)P(E|E')$$

$$P(H|E') = P(H|\neg E) + [P(H|E) - P(H|\neg E)]P(E|E')$$

$$y = P(H|\neg E) + [P(H|E) - P(H|\neg E)]x$$

Výsledkom uvedeného odvodenia je priamka, ktorá je znázornená v nasledovnom obrázku.

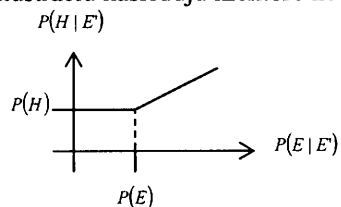


Túto priamku zalomíme takým spôsobom, aby obsahovala aj apriórne pravdepodobnosti predpokladu a záveru, teda $P(E)$ a $P(H)$. Interpoláciou týchto troch bodov vznikne lineárna lomená funkcia, ktorá predstavuje **definíciu kombinačnej funkcie CTR** v nasledovnom tvare:

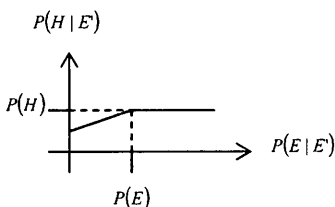
$$0 \leq P(E|E') \leq P(E) \Rightarrow P(H|E') = P(H|\neg E) + \frac{P(H) - P(H|\neg E)}{P(E)} P(E|E')$$

$$P(E) \leq P(E|E') \leq 1 \Rightarrow P(H|E') = P(H) + \frac{P(H|E) - P(H)}{P(E)} [P(E|E') - P(E)]$$

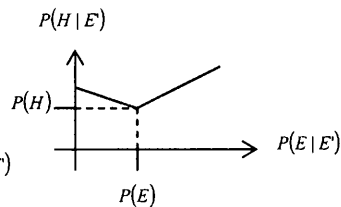
Funkcia CTR môže mať aj iný tvar, ako je uvedené na predchádzajúcom obrázku. Pre ilustráciu nasledujú niektoré neobvyklé prípady funkcie CTR.



Nesplnenie predpokladu nemá vplyv na záver



Splnenie predpokladu nemá vplyv na záver



Rozporná funkcia, nesplnenie aj splnenie podporuje záver

21. Ako je v Bayes-ovskej metóde definovaná funkcia GLOB a ostatné kombinačné funkcie?

Skladanie príspevkov jednotlivých pravidiel s tým istým záverom do aposteriórnej pravdepodobnosti záveru $P(H | E_1', E_2', \dots, E_3')$ je realizované v relatívnom tvare. Určia sa váhy jednotlivých pravidiel, kde váha j-teho pravidla je definovaná: $L_j = \frac{O(H | E_j)}{O(H)}$, pričom L predstavuje mieru postačiteľnosti LS. Pri splnení predpokladu štatistickej nezávislosti jednotlivých predpokladov, možno **definovať funkciu GLOB** nasledovne:

$$O(H | E_1', E_2', \dots, E_3') = \left(\prod_{j=1}^n L_j \right) O(H).$$

Táto definícia funkcie GLOB nám poskytuje výslednú aposteriórnu šancu záveru, ktorú je potrebné transformovať na výslednú aposteriórnu pravdepodobnosť záveru. Pre túto transformáciu môže poslúžiť nasledovná formula:

$$P(H | E_1', E_2', \dots, E_3') = \frac{O(H | E_1', E_2', \dots, E_3')}{1 + O(H | E_1', E_2', \dots, E_3')}.$$

Ostatné kombinačné funkcie sú v Bayes-ovskej metóde spracovania neurčitosti definované nasledovne:

- ✓ **Funkcia NEG** je definovaná formulou: $P(\neg H) = 1 - P(H)$.
- ✓ Pre **funkciu CONJ** zase platí: $P(H_1 \wedge H_2) = \min(P(H_1), P(H_2))$.
- ✓ Napokon **funkcia DISJ** je definovaná: $P(H_1 \vee H_2) = \max(P(H_1), P(H_2))$

22. Čo predstavuje takzvaný intuitívny model spracovania neurčitosti?

Intuitívny model spracovania neurčitosti predstavuje úvod ku algebraickej teórii, t.j. ďalšej metóde spracovania neurčitosti.

Aj keď jednotlivé kombinačné funkcie môžu byť definované rôzne, intuitívne možno stanoviť ich interpretáciu. V ďalšom sa budeme zaoberať iba funkciami CTR a GLOB. Predpokladáme existenciu troch špeciálnych hodnôt:

\Downarrow -minimálny prvok – úplná neplatnosť, napríklad -1

0 - neutrálny prvok

\Uparrow - maximálny prvok – úplná platnosť, napríklad +1.

Produkčné pravidlo s neurčitou môže potom interpretovať nasledovne: ak je predpoklad úplne splnený (je pravdivý), potom záver platí s váhou w : $P \xrightarrow{w} Z$. Ak predpoklad nie je splnený úplne, potom príspevok pravidla k posilneniu dôvery v záver pravidla je menší ako w . Čím väčšia je neurčitost' predpokladu, tým menší je príspevok k platnosti záveru.

Z uvedeného vyplývajú určité podmienky pre funkcie CTR a GLOB. Pre sekvenčnú kombinačnú funkciu CTR musia byť splnené predpoklady spojitosti a monotónnosti. Ďalej platí:

- ✓ Niekedy sa funkcia CTR uvažuje iba v intervale $(0, \uparrow)$. V intervale $(\downarrow, 0)$ potom platí $CTR(e, w) = 0$.
- ✓ Inokedy sa uvažujú obidva intervaly, a teda uvažuje sa aj komplementárne pravidlo $\sim P \rightarrow Z$.

Pre paralelnú kombinačnú funkciu GLOB musí platiť:

- ✓ Ak prvé aj druhé produkčné pravidlo podporuje záver, potom výsledná váha je posilňovaná.
- ✓ Ak prvé aj druhé produkčné pravidlo oslabuje záver, potom výsledná váha je oslabovaná.
- ✓ Ak jedno pravidlo záver podporuje a druhé ho vyvracia, potom sa vplyvy eliminujú a výsledná váha odpovedá vplyvu silnejšieho produkčného pravidla, oslabovaného vplyvom slabšieho pravidla.

Nasledujú ďalšie vlastnosti funkcie GLOB za predpokladu že e_1, e_2 a e_3 sú príspevky troch produkčných pravidiel k platnosti záveru:

- 1) **Komutatívnosť** znamená, že výsledok nezávisí na poradí uvažovania príspevkov a je definovaná: $GLOB(e_1, e_2) = GLOB(e_2, e_1)$.
- 2) **Asociatívnosť** umožňuje spracovávanie príspevkov vždy po dvoch: $GLOB(e_1, GLOB(e_2, e_3)) = GLOB(GLOB(e_1, e_2), e_3)$.
- 3) **Neutrálny prvok** má tú vlastnosť, že výsledná váha na ňom nezávisí: $GLOB(0, e_1) = e_1$.
- 4) **Opačný prvok** zohráva svoju úlohu v prípade, ak jedno pravidlo posilňuje a druhé oslabuje záver. Navzájom sa eliminujú, ak sú ich vplyvy rovnaké, čo je definované: $e_1 = -e_2 \Rightarrow GLOB(e_1, e_2) = 0$.
- 5) **Monotónnosť** je vlastnosť, ktorá sa dá interpretovať tvrdením, že čím je vyšší príspevok predpokladu pravidla, tým je vyššia váha záveru pravidla: $e_1 \geq e_2 \Rightarrow GLOB(e_1, e_3) \geq GLOB(e_2, e_3)$.

23. Ako je možné spracovať extrémne hodnoty neurčitosti?

Pod extrémnymi hodnotami neurčitosti rozumieme hodnotu veľmi malú \Downarrow (blízku 0) a hodnotu veľmi veľkú \Uparrow (blízku možnému maximu). Pre tieto hodnoty platí:

$$GLOB(e, \Downarrow) = \Downarrow$$

$$GLOB(e, \Uparrow) = \Uparrow$$

$$GLOB(\Downarrow, \Uparrow) = ?$$

Ak máme viacero príspevkov k platnosti záveru, ktoré nie sú extrémne, tak sa zložia podľa nejakej vopred definovanej funkcie (napríklad sa vypočíta ich priemerná hodnota). Ak iba jeden z príspevkov predstavuje extrémnu hodnotu, potom táto istá hodnota bude priradená aj platnosti záveru. Akoby extrémne hodnoty prebýjali neextrémne hodnoty. Ak máme viacero príspevkov extrémnych, ale všetky sú toho istého druhu (všetky sú maximá, alebo všetky sú minimá), potom aj záveru bude priradená maximálna, resp. minimálna neurčitosť. Problém nastáva, keď máme viacero extrémnych príspevkov z opačného spektra, lebo musíme riešiť ako skladať veľké maximum s veľkým minimom. V týchto prípadoch totiž neplatí asociatívnosť, pretože ak by sme predpokladali, že: $GLOB(\Downarrow, \Uparrow) = 0$, potom dostávame dva navzájom si protirečiace závery:

$$GLOB(\Uparrow, GLOB(\Uparrow, \Downarrow)) = \Uparrow$$

$$GLOB(GLOB(\Uparrow, \Uparrow), \Downarrow) = 0$$

Z uvedeného vyplýva ďalšia vlastnosť kombinačnej funkcie GLOB, a to jej nearchimedovský charakter. Funkcia je **archimedovská**, ak skladaním veľkého počtu malých podporujúcich váh sa blížíme k absolútnemu potvrdeniu záveru. Alebo skladaním veľkého počtu malých popierajúcich váh sa blížíme k absolútnemu vyvráteniu záveru, čo sa v jazyku matematiky zapíše nasledovne:

$$\Downarrow \ll e < 0 \Rightarrow GLOB(GLOB(\dots GLOB(e, e), \dots, e), e) \rightarrow \Downarrow.$$

Avšak je otázkou, či nás mnoho slabých tušení naozaj privedie blízko istote.

24. Čo je to algebraická teória?

Algebraická teória je model spracovania neurčitosti založený na algebraických štruktúrach. Táto teória vznikla v Prahe. Rozpracoval ju Prof. Hájek (Hájek, 1987). Predstavuje zovšeobecňujúci prístup k spracovaniu neurčitosti. Sústreďuje sa na kombinačnú funkciu GLOB a jej algebraické vlastnosti.

Ak funkcia GLOB má požadované vlastnosti komutatívnosti, asociatívnosti, neutrálneho prvku, opačného prvku a monotónnosti, potom tvorí usporiadanú komutatívnu grupu, alebo **Ábelovu grupu - OAG (Ordered Abelian Group)**. Podľa toho, či sa použije grupová operácia plus, alebo krát, rozoznávame dva najznámejšie prípady Ábelovej grupy:

operácia:	plus (+)	krát (.)
interval definície:	$(-\infty, \infty)$	$(0, \infty)$
neutrálny prvok	$N = 0$	$N = 1$
opačný prvok	$Q(x) = -x$	$Q(x) = \frac{1}{x}$

Zatiaľ čo Ábelova grupa má definičný obor $(-\infty, \infty)$, resp. $(0, \infty)$, neurčitosť je definovaná na intervale $\langle -1, 1 \rangle$, kde platí:

-1	určite nie
0	neviem
1	určite áno.

Teda ak chceme použiť Ábelovu grupu, musíme neurčitosť transformovať z intervalu $\langle -1, 1 \rangle$ na interval $(-\infty, \infty)$, vykonať skladanie príspevkov pomocou funkcie GLOB a výsledok transformovať späť na interval $\langle -1, 1 \rangle$. Na to slúžia nasledovné funkcie:

- ✓ transformácia z intervalu $\langle -1, 1 \rangle$ na interval $(0, \infty)$ alebo $(-\infty, \infty)$

$$f : \langle -1, 1 \rangle \rightarrow (0, \infty)$$

$$f : \langle -1, 1 \rangle \rightarrow (-\infty, \infty)$$

- ✓ spätná transformácia z intervalu $(0, \infty)$ alebo $(-\infty, \infty)$ na interval $\langle -1, 1 \rangle$

$$g : (0, \infty) \rightarrow \langle -1, 1 \rangle$$

$$g : (-\infty, \infty) \rightarrow \langle -1, 1 \rangle$$

- ✓ postup práce pri transformovaní

$$\left. \begin{array}{l} x \in \langle -1, 1 \rangle \xrightarrow{f} x' \in (-\infty, \infty) \\ y \in \langle -1, 1 \rangle \xrightarrow{f} y' \in (-\infty, \infty) \end{array} \right\} GLOB(x', y') = z' \in (-\infty, \infty) \xrightarrow{g} \langle -1, 1 \rangle$$

Funkcia GLOB môže byť definovaná dvojako, podľa toho či je založená na grupovej operácii plus, alebo krát. Nasledujú možné definície funkcie GLOB:

- ak grupovou operáciou je „+“: $GLOB(x, y) = g(f(x) + f(y)) = g(z)$
- ak grupovou operáciou je „*“: $GLOB(x, y) = g(f(x) \cdot f(y)) = g(z)$.

25. Ako môže vyzerat' funkcia GLOB v algebraickej teórii?

V ďalšom si uvedieme príklady konkrétnych definícií funkcie GLOB v algebraickej teórii, reprezentované Ábelovou grupou:

I. Majme dané transformačné funkcie v tvare:

$$f = \ln \frac{1+x}{1-x} \quad g = \frac{e^z - 1}{e^z + 1}, \text{ pričom sa operuje na grupovej operácii „+“}. \text{ Potom je}$$

možné odvodiť funkciu GLOB, používanú v systéme MYCIN, nasledovným postupom:

$$\begin{aligned} GLOB(x, y) &= g(f(x) + f(y)) = \frac{e^{\ln \frac{1+x}{1-x} + \ln \frac{1+y}{1-y}} - 1}{e^{\ln \frac{1+x}{1-x} + \ln \frac{1+y}{1-y}} + 1} = \frac{e^{\ln \left(\frac{1+x}{1-x} \cdot \frac{1+y}{1-y} \right)} - 1}{e^{\ln \left(\frac{1+x}{1-x} \cdot \frac{1+y}{1-y} \right)} + 1} = \frac{\frac{1+x}{1-x} \cdot \frac{1+y}{1-y} - 1}{\frac{1+x}{1-x} \cdot \frac{1+y}{1-y} + 1} = \\ &= \frac{(1+x)(1+y) - (1-x)(1-y)}{(1+x)(1+y) + (1-x)(1-y)} = \frac{1+x+y+xy - 1+x+y-xy}{1+x+y+xy + 1-x-y+xy} = \frac{2x+2y}{2+2xy} = \frac{x+y}{1+xy} \end{aligned}$$

II. Majme dané transformačné funkcie v tvare:

$$f = \ln \frac{1}{1-x} \quad g = \frac{e^z - 1}{e^z}, \text{ pričom sa operuje na grupovej operácii „+“}. \text{ Potom je}$$

možné odvodiť funkciu GLOB, ktorá odpovedá špecifickému prípadu – metóde Činiteľ istoty, a taktiež sa využíva ako jedna z možností spracovania neurčitosti v systéme EXSYS:

$$\begin{aligned} GLOB(x, y) &= g(f(x) + f(y)) = \frac{e^{\ln \frac{1}{1-x} + \ln \frac{1}{1-y}} - 1}{e^{\ln \frac{1}{1-x} + \ln \frac{1}{1-y}}} = \frac{e^{\ln \left(\frac{1}{1-x} \cdot \frac{1}{1-y} \right)} - 1}{e^{\ln \left(\frac{1}{1-x} \cdot \frac{1}{1-y} \right)}} = \frac{\frac{1}{1-x} \cdot \frac{1}{1-y} - 1}{\frac{1}{1-x} \cdot \frac{1}{1-y}} = \\ &= \frac{1 - (1-x)(1-y)}{1} = 1 - 1 + x + y - xy = x + y - xy \end{aligned}$$

III. Majme dané transformačné funkcie v tvare:

$$f = \frac{x}{1-x} \quad g = \frac{z}{1+z}, \text{ pričom sa operuje na grupovej operácii „*“}. \text{ Potom je}$$

možné odvodiť funkciu GLOB, ktorá odpovedá špecifickému prípadu algebraickej teórie – Dempster-Shafferovej metóde, ktorá pracuje bez vágnosti:

$$GLOB(x, y) = g(f(x)f(y)) = \frac{\frac{x}{1-x} \cdot \frac{y}{1-y}}{1 + \frac{x}{1-x} \cdot \frac{y}{1-y}} = \frac{xy}{(1-x)(1-y) + xy} = \frac{xy}{1-x-y+xy+xy} = \frac{xy}{1-x-y+2xy}$$

IV. V tomto prípade si transformačné funkcie musíme odvodiť.

V ďalšom ukážeme, že aj Bayesova metóda je špeciálnym prípadom Algebraickej teórie. Bayes používa interval $(0, \infty)$, z toho vyplýva, že pri odvádzaní pôjde o Ábelovu grupu 2. typu s operáciou násobenia. Teraz si odvodíme návrh transformačnej funkcie g pre spätnú transformáciu. Nech: o je šanca a p je pravdepodobnosť. Potom transformácia šance na

pravdepodobnosť predstavuje nasledovný posun: $O \rightarrow P$
 $(0, \infty) \rightarrow (0, 1)$. Šanca je definovaná funkciou

$O = \frac{P}{1-P}$. Táto definícia predstavuje transformáciu pravdepodobnosti na šancu. K nej

inverzná funkcia potom transformuje šancu na pravdepodobnosť $P = \frac{O}{1+O}$.

V transformácii musíme pokračovať až po interval $\langle -1, 1 \rangle$. Transformácia $(0, 1) \rightarrow \langle -1, 1 \rangle$ sa dá vykonať pomocou funkcie $w = 2P - 1$. Zložením obidvoch transformačných krokov dostaneme hľadanú funkciu g :

$$w = 2P - 1 = 2 \frac{O}{1+O} - 1 = \frac{2O}{1+O} - \frac{1+O}{1+O} = \frac{2O - 1 - O}{1+O} = \frac{O - 1}{O + 1} = \frac{z - 1}{z + 1} = g.$$

Transformačná funkcia f je inverznou k funkcii g , preto ju môžeme zložiť z funkcií, ktoré sú inverzné k tým, ktoré sme použili pri odvádzaní funkcie g . Transformácia z intervalu $\langle -1, 1 \rangle$

na interval $(0, 1)$ sa dá realizovať pomocou funkcie $P = \frac{w+1}{2}$ a následná transformácia z

intervalu $(0, 1)$ na interval $(0, \infty)$ pomocou funkcie $O = \frac{P}{1-P}$. Zložením týchto dvoch funkcií dostaneme hľadanú transformačnú funkciu f :

$$O = \frac{\frac{w+1}{2}}{1 - \frac{w+1}{2}} = \frac{w+1}{2-w-1} = \frac{w+1}{1-w} = \frac{1+x}{1-x} = f.$$

Nuž teda máme dané transformačné funkcie f a g a vieme, že sa používa grupová operácia „*“. Teda môžeme odvodiť funkciu GLOB:

$$GLOB \quad (x, y) = \frac{\frac{1+x}{1-x} \cdot \frac{1+y}{1-y} - 1}{\frac{1+x}{1-x} \cdot \frac{1+y}{1-y} + 1} = \frac{(1+x)(1+y) - (1-x)(1-y)}{(1+x)(1+y) + (1-x)(1-y)} = \frac{x+y}{1+xy}$$

Napokon môžeme skonštatovať, že Algebraická teória dokáže emulovať Dempster-Shafferovu metódu, ktorá je implementovaná v známom znalostnom systéme MYCIN, ako aj Subjektívnu Bayesovu metódu, ktorá je implementovaná v znalostnom systéme EXSYS.

26. Ako sú v algebraickej teórii definované ostatné kombinačné funkcie?

Definície ostatných kombinačných funkcií: negácie, konjunkcie a disjunkcie, vychádzajú z intuitívneho modelu. Sú definované nasledovne:

$$\begin{aligned} NEG(w) &= -w \\ CONJ(w_1, w_2) &= \min(w_1, w_2) \\ DISJ(w_1, w_2) &= \max(w_1, w_2) \end{aligned}$$

Funkcia CTR je definovaná dvojako: vo forme jednoduchšej a častejšie používanej definície CTR₁ a zložitejšej definície CTR₂. Nasleduje definícia CTR₁:

$$\begin{aligned} CTR_1 &= 0 && \text{pre } a < 0 \\ CTR_1 &= w * a && \text{pre } a \geq 0. \end{aligned}$$

Funkcia CTR₂ je definovaná nasledovne:

$$CTR_2 = \begin{cases} 0; & a < 0 \\ \min(a, w) & a \geq 0 \\ \max(0, a + w - 1) & w \geq 0 \\ \min(a, -w) & a \geq 0 \\ \max(0, a - w - 1) & w \leq 0 \end{cases}$$

27. Čo je to Dempster-Shafferova metóda spracovania neurčitosti?

Táto teória vznikla pred objavením sa prvých Znalostných systémov mimo rámec umelej inteligencie. Bola prispôsobená na manipuláciu s neurčitou. Neurčitosť reprezentuje konfidenčným intervalom, teda dvoma numerickými hodnotami. Pri skúmaní šírenia neurčitosti inferenčnou sieťou sa zaoberá iba funkciou GLOB, ostatné kombinačné funkcie preberá z iných modelov, pričom ich prispôsobuje pre prácu s intervalovo zadanou neurčitou.

Metóda je založená na práci s Dempsterovými priestormi (Hájek, 1987). Jej základy môžeme ilustrovať nasledujúcim príkladom.

- ✓ Predpokladajme, že na stole máme rozloženú mapu sveta. Potom pravdepodobnosť toho, že poslepiacky trafím pevninu (oceán) závisí od plochy (počtu bodov) pevniny (oceánu).
- ✓ Ak stojím ďalej a triafam ju napríklad kriedou, potom nejde len o počet bodov, pretože každý bod má navyše vlastnú pravdepodobnosť zásahu. Potom pravdepodobnosť, že trafím pevninu je rovná súčtu pravdepodobností zásahu všetkých bodov pevniny.
- ✓ Potom zakryjem časť mapy knihou. Potom sú možné dva extrémne prípady:
 - 1) Pod knihou je iba oceán. Potom pravdepodobnosť P_1 , že trafím pevninu sa rovná súčtu pravdepodobností všetkých bodov pevniny, ktoré vidím.
 - 2) Pod knihou je iba pevnina. Potom pravdepodobnosť P_2 , že trafím pevninu sa rovná súčtu pravdepodobností všetkých bodov pevniny a pravdepodobností všetkých bodov pod knihou.

Keď to zhrnieme, pravdepodobnosť že trafím pevninu sa nachádza niekde medzi týmito extrémami, teda v **konfidenčnom intervale** $\langle P_1, P_2 \rangle$, ktorý vyjadruje vágnosť nášho poznania.

Následne môžeme definovať funkciu negácie:

$$P(\text{pevnina}) = \langle P_1, P_2 \rangle$$
$$P(\neg \text{pevnina}) = \langle 1 - P_2, 1 - P_1 \rangle.$$

V tejto definícii rozoznávame:

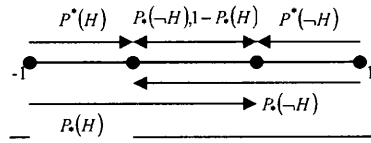
- ✓ P_1 alebo $1 - P_2$ ako **dolnú pravdepodobnosť** alebo zaručenú hodnotu či stupeň dôvery (degree of believe) $P_*(H) = P\{\text{ano}\}$
- ✓ P_2 alebo $1 - P_1$ ako **hornú pravdepodobnosť** alebo stupeň plausibility, možnosti, prípustnosti (degree of plausibility) $P^*(H)$.

V tomto ponímaní je konfidenčný interval prechod od zaručeného po možné. Platí:

$$P_*(H) + P^*(\neg H) = 1$$
$$P^*(H) = 1 - P_*(\neg H)$$
$$P_*(H) \leq P(H) \leq P^*(H) = 1 - P_*(\neg H).$$

Z toho vyplýva definícia konfidenčného intervalu: $\langle P_*(H), P^*(H) \rangle$.

Vzťahy medzi dolnou a hornou pravdepodobnosťou v konfidenčnom intervale, je možné ilustrovať nasledovným obrázkom.

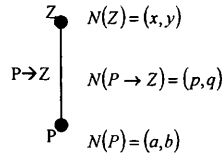


Šírka konfidenčného intervalu sa označuje ako vágnosť. Ak je vágnosť nulová, platí:

$$P(H) + P(-H) = 1. \text{ Ak je vágnosť nenulová platí: } P(H) + P(-H) < 1.$$

Čím väčšia je vágnosť, tým menej je isté a tým väčší je rozsah neurčitosti. Čím je vágnosť menšia, tým vyššia je miera istoty a tým menší je rozsah neurčitosti.

Teraz bude nasledovať krátke zhrnutie. Dempster-Shafferova metóda je reprezentovaná numericky, používa absolútne hodnoty, pracuje s dvoj hodnotovou logikou (tzv. Dempsterova dvojica). Spracováva neurčité v báze znalostí i báze dát, čo prezentuje nasledovný obrázok.



Kde $a = P_*(P)$ a $b = P^*(-P)$. Súvis Dempsterovej dvojice a konfidenčného intervalu vyjadruje nasledovný vzťah: $\langle a, 1 - b \rangle = \langle P_*(P), P^*(P) \rangle$.

28. Ako je definovaný Dempster-Shafferov priestor?

V ďalšom si uvedieme zjednodušenú podobu definície Dempster-Shafferovho priestoru, ktorá je používaná v znalostných systémoch (Hájek, 1987).

Majme množinu X argumentov, alebo svedectiev o výroku V . Nech T je množina pravdivostných hodnôt týchto svedectiev $T = \{ano, nie\}$ a U je podmnožina T , teda:

$U = \{ \} \quad U = \{ano\} \quad U = \{nie\} \quad U = \{ano, nie\}$. Jednotlivé svedectvá môžu podporovať platnosť / neplatnosť výroku V . Z toho vyplýva, že svedectvá môžu byť štyroch typov: zlučiteľné s ANO, zlučiteľné s NIE, zlučiteľné s ANO aj NIE a napokon nezlučiteľné ani s ANO ani s NIE:

$\Gamma(ano) = \{x_i\}$ - argumenty x_i hovoria, že výrok platí

$\Gamma(nie) = \{x_j\}$ - argumenty x_j hovoria, že výrok neplatí

$\Gamma(ano, nie) = \{x_k\}$ - argumenty x_k hovoria, že výrok platí alebo neplatí

$\Gamma\{ \} = \{x_l\}$ - argumenty x_l hovoria, že nie sú zlučiteľné ani s ANO ani s NIE.

Jednotlivé svedectvá nemusia byť absolútne platné/neplatné. Ich platnosť môžeme vyjadriť pravdepodobnostnou mierou ν . Dempsterov priestor je potom definovaný štvoricou $\langle X, T, \Gamma, \nu \rangle$. Pravdepodobnosť nejakej podmnožiny $U \subset T$ je možné určiť ako súčet pravdepodobností svedectiev z x zlučiteľných s U . Predpokladajme, že máme $P_1(ano), P_2(nie), P_3(ano, nie)$, ktoré spolu vytvárajú informačný zdroj, alebo úplný súbor svedectiev, ak platí: $\sum_{U \subset T} P_n = 1$. Potom môžeme vypočítať nasledovné pravdepodobnosti:

$$P\{V : ano\} = \sum_{ano} \nu$$

$$P\{V : nie\} = \sum_{nie} \nu$$

$$P\{V : ano, nie\} = \sum_{ano, nie} \nu$$

$$P\{V : \{ \}\} = \sum \nu_s$$

kde ν_s sú singulárne svedectvá (nie sú kompatibilné ani s *ano* ani s *nie*).

Tento Dempsterov priestor je potrebné normovať za účelom vylúčenia singulárnych svedectiev. Teda, z množiny X vyberieme všetky singulárne svedectvá

$$X' = X - S$$

$\nu' = \frac{\nu}{1 - \nu_s}$. Dôsledkom bude normovaný Dempsterov priestor:

$$\sum P = 1$$

$$\Gamma(0) = \{ \}$$

$$P\{V : ano\} = P_*(V)$$

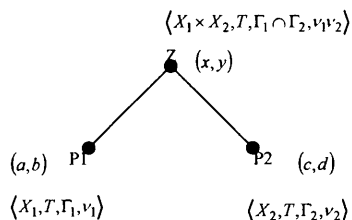
$$P\{V : nie\} = P_*(-V)$$

$$P\{V : ano, nie\} = 1 - P_*(V) - P_*(-V)$$

29. Ako je definovaná funkcia GLOB v Dempster-Shafferovom modeli spracovania neurčitosti?

Pre no

Majme dva predpoklady $P1$ (charakterizovaný dempsterovou dvojicou (a,b)) a $P2$ (charakterizovaný dempsterovou dvojicou (c,d)). Úlohou je získať dempsterovú dvojicu (x,y) pre záver Z . Vzťahy medzi predpokladmi a záverom sú ilustrované nasledovným obrázkom, na ktorom sú uvedené aj dempsterove priestory v hranatých zátvorkách.



Takto
Ak ne

Teraz budeme riešiť problém, ako získať dempsterov priestor odpovedajúci záveru. Vieme, že nová množina svedectiev X pre záver Z je tvorená karteziánskym súčinom $X_1 \times X_2$, pri ktorom môžu vzniknúť nové singulárne dvojice. Singulárne dvojice vzniknú vtedy, ak x_1 je zlučiteľné s $\{ano\}$ a zároveň x_2 je zlučiteľné s $\{nie\}$. Potom v súlade s tým, čo bolo uvedené v definícii dempsterovho priestoru, dempsterov priestor odpovedajúci záveru bude definovaný nasledovne:

$x = 1$

$y = 1$

Ak ne

$$\begin{aligned}
 P(Z : 0) &= ad + bc \\
 P(Z : ano) &= ac + a(1 - c - d) + (1 - a - b)c \\
 P(Z : nie) &= bd + b(1 - c - d) + (1 - a - b)d \\
 P(Z : ano, nie) &= (1 - a - b)(1 - c - d)
 \end{aligned}$$

$x = 1 -$

čo je

Demp

Kde $P(Z : 0)$ predstavuje pravdepodobnosť singulárnych bodov, v ktorých sú obidva predpoklady v rozpore. Prvý singulárny bod odpovedá situácii, keď prvý predpoklad určite platí a druhý určite neplatí, čo reprezentuje pravdepodobnosť ad . Druhý singulárny bod odpovedá situácii, keď prvý predpoklad určite neplatí a druhý určite platí, čo reprezentuje pravdepodobnosť bc . Pravdepodobnosť $P(Z:ano)$ sa skladá z troch možností: z prípadu, keď obidva predpoklady určite platia ac ; z prípadu, keď prvý predpoklad určite platí a druhý predpoklad možno platí $a(1-c-d)$ (pretože $(1-c-d)$ reprezentuje plausibilitu – možnosť platnosti aj neplatnosti druhého predpokladu); a z prípadu, keď prvý predpoklad možno platí a druhý predpoklad určite platí $(1-a-b)c$. Pravdepodobnosť $P(Z:nie)$ sa skladá z troch možností: z prípadu, keď obidva predpoklady určite neplatia bd ; z prípadu, keď prvý predpoklad určite neplatí a druhý predpoklad možno neplatí $b(1-c-d)$; a z prípadu, keď prvý predpoklad možno neplatí a druhý predpoklad určite neplatí $(1-a-b)d$. Pravdepodobnosť $P(Z:ano,nie)$ predstavuje prípad, keď obidva predpoklady možno platia $(1-a-b)(1-c-d)$.

Pre normovaný Dempsterov priestor bude platiť:

$$P(Z : ano) = \frac{ac + a(1-c-d) + (1-a-b)c}{1-(ad+bc)} = \frac{ac + a - ac - ad + c - ac - bc}{1-ad-bc} =$$

$$= \frac{1-ad-bc - (1-a-c+ac)}{1-ad-bc} = 1 - \frac{(1-a)(1-c)}{1-(ad+bc)} = x$$

$$P(Z : nie) = \frac{bd + b(1-c-d) + (1-a-b)d}{1-(ad+bc)} = \frac{bd + b - bc - bd + d - ad - bd}{1-ad-bc} =$$

$$= \frac{1-ad-bc - (1-b-d+bd)}{1-ad-bc} = 1 - \frac{(1-b)(1-d)}{1-(ad+bc)} = y$$

Takto sme dostali **Dempsterovo pravidlo** (Hájek, 1987) ako vypočítať dvojicu (x, y) .

Ak neuvažujeme neplatnosť, t.j. ak $b=0$ a $d=0$, potom dostaneme nezávislú pravdepodobnosť :

$$x = 1 - \frac{(1-a)(1-c)}{1} = a + c - ac$$

$$y = 1 - \frac{1}{1} = 0$$

, ktorá platí pre systémy MYCIN aj EXSYS.

Ak nebudeme uvažovať vágnosť, teda $b=1-a$ a $d=1-c$, dostaneme:

$$x = 1 - \frac{(1-a)(1-c)}{1-(a(1-c)-(1-a)c)} = 1 - \frac{1-a-c+ac}{1-a+ac-c+ac} = \frac{2ac-ac}{1-a-c+2ac} = \frac{ac}{1-a-c+2ac},$$

čo je grupová operácia násobenia v Algebraickej teórii, ktorá je teda všeobecnejšia ako

Dempster-Shaferova metóda: $GLOB(x, y) = \frac{xy}{1-x-y+2xy} \Rightarrow f = \frac{1}{1-x}; g = \frac{z}{1+z}.$

30. Ako sú definované ostatné funkcie v Dempster-Shafferovom modeli spracovania neurčitosti?

Ostatné kombinačné funkcie sú definované nasledovne:

$$\begin{aligned} NEG((a,b)) &= (b,a) \\ CONJ((a,b),(c,d)) &= \min((a,b),(c,d)) \\ DISJ((a,b),(c,d)) &= \max((a,b),(c,d)) \end{aligned}$$

Funkcia CTR je definovaná spôsobom: $N(Z) = \min(N(P), N(P \rightarrow Z))$, pričom $N(P)$ je neurčitosť predpokladu P reprezentovaná dempsterovou dvojicou (a,b) , $N(P \rightarrow Z)$ je neurčitosť pravidla reprezentovaná dempsterovou dvojicou (p,q) a výsledná neurčitosť záveru $N(Z)$ je reprezentovaná dempsterovou dvojicou (x,y) . Teraz musíme vyriešiť ďalší problém. Ako určíme, ktorá z dvoch dempsterových dvojíc je menšia, a ktorá je väčšia. Na to nám posluží takzvané centrum dempsterovej dvojice. Výpočet tohto centra predstavuje transformáciu neurčitosti dvoj hodnotovej (dempsterovej dvojice) na neurčitosť jedno hodnotovú (centrum dempsterovej dvojice). Neurčitosti reprezentované jednou hodnotou sa dajú ľahko porovnávať. Ako sa také centrum počíta? Postupujeme tak, že pomocou funkcie GLOB združíme dvojicu (a,b) , pre ktorú počítame centrum, s neutrálnou dvojicou $(0.5,0.5)$. Tak dostaneme novú dvojicu (x,y) :

$(x,y) = h(a,b) = GLOB((a,b),(0.5,0.5))$. Pre túto dvojicu platí:

$$\begin{aligned} x &= 1 - \frac{(1-a)(1-0.5)}{1 - (0.5a + 0.5b)} = 1 - \frac{0.5(1-a)}{0.5(2-a-b)} = \frac{2-a-b-1+a}{2-a-b} = \frac{1-b}{2-a-b} \\ y &= 1 - \frac{(1-b)(1-0.5)}{1 - (0.5a + 0.5b)} = 1 - \frac{0.5(1-b)}{0.5(2-a-b)} = \frac{2-a-b-1+b}{2-a-b} = \frac{1-a}{2-a-b} \end{aligned}$$

Dá sa dokázať, že dempsterova dvojica (x,y) nemá vágnosť, pretože platí:

$$x + y = \frac{1-b}{2-a-b} + \frac{1-a}{2-a-b} = \frac{2-a-b}{2-a-b} = 1$$

To znamená, že dvojici (x,y) odpovedá jednobodový interval:

$$\langle x, 1-y \rangle = \left\langle \frac{1-b}{2-a-b}, \frac{1-b}{2-a-b} \right\rangle, \text{ keďže } 1-y = 1 - \frac{1-a}{2-a-b} = \frac{2-a-b-1+a}{2-a-b} = \frac{1-b}{2-a-b}$$

Teda centrum dempsterovej dvojice (a,b) môžeme vypočítať nasledovne:

$$C = h(a,b) = \frac{1-b}{2-a-b}$$

Potom porovnanie dvoch dempsterových dvojíc (a,b) a (c,d) platí:

ak $h(a,b) < h(c,d)$ potom $(a,b) < (c,d)$

ak $h(a,b) = h(c,d)$ & $a < c$ potom $(a,b) < (c,d)$.

31.

slovo alebo (1965), množina príslušných množín

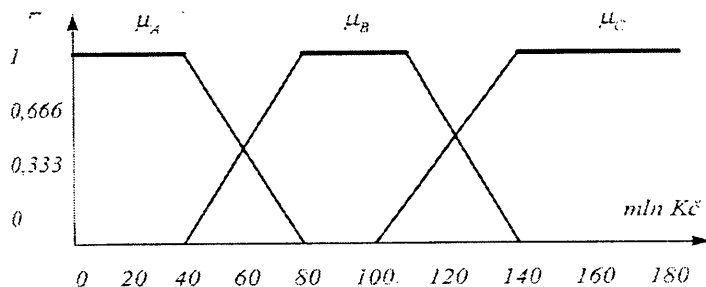
množina fuzzy

Na to funkcia $\mu_B(x)$ týchto množín $A = \{ \dots \}$ určite zisk. Podobne milión ešte n. Napo je veľ veľký. Nebu ešte postu [pod]

31. Na čom je založený fuzzy model spracovania neurčitosti?

Fuzzy model spracovania neurčitosti je založený na princípoch fuzzy logiky. Fuzzy je slovo anglického pôvodu, ktoré zdomácnelo. Dalo by sa preložiť slovom neurčitý, neistý alebo zahmlený. Základy fuzzy logiky boli položené pojmom „fuzzy množina“ (Zadeh, 1965), (Zadeh, 1973), ako pojmu protikladnému pojmu „ostrá množina“. Zatiaľ čo do ostrej množiny prvok buď patrí alebo nepatrí, do fuzzy množiny každý prvok patrí s určitou mierou príslušnosti k tejto množine. Z toho vyplýva definícia fuzzy množiny. Napríklad fuzzy množinu A môžeme matematicky zapísať takto:

$A = \{(x, \mu_A(x)) : \mu_A(x) \in L, \forall x \in U\}$. Teda, každý prvok x z univerza U patrí do množiny A s istou mierou neurčitosti danej jeho hodnotou príslušnosti $\mu_A(x)$. Príklad takejto fuzzy množiny je uvedený na nasledovnom obrázku.



Na tomto obrázku sú uvedené tri fuzzy množiny: A – reprezentuje malý zisk so svojou funkciou príslušnosti $\mu_A(x)$, B – reprezentuje stredný zisk so svojou funkciou príslušnosti $\mu_B(x)$ a C – reprezentuje veľký zisk so svojou funkciou príslušnosti $\mu_C(x)$. Funkcie príslušnosti týchto množín: μ_A , μ_B a μ_C sú reálne čísla z intervalu $\langle 0,1 \rangle$. Extenzionálne definície týchto množín vymenovaním prvkov sú nasledovné:

$A = \{(0,1), (20,1), (40,1), (60,0.5), (80,0)\}$, čo znamená, že 0, 20 alebo 40 miliónov korún je určite nízky zisk, že 60 miliónov korún je nízky zisk na 50%, a že 80 miliónov už nie je nízky zisk.

Podobne $B = \{(40,0), (60,0.5), (80,1), (100,1), (120,0.5), (140,0)\}$ znamená, že 80 a 100 miliónov je určite stredný zisk, že 60 a 120 miliónov je stredný zisk na 50%, a že 40 miliónov ešte nie je stredný zisk a 140 miliónov už nie je stredný zisk.

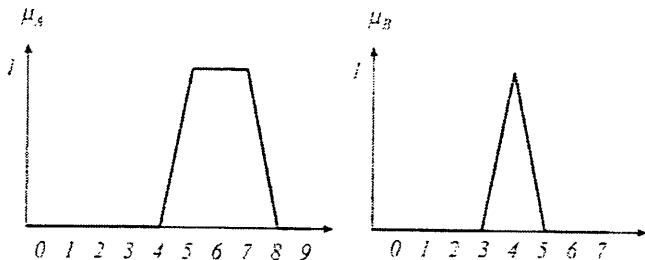
Napokon pre $C = \{(100,0), (120,0.5), (140,1), (160,1), (180,1)\}$ platí, že 100 miliónov ešte nie je veľký zisk, že 120 miliónov je veľký zisk na 50% ale 140, 160 a 180 miliónov je už určite veľký zisk.

Nebudeme tu uvádzať všetky definície známe z teórie fuzzy množín, ale jednu definíciu si ešte uvedieme. **Lingvistická premenná** je premenná, ktorej hodnoty sú slová, resp. postupnosť alfa-numerických znakov. Napríklad:

[podiel_na_trhu] = nízky, stredný, vysoký, monopolný.

32. Ako je možné s fuzzy množinami operovať?

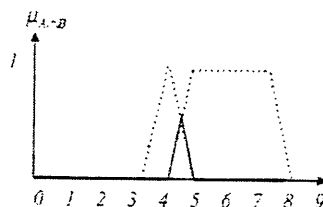
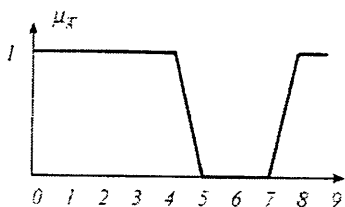
Nad fuzzy množinami je možné definovať operácie negácie, konjunkcie a disjunkcie, ktoré tvoria základ fuzzy logiky. Majme dané dve fuzzy množiny A a B , ktorých funkcie príslušnosti sú ilustrované nasledovným obrázkom.



Potom funkcia negácie množiny A funkcia konjunkcie, resp. prieniku množín A a B , t.j. $A \cap B$ sa realizuje podľa definície a graficky tak, ako je to znázornené nižšie.

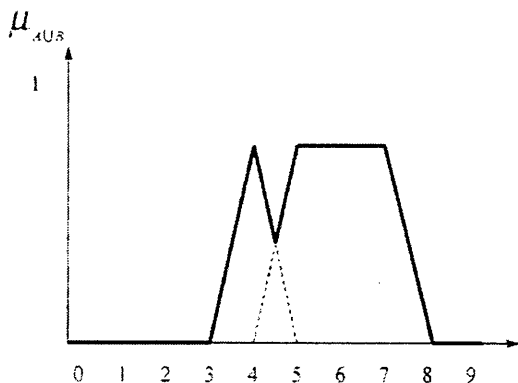
$$\mu_{\bar{A}} = 1 - \mu_A$$

$$\mu_{A \cap B} = \min\{\mu_A(x), \mu_B(x)\}$$



Nasleduje definícia a grafické znázornenie funkcie disjunkcie, resp. zjednotenia množín A a B , t.j. $A \cup B$

$$\mu_{A \cup B} = \max\{\mu_A(x), \mu_B(x)\}$$



majú v
a z fuzzy
Podrobn
jednot

odpov

pravid
prísluš
prísluš
vyjadr

funkci
pravid

výsled

33. Ako sa realizuje neurčité rozhodovanie vo fuzzy znalostných systémoch?

Na úvod by bolo vhodné spomenúť, že fuzzy znalostné systémy a fuzzy regulátory majú v praxi dobré výsledky. Fuzzy znalostný systém sa skladá zo súboru funkcií príslušnosti a z fuzzy pravidiel, ktoré sa používajú na fuzzy inferenciu, resp. na fuzzy rozhodovanie. Podrobnejšie o fuzzy rozhodovaní sa pojednáva v práci (Kelemen a kol., 1999). Inferencia je jednou zo štyroch základných operácií: fuzzifikácie, inferencie, agregácie a defuzzifikácie.

- 1) Fuzzifikácia je proces, v ktorom sa vstupné dáta s neurčitou transformujú na fuzzy množiny, charakterizované svojimi funkciami príslušnosti.
- 2) Inferencia sa dá prirovnať ku sekvenčnej kombinácii – funkcii CTR. V rámci inferencie sa ku fuzzifikovaným faktom hľadajú predpoklady fuzzy pravidiel, ktoré im najviac odpovedajú. Na základe tejto zhody sa odvádzajú dôsledky pravidiel v podobe fuzzy množín. Výsledkom inferencie je niekoľko fuzzy množín.
- 3) Agregácia sa dá prirovnať k paralelnej kombinácii – funkcii GLOB. Výsledkom agregácie je jediná fuzzy množina, ktorá vzniká zložením fuzzy množín dôsledkov použitých pravidiel.
- 4) Defuzzifikácia je pokus reprezentovať funkciu príslušnosti fuzzy množiny (teda zložitú krivku, resp. plochu) pomocou jedinej ostrej hodnoty (jedného reálneho čísla).

Napríklad, majme všeobecné pravidlo: *IF predpoklad A_i THEN dôsledok B_i* a jemu odpovedajúce dve fuzzy pravidlá:

$$\mu_{A1} \rightarrow \mu_{B1}$$

$$\mu_{A2} \rightarrow \mu_{B2}$$

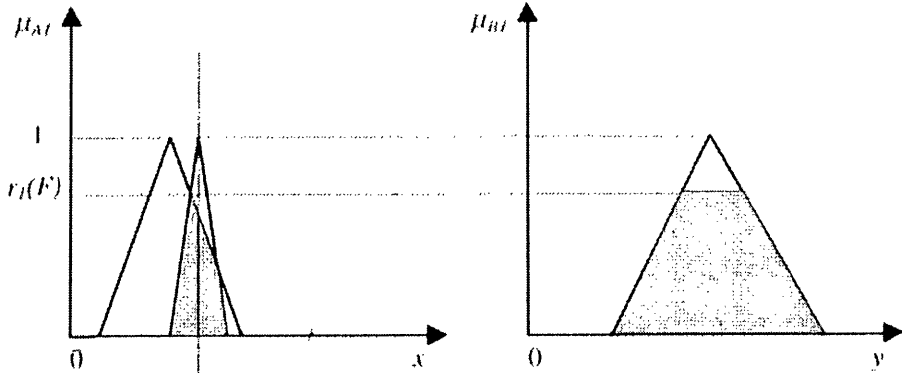
Ďalej majme fuzzy fakt μ_F .

Potom inferencia prebieha nasledovne. Miera zhody medzi faktom F a predpokladom pravidla μ_{A1} alebo μ_{A2} sa vyhodnotí ako prienik funkcie príslušnosti faktoru a funkcie príslušnosti predpokladu pravidla. Neurčitosť záveru je potom vyjadrená ako tá časť funkcie príslušnosti dôsledku pravidla, ktorá nepresahuje maximum spomínaného prieniku. To je vyjadrené v nasledujúcom obrázku.

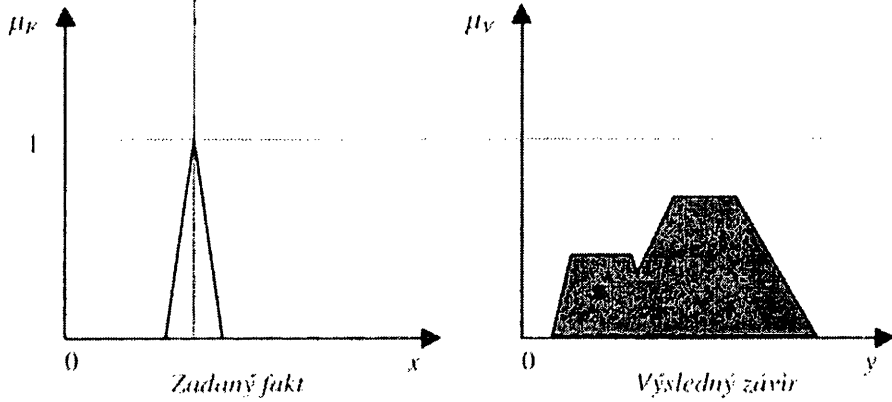
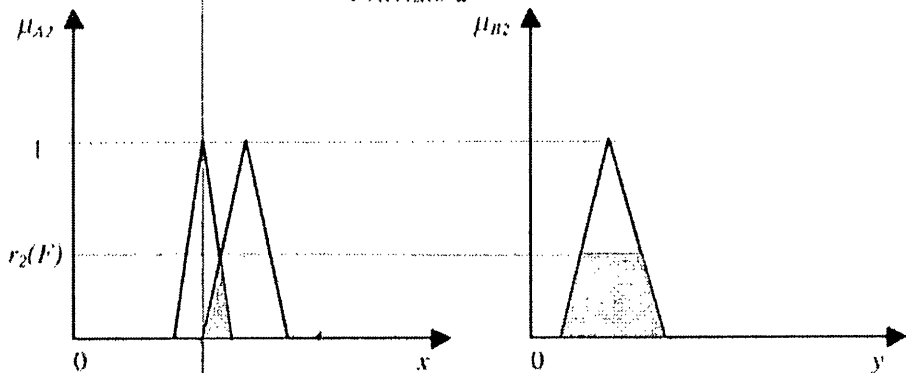
Dolná časť toho istého obrázku ilustruje agregáciu tj. proces, v ktorom je tvorená funkcia príslušnosti záveru a to zjednotením všetkých dielčích funkcií príslušnosti dôsledkov pravidiel, ktoré boli aktivované (v našom prípade dva).

Napokon defuzzifikácia by mohla predstavovať nájdenie ťažiska celkovej plochy výsledného záveru. Ťažisko ako jediná ostrá hodnota by vyjadrovala neurčitosť záveru.

Pravidlo 1



Pravidlo 2

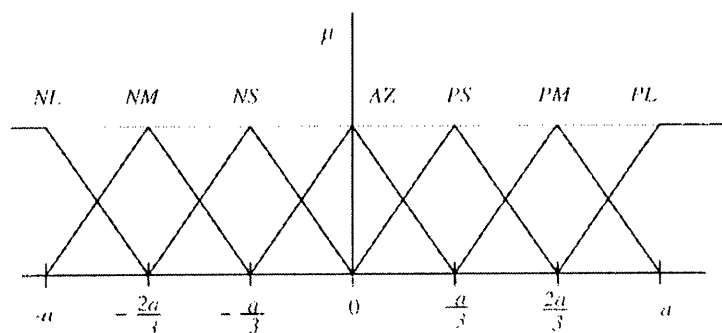


fuzzy pomocí náklad tvaru prirad univer je nor

Na to NL NM NS AZ PS PM PL Sprav Avšak vychá nejed typu f

34. Ako môže prebiehať fuzzifikácia v znalostných systémoch?

Prvá vec a základný problém fuzzifikácie, ktorý musíme zvládnuť je špecifikácia fuzzy množiny určením jej funkcie príslušnosti. Touto špecifikáciou riešime, ako vyjadriť pomocou funkcie príslušnosti také termíny, ako napríklad: nadpriemerný zisk alebo vysoké náklady atď. Prečo? Lebo potrebujeme transformovať lingvistický term do normalizovaného tvaru čísla z intervalu $\langle 0,1 \rangle$ alebo čísla z intervalu $\langle -1,1 \rangle$. Teda jednej ostrej hodnote priradíme stupeň príslušnosti k jednej alebo k viacerým fuzzy množinám. Celé normalizované univerzum pokryjeme nosičmi fuzzy množín, ako je to uvedené na nasledovnom obrázku, kde je normalizácia uskutočnená na intervale $\langle -a,a \rangle$.

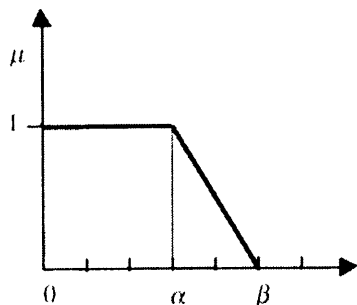


Na tomto obrázku je definovaných sedem nosičov fuzzy množín podľa (Klir-Yuan,1995):

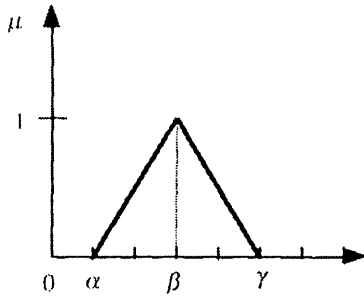
- NL veľká záporná hodnota (Large Negative)
- NM stredná záporná hodnota (Medium Negative)
- NS malá záporná hodnota (Small Negative)
- AZ približne nulová hodnota (Approximately Zero)
- PS malá kladná hodnota (Small Positive)
- PM stredná kladná hodnota (Medium Positive)
- PL veľká kladná hodnota (Large Positive)

Spravidla volíme nepárny počet nosičov (3,5,7,9), málokedy viac ako 9.

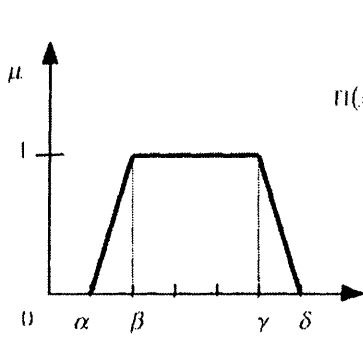
Avšak potrebujeme mať k dispozícii aj iné tvary funkcií príslušnosti. Pri ich návrhu sa vychádza z expertných odhadov. Pritom sa berie na zreteľ, že funkcia príslušnosti má byť čo najjednoduchšia, t.j. má mať čo najmenší počet lineárnych úsekov. Štyri nasledovné vzorové typy fuzzy funkcií boli navrhnuté v publikácii (Driakov at all, 1993).



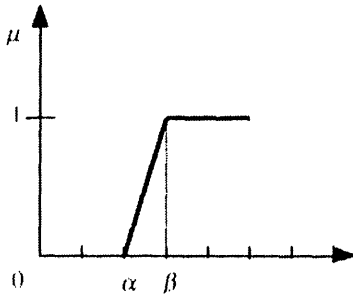
$$l(x, \alpha, \beta) = \begin{cases} 0 & x < \alpha \\ (\alpha - x) / (\beta - \alpha) & \alpha \leq x \leq \beta \\ 1 & x > \beta \end{cases}$$



$$\Lambda(x, \alpha, \beta, \gamma) = \begin{cases} 0 & x < \alpha \\ (x - \alpha) / (\beta - \alpha) & \alpha \leq x \leq \beta \\ (\alpha - x) / (\gamma - \beta) & \beta \leq x \leq \gamma \\ 0 & x > \gamma \end{cases}$$



$$\Pi(x, \alpha, \beta, \gamma, \delta) = \begin{cases} 0 & x < \alpha \\ (x - \alpha) / (\beta - \alpha) & \alpha \leq x \leq \beta \\ 1 & \beta \leq x \leq \gamma \\ (\alpha - x) / (\gamma - \beta) & \gamma < x \leq \delta \\ 0 & x > \delta \end{cases}$$



$$\Gamma(x, \alpha, \beta) = \begin{cases} 0 & x < \alpha \\ (x - \alpha) / (\beta - \alpha) & \alpha \leq x \leq \beta \\ 1 & x > \beta \end{cases}$$

podpri
premer

Kde μ
zisk. U

(defen
<od>

((<naz

(<naz

Apliku
definíc

(defen

0 200

((nizky

(podp

(prien

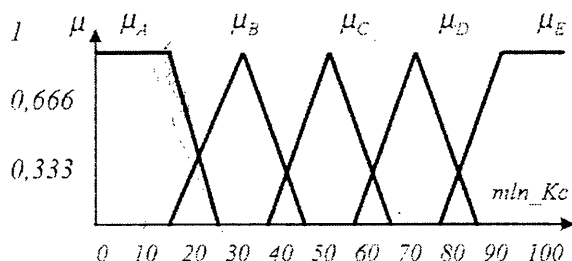
(nadv

(vyso

)).

35. Ako môžeme reprezentovať jazykové premenné vo FuzzyClipse?

Uvažujme jazykovú premennú *ZISK*, ktorá môže nadobúdať hodnoty: nízky, podpriemerný, priemerný, nadpriemerný a vysoký. Funkcie príslušnosti tejto jazykovej premennej sú znázornené na nasledujúcom obrázku.



Kde μ_A je nízky zisk, μ_B je podpriemerný, μ_C je priemerný, μ_D je nadpriemerný a μ_E je vysoký zisk. Uvažujme ďalej definíciu fuzzy šablóny vo FuzzyClipse:

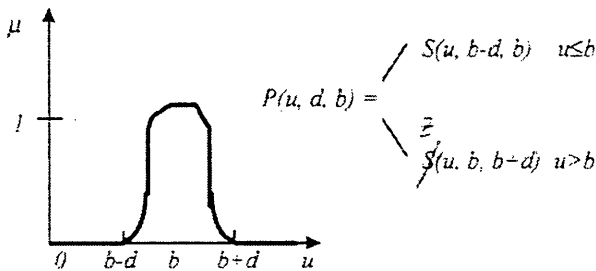
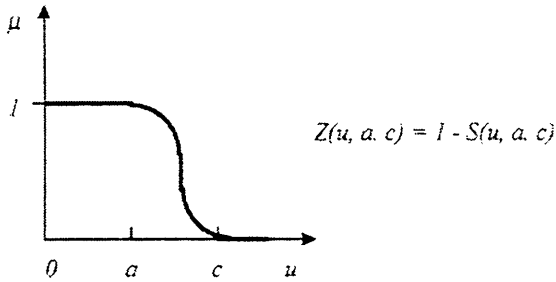
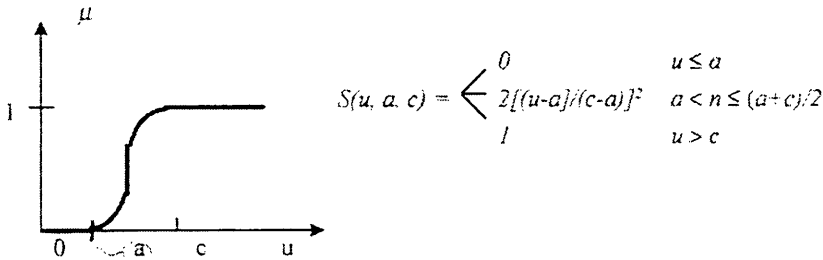
```
(deftemplate zisk [komentar]
  <od> <do> [<jednotka>]
  ((<nazov_hodnoty_t1> <popis_fuzzy_mnoziny>)
   (<nazov_hodnoty_tn> <popis_fuzzy_mnoziny>))).
```

Aplikujúc túto šablónu na vyššie uvedenú jazykovú premennú *ZISK* dostávame nasledovnú definíciu:

```
(deftemplate zisk ; definicia jazykovej premennej
  0 200 mln_Sk
  ((nizky (0,1) (10,1) (15,0.666) (20,0.333) (25,0))
   (podpriemerny (15,0) (20,0.333) (25,0.666) (30,1) (35,0.666) (40,0.333) (45,0))
   (priemerny (35,0) (40,0.333) (45,0.666) (50,1) (55,0.666) (60,0.333) (65,0))
   (nadpriemerny (55,0) (60,0.333) (65,0.666) (70,1) (75,0.666) (80,0.333) (85,0))
   (vysoky (75,0) (80,0.333) (85,0.666) (90,1) (100,1))
  )).
```


36. Ako môžeme reprezentovať jazykové premenné štandardnými funkciami?

V odpovedi na otázku č. 34 uvedené štandardné fuzzy funkcie *L*, *PI* a *Gama* boli pre FuzzyClips modifikované na spojité funkcie *Z*, *P* a *S*, ktoré sú znázornené na nasledujúcom obrázku.



Jednotlivé spojité štandardné fuzzy funkcie sa vo FuzzyClipse zadávajú podľa nasledovnej šablony:
 funkcia *S* ako (*S* *a* *c*)
 funkcia *Z* ako (*Z* *a* *c*)
 funkcia *PI* ako (*PI* *d* *b*).

Definícia jazykovej premennej bude potom oveľa jednoduchšia:

```
(deftemplate zisk
  0 200 mln_Sk
  ((nizky (Z 10 25))
   (podpriemerny (PI 15 30))
   (priemerny (PI 15 50))
   (nadpriemerny (PI 15 70))
   (vysoky (S 75 90))
  )).
```

37. Ktoré sú základné techniky neurčitého rozhodovania?

Neurčité rozhodovanie v zásade delíme na rozhodovanie pomocou jednoduchých pravidiel a rozhodovanie pomocou zložených pravidiel. K jednoduchým pravidlám patria:

- ✓ jednoduché pravidlo GRISP
- ✓ jednoduché pravidlo FUZZY_GRISP
- ✓ jednoduché pravidlo FUZZY_FUZZY.

Zložené pravidlá delíme na:

- ✓ zložené pravidlá s viacnásobným dôsledkom
- ✓ zložené pravidlá s viacnásobným predpokladom
- ✓ zložené pravidlá s viacnásobne vloženým faktorom.

JEDNODUCHÉ PRAVIDLO GRISP

Ak je požadované použitie neurčitosti vo fakte alebo pravidle, FuzzyClips pripojí k danej entite činiteľ istoty CF t.j. číslo z intervalu $\langle 0,1 \rangle$. Uvažujme jednoduché pravidlo GRIPS, ktoré neobsahuje fuzzy entitu:

IF F_1 THEN F_2 s činiteľom istoty CF_R .

Aktivujme toto pravidlo zadaním faktu F_1 s činiteľom istoty CF_{F1} . Do bázy dát sa následne vloží fakt F_2 s činiteľom istoty CF_{F2} , pričom platí: $CF_{F2} = CF_R * CF_{F1}$.

Uvažujme nasledovný príklad vo FuzzyClipse.

Majme dané pravidlo:

```
(defrule      jednoduche_pravidlo_grisp
  (declare    (CF 0.9)
  (vynosy prevysuju_naklady)
  =>
  (assert (zisk nenulovy)))
```

Nech k danému pravidlu pristúpi fakt:

```
(vynosy prevysuju_naklady) CF 0.8
```

Po aplikácii pravidla nad daným faktorom sa do bázy dát vloží fakt:

```
(zisk nenulovy) CF 0.72.
```

38. Ako pracuje jednoduché pravidlo FUZZY_GRISP?

Pravidlo FUZZY_GRISP je pravidlo, u ktorého iba predpokladová časť obsahuje fuzzy entitu. Z toho vyplýva, že aj fakt ktorý je stotožňovaný s predpokladovou časťou pravidla, musí mať fuzzy charakter. Navyiac fakt ako aj predpoklad pravidla musia byť definované rovnakou jazykovou premennou. V hodnotách tejto premennej sa musia aspoň čiastočne prekryvať. Tieto podmienky, napríklad, nespĺňa dvojica: (*riziko vysoke*) a (*naklady nizke*). Ale spĺňa ich dvojica: (*zisk vysoky*) a (*zisk nadpriemerny*).

Uvažujme pravidlo: IF F_1 THEN F_2 s činiteľom istoty CF_R .

Aktivujme toto pravidlo zadaním faktu F_1' s činiteľom istoty CF_{F_1} . Do bázy dát sa následne vloží fakt F_2 s činiteľom istoty CF_{F_2} , pričom platí: $CF_{F_2} = CF_R * CF_{F_1} * S$, kde S je miera zhody predpokladu F_1 s s faktorom F_1' . Mieru zhody vypočítame nasledovne:

Ak $N(F_1/F_1') > 0.5$ potom $S = P(F_1/F_1')$.

Ak $N(F_1/F_1') \leq 0.5$ potom $S = P(F_1/F_1')[N(F_1/F_1') + 0.5]$.

Kde: $N(F_1/F_1') = 1 - P(negF_1/F_1')$

a $P(negF_1/F_1') = \max(\min(neg\mu_{F_1}, \mu_{F_1'}))$.

Uvažujme nasledovný príklad.

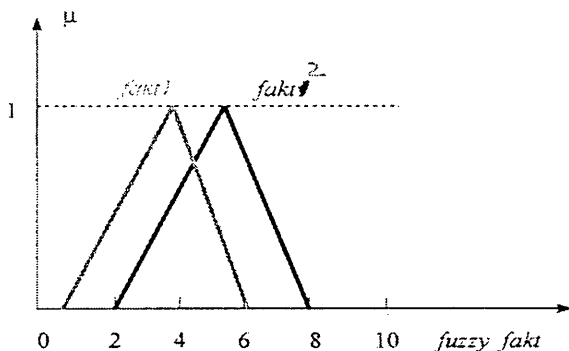
Majme dané pravidlo:

```
(defrule      jednoduche_pravidlo_fuzzy_grisp
  (declare    (CF 0.7)
  (fuzzy_fakt fakt2)
  =>
  (assert (ostry_fakt fakt3)))
```

Nech k danému pravidlu pristúpi fakt:

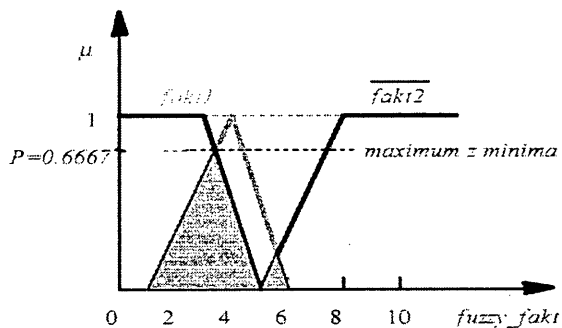
```
(fuzzy_fakt fakt1) CF 0.8
```

Fuzzy predpoklad pravidla *fakt2* ako aj s nim porovnávaný *fakt1*, sú znázornené na nasledovnom obrázku.



Z ďalšieho obrázku vyplýva: $P(negF_1/F_1') = \max(\min(neg\mu_{F_1}, \mu_{F_1'}))$

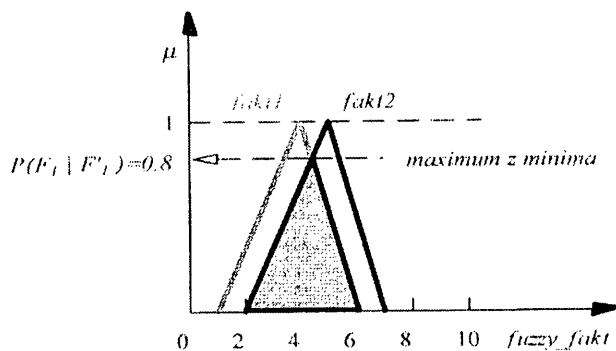
Teda pre náš príklad: $P(neg_fakt2/fakt1) = \max(\min(neg\mu_{neg_fakt2}, \mu_{fakt1})) = 0.6667$.



Potom: $N(F1/F1') = 1 - P(\text{neg}F1/F1') = 1 - 0.6667 = 0.333$.

Keďže $N(F1/F1') < 0.5$, potom: $S = P(F1/F1') [N(F1/F1') + 0.5] = (0.333 + 0.5) * 0.8 = 0.666$.

To, že: $P(F1/F1') = 0.8$ vyplýva z nasledujúceho obrázku.



$$S = (0.3333 + 0.5) * 0.8 = 0.6667$$

Činiteľ istoty: $CF_{F2} = CF_{fakt3} = CF_R * CF_{F1} * S = CF_R * CF_{fakt2} * S = 0.7 * 0.8 * 0.666 = 0.3733$.

Po aplikácii pravidla nad daným faktorom fakt1 sa do bazy dát vloží fakt:

(ostry_fakt fakt3) CF 0.3733.

39. Ako pracuje jednoduché pravidlo FUZZY_FUZZY?

40

Pravidlo FUZZY_FUZZY je pravidlo, ktoré pracuje s fuzzy entitami tak na strane predpokladov, ako aj na strane dôsledkov. Na strane predpokladov sa s fuzzy faktami pracuje tak, ako v predchádzajúcom prípade. Na strane dôsledkov platí $R = F1 * F2$,

kde: $F1$ je fuzzy množina predpokladu $F1$

a $F2$ je fuzzy množina dôsledku $F2$.

Platí: $\mu R(u,v) = \min(\mu F1(u), \mu F2(v))$ pre všetky hodnoty u a v univerza $U \times V$.

Výsledný činiteľ istoty sa následne vypočíta podľa vzťahu: $CF_{F2} = CF_r * CF_{F1}$.

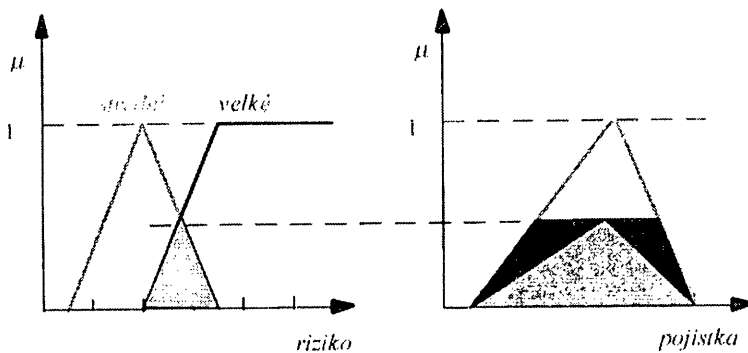
Majme daný príklad:

Pravidlo: (defrule *jednoduche_pravidlo_fuzzy_fuzzy*
(riziko veľké)
=>
(poistka vysoka))

Nech k danému pravidlu pristúpi fakt:

(riziko stredne)

Do databázy bude vložený fuzzy fakt *poistka*, ktorého funkcia príslušnosti je znázornená na nasledujúcom obrázku.



40. Ako sa spracovávajú zložené fuzzy pravidlá?

Doteraz sme uvažovali práve jednu entitu tak v predpokladovej časti, ako aj v dôsledkovej časti pravidla. Teraz budeme uvažovať ľubovoľný počet entít. Podľa toho, kde sa viacnásobná entita nachádza, rozlišujeme:

- ✓ **viacnásobný dôsledok.** Pravidlo s viacnásobným dôsledkom môže mať formu: *IF predpoklad THEN C_1, \dots, C_N* . Toto pravidlo sa vyhodnotí tak, že sa vykoná N jednoduchých pravidiel:

IF predpoklad THEN C_1

atď.

IF predpoklad THEN C_N .

- ✓ **viacnásobný predpoklad.** Pravidlo s viacnásobným predpokladom môže mať formu: *IF F_1 & F_2 THEN C (CF_r)*. V zátvorke je uvedený činiteľ istoty pravidla. Predpokladajme, že v báze dát sa nachádzajú tieto fakty s ich činiteľmi istoty:

F_1' (CF_{F_1})

F_2' (CF_{F_2}).

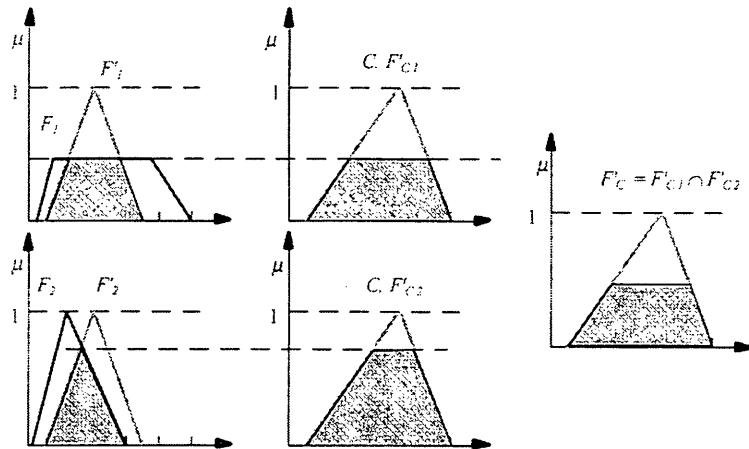
Po vykonaní pravidla pribudne v BD fakt C' s činiteľom istoty CF_C .

Postupnosť výpočtu CF_C je nasledovná.

Fuzzy hodnota faktu C' je daná formulou $F_C' = F_{C_1}' \cap F_{C_2}'$, kde:

F_{C_1}' je výsledok fuzzy inferencie pravidla $F_1 \rightarrow C$ pre fakt F_1' a

F_{C_2}' je výsledok fuzzy inferencie pravidla $F_2 \rightarrow C$ pre fakt F_2' . Zjednotená funkcia príslušnosti záveru od oboch predpokladov je ilustrovaná nasledovným obrázkom.



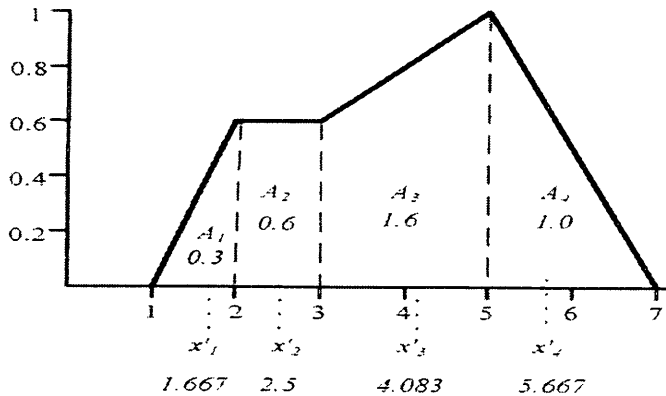
Napokon

$$CF_C = \min(CF_{F_1}', CF_{F_2}') * CF_r.$$

41. Ako sa uskutočňuje defuzzifikácia?

Defuzzifikácia predstavuje transformáciu výsledku rozhodovania v tvare fuzzy množiny na konkrétnu číselnú hodnotu. Dve najznámejšie metódy defuzzifikácie sú:

- ✓ **metóda stredú maxima** (Middle of the Maximum - MOM). Pri tejto metóde hľadáme strednú hodnotu všetkých lokálnych maxim funkcie príslušnosti výslednej fuzzy množiny. V najjednoduchšom prípade, ak je známe iba jedno globálne maximum, je výsledkom hodnota na osi x , odpovedajúca globálnemu maximu funkcie príslušnosti. V nasledovnom obrázku je $\text{StredMaxima} = 5$.



Ak je lokálne maximum tvorené úsečkou s konštantnou hodnotou y , potom výsledok je stred tejto úsečky a jemu odpovedajúca hodnota na osi x .

- ✓ **metóda ťažiska plochy** (Center of Balance - COB). Ťažisko plochy je bod, v ktorom by sme vedeli plochu vybalansovať na ukazováku ruky. Ako defuzzifikovaná hodnota výslednej fuzzy množiny sa použije súradnica x ťažiska. Konkrétne sa počíta ťažisko geometrického útvaru, ohraničeného osou x a krivkou funkcie príslušnosti fuzzy záveru. Pre vyššie uvedený obrázok sa ťažisko vypočíta nasledovným spôsobom:

$$x' = (1.667 \cdot 0.3 + 2.5 \cdot 0.6 + 4.083 \cdot 1.6 + 5.667 \cdot 1.0) / (0.3 + 0.6 + 1.6 + 1.0) = 4.0568,$$

kde: $A_1 = 0.3$, $A_2 = 0.6$, $A_3 = 1.6$ a $A_4 = 1.0$ sú obsahy dielčích plôch a $x_1 = 1.667$, $x_2 = 2.5$, $x_3 = 4.083$ a $x_4 = 5.667$ sú súradnice ťažísk dielčích plôch.

42. Čo sú to metaznalosti a ako ich delíme?

Metaznalosti sú znalosti o znalostiach, ktorých použitím sa stáva práca so znalostným systémom efektívnejšou. Nie sú to znalosti o riešenej úlohe. Sú lokalizované mimo inferenčný mechanizmus v samostatnom bloku. Výnimkou sú metaznalosti o neurčitosti, ktoré sú zabudované priamo v inferenčnom mechanizme. Sú reprezentované takzvanými metapravidlami. Uvedieme príklad takýchto metapravidiel:

MP1: Pravidlá zabezpečujúce bezpečnosť ľudí majú prednosť pred ostatnými.

- Dáva strategický návod pre výber pravidiel.
- Zužuje priestor prehľadávania, čím zlepšuje prácu systému.

MP2: Pravidlá týkajúce sa rozpustnosti a pH sa budú meniť zriedkavo.

- Slúži pri tvorbe a ladení bázy znalostí.

MP3: Pravidlo R6 pracuje preto, lebo vápno s kyselinou sírovou vytvára usadeninu.

- Objasňuje podmienky použitia niektorého pravidla.
- Môže slúžiť používateľovi na vysvetlenie záveru znalostného systému.
- Môže slúžiť systému, napr. došlo vápno a systém chce vedieť, či ho môže nahradiť lúhom.

Metaznalosti môžu mať charakter:

- ✓ znalostí o úlohe, ktoré sa získavajú od znalostného experta, a
- ✓ znalostí o systéme, ktoré sa získavajú od experta - tvorca prostriedkov.

V rannom štádiu návrhu znalostný inžinier zhromažďuje znalosti o úlohe a znalosti o vytváranom programe. V pokročilom štádiu systém rastie a preberá na seba úlohy chápania svojej činnosti, dokumentovania a zdôvodňovania svojej činnosti, a prípadne aj úlohu vlastnej modifikácie.

Reprezentácia metaznalostí je zhodná s reprezentáciou znalostí v báze znalostí. Ak je príslušná znalosť v BZ, po odvodení sa môže sprístupniť používateľovi vo forme výsledku. Ak je znalosť z bloku Metaznalosti, jej záver ovplyvní výber produkčného pravidla v inferenčnom mechanizme, ale navonok, smerom k používateľovi, sa neprejaví.

Podľa spôsobu použitia rozoznávame niekoľko druhov metaznalostí:

- 1) Metaznalosti pre výber pravidiel
- 2) Podporné znalosti o znalostiach
- 3) Metaznalosti definujúce zmysel pravidiel
- 4) Detekcia chýb v pravidlách
- 5) Zdôvodňovanie architektúry systému
- 6) Adaptácia znalostí na novú situáciu
- 7) Modelovanie možností znalostného systému.

43. Čo sú to strategické metaznalosti?

Strategické metaznalosti sú vlastne metaznalosti pre výber pravidiel. Nasleduje príklad takýchto metaznalostí. Majme produkčné pravidlá:

- PP1: Ak bol únik kyseliny sírovej
Potom použi látku x
- PP2: Ak bol únik kyseliny sírovej
Potom použi látku y

Aby sme vedeli ktoré z pravidiel PP1 a PP2 použiť, musíme definovať metaprávidlá pre výber pravidiel, resp. strategické metaprávidlá:

- MP3: Ak je viac alternatív
Potom použi lacnejšiu alternatívu
- MP4: Ak je viac alternatív
Potom použi menej riskantnú alternatívu
- MP5: Ak je viac alternatív
Potom použi alternatívu od väčšej autority

Avšak, produkčný systém stále nevie s týmito pravidlami pracovať, kým mu nedodáme nasledovné informácie

PP1	drahšia	menej riziková	menšia autorita	X
PP2	lacnejšia	viac riziková	väčšia autorita	Y

- Stratégia metaprávidla MP3 vyberie PP2.
- Stratégia metaprávidla MP4 nás zavedie na PP1.
- Stratégia metaprávidla MP5 odpáli PP2.

Či sa vykoná najprv PP1 alebo PP2 závisí od toho, ktoré metaprávidlo sa použije. Teda musíme vyriešiť problém, ktorú metaznalosť uprednostniť. Na to potrebujeme metaprávidlá o metaprávidlách, t.j. metaprávidlá druhého stupňa. Metaprávidlá druhého stupňa sú znalosti o metaprávidlách prvého stupňa. Metaprávidlá prvého stupňa sú znalosti o metaprávidlách nultého stupňa t.j. produkčných pravidiel. Metaprávidlá druhého stupňa sú aplikovateľné samé na seba. Uvedieme príklad.

- ✓ Predpokladajme, že MP3 sformuloval ekonóm závodu, lebo uprednostňuje lacnejšiu alternatívu. Ďalej predpokladajme, že MP4 sformuloval špecialista a ten uprednostňuje menšie riziko. Ak na tieto metaprávidlá aplikujeme MP5, ktoré uprednostňuje väčšiu autoritu, MP5 vyberie MP4 a to odpáli PP1.
- ✓ Predpokladajme, že MP3 vedie často ku katastrofám, lebo uprednostňuje lacnejšie materiály, zatiaľ čo MP5 je menej riskantné. Ak na tieto metaprávidlá aplikujeme metaprávidlo MP4, ktoré uprednostňuje menšie riziko, MP4 vyberie MP5 a to odpáli PP2.

Strategické metaprávidlá môžu mať aj všeobecnejší charakter, napríklad:

- Ak chcete pravidlá, ktoré vyriešia y
Potom hľadajte y
- Inak hľadajte všeobecnejšie pravidlá obsahujúce y.

44. Na čo sa používajú podporné znalosti o znalostiach?

Na rozdiel od strategických metaznalostí, podporné metaznalosti nehrajú aktívnu úlohu v rozhodovaní o výbere pravidiel. Majú skôr pomocný charakter. Príkladom podporných metaznalostí sú dodatočné informácie o miere rizika, o autorite pôvodcu a o cene chemickej komponenty (môže byť zadaná priamo alebo spôsobom výpočtu), ktoré boli uvedené v príklade strategických metaznalostí. Ak by nebola zabezpečená dostupnosť takýchto informácií nebolo by možné dané strategické metaznalosti vôbec použiť. Na druhej strane, úloha by mohla byť riešiteľná aj bez metapravidiel, aj keď ťažkopádnejšie.

Podporné znalosti môžu mať aj formu automatických štatistických pozorovaní o pravidlách, ktoré sa v niektorých znalostných systémoch realizujú. Sú to informácie druhu:

- koľkokrát bolo pravidlo použité
- koľkokrát bolo pravidlo použité s úspechom a koľkokrát s neúspechom
- aký je priemerný čas vykonania pravidla
- počet prípadov, kedy systém prerušuje svoju prácu a obracia sa s otázkou na používateľa.

Vznikajú tak empirické dáta o využití pravidiel, resp. zdrojov. Tieto údaje vo forme metaznalostí môžu byť využité pri návrhu znalostného systému.

Podobným typom metaznalostí sú znalosti ktoré určujú, nakoľko je presný výsledok pri použití danej metódy.

Zápisy o tom, čo sa v systéme dialo, o cene, o stupni rizika predstavujú teda nový typ metaznalostí t.j. podporných metaznalostí, ktoré majú opisný, objasňujúci nie aktívny charakter. Teda typ metaznalostí odlišných od strategických, ktoré vytvárajú efektívnu a vykonateľnú radu systému, používanú pre riadenie procesu riešenia.

45. Čo sú to metaznalosti definujúce zmysel pravidiel?

Metaznalosti definujúce zmysel pravidiel predstavujú dodatočnú informáciu, ktorá objasňuje zmysel pravidla, reprezentujúceho znalosť. Napríklad majme produkčné pravidlo PP, ktoré hovorí, že keď došlo k znečisteniu látkou X je potrebné použiť na jej neutralizáciu látku A. Ideme do skladu, a tam látka A nie je, ale je tam látka C. Otázka znie: môžeme použiť látku C? Nevieme sa rozhodnúť, lebo v BZ nie sú o látke C žiadne informácie. Navyiac nepoznáme väzbu medzi X a A. Keby sme vedeli ako spolu reagujú, mohli by sme zistiť, či aj C reaguje s X tak isto. Ak reaguje tak isto, môže zlúčenina klesnúť na dno, resp. neklesne, ale sa pozbera z hladiny. Nasleduje príklad troch pravidiel, ktorých súčasťou je aj definovaný ich zmysel.

PP6: Ak bol únik kyseliny sírovej

Potom použi vápno

Zmysel: Vápno zneutralizuje túto kyselinu a vytvorí nerozpustnú zrazeninu

PP7: Ak bol únik kyseliny octovej

Potom použi vápno

Zmysel: Vápno zneutralizuje túto kyselinu a vytvorí nerozpustnú zrazeninu

PP8: Ak bol únik kyseliny soľnej

Potom použi vápno

Zmysel: Vápno zneutralizuje túto kyselinu a vytvorí nerozpustnú zrazeninu

Ak sa stane, že nemáme k dispozícii vápno, ale lúh. Musíme nájsť odpoveď na otázku: či aj lúh zneutralizuje príslušnú kyselinu a vytvorí s ňou nerozpustnú zrazeninu. Z bázy znalostí by sme sa dozvedeli:

lúh + kyselina sírová tvoria rozpustnú zrazeninu

lúh + kyselina octová tvoria nerozpustnú zrazeninu

lúh + kyselina soľná tvoria nerozpustnú zrazeninu

Z toho vyplýva, že lúh nemôžeme použiť na neutralizáciu kyseliny sírovej, ale môžeme ho použiť na neutralizáciu kyseliny octovej a soľnej.

Aj návrh o zámene vápna lúhom môže mať svoj zmysel, čo je ilustrované nasledujúcim príkladom:

MP: Ak nie je vápno

Potom preskúmajte možnosť jeho zámene lúhom

Zmysel: Lúh môže neutralizovať kyseliny

Čo všetko sa môžeme dozvedieť zo zmyslu pravidla?

Často chceme vedieť, na čom spočíva dôvera v nejaké pravidlo:

- či pravidlo pochádza z dedukcie, alebo indukcie,
- či bolo získané z experimentu alebo od autority.

Pri neurčitých pravidlách:

- odvádzame šancu pravidla na úspech,
- odhadujeme riziko použitia pravidla.

Používateľ môže chovať nedôveru k výsledkom znalostného systému a chcel by vedieť:

- nakoľko je pravidlo spoľahlivé v praxi?
- od koho boli produkčné pravidlá získané?
- prečo sa predpokladá, že pracuje správne?

46. Môžu metapravidlá detekovať chyby v pravidlách?

V princípe je to možné. Uvažujme dva pomerne často sa vyskytujúce prípady chybných pravidiel.

Po prvé, predpokladajme, že sa v BZ z nejakých dôvodov vyskytlo pravidlo:

PP9: THEN Použite vápno.

Takéto pravidlo, ktorému chýba podmienková časť, musí byť systémom zachytené a systém musí následne požiadať znalostného inžiniera, aby pravidlo potvrdil, alebo opravil. Mohlo sa stať, že pre dané pravidlo má platiť tá istá podmienková časť ako pre predchádzajúce, ale znalostný inžinier ju zabudol doplniť.

Po druhé, často sa vyskytujúca chyba je, že pravidlo obsahuje podmienky, ktoré sa nedajú splniť, napríklad:

PP10: Ak potrebujeme neutralizovať kyselinu
a súčasne máme lieh
a súčasne nemáme nič s vysokým pH
Potom ...

Tretia podmienka sa ťažko overuje. Lepšie by bolo nahradiť ju podmienkou „nemáme nič **iné** s vysokým pH“. Bez toho slovíčka „iné“, nebude dané pravidlo nikdy odpálené. Bez toho slovíčka „iné“ je v pravidle rozpor: ak máme lieh, tak máme niečo s vysokým pH.

Tieto chyby by mohli byť korigované nasledovnými metapravidlami:

MP11: Ak pravidlo neobsahuje IF časť
potom navrhni korekciu, alebo vyžiadať potvrdenie
MP12: Ak počas behu systému nebolo niektoré pravidlo použité
potom sa spýtaj, či jeho podmienky nepotrebujú korekciu.

47. Môžu metaprávidlá zdôvodňovať architektúru znalostného systému?

48

V princípe je to možné, pretože otázky na systém môžu ísť niekedy do takej hĺbky, že sa týkajú jeho architektúry. Na ich zodpovedanie sú potrebné znalosti tak experta ako aj človeka zodpovedného za vytvorenie systému, zväčša znalostného inžiniera. Môže ísť o otázky typu: Prečo bola vybraná taká pomalá štruktúra riadenia? Prečo sa systém nevie vrátiť niekoľko krokov dozadu? Na zodpovedanie otázok tohto druhu je potrebné systém vybaviť veľkou zásobou vedomostí, týkajúcich sa samotnej konštrukcie ZS. Takých vedomostí ako napríklad: Z čoho pozostáva úloha? Ako ju treba riešiť? Aké sú varianty reprezentácie riadiacich štruktúr? Aké sú výhody a nedostatky riadiacich štruktúr? Ďalej sú uvedené príklady metaznalostí zdôvodňujúcich architektúru systému:

- MP13: Ak je priestor riešení malý
Potom použi metódu úplného prehľadávania
- MP14: Ak je produkčný systém zriedkavo modifikovaný
Potom ho skompiluj

- Metaznalosť MP13 umožňuje konštruktérovi, ktorý pozná rozmer úlohy, vybrať si spôsob ovládania, riadenia riešenia, resp. spôsob prehľadávania priestoru riešení. Konkrétny používateľ sa môže spýtať: Prečo bolo použité toto pravidlo? Systém odpovie: Lebo používam úplné prehľadávanie. Odpoveď je inšpirovaná metaprávidlom MP13. Teda, MP13 môže byť použité trojako:
 - o ako pomoc konštruktérovi systému
 - o ako pomoc pri riešení úlohy
 - o ako pomoc pri odpovedi na otázky používateľa.

48. Ako sa uskutočňuje adaptácia znalostného systému na novú situáciu pomocou metapravidiel?

V prvom rade si musíme položiť otázku: Môže systém sám meniť svoju konfiguráciu na základe získaných skúseností? Skôr než odpovieme, uvažujme nasledovné metapravidlá:

- MP15: Ak sa ukáže, že niektorá podmienka je častejšie nespĺnená ako ostatné
Potom zarad' túto podmienku na začiatok podmienkovej časti
- MP16: Ak sa časť kódu programu systému vyvoláva často
Potom je účelné sa zaoberať jeho optimalizáciou
- MP17: Ak určitý typ používateľa používa znalostný systém iba zriedkavo
Potom je vhodné vymazať model tohto používateľa zo systému

Podmienkové časti týchto metapravidiel popisujú novú situáciu, v ktorej sa systém nachádza, napríklad: V poslednej dobe je niektorá podmienka častejšie nespĺnená. Alebo, časť kódu programu sa vyvoláva často. Resp. tento používateľ v poslednej dobe systém takmer nepoužíva.

Záverové časti týchto metapravidiel predstavujú reakciu znalostného systému na túto novú situáciu: preusporiadajú podmienky, optimalizujú kód, vymaž model používateľa. Dá sa taktiež povedať, že predstavujú formu adaptácie systému na novú situáciu.

Takéto pravidlá môžu fungovať iba v systéme s evidenciou frekvencie používania jednotlivých častí kódu, s evidenciou strednej doby práce jednotlivých kódov. Z toho vyplýva, že metaznalosti umožňujúce adaptáciu na novú situáciu pracujú v súčinnosti s podpornými, opisnými metaznalosťami.

49. Je možné modelovať možnosti systému pomocou metaznalostí?

V prvom rade by sme si mali objasniť, čo to znamená „modelovanie možnosti systému“. V podstate ide o to, že systém má zmapované vlastné možnosti. To znamená, že vie, ktoré úlohy dokáže riešiť, a ktoré nie, prípadne ktoré problémy vyrieši ale riešenie pravdepodobne nebude dostatočne validné. Uvedieme príklad: Keď sa vás spýtajú na vaše telefónne číslo, hneď odpoviete. Keď sa vás spýtajú na telefónne číslo známeho, povieť, že sa musíte pozrieť do zoznamu. Keď sa vás opýtajú na telefónne číslo prezidenta, uvedomíte si, že je zbytočné pozeráť sa do zoznamu a jednoducho povieť: „neviem“. Pri rozhodovaní o tom, či použijete pamäť, zoznam, alebo sa vzdáte, používate znalosti o tom, čo poznáte a čo nie, ktoré úlohy ste schopný riešiť a ktoré nie, koľko času Vám zaberie riešenie úlohy, a teda, či má zmysel začínať s riešením. Slovíčko „neviem“ je prejavom vašej sebareflexie. Modelovanie možnosti systému je modelovaním sebareflexie systému. Uvedieme príklady metaznalostí, modelujúcich možnosti systému:

- MP18: Ak je vzniknutá látka biologicky aktívna
Potom báza znalostí nie je adekvátna na riešenie úlohy
(dokáže riešiť len chemické znečistenie)
- MP19: Ak sa vzniknutá látka nenachádza v zozname látok
Potom báza znalostí nie je kompetentná pre riešenie úlohy

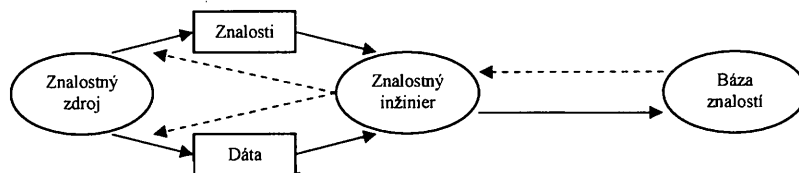
Program, ktorý je vybavený takýmito metaznalosťami, môže na niektoré situácie reagovať pružnejšie a rýchlejšie. MP18 oznamuje, že eventúálne prijaté závery nie sú adekvátne situácii. MP19 umožňuje systému vyhnúť sa odvodeniam, ktoré by aj tak nevedli k výsledku. Modelovanie sebareflexie znalostného systému je nevyhnutné v oblastiach, v ktorých chyby nie sú prípustné.

50. Ako je možné získavať znalosti pre znalostný systém?

Podľa cieľa môžeme metódy používané pre získavanie znalostí deliť na:

- ✓ získavanie nových znalostí (BZ je na začiatku prázdna)
- ✓ zjemňovanie (modifikácia, dopĺňanie) existujúcich znalostí (BZ nie je na začiatku prázdna).

Báza znalostí predstavuje explicitné vyjadrenie znalostí obsiahnutých v nejakom znalostnom zdroji. Znalostný zdroj môže byť predstavovaný expertom, databázou, empirickými dátami, knihou, expertízou a podobne. Explicitné vyjadrenie znalostí môže mať formu produkčných pravidiel, predikátovej logiky, rámcov a podobne. Proces získavania znalostí je ilustrovaný nasledovným obrázkom:



Subjekt získavania znalostí, teda znalostný inžinier, hrá centrálnu úlohu v procese získavania znalostí. Znalostný inžinier v spolupráci so znalostným zdrojom získava buď priamo znalosti, ktoré transformuje do BZ, alebo dáta, ktorých extrakciou získa pravidlá pre BZ.

V princípe existujú tri najznámejšie spôsoby získavania znalostí:

- ✓ **klasickou cestou.** Je to historicky najstarší model. V tomto modeli robí znalostný inžinier spojovací medzičlánok medzi BZ a expertom ako znalostným zdrojom. Konzultuje s expertom jeho znalosti a vkladá ich do BZ. Slabým miestom tohto spôsobu získavania znalostí je tak expert ako aj znalostný inžinier. Znalostný inžinier preto, lebo je laikom v danej oblasti. Musí pochopiť aspoň základy domény a toto učenie je zdĺhavé a neefektívne. Expert preto, lebo nie je vždy schopný vyjadriť svoje znalosti ktoré používa na riešenie úloh. Jeho riešenia sú často podvedomé, vychádzajúce zo skúsenosti. Často je zaneprázdnený a niekedy sa obáva konkurencie znalostných systémov.
- ✓ **priamym prenosom.** Priamy prenos je pokusom vylúčiť z procesu získavania jedno jeho slabé miesto, a to znalostného inžiniera. Pri tomto spôsobe je BZ vytváraná priamo expertom v príslušnom editore a v dohodnutom formáte. BZ vytvára expert, oboznámený s problematikou programovania. Nevyhnutný je inteligentný modul priameho získavania znalostí, ktorý nie je pasívnym príjemcom znalostí, ale pomáha expertovi vyjadrovať znalosti, navádza ho a kontroluje, dokáže odhaliť logický rozpor, musí mať rozsiahle dialógové schopnosti a musí mať znalosti o štruktúre BZ.
- ✓ **strojovým učením.** Strojové učenie je pokusom vylúčiť z procesu získavania ďalšie jeho slabé miesto, a to experta. Strojové učenie sa v princípe delí na indukčné a deduktívne. V znalostných systémoch sa používa indukčné strojové učenie, ktoré indukuje, resp. generuje produkčné pravidlá pre BZ priamo z tréningových príkladov reprezentujúcich typické situácie.

Všetky metódy získavania znalostí majú svoje výhody a nevýhody. Niektoré sa používajú v počiatočných fázach procesu budovania BZ, iné v neskorších. Väčšinou sa používa klasická cesta doplnená indukčnými metódami strojového učenia.

Pri dopĺňaní BZ a jej modifikácii sa spravidla aplikuje klasická cesta. Spravidla je BZ dopĺňaná znalosťami, ktoré súvisia s už danými znalosťami v BZ. Práca (Bieliková a kol., 1992) pojednáva o podpore vývoja znalostných systémov.

51. Čo môžeme získať strojovým učeníím?

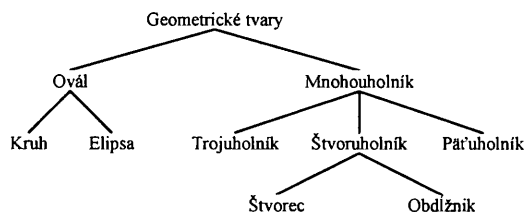
Pomocou algoritmov strojového učenia môžeme získať produkčné pravidlá pre BZ priamo z typických príkladov. Strojové učenie je vo svojej podstate induktívnym učeníím (Michalski at all, 1983). Nevychádza z expertných znalostí, ale z typických príkladov. V terminológii strojového učenia sa typický príklad označuje častejšie pojmom tréningový príklad. Množina spravidla veľkého počtu tréningových príkladov sa nazýva tréningová množina. Tréningový príklad pozostáva z dátovej a rozhodovacej časti. Dátová časť popisuje prípad. **Dátová časť** obsahuje množinu atribútov (v nasledovnej tabuľke je množina atribútov = {Inteligencia, Tvorivosť, Usilovnosť}) s ich hodnotami (v nasledovnej tréningovej tabuľke napríklad: Inteligencia = výnimočná). **Rozhodovacia časť** slúži pre klasifikáciu daného prípadu do triedy. Ide o rozhodnutie, resp diagnostikovanie napríklad choroby alebo poruchy. V nasledujúcej tréningovej tabuľke ide o diagnostikovanie úspechu na skúške, pretože trieda do ktorej môže byť príklad diagnostikovaný je: Výsledok.

Študent	Inteligencia	Tvorivosť	Usilovnosť	Výsledok
1	výnimočná	výnimočná	lenivý	uspeje
2	priemerná	priemerná	usilovný	uspeje
3	priemerná	výnimočná	lenivý	uspeje
4	podpriemerná	žiadna	lenivý	neuspeje
5	priemerná	žiadna	lenivý	neuspeje

Tento typ učenia je takzvané **kontrolované učenie**, resp. učenie s učiteľom. Predpokladá sa, že je vopred známa množina riešení, množina tried. Našou úlohou je vybrať jedno z možných riešení, resp. klasifikovať úlohu do triedy. Existuje aj opačná alternatíva, keď nie je známa množina možných riešení a ide o **nekontrolované učenie**. Príkladom nekontrolovaného učenia je zhlukovanie.

Atribúty v dátovej časti tréningového príkladu môžu byť rôzneho druhu:

- ✓ binárne – majú dvoj prvkovú množinu hodnôt napríklad: {ano, nie}, {platí, neplatí}
- ✓ nominálne – majú konečnú neusporiadanú množinu hodnôt, napríklad: {oheň, voda, vzduch}
- ✓ ordinárne – majú konečnú usporiadanú množinu hodnôt, napríklad množina vlnových dĺžok farieb, alebo množina: {nízky, stredný, vysoký}
- ✓ numerické – majú numerické hodnoty, spravidla usporiadané
- ✓ reálne – majú nekonečný počet reálnych hodnôt
- ✓ hierarchické – majú hodnoty, medzi ktorými sú hierarchické vzťahy, napríklad: {ovál, kruh, elipsa, mnohouhol., trojuhol., štvoruhol., štvorec, obdĺžnik, päťuhol.};



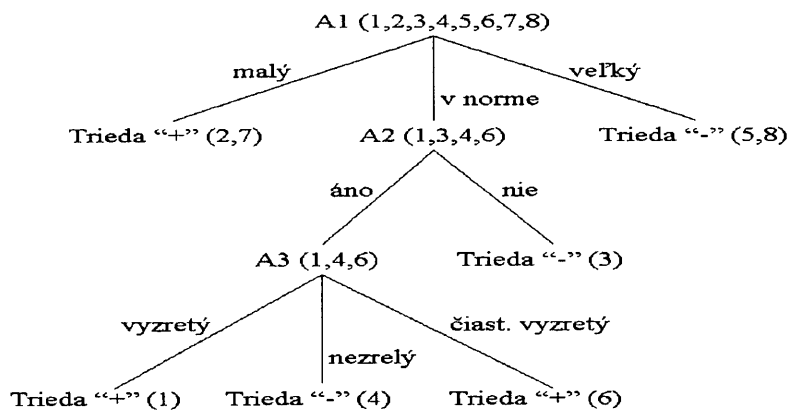
Cieľom strojového učenia je nájsť rozhodovaciu štruktúru (produkčné pravidlo, rozhodovací strom), ktorá pre dátovú časť každého príkladu nájde triedu, t.j. jeho rozhodovaciu časť.

52. Prečo a ako generujeme rozhodovacie stromy?

Rozhodovacie stromy (Quinlan, 1990), (Quinlan, 1993) generujeme preto, lebo z nich ľahko získame produkčné pravidlá pre BZ znalostného systému. Navyiac, rozhodovacie stromy sú prehľadné. Rozhodovací strom je grafická reprezentácia rozhodovacej procedúry. Služi na klasifikáciu tréningových príkladov do tried, pričom klasifikácia môže byť dvojtriedna, alebo viactriedna. Napríklad z nasledovnej tréningovej množiny:

Por.č.	A1	A2	A2	T
1.	v norme	áno	vyzretý	+
2.	malý	áno	vyzretý	+
3.	v norme	nie	vyzretý	-
4.	v norme	áno	nezrelý	-
5.	veľký	nie	vyzretý	-
6.	v norme	áno	čiasť. vyzretý	+
7.	malý	áno	čiasť. vyzretý	+
8.	veľký	nie	čiasť. vyzretý	-

Je možné generovať takýto rozhodovací strom:



Rozhodovací strom je teda orientovaný acyklický graf, ktorý obsahuje koreňové, medzilahlé a listové uzly. Listový uzol odpovedá triede. Nelistový uzol obsahuje špecifický test, ktorý rozdeľuje priestor príkladov podľa výsledku testu, ktorý má spravidla tvar: atribút = hodnota. Atribút, ktorý vyberieme zo všetkých atribútov a umiestnime v nelistovom uzle stromu sa nazýva testovací atribút. Podmienková časť produkčného pravidla potom predstavuje konjunkciu podmienok na ceste od koreňa k listom. Záver pravidla je trieda, ktorá sa nachádza v listovom uzle na konci cesty. Z vyššie uvedeného stromu je možné generovať nasledovné pravidlá: AK A1=malý POTOM trieda (+)
 AK A1=v norme & A2=áno & A3=vyzretý POTOM trieda (+)
 AK A1=v norme & A2=áno & A3=čiasť vyzretý POTOM trieda (+)

Induktívne algoritmy budujú strom zhora nadol, pričom delia priestor príkladov rekurzívne na podpriestory výberom testovacích atribútov, až kým nie je splnená ukončovacia podmienka (spravidla kým všetky príklady v danom uzle nebudú klasifikované do jednej triedy).

53. Ako generuje rozhodovacie stromy algoritmus ID3?

Najznámejším algoritmom generujúcim rozhodovací strom metódou zhora nadol je algoritmus **ID3 (Iterative Dichotomizer 3)** (Quinlan, 1990). Voľne by sa dal názov algoritmu preložiť ako Iteratívny dvojtriedny klasifikátor. Bol vyvinutý Rossom Quinlainom v 1979 v doméne znalostí šachistu o hraní šachových koncoviek.

Ukončovacie kritérium tohto algoritmu je, že každý podpriestor obsahuje iba príklady jednej triedy. Ak je množina atribútov dostatočná, možno uvedeným postupom zostrojiť rozhodovacie stromy korektne klasifikujúce trénovacie príklady. Hovoríme že rozhodovací strom generovaný algoritmom je:

- **perfektný** keď správne klasifikuje všetky trénovacie príklady
- **rozmerovo optimálny resp. minimálny** keď je rozhodovacia procedúra čo najjednoduchšia, s čo najmenším počtom testov hodnôt atribútov.

Algoritmus ID3 pre výber testovacieho atribútu využíva Shannonovu teóriu informácie (Russell-Norvig, 1995), ktorá na meranie množstva informácie používa entropiu. Claude Shannon objavil informačnú teóriu ešte v prvej polovici dvadsiateho storočia. Spôsobila revolúciu v oblasti komunikácie a taktiež prispela k úspechu učenia pomocou rozhodovacích stromov.

Ak správy x_1, x_2, \dots, x_n sú možné s pravdepodobnosťami $p(x_1), p(x_2), \dots, p(x_n)$ a tieto vytvárajú úplný súbor pravdepodobností, teda:

$$\sum_{j=1}^N p(x_j) = 1$$

potom entropiu, resp. neurčitost' daného súboru správ možno vyjadriť nasledovne:

$$H = -\sum_{j=1}^N p(x_j) \log_2 p(x_j)$$

Tento vzťah je známy ako Shannonova entropia. Čím je entropia súboru správ vyššia, tým menej určitý je obsah budúcej správy a tým väčšie množstvo informácie získame, keď správu prijmem.

Rozhodovací strom môže byť považovaný za zdroj informácie, ktorá pre nejaký konkrétny príklad formuje správu o klasifikácii daného príkladu do nejakej triedy. Keď niektorý uzol stromu obsahuje iba príklady tej istej triedy, entropia v tomto uzle je nulová, pretože klasifikačné rozhodnutie pre príklady prislúchajúce tomuto uzlu je pevne dané. (Teda pravdepodobnosť jednej triedy je 1 a ostatných nulová.)

Z toho vyplýva, že entropia v listových uzloch je nulová, zatiaľ čo v koreňovom uzle a medziľahlých uzloch má nenulovú hodnotu. Pre vygenerovanie minimálneho rozhodovacieho stromu by entropia mala čo najrýchlejšie klesnúť na nulovú hodnotu. Algoritmus ID3 používa heuristiku, ktorá usiluje o čo najvyšší pokles entropie lokálne v každom kroku.

Nech uzol **S** obsahuje n_1 príkladov klasifikovaných do triedy **T1** a n_2 príkladov zaradených do triedy **T2**. Potom pravdepodobnosť, že nejaký príklad prislúchajúci uzlu **S** rozhodovacieho stromu bude klasifikovaný do triedy **T1** alebo **T2** je

$$\frac{n_1}{n_1 + n_2} \quad \text{resp.} \quad \frac{n_2}{n_1 + n_2}$$

Teda entropia v danom uzle S , presnejšie povedané entropia súboru tréningových príkladov prislúchajúcich uzlu S , bude určená nasledovne:

$$H(S) = - \sum_{j=1}^2 \frac{n_j}{n_1 + n_2} \log_2 \frac{n_j}{n_1 + n_2}$$

Predpokladajme, že je možné v uzle S použiť atribút A s hodnotami a_1 a a_2 pre rozdelenie príkladov z uzla S do dvoch disjunktných podmnožín S_1 a S_2 . Ak v uzle S má m_1 príkladov hodnotu atribútu A rovnú a_1 a m_2 príkladov má hodnotu a_2 , potom celková entropia v uzle S s použitím atribútu A sa určí ako:

$$H(S, A) = \sum_{j=1}^2 \frac{m_j}{m_1 + m_2} H(S_j)$$

Informácia získaná použitím atribútu A v uzle S je označovaná ako informačný zisk a počíta sa nasledovne:

$$I(A) = H(S) - H(S, A)$$

V každej iterácii algoritmu pre generovanie rozhodovacieho stromu metódou zhora nadol sa vyšetrujú všetky ešte nepoužitú atribúty a ako testovací atribút sa vyberá ten, ktorý maximalizuje informačný zisk resp. minimalizuje entropiu t.j. neurčitost'. Stratégiou algoritmu ID3 je čo najrýchlejší pokles entropie na nulu a tak generovanie minimálneho rozhodovacieho stromu.

Pre použitie algoritmu musí platiť podmienka neprotirečivosti (nekontradikčnosti) tréningových príkladov. Nutnosť platnosti tejto podmienky vyplýva z toho, že ID3 nie je odolný voči zašumeným údajom na vstupe. Zo spôsobu určovania pravdepodobností vyplýva, že iba dodržanie dvoch dodatočných podmienok, a to podmienky neredundantnosti príkladov (nepovolenie viacnásobného výskytu tréningového príkladu v tréningovej množine) a podmienky vzájomnej nezávislosti atribútov, garantuje nájdenie minimálneho rozhodovacieho stromu.

54. Čo je to Kombinačná analýza dát?

Kombinačná analýza dát (KAD) je metóda bežná v štatistike. Výskumníci znalostných systémov ju prispôbili požiadavkám vlastnej oblasti. KAD sa zaoberá výskytom rôznych kombinácií selektorov (Atribút = hodnota) v dátach. Zisťuje **frekvencie**, t.j. početnosti výskytov kombinácií selektorov v dátach. Na vyššej úrovni zisťuje **koincidencie**, t.j. zhody výskytov kombinácií selektorov a daných tried v dátach. Podstatou použitia KAD je vyhľadávanie takých kombinácií selektorov v dátach, ktoré smerujú k (resp. sú späté so) zadanej cieľovej kombinácii (pre znalostné systémy klasifikačnej triede). Dáta sú dané väčšinou vo forme tabuľky, kde stĺpce obsahujú hodnoty atribútov (vrátane tried – posledný) a riadky obsahujú pozorovania vo forme trénovacích príkladov. Nasleduje príklad takej tabuľky.

A1	A2	A3	A4	T
a	b	r	t	T1
a	b	x	y	T2
a	b	p	q	T1
c	d	x	y	T2

Frekvencie sa skúmajú iba v dátovej časti, bez uvažovania tried. Koincidencie definujú vzťah kombinácie hodnôt atribútov voči hodnotám triedy. (Pôvodná metóda skúmala vzťah kombinácie hodnôt atribútov voči kombinácii hodnôt tried. Toto chápanie metódy KAD bolo zúžené na hodnoty tried v jednom stĺpci, čím bola pôvodná metóda prispôbená na použitie v znalostných systémoch.)

Majme N atribútov a jednu klasifikačnú triedu. Nech $V_i(Atr_i)$ je množina všetkých hodnôt, ktoré môže nadobúdať Atr_i . Potom kombinácia dĺžky k je:

$$C = (Atr_1 = a_1) \wedge (Atr_2 = a_2) \wedge \dots \wedge (Atr_k = a_k), \text{ pričom } a_i \in V_i(Atr_i)$$

Nech d je dĺžka skúmanej kombinácie. Potom príklady kombinácie dĺžky $d = 2$ sú:

$$d = 2 \Rightarrow C_2 = (A_2 = h_{21}) \wedge (A_6 = h_{63})$$

$$d = 2 \Rightarrow C_2 = (A_4 = h_{42}) \wedge (A_1 = h_{18})$$

pričom d môže nadobúdať hodnoty z intervalu $\langle 1, N \rangle$. Počet možných kombinácií sa dá určiť podľa vzťahu: $X = (|V_1| + 1)(|V_2| + 1) \dots (|V_N| + 1)$.

Výpočet frekvencie kombinácie. Výskytom kombinácie v dátach rozumieme každý trénovací príklad, v ktorom sa v príslušných stĺpcoch nachádzajú hodnoty atribútov zo zvolenej kombinácie. Frekvencia kombinácie v dátach sa označuje $|C|_D$ a predstavuje počet príkladov v ktorých sa daná kombinácia nachádza. **Fundovanosť kombinácie** predstavuje dolnú medzu počtu trénovacích príkladov, v ktorých sa daná kombinácia musí vyskytovať, aby bola fundovaná, t.j. aby sme sa ňou vôbec zaoberali.

Výpočet koincidencie. Tento výpočet smeruje k zisteniu počtu výskytov empirických implikácií v tvare: $C \rightarrow T_i(P)$ alebo $C \rightarrow T_i(w_D)$. Pojem empirická implikácia znamená platnosť s výnimkami (napríklad na 90%). Platnosť implikácie počítame podľa vzťahu:

$w_D = P = \frac{|C \wedge T_i|_D}{|C|_D}$, kde v čitateli zlomku je vyčíslená frekvencia súčasného výskytu kombinácie C a triedy T_i , zatiaľ čo v menovateli iba frekvencia výskytu kombinácie C . Platí, že $P \in \langle 0, 1 \rangle$. Aj v tomto prípade sa zadáva dolná medza platnosti, ktorá definuje aká minimálna musí byť platnosť implikácie, aby sa s ňou počítalo, t.j. aby bola platná.

55. Ako je možné použiť Kombinačnú analýzu dát v znalostných systémoch?

Kombinačnú analýzu dát (KAD) (Mach, 1997) je možné použiť na vyhľadávanie implikácií v dátach, tieto implikácie je možné prehodnotiť, stanoviť ich neurčitosť a vložiť do BZ znalostného systému. Pomocou tejto metódy je možné získať produkčné pravidlá priamo z dát, bez asistencie experta. Princíp je podobný tomu, ktorý sa používa v strojovom učení.

To, ktoré platné empirické implikácie existujúce v dátach budú zaradené do bázy znalostí, závisí na zvolenom inferenčnom mechanizme (aká operácia skladania neurčitých príspevkov GLOB sa používa), na požiadavke presnosti, s ktorou má byť BZ v súlade s dátami a na požadovaných hodnotách platnosti a fundovanosti. Pre rozhodnutie o zaradení implikácie do BZ možno použiť analógiu s vytváraním axiomatickej teórie nejakého odboru. Z tohto hľadiska musí BZ spĺňať dve požiadavky:

- ✓ **úplnosť**. Medzi axiomy sú zaradované všetky najjednoduchšie tvrdenia. Všetky ostatné zákony odboru, resp. domény sú z nich odvoditeľné. Napríklad, majme dve axiomy A1 a A2.

A1: Sčítanie jednociferných čísel

A2: Prenos cez desiatku

Z nich sa dá odvodiť sčítavanie dvojiciferných čísel.

- ✓ **neredundantnosť**. Zákony, ktoré sa dajú odvodiť z iných axióm sú odstraňované z bázy znalostí.

Uvádzaná vlastnosť neredundantnosti sa dá v rámci znalostných systémov interpretovať nasledovne. Do bázy znalostí budú zaradené iba tie implikácie, ktorých platnosť v dátach sa významne líši od váhy odvodenej inferenčným mechanizmom z váh vzťahov do bázy znalostí už zaradených. Odlišnosť váh akoby mala zabezpečiť, že vkladaná implikácia sa nedá odvodiť z implikácií, ktoré už v BZ sú. Napríklad ak riešime problém, či je prípustné zaradiť do BZ nasledovnú implikáciu, ktorej váha w_D je väčšia, než požadovaná platnosť implikácie:

$(A_1 = h_{16}) \wedge (A_2 = h_{24}) \rightarrow T_3(w_D)$. V BZ sa však už nachádzajú implikácie:

$$A_1 = h_{16} \rightarrow T_3(w_1)$$

$$A_2 = h_{24} \rightarrow T_3(w_2)$$

Združíme váhy týchto implikácií w_1, w_2 a vytvoríme združenú váhu w_z pomocou grupovej

operácie na združovanie príspevkov \otimes_{GLOB} : $w_z = w_1 \otimes_{GLOB} w_2$.

Ak $w_z = w_D$, potom nové pravidlo do bázy znalostí nezaradíme, pretože predpokladáme, že sa dá odvodiť z axióm A1 a A2. Ak $w_z \neq w_D \Rightarrow$ nové pravidlo do bázy znalostí zaradíme, pričom predpokladáme výraznú odlišnosť týchto váh.

Teda, aby sme to zhrnuli vzťah $C = (Atr_1 = a_1) \wedge (Atr_2 = a_2) \wedge \dots \wedge (Atr_k = a_k) \rightarrow T_i$ budeme považovať pre dané dáta za odvoditeľný z BZ (konkrétne z množiny vzťahov vedúcich na T_i), ak platnosť tohto vzťahu v dátach sa významne nelíši od hodnoty odvodenej zložením príspevkov tých pravidiel z BZ, ktorých podmienka je podkombináciou kombinácie C . Ak je nejaký vzťah zaradovaný do BZ, tak s váhou w , ktorá koriguje rozdiel medzi platnosťou w_D zaradovaného vzťahu a zloženou váhou w_z podľa definície $w_z \otimes w = w_D$.

BZ získaná KAD je použiteľná vo všetkých znalostných systémoch, pracujúcich s numerickým prístupom ku skladaniu neurčitých príspevkov jednotlivých pravidiel do neurčitého záveru.

56. Ako je definovaný algoritmus Kombinačnej analýzy dát?

Algoritmus vytvárania BZ pomocou KAD má dve etapy:

1) **inicializačnú fázu**. Inicializačná fáza pozostáva s nasledovných krokov:

a) Inicializujeme $d = 0$, kde d je momentálne skúmaná dĺžka kombinácie ľavej strany hľadanej implikácie.

$$0 \rightarrow T_i(w_0)$$

b) Nasledovným spôsobom sa určí w_0 platnosť prázdneho vzťahu

$$w_0 = \frac{|T_i|_D}{m}$$

v čitateli je početnosť výskytov T_i v dátach a v menovateli je počet riadkov tabuľky, pretože prázdny vzťah platí pre všetky tréningové príklady.

c) Ak sa w_0 líši signifikantne od numerickej hodnoty 0.5, potom sa prázdny vzťah zaradí do BZ s váhou w_0 . Ak sa w_0 nelíši signifikantne od numerickej hodnoty 0.5, potom sa prázdny vzťah zaradí do BZ s váhou 0.5.

2) **fázu vyhľadávania vzťahov**. Táto fáza pozostáva z nasledovných krokov:

a) Inkrementuje sa d na $d + 1$.

Ak $d > d_{\max}$ (stanovená maximálna dĺžka kombinácie), potom končíme.

Ak $d > m$ (počet atribútov), potom končíme.

Druhé pravidlo je alternatívou pre prípad, že d_{\max} nebolo stanovené.

b) Pre všetky kombinácie dĺžky d sa nájde fundovaná kombinácia C dĺžky d , definuje sa

$$\text{implikácia } C \rightarrow T_i \text{ a vypočíta sa jej platnosť, podľa vzťahu } w_D = P = \frac{|C \wedge T_i|_D}{|C|_D}.$$

c) Ak je w_d menšie ako požadovaná platnosť, potom ideme na krok 2b).

Ak je w_d väčšie ako požadovaná platnosť, potom ideme na krok 2d).

d) Nájdu sa váhy $w_0(0), w_1(A_1 = a_1), \dots$ všetkých implikácií v BZ, ktorých ľavá strana je podkombináciou momentálne skúmanej kombinácie (v podkombinácii smie byť maximálne $d - 1$ podmienok).

e) Z týchto nájdených váh sa určí w_z opätovným použitím operácie GLOB pre skladanie neurčitých príspevkov do výslednej neurčitosti:

$$w_{z_1} = w_0 \underset{GLOB}{\otimes} w_1$$

$$w_{z_2} = w_{z_1} \underset{GLOB}{\otimes} w_2$$

⋮

$$w_z = w_{z_{n-1}} \underset{GLOB}{\otimes} w_n$$

f) Realizuje sa test odlišnosti:

Ak $\text{dif}(w_D, w_z) = \text{true}$, potom sa do BZ zaradí implikácia $C \rightarrow T_i$ s váhou W .

Ak $\text{dif}(w_D, w_z) = \text{false}$, potom do bázy znalostí implikáciu nezaradíme.

Váhu W počítame podľa vzťahu: $w_z \underset{GLOB}{\otimes} w = w_D$.

57. Čo predstavujú priame metódy získavania znalostí?

V princípe môžeme znalosti pre BZ znalostného systému získavať priamo a nepriamo, t.j. priamymi metódami získavania znalostí a nepriamymi (Mach, 1997). Priame metódy sú založené na vedomých znalostiach experta. U experta predpokladajú prístup experta k svojim mentálnym procesom (expert dokáže formulovať vedomosti, ktoré používa) ako aj schopnosť a ochotu odovzdať svoje znalosti znalostnému inžinierovi. Rozlišujeme nasledovné triedy priamych metód získavania vedomostí:

- 1) rozhovory
- 2) prípadové štúdie
- 3) pozorovania
- 4) získavanie pojmov
- 5) špeciálne techniky. Postupne si preberieme každú z týchto tried.

1) ROZHOVORY predstavujú plošné pokrytie oblasti. Majú retrospektívny charakter, pretože sú postavené na minulej praxi. Delia sa nasledovne:

- ✓ **Neštruktúrovaný rozhovor** je voľné „interview“, pričom ani obsah ani sekvenčnosť rozhovoru nie sú vopred určené. Cieľom je získať kompletný prehľad domény, resp. oblasti. Používa sa hlavne v počiatočných fázach získavania znalostí.
- ✓ **Štruktúrovaný rozhovor** môže mať formu **vysoko štruktúrovaného** rozhovoru, v ktorom aj obsah aj poradie udalostí sú dané, alebo **semištruktúrovaného**, v ktorom je obsah daný, ale poradie otázok sa môže meniť.
- ✓ **Sondy** sú krátke zásahy do konverzácie. Znalostnému inžinierovi umožňujú ovplyvňovať konverzáciu. **Aditívna** sonda povzbudzuje experta k poskytnutiu ďalších informácií a naznačuje, že znalostný inžinier pochopil, čo expert doposiaľ povedal. **Reflexívna** sonda predstavuje nedirektívnu výzvu pre experta k hlbšiemu ponoreniu sa do námetu. Znalostný inžinier urobí syntézu expertovej predchádzajúcej odpovede a postaví ju ako otázku. **Direktívna** sonda špecifikuje smerovanie a úroveň všeobecnosti ďalšej konverzácie. **Sonda zmeny módu** inicializuje prechod na iné hľadisko. **Definičná** sonda zase inicializuje vysvetlenie významu nejakého pojmu.
- ✓ **Spätné učenie** je metódou verifikácie. Znalostný inžinier spätne vysvetľuje expertovi niečo, čo mu on predtým vysvetlil. Robí to dovedy, kým expert nie je s vysvetlením spokojný.
- ✓ **Dvadsať otázok**. Znalostný inžinier volí v duchu nejakú situáciu (diagnózu, poruchu, stav) a expert sa snaží prísť na to, o čo ide. Môže položiť dvadsať otázok, na ktoré dostane odpoveď áno, resp. nie. Typy expertových otázok prezradia znalostnému inžinierovi sekvencie riešiacich krokov a stratégie, ktoré expert používa pri práci.
- ✓ **Dotazníky**. Ich výhodou je, že ich expert môže vyplniť v pohodlí a v čase, keď mu to vyhovuje.

2) PRÍPADOVÉ ŠTÚDIE sa sústreďujú na jednotlivé riešené prípady. Kritickým miestom tejto metódy je výber vhodných reprezentatívnych prípadov. Prípady nemajú byť rutinné, ale zaujímavé, zriedkavé, ľahko zapamätateľné, obtiažne, prípadne zábavné. Patria sem nasledovné techniky:

- ✓ **Analýza prípadových štúdií.** Expert sa rozepamätáva na konkrétny prípad. Znalostný inžinier spolu s expertom analyzujú krok za krokom hľadanie riešenia. K úskaliam tejto metódy patrí deformované vybavovanie – tzv. spomienkový optimizmus, keď rekonštrukcia riešenia zahŕňa aj momentálne znalosti, t.j. znalosti získané po vyriešení prípadu.
- ✓ **Simulácia podľa scenára.** Znalostný inžinier poskytne náčrt prvotnej situácie. Expert rieši prípad daný načrtnutou situáciou. Riešenie môže byť reprodukované (riešenie použité v minulosti), alebo konštruované (nový prípad).
- ✓ **Metóda minimálneho scenára.** V predloženom scenári chýbajú niektoré kritické informácie. Expert je nútený ich doplniť, alebo sa bez nich zaobísť.
- ✓ **Dynamické spätné učenie.** Znalostný inžinier vykonáva simuláciu expertovej práce, pričom už dostatočne chápe riešenia úloh v oblasti. Expert spočiatku navádza znalostného inžiniera na správny postup.
- ✓ **Verbálny off-line protokol.** Expert najprv prípad vyrieši bez rušenia znalostným inžinierom. Riešenie je archivované. Expert vytvorí komentár k svojmu postupu.

3) **POZOROVANIA.** Znalostný inžinier sa stáva pasívnym účastníkom procesu riešenia. Do tejto skupiny patria dve techniky: pozorovanie a verbálny on-line protokol.

- ✓ **Pozorovanie.** Expert vykonáva svoju prácu a znalostný inžinier ho v tichosti pozoruje za účelom odhalenia stratégií, ktoré nie sú expertovi vedome dostupné, motorických schopností a automatických procedúr pre verifikáciu expertného popisu. Znalostný inžinier už musí mať znalosti z domény, aby správne interpretoval údaje a pravidlá. Cieľom pozorovania môžu byť známe úlohy, simulované známe úlohy, úlohy s ohraničenou informáciou, úlohy s ohraničeným spracovaním a ťažké prípady.
- ✓ **Verbálny on-line protokol.** Expert vykonáva svoju prácu a zároveň slovne komentuje svoj postup riešenia teda „myslí nahlas“. V rámci verbálneho on-line protokolu rozoznávame samoreport a tieňový report. Pri **samoreporte** expert súčasne rieši problém a poskytuje informácie o jednotlivých krokoch riešenia. Expert má niekedy tendencie poskytovať knižné vedomosti. Pri **tieňovom reporte** jeden expert úlohu rieši a druhý expert jeho postup komentuje. Výhodou je, že sú kladené menšie nároky na riešiaceho experta. Na druhej strane, znalosti expertov sa musia zhodovať v dostatočnej miere. Existujú nasledovné modifikácie techniky samoreportu. V metóde kritickej odozvy expert komentuje iba niektoré – kritické časti úlohy. V metóde periodickej odozvy sa stanoví časový interval, po uplynutí ktorého expert preruší svoju prácu a okomentuje, čo do toho času robil. V metóde prerušení expert preruší riešenie úlohy, keď je požiadaný znalostným inžinierom o vysvetlenie svojej činnosti, pretože znalostný inžinier prestal niečomu rozumieť.

4) **ZÍSKAVANIE POJMOV** je metóda používaná v počiatkových fázach návrhu BZ na odhalenie štruktúry oblasti. Pojmy sa môžu získavať rôznymi spôsobmi:

- ✓ **Priamym získavaním pojmov**, pričom sa môže získavať zoznam pojmov, zoznam krokov alebo zoznam kapitol. V technike zoznam pojmov je expert požiadaný, aby vymenoval všetky kritické pojmy oblasti. V technike zoznam kapitol expert vymenuje všetky kroky, ktoré používa pri riešení rôznych úloh. V technike zoznam kapitol expert vymenuje názvy kapitol a podkapitol hypotetickej knihy.
- ✓ **Landeringom**, metódou zameranou na zisťovanie hierarchického usporiadania medzi pojmami, generovaním nadriadených a podriadených pojmov a atribútov. Nehodí sa na nehierarchické oblasti.

- ✓ **Triádové porovnávanie** je používané hlavne v spojitosti s repertoárovou sieťou resp. s nepriamymi technikami. Repertoárová sieť bude opísaná v odpovedi na otázku č. 62. Expert vytvára z pojmov oblasti trojice. Zvolí pojem, ktorý sa najviac odlišuje od ostatných pojmov. Potom charakterizuje ostatné dva pojmy tak, aby ich odlišil od tretieho. Charakteristika má tvar vlastnosti, resp. viacerých vlastností.
- ✓ **Klasifikácia.** Expert triedi objekty do skupín na základe ich príbuznosti. Nie je obmedzovaný počtom skupín, alebo počtom objektov v skupine. Často pomenováva skupiny po skončení triedenia, čo je obdoba zhlukovania.

5) **ŠPECIÁLNE TECHNIKY** nie sú všeobecne použiteľné. Často sú viazané na nejaký konkrétny spôsob reprezentácie znalostí. Predstavujú výhodnú skratku. Ich nevhodné použitie vedie ku skresleným znalostiam. Patria sem:

- ✓ **Analýza rozhodnutí**, ktorá sa zameriava na vyhodnocovanie alternatív. Vyhodnocuje sa kedy sa problém vyskytuje, aké sú jeho parametre, uskutočňuje sa váženie faktorov a ich kombinovanie, zvažujú sa pravdepodobné následky alternatívnych rozhodnutí. Riešením je výber rozhodnutia z množiny aplikovateľných rozhodnutí.
- ✓ **Rozklad cieľov.** Znalostný inžinier stanoví cieľ a expert popisuje, aké úlohy je potrebné vykonať, alebo ktoré podmienky je potrebné splniť. Ide vlastne o postupný rozklad problému na podproblémy.
- ✓ **Systematické spájanie symptómov a diagnóz.** Pri tejto metóde sa predpokladá, že sú k dispozícii zoznamy všetkých diagnóz a všetkých symptómov. Systematicky sa preberajú všetky možné kombinácie porúch voči symptómom a expert sa vyjadruje k ich zmysluplnosti.

58. Aké sú to nepriame metódy získavania znalostí?

Nie vždy má expert prístup k svojim mentálnym procesom. Robí závery, ale nevie exaktne popísať spôsob akým to robí. V takom prípade je vhodné použiť nepriame metódy, v ktorých expert vyjadruje svoje znalosti nepriamo. Znalostný inžinier tak získa to, čo musí vedieť, aby sa správne rozhodoval pri riešení úloh domény.

Pri nepriamych technikách spravidla vychádzame z existujúcej množiny pojmov. Hľadáme podobnosť medzi doménovými pojmami, vyberáme reprezentatívnu množinu pojmov. Existujú rôzne nepriame techniky s rôznou reprezentáciou a rôznymi obmedzujúcimi predpokladmi. Konkrétna metóda sa vyberá na základe „interview“ s expertom. Vo všeobecnosti má nepriame získavanie znalostí nasledovné etapy:

- ✓ **Získavanie dát** sa uskutočňuje prostredníctvom odhadov podobnosti pojmov. Tieto odhady majú formu matice podobnosti. Matica podobnosti spravidla reprezentuje podobnosť medzi dvojicami tej istej skupiny pojmov. Riadky a stĺpce matice reprezentujú tie isté pojmy. Je možný aj iný prístup, pri ktorom jedna skupina pojmov slúži na vyjadrenie inej skupiny pojmov. Teda riadky a stĺpce reprezentujú rôzne doménové pojmy. Maticová reprezentácia umožňuje kombinovať dáta, resp. vedomosti viacerých expertov.
- ✓ **Štruktúrálna analýza.** Jej cieľom je získať predstavu o štruktúrálnych vzťahoch medzi pojmami. Štruktúrálny vzťahy môžu mať podobu hierarchického usporiadania, alebo typológie vzťahov medzi dvojicami pojmov. Umožňuje zistiť, či medzi nejakou skupinou pojmov existuje nejaký vzťah, odrazový mostík k popisu a charakterizácii tohto vzťahu expertom.
- ✓ **Odvodzovanie kompilovaných znalostí.** Štruktúrálny znalosti sa nedajú priamo využiť ako procedurálne znalosti pre riešenie úloh. Z toho vyplýva potreba získania produkčných pravidiel, ktoré kompilujú zložité závislosti medzi pojmami do jednoduchých asociácií – pravidiel.

Nepriame metódy sú podobné metódam strojového učenia, ale nevychádzajú z typických príkladov. K metódam nepriameho získavania znalostí patria:

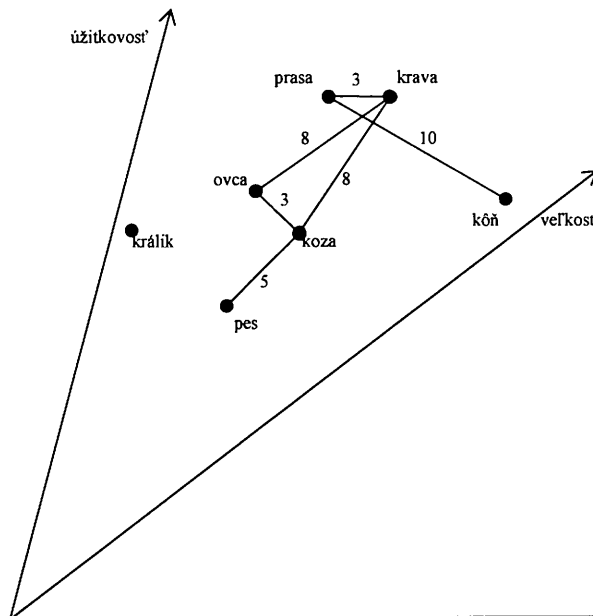
- ✓ Mnohorozmerné škálovanie
- ✓ Johnsonovo hierarchické zhlukovanie
- ✓ Repertoárová sieť.

59. Čo je to Mnohorozmerné škálovanie?

Mnohorozmerné škálovanie je nepriama metóda získavania znalostí, ktorá škálu pojmov umiestňuje do mnohorozmerného priestoru, v ktorom každý rozmer predstavuje nejakú vlastnosť, na základe ktorej sa jednotlivé pojmy odlišujú. Metóda vychádza z matice podobnosti dvojíc pojmov a generuje rozloženie týchto pojmov v k – rozmernom priestore. Metóda je vhodná pre odкрытие zhlukov pojmov a relácií podobností. Postupuje sa nasledovne. Expert je požiadaný, aby vyjadril podobnosť dvojíc pojmov číselne. Miera podobnosti je zvyčajne diskretná a nezáporná, z intervalu $\langle 0, m \rangle$, kde 0 znamená totožnosť a m maximálnu nepodobnosť. Teda expert vymenuje charakteristické pojmy oblasti a vyplní polmaticu podobnosti. Najprv si zvolí referenčnú dvojicu, podľa ktorej určí podobnosť ostatných dvojíc. Pre neporovnateľné dvojice určí podobnosť vo forme nepodobnosti – m . Nasleduje príklad takejto polmatice.

	koza	krava	ovca	prasa	kôň	pes	králik
koza	0	8	3	8	7	5	5
krava		0	8	3	8	14	11
ovca			0	2	7	9	4
prasa				0	10	13	10
kôň					0	11	12
pes						0	5
králik							0

Získané hodnoty polmatice sa zobrazia znalostným inžinierom v mnohorozmernom priestore v takej forme, aby hodnota podobnosti odpovedala vzdialenosti medzi bodmi. Pre vyššie uvedenú polmaticu podobnosti môže mať zobrazenie pojmov v dvojrozmernom priestore nasledovný tvar.



Pre voľbu dimenzionality cieľového priestoru existuje empirická heuristika: čím viac dimenzií, tým menšia chyba rozmiestnenia pojmov. Pri zvolenej dimenzionalite existuje niekoľko rôznych rozložení pojmov. Znalostný inžinier vyberie najvhodnejšie rozloženie, pre ktoré potom expert navrhne polohu osí a ich možné označenie. Osí odrážajú vlastnosti, pozdĺž ktorých sa pojmy menia.

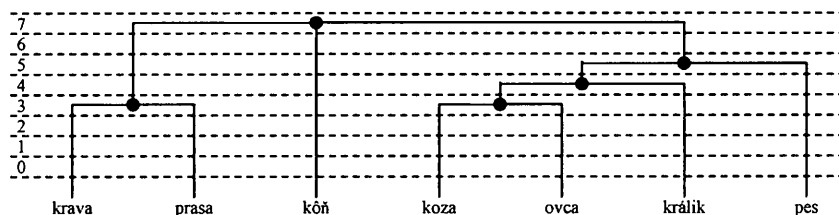
Metóda má jednoduché použitie, ale objavovanie dimenzií a ich interpretácia sú obtiažne. Plusom tejto metódy je možné objavenie nových neobvyklých zhlukov, resp. nových podobností, ktoré po objasnení expertom môžu viesť k novým znalostiam, teda novým doteraz nepoužívaným pravidlám.

60. Ako funguje Johnsonovo mnohorozmerné zhlukovanie?

Johnsonovo mnohorozmerné zhlukovanie vytvára zhluky pojmov podľa príbuznosti, vychádzajúc z polmatice podobností, podobne ako mnohorozmerné škálovanie. Podobnosť sa v tejto metóde interpretuje ako úroveň abstrakcie, na ktorej možno považovať dané pojmy za rovnaké, teda je možné ich zaradiť do toho istého zhluku. Je to čosi ako výška, v ktorej sa pojmy stávajú členmi tej istej kategórie. Postupuje sa nasledovne. Vytvorí sa polmatica podobnosti a tá sa postupne modifikuje – znižuje, pretože pojmy, ktoré sú navzájom najpodobnejšie, sa spájajú do jedného zhluku, ktorý v ďalšom vystupuje ako jeden pojem. Napríklad modifikáciou polmatice uvedenej pri predchádzajúcej metóde mnohorozmerného škálovania dostaneme nasledovnú polmaticu.

	koza+ovca	krava+prasa	kôň	pes	králik
koza+ovca	0	7	7	5	4
krava+prasa		0	8	13	10
kôň			0	11	12
pes				0	5
králik					0

Podobnosti novovytvorených zhlukov sa určia ako priemer, minimum alebo maximum hodnôt podobností každého člena novovytvoreného zhluku voči danému pojmu. Algoritmus metódy končí vytvorením hierarchickej reprezentácie podobností všetkých pojmov. Nasleduje príklad takejto hierarchickej reprezentácie podobností pojmov z vyššie uvedenej polmatice podobnosti.



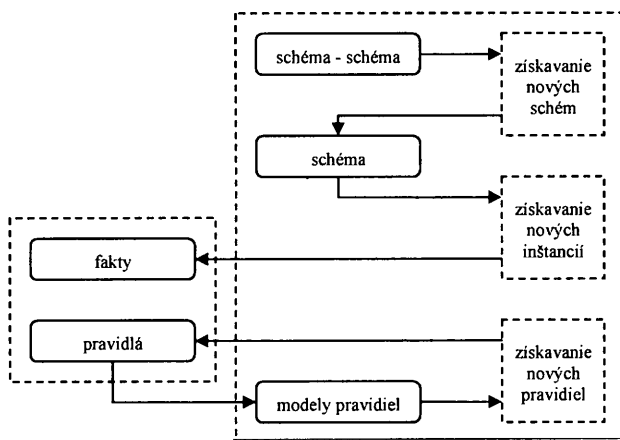
Metóda vychádza z predpokladu, že pojmy sú navzájom rozlíšiteľné pomocou diskretných vlastností, a teda môže byť použitá aj pre skupinu heterogénnych pojmov. Výhodou tejto metódy je, že znalostnému inžinierovi stačí ceruzka a papier. Nevýhodou je vysoký počet vyžadovaných podobností ako aj skutočnosť, že výsledok silne závisí od použitej operácie pre výpočet nových podobností. Rôzne operácie vedú k rôznym hierarchickým zhlukom.

61. Ako sa uskutočňuje Priamy prenos znalostí?

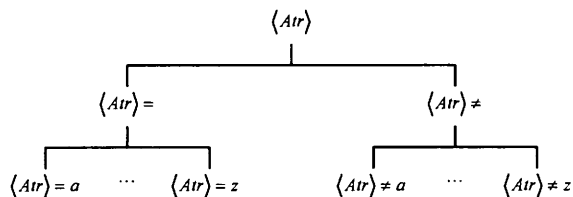
Priamy prenos znalostí uskutočňuje priamo expert bez pomoci znalostného inžiniera. Je to jeden z troch možných spôsobov získavania znalostí spolu so strojovým učením a klasickou cestou. Klasická cesta predstavuje všetky priame a nepriame metódy získavania znalostí od experta.

Teda pri priamom prenose znalostí vytvára BZ expert sám v príslušnom editore a dohodnutom formáte. Tento editor je súčasťou inteligentného modulu priameho získavania znalostí od experta. Inteligentný modul získavania znalostí od experta by mal rozumieť znalostiam v syntaktickom zmysle ale aj odhaliť sémantické chyby, mal by navádzať experta pri vkladaní znalostí a určovať, ktorá časť oblasti je slabo pokrytá znalosťami. Ďalej by mal automaticky udržiavať konzistenciu znalostí. Ide o zložitý a nie bežný spôsob získavania znalostí.

Pokusom o takýto typ systému je systém TEIRESIAS (Berka, 1994), ktorý predstavoval modul získavania znalostí pre znalostný systém EMYCIN. Používa metaznalosti, pretože vytvára nielen BZ, ale aj znalosti o reprezentácii znalostí. Metaznalosti o znalostiach vytvárajú očakávania, ako by mala nová znalosť vyzerieť. Na nasledovnom obrázku je uvedená schéma činnosti systému TEIRESIAS.



Rozlišujú sa dva odlišné typy získavania znalostí: získavanie nových pojmov, resp. konceptov a získavanie vzťahov medzi existujúcimi konceptmi. Pri **získavaní vzťahov medzi konceptmi** vytvára systém vnútorné modely pravidiel. Modely reprezentujú „typické pravidlá“ v danej podmnožine a sú vytvárané zovšeobecnením pravidiel tejto podmnožiny. Expert nemá potuchy o tom, že sú vytvárané nejaké modely pravidiel. Nasledovný obrázok ilustruje stromovú štruktúru, do ktorej je organizovaná množina vytváraných modelov.



62. Čo je to Repertoárová sieť?

Repertoárová sieť je metóda, ktorá nám umožní z popisu vzťahov medzi pojmami generovať pravidlá pre BZ znalostného systému. Táto metóda je založená na teórii osobných konštruktov. Pod pojmom „konštrukt“ sa rozumie séria bipolárnych vlastností. Táto metóda bola navrhnutá v roku 1995 psychológom Georgom Kellym, ktorý predpokladal, že pomocou konštruktov je človek schopný vyjadriť svoju vnútornú štruktúru pojmov. Na základe tejto štruktúry potom klasifikuje svoje okolie, predvída budúce javy a riadi svoje jednanie. Kelly predpokladal, že problémy jeho pacientov vysporiadať sa s okolitým svetom, sú spôsobené nekonzistenciou v ich sieti konštruktov.

Každý konštrukt je reprezentovaný dvoma opačnými pólmi: ľavým a pravým pólom. Miera príslušnosti pojmu ku konštruktovi leží medzi pólmi konštruktov. V nasledovnej tabuľke sú uvedené konštrukty: veľkosť, úžitkovosť, rýchlosť ako aj miera príslušnosti pojmov: pes, kôň, prasa, ovca, králik k týmto konštruktom.

	ľavý pól	pes	kôň	prasa	ovca	králik	pravý pól
veľkosť	veľká	1	5	3	2	1	malá
úžitkovosť	veľká	1	2	5	3	3	malá
rýchlosť	veľká	4	5	2	2	3	malá

Konštrukty, pojmy aj ich mieru príslušnosti ku konštruktom navrhuje expert. Takto nepriamo vyjadruje svoje vedomosti z danej oblasti. Miera príslušnosti je spravidla stupňovitá a počet stupňov je spravidla nepárny, aby existovala stredná hodnota. Konštrukt reprezentuje ohodnotenie pojmov pozdĺž nejakej dimenzie, a to celočíselnou váhou od 1 po N , kde N je nepárne číslo spravidla rovné piatim. 1 – znamená, že pre daný pojem platí určite pravý pól. N

– znamená, že pre daný pojem platí určite ľavý pól. Stredná hodnota je $S = \frac{n+1}{2}$. Ak element nie je ohodnotený, potom konštrukt nie je zmysluplne použiteľný. Výber konštruktov realizuje spravidla expert, ale môže to robiť aj znalostný inžinier, a to metódou triádového porovnávania. Základná sada konštruktov má počet (spravidla tri) rovný počtu pojmov (resp. elementov) zmenšeného o dva.

Podstata triádového porovnávania spočíva v hľadaní takej vlastnosti, ktorá jednoznačne vyčlení z trojice pojmov jeden pojem. Ide o porovnávanie jedného elementu voči zvyšným dvom. Napríklad:

- {králik, pes, kôň} → veľkosť (kôň)
- {pes, kôň, prasa} → úžitkovosť (prasa)
- {kôň, prasa, ovca} → rýchlosť (kôň).

Repertoárová sieť sa analyzuje pomocou

- ✓ podobnosti pojmov
- ✓ podobnosti konštruktov
- ✓ implikačnej analýzy.

Výsledky analýzy sú predkladané expertovi na posúdenie potreby modifikácie Repertoárovej siete. Spravidla je modifikácia žiaduca pri prekročení prahových hodnôt podobnosti pojmov a konštruktov a pri výskyte dvojznačných vzťahov v implikačnej analýze. Modifikácia sa spravidla uskutočňuje zmenou hodnôt príslušností, dopĺňaním a vypúšťaním pojmov, resp. konštruktov, zlučovaním konštruktov a pod. Ak je expert spokojný, resp. ak je repertoárová sieť vyvážená, potom sa zo siete generujú produkčné pravidlá pre BZ znalostného systému.

63. Ako sa určuje podobnosť pojmov a konštruktov?

- 1) **Podobnosť pojmov:** Dva pojmy sú si najpodobnejšie (100%) vtedy, keď stĺpce oboch pojmov obsahujú rovnaké hodnoty a najmenej podobné (0%) vtedy, keď pre každý konštrukt v jednom stĺpci je hodnota 1 a v druhom N . Pre podobnosť pojmov P_k a P_l platí:

$$P_p = 100 - \frac{100}{N-1} \frac{1}{PK} \sum_{j=1}^{PK} |v_{jk} - v_{jl}|, \text{ kde } PK$$

je počet konštruktov, v_{jk} je váha k -teho pojmu a j -teho konštruktu a v_{jl} je váha l -teho pojmu a j -teho konštruktu. Tak napríklad podobnosť pojmov kôň a prasa vo vyššie uvedenej repertoárovej sieti sa určí nasledovne:

$$P_p = 100 - \frac{100}{4} \cdot \frac{1}{3} (|5-3| + |2-5| + |5-2|) = 100 - 25 \cdot \frac{(2+3+3)}{3} = 100 - 25 \cdot \frac{8}{3} = 33,3\%.$$

- 2) **Podobnosť konštruktov:** Tu vystupuje navyše problém určenia, ktoré póly konštruktov patria k sebe. Napríklad, či k sebe patria malá veľkosť a veľká rýchlosť alebo veľká veľkosť a veľká rýchlosť. Alebo či k sebe patria veľká veľkosť a malá rýchlosť alebo malá veľkosť a malá rýchlosť. Najväčšia odlišnosť medzi konštruktami je $[(N+1)/2 - 1] \cdot PP$, kde PP je počet pojmov. Ak je pri určitom priradení pólov väčšia odlišnosť medzi konštruktami, stačí prevrátiť póly. Hodnoty konštruktov s prevrátenými pólmi sa nazývajú reverzné hodnoty. Pre podobnosť konštruktov C_k a C_l a reverznú hodnotu v_{lj}^r platí:

$$P_K = 100 - \frac{100}{\frac{N+1}{2} - 1} \cdot \frac{1}{PP} \min \left(\sum_{j=1}^{PP} |v_{kj} - v_{lj}|, \sum_{j=1}^{PP} |v_{kj} - v_{lj}^r| \right).$$

Tak napríklad pre podobnosť konštruktov veľkosť a úžitkovosť platí:

$$\begin{aligned} P_K &= 100 - \frac{100}{2} \cdot \frac{1}{5} \min ([|1-1| + |5-2| + |3-5| + |2-3| + |1-3|], [|1-5| + |5-4| + |3-1| + |2-3| + |1-3|]) = \\ &= 100 - 10 \min(8,10) = 100 - 80 = 20\% \end{aligned}$$

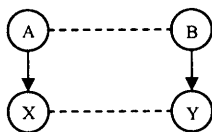
64. Čo je to Implikačná analýza?

Implikačná analýza je metóda štrukturálnej analýzy, ktorá sa môže použiť aj ako samostatná metóda, nie len v rámci repertoárovej siete. Nevýhodou tejto metódy je vysoký počet vyžadovaných podobností. Implikačná analýza reprezentuje špeciálny typ analýzy dát v repertoárovej sieti. V pôvodnom chápaní bola implikačná analýza cieľom, pretože sa analyzoval vnútorný systém konštruktov pacienta. V znalostnom inžinierstve sa stáva prostriedkom kontroly znalostí uložených v repertoárovej sieti. Implikačná analýza odкрýva závislosti medzi jednotlivými konštruktami v tvare implikácií. Ak sú dané dva rôzne konštrukty s pólmi $A - B$ a $X - Y$, potom medzi nimi môžu existovať nasledovné vzťahy:

- ✓ **paralelný vzťah** $A \Rightarrow X$ a $B \Rightarrow Y$, ktorý je prípustným vzťahom v repertoárovej sieti.

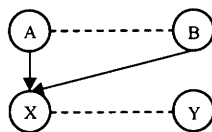
Príkladom sú konštrukty:

- milovaný – príjemný
- nenávidený – nepríjemný



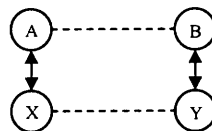
- ✓ **ortogonálny vzťah** $A \Rightarrow X$ a $B \Rightarrow X$, ktorý je taktiež prípustným vzťahom v repertoárovej sieti. Príklad:

- zamestnaný – nezamestnaný
- prijíma potravu – neprijíma potravu



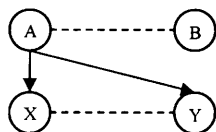
- ✓ **recipročný vzťah** $A \Rightarrow X$ a $X \Rightarrow A$, ako aj $B \Rightarrow Y$ a $Y \Rightarrow B$, ktorý je zbytočným vzťahom, keďže ide o ekvivalentnosť konštruktov. Je lepšie ho z repertoárovej siete odstrániť. Príklad:

- nervózny – pokojný
- napätý – uvoľnený

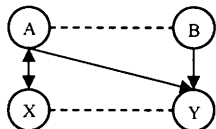


- ✓ **dvojznačný vzťah** $A \Rightarrow X$ a $A \Rightarrow Y$, ktorý z repertoárovej siete „musíme“ odstrániť. Nie je prípustné, aby z jedného predpokladu bolo možné odvodiť viac záverov. Subjekt má v tomto prípade nekompletnú predstavu o rozdiel medzi danými konštruktami. Príklad:

- realizmus – idealizmus
- žiadúci – nežiadúci



Ak sa vyskytne viac implikačných vzťahov v rámci jedného prípadu, o ďalšom postupe budeme rozhodovať podľa najnepriaznivejšieho vzťahu. Príklad:



Vzťahy, ktoré sa v tomto príklade vyskytujú sú nasledovné: paralelný vzťah, recipročný vzťah a dvojznačný vzťah. Keďže v tomto príklade sa vyskytuje dvojznačný vzťah, tak musíme repertoárovú sieť zmeniť.

Implikačná analýza je dosť zdĺhavá, keďže pre dva konštrukty je potrebné overiť 8 implikácií. Sila každej jednotlivej implikácie sa určí podľa nasledovnej tabuľky:

Sila		Implikácia A→X				
		X				
		1	2	3	4	5
A	4	0,25	0,50	0,75	1,00	1,00
	5	0,00	0,25	0,50	0,75	1,00

Implikácia sa určuje iba medzi ľavými pólmi konštruktov. Pre pravé strany sa uvažujú reverzné hodnoty. Výsledná sila implikácie sa určí ako priemer za jednotlivé stĺpce. Príklad:

veľká rýchlosť ⇒ veľká veľkosť

4 ⇒ 1 → 0,25

5 ⇒ 5 → 1,00

$1,25/2 = 0,625$

Výsledná hodnota sily implikácie „veľká rýchlosť ⇒ veľká veľkosť“ sa rovná 0,625.

Sila implikácie musí prekročiť prahovú hodnotu, aby daná implikácia platila.

65. Ako sa generujú pravidlá z repertoárovej siete?

Repertoárová sieť môže byť použitá pre generovanie znalostí v tvare produkčných pravidiel. Ku každej dvojici konštrukt – pojem je možné vygenerovať dve pravidlá:

AK ľavá strana konštrukt POTOM pojem
 AK pravá strana konštrukt POTOM pojem

Ak označíme konštrukty K_i a pojmy E_j a váhu l -teho pravidla p_l , potom môžeme tieto pravidlá zapísať aj takto:

AK ľavý pól K_i POTOM $E_j(p_1)$
 AK pravý pól K_i POTOM $E_j(p_1)$,

alebo: AK K_i POTOM $E_j(p_1, p_2)$

Pomocou repertoárovej siete sa nedá tvoriť hierarchicky štruktúrovaná BZ. Vyniknutá BZ bude iba dvojstupňová. Vygenerované pravidlá budú mať priradenú váhu – mieru platnosti, ktorá sa určuje podľa vzorca:

$$p = F_1 F_2 F_3 .$$

Jednotlivé faktory sa určujú nasledovne:

- ✓ **Faktor F_1** reflektuje spôsob narábania s neurčitou. Predstavuje spôsob skladania príspevkov od jednotlivých pravidiel podľa modelu, ktorý používa ZS EMYCIN. Ak skladáme dve pravidlá s váhami x_1 a x_2 , výsledkom bude:

$$x_1 + x_2 - x_1 x_2 = x_1 + x_2(1 - x_1) + 1 - 1 = 1 - (1 - x_1) + x_2(1 - x_1) = 1 + (1 - x_1)(x_2 - 1) = 1 - (1 - x_1)(1 - x_2)$$

Ak skladáme n neurčitých svedectiev do jednej váhy w výsledného pravidla:

$$PP : P \rightarrow Z (w) ,$$

a ak v rámci určitej tolerancie platí, že všetky príspevky sú rovnaké, potom:

$$1 - (1 - w)^n = 0,95$$

$$F_1 = w = 1 - \sqrt[n]{1 - 0,95} .$$

- ✓ **Faktor F_2** predstavuje váhu pojmu vzhľadom na nejaký konštrukt. Táto váha je číslo z diskrétného intervalu $\langle 1, N \rangle$. Je potrebné pretransformovať ho na interval $\langle -1, 1 \rangle$. Z toho dôvodu vzťah pre výpočet faktoru F_2 je:

$$F_2 = \frac{x - \frac{N+1}{2}}{N-1} - \text{pre ľavý pól konštrukt, a}$$

$$F_2 = -\frac{x - \frac{N+1}{2}}{N-1} - \text{pre pravý pól konštrukt.}$$

- ✓ **Faktor F_3** reprezentuje dôležitosť konštrukt: $F_3 = \frac{d}{d_{\max}}$, kde d je dôležitosť aktuálneho konštrukt a d_{\max} je maximálna dôležitosť konštrukt. Čím väčšia je dôležitosť konštrukt d , tým vyššia je váha pravidla, v ktorom sa vyskytuje.

66. Ako je možné vylepšovať znalosti?

Budovanie BZ neznamena iba neustále rozširovanie znalostí vkladaním nových znalostí. Niekedy je potrebné spätne sa vrátiť k BZ za účelom opravy alebo zjemňovania znalostí. Zjemňovanie znalostí môže mať formu:

- ✓ **dopĺňania znalostí.** Nové znalosti sú zaraďované tak, aby nadväzovali na tie znalosti, ktoré už v BZ sú.
- ✓ **redukcie znalostí.** V BZ sa nachádzajú okrem globálne platných zákonitostí aj náhodné súvislosti, resp. lokálne platné zákonitosti. Náhodné súvislosti sa do BZ môžu dostať pri použití niektorej metódy strojového učenia. Znalosti, ktoré predstavujú náhodné súvislosti, sú z BZ odstraňované. Ponechané sú iba globálne platné zákonitosti.
- ✓ **modifikácie znalostí.** Celkový objem znalostí sa zachováva. Znalosti sa spresňujú, aby lepšie odrážali pomery v doméne t.j. predmetnej oblasti. Modifikáciu znalostí je možné uskutočniť buď zmenou váh jednotlivých pravidiel, alebo zmenou počtu podmienok v pravidlách. Zmena počtu podmienok je najviac využívaná v prípade, keď modifikované pravidlá boli získané strojovým učením z množiny trénovacích príkladov. Tieto pravidlá sa ešte overujú na testovacej množine. Nasleduje zosúladenie BZ na základe dodatočne získaných informácií.

Príkladom dopĺňania znalostí je postup, pri ktorom sa nedopĺňajú celkom nové znalosti, ale generujú sa rôzne varianty pravidiel, ktoré už v BZ sú, ale nedávali by dobré výsledky, keby sa používali veľmi striktné. Napríklad, keď lekár popisuje zoznam symptómov nejakej choroby, neznamená to, že u každého pacienta trpiaceho danou chorobou sa musia vyskytnúť všetky symptómy. To znamená, že dané pravidlo by mohlo fungovať aj keby niektoré podmienky v ňom boli negované. Spomenutý postup sa používa v prípade, keď chápanie znalostí expertom je širšie, ako jej explicitné vyjadrenie. Predpokladajme, že znalostný inžinier získa od experta nasledovné pravidlo:

AK p_1 & p_2 & ... & p_n POTOM Z

Expert explicitne predpokladá, že záver môže byť splnený aj vtedy, ak niektorá podmienka splnená nie je. Preto by BZ mala byť doplnená o pravidlá, ktoré niektoré podmienky neuvažujú. Postup je nasledovný:

- 1) **Zoradenie podmienok podľa dôležitosti.** Podmienky rozdelíme na dominantné a nedominantné. **Dominantné** sú tie podmienky, bez ktorých splnenia záver neplatí. **Nedominantné** sú tie podmienky, bez ktorých splnenia záver môže platiť. Každéj nedominantnej podmienke priradíme číselnú hodnotu, ktorá vyjadruje dôležitosť tejto podmienky. Potom nedominantné podmienky zoradíme podľa dôležitosti, teda priradenej číselnej hodnoty.
 - najmenej významné 1
 - významnejšie 2
 - atď.
 - najvýznamnejšie m (počet nevýznamných podmienok)

- 2) **Generovanie doplnkových pravidiel.** Každá podmienka ktorá nie je dominantná, môže byť negovaná. Ak máme m nedominantných podmienok, môžeme generovať $2^m - 1$ podmienok.
- 3) **Usporiadanie doplnkových pravidiel.** Každé doplnkové pravidlo sa ohodnotí číslom, ktoré predstavuje sumu číselných ohodnotení z kroku 1) všetkých nedominantných podmienok, ktoré sú v danom pravidle negované. Táto číselná hodnota odráža nakoľko sa doplnkové pravidlo líši od pôvodného pravidla. Doplnkové pravidlá sa zoradia podľa hodnôt od najnižšej po najväčšiu.
- 4) **Určenie hraničného pravidla.** Vystáva otázka, kedy je odlišnosť doplnkového pravidla od pôvodného pravidla. Odpoveď na túto otázku dá expert tým, že vyberie hraničné pravidlo. Pravidlá s nižším ohodnotením ako má hraničné pravidlo sa do BZ nezaradia.
- 5) **Určenie váh.** Pre akceptované doplnkové pravidlá je potrebné určiť váhy, s ktorými budú zaradené do BZ. Expert určí váhu hraničného pravidla. Váhy ostatných pravidiel sa určia z lineárnej závislosti:

$$w = \frac{W - w_{hran}}{\sum_{i \in NEG(hran)} d_i} \sum_{i \in NEG} d_i$$

kde: W - je hodnota pôvodného pravidla

w_{hran} - je váha hraničného pravidla

$\sum_{i \in NEG(hran)} d_i$ - suma hodnôt negovaných podmienok pre hraničné pravidlo

$\sum_{i \in NEG} d_i$ - suma hodnôt negovaných podmienok pre aktuálne pravidlo

Nasleduje príklad, ktorý bude ilustrovať jednotlivé kroky postupu dopĺňania znalostí. Predpokladajme, že pôvodné pravidlo:

AK $p_1 \& p_2 \& p_3$ POTOM $z(0,8)$

- 1) Zoradenie podmienok podľa dôležitosti: expert určil, že p_3 je dominantná podmienka
 p_1, p_2 sú nedominantné podmienky
 p_1 je dôležitejšia ako p_2 , preto ich ohodnotenie je určené nasledovne:

$$\text{hodnota}(p_1) = 2 = d_1$$

$$\text{hodnota}(p_2) = 1 = d_2$$

- 2) Generovanie doplnkových pravidiel:

AK $\neg p_1 \& p_2 \& p_3$ POTOM z $d = 2$

AK $p_1 \& \neg p_2 \& p_3$ POTOM z $d = 1$

AK $\neg p_1 \& \neg p_2 \& p_3$ POTOM z $d = 3$

3) Usporiadanie doplnkových pravidiel:

AK $p_1 \& \neg p_2 \& p_3$ POTOM z $d = 1$

AK $\neg p_1 \& p_2 \& p_3$ POTOM z $d = 2$

AK $\neg p_1 \& \neg p_2 \& p_3$ POTOM z $d = 3$

4) Určenie hraničného pravidla: expert určí hraničné pravidlo

AK $\neg p_1 \& p_2 \& p_3$ POTOM z $d = 2$

a určí mu váhu napríklad $0,4$.

5) Určenie váh. Do bázy znalostí pôjdu okrem pôvodného pravidla aj doplnkové pravidlá:

AK $p_1 \& \neg p_2 \& p_3$ POTOM z $d = 1$ s váhami: $0,6$

AK $\neg p_1 \& p_2 \& p_3$ POTOM z $d = 2$ $0,4$

Váhu druhého, hraničného pravidla zadal expert. Váha prvého pravidla sa vypočíta:

$$w_1 = 0,8 - \frac{0,8 - 0,4}{2} \cdot 1 = 0,6.$$

67. Načo slúži vysvetľovací mechanizmus?

Existujú dva hlavné dôvody, prečo by mal byť znalostný systém vybavený vysvetľovacím mechanizmom. Po prvé, nedá sa očakávať, že používateľ bude poznať všetky možnosti práce so zložitým znalostným systémom. Preto potrebuje vysvetlenie, čo sa vlastne v systéme robí. Po druhé, znalostné systémy sa najčastejšie používajú v riešení slabó formalizovateľných úloh, ktoré nemajú algoritmické riešenie, preto sa používateľ potrebuje utvrdiť v dôveryhodnosti výsledku a použitých metód. Vysvetľovanie je charakterizované:

- 1) cieľom vysvetľovania
- 2) objektom vysvetľovania
- 3) spôsobom vysvetľovania
- 4) adresátom vysvetľovania.

Cieľ vysvetľovania vyplýva zo spôsobu použitia znalostného systému. Spôsob použitia je zase daný typom používateľa, ktorý systém používa. Súvislosti medzi typom používateľa, spôsobom použitia a cieľom vysvetľovania sú ilustrované nasledovnou tabuľkou.

Spôsob použitia expertného systému	Cieľ vysvetľovania
získať riešenie	čo riešenie znamená objasniť základné súvislosti objasniť vstupy a výstupy
znižovať náročnosť získania riešenia zvýšiť kvalitu riešenia	objasniť spôsob získania riešenia
chce si objasniť spôsob riešenia problému	objasniť spôsob získania riešenia potrebuje obsah celej bázy znalostí v zrozumiteľnej forme
ladenie bázy znalostí modifikácia expertného systému	lokalizácia chyby nájsť slabé miesta systému zistiť ako vznikla chyba čo sa na vzniku chyby podieľalo
ladenie bázy znalostí modifikácia expertného systému	ladenie chýb zistiť či je správne reprezentovaná znalosť zistiť či je dobrá inferenčná sieť

Objekt vysvetľovania predstavuje to, čo sa bude objasňovať. Od toho, čo bude objasňované závisí spôsob objasňovania. Okruh toho, čo môže byť objasnené, nezahŕňa všetky znalosti a procedúry. Je potrebné špecifikovať, čo môže byť objasnené a čo nie.

Spôsob vysvetľovania má niekoľko aspektov, a to úroveň detailnosti, jazyk objasnenia a typ objasnenia. Nevyhnutnosť poskytovať objasnenia na rôznych úrovniach detailnosti je spôsobená rôznosťou cieľov používateľov, rôznosťou úrovni ich znalostí ako aj zmenou v čase. Jazykom objasnenia môže byť buď ohraničený prirodzený jazyk, alebo jazyk grafických obrazov. Spravidla sa rozlišuje päť rôznych typov objasnení:

- 1) Kauzálne vysvetľovanie je všeobecne použiteľné, lebo predstavuje príčinné súvislosti. Nieto javu, ktorý by nemal svoje príčiny a nevyvolával by nejaké následky.

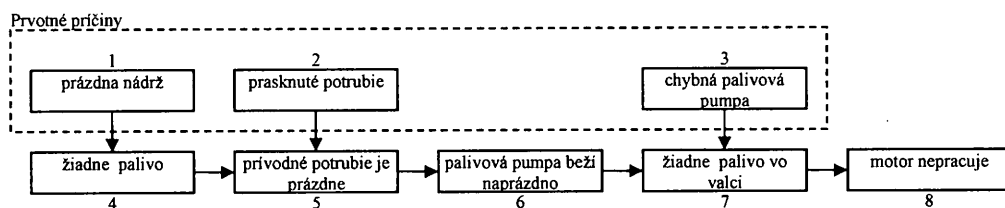
- 2) Objasnenie pomocou zákona stanovuje, v súvislosti s akým zákonom (teóriou, modelom) vznikol alebo prebehol vysvetľovaný jav. Na takéto objasnenie sa možno pozerat' ako na logickú operáciu dedukcie, teda získanie konkrétnych záverov zo všeobecného zákona.
- 3) Funkcionálne vysvetlenie vyjadruje funkciu, účel nejakého procesu. Napríklad: „Mimikry je pre hmyz dôležité preto, aby sa skryl pred nepriateľom.“ V sociálnych vedách je toto objasnenie cieľové, motivované cieľavedomou činnosťou ľudí.
- 4) Štrukturálne vysvetlenie obsahuje opis štruktúry, ktorá zabezpečuje realizáciu funkcií a chod systému ako celku. Z tejto štruktúry používateľ vysleduje spôsob použitia.
- 5) Genetické vysvetlenie sa niekedy označuje názvom historické. Je to objasnenie podmienok, príčin a zákonov, ktoré priviedli systém do daného stavu. Historické preto, lebo vychádza z postupnosti, teda histórie javov, ako idú za sebou.

Adresát vysvetľovania je ten, kto prijíma vysvetlenie. Logika práce znalostného systému a jeho spôsob reprezentácie znalostí nie sú identické reprezentácii u používateľa. Preto je potrebné vysvetlenie v tvare relevantnom otázke používateľa, teda pochopiteľné pre používateľa, týkajúce sa toho, na čo sa pýtal a na odpovedajúcej úrovni abstrakcie a detailnosti. Vysvetlenie by sa malo prispôbiť aj tomu fakt, že kvalifikácia používateľa sa mení s časom. Na začiatku, ako laik, potrebuje detailné vysvetlenie. Neskôr už tak podrobné vysvetlenie nepotrebuje.

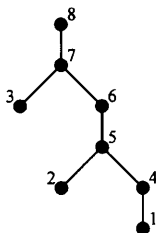
68. Ako delíme jednotlivé vysvetlenia?

Existujú tri základné prístupy k objasňovaniu v reálnych znalostných systémoch:

- 1) **Vopred pripravené vysvetlenia** predstavujú najjednoduchší spôsob. S každým krokom činnosti programu je zviazaný nejaký vysvetľujúci text vo forme textu v prirodzenom jazyku. Príkladom môžu byť chybové hlášky. Nevýhodou tohto prístupu je, že ho možno použiť iba v prípade, keď je možné predvídať všetky otázky používateľa. Systém nemá žiadny konceptuálny model toho, čo oznamuje. Všetky texty sú pre neho na rovnakej úrovni. Texty nemôžu zohľadniť dynamickú podstatu vysvetľovania.
- 2) **Vysvetlenia generované z behu programu.** Používateľ najčastejšie žiada o vysvetlenie, keď je dotazovaný (Vysvetli, prečo sa pýtaš?). Rozoznávame tieto druhy vysvetlení, generovaných z behu programu:
 - ✓ GOAL – Používateľ sa pýta, prečo má odpovedať na otázku v danom listovom uzle. Systém odpovedá tak, že zverejní hypotézu, ktorú práve dokazuje.
 - ✓ TRACE – Systém zverejní celú cestu.
 - ✓ WHY – Systém sa pýta na konkrétny listový uzol, lebo chce vedieť, či platí nadradený uzol ten a ten. Rekurzívnym používaním WHY sa vysvetľuje stále vyššia úroveň až po cieľový uzol (čo odpovedá vysvetlieniu TRACE).
 - ✓ HOW – Je to komplementárny spôsob k WHY. Pri rekurzívnom použití zostupuje o úroveň nižšie až po posledný predpoklad – listový uzol.
 - ✓ WHAT – Uvádza, aké znalosti sa používajú.
 - ✓ WHAT – IF je najzložitejšie vysvetľovanie. Vysvetlí, čo sa stane, ak nejaký uzol neplatí. Teda bude chodiť po konjunkciách a hľadať, ktoré uzly nebudú v dôsledku toho platiť. Taktiež vysvetlí, čo sa stane, ak nejaký uzol platí. Teda bude chodiť po disjunkciách a hľadať, ktoré uzly budú v dôsledku toho platiť.
- 3) **Vysvetľovanie podľa modelu.** Systém má k dispozícii nielen inferenčnú sieť (ktorú používa výlučne na odvádzanie riešení), ale aj kauzálny model znalostí (ktorý používa na formuláciu vysvetlení). Nasleduje príklad takéhoto modelu.



Tomuto kauzálnemu modelu odpovedá nasledovná inferenčná sieť:



69. Ako sa zabezpečujú znalostné systémy?

Tým, že znalostné systémy prešli do komerčného využívania, do popredia sa dostala otázka ich bezpečnosti. Najnebezpečnejšie spôsoby poškodenia znalostného systému sú: zmena jeho obsahu, neoprávnený prístup k niektorým jeho častiam, hlavne BZ, neoprávnený prístup k vonkajším zdrojom informácií, napríklad k externým databázam. Ktoré sú teda slabé miesta znalostného systému, ktoré umožňujú únik informácií:

- ✓ **Používateľské rozhranie** sa zvyčajne navrhuje tak, aby používateľ a čo najmenej zaťažovalo, preto môže predstavovať jednoduchý prostriedok na preniknutie do systému.
- ✓ **Špeciálne typy znalostí**, ktoré nemajú formalizmus produkčných pravidiel, ako napríklad externé procedúry, je potrebné chrániť oddelene.
- ✓ **Vysvetľovanie činnosti systému** umožňuje spätne získať tie znalosti, ktoré boli uložené v báze znalostí a to jednoducho trasovaním.
- ✓ **Databázy**. Používateľ môže prostredníctvom znalostného systému získať oprávnenie k prístupu do nejakej externej databázy, z ktorej znalostný systém čerpá informácie.
- ✓ **Shell**. Znalostné systémy často pracujú v prostredí veľkých „main frame“ pracovných staníc. V tom prípade by sa mohli spoľahnúť na ochranu realizovanú v operačnom systéme. Veľmi málo shellov však poskytuje možnosti ochrany. Zvyčajne sú také jednoduché, že sami otvárajú dvere neautorizovaným používateľom.

Na ochranu znalostných systémov je možné použiť bežné prostriedky zabezpečujúce informačné systémy, ako napríklad **ochrana heslom**, ktorá však nie je veľmi účinná. Avšak špecifiká znalostných systémov vytvárajú podmienky pre vývoj nasledovných špeciálnych metód ochrany:

- ✓ **Numerické informácie v BZ** umožňujú uskutočňovať kontrolné sumy, napríklad váh jednotlivých produkčných pravidiel.
- ✓ **Symbolické informácie v BZ** taktiež umožňujú použitie niektorých numerických metód, napríklad kontrolné sumy počtu slov v pravidlách, počtu pravidiel, alebo počtu podmienok v podmienkovej časti pravidla.
- ✓ **Chránená verzia systému** je podmienená existenciou dvoch verzií systému. Jedna verzia umožňuje plnú manipuláciu so znalosťami. Druhá verzia je určená konečnému používateľovi a umožňuje iba používanie už dotvorenej bázy znalostí.
- ✓ **Rôzne verzie bázy znalostí**. V tomto prípade sú pre zmenu k dispozícii dve verzie bázy znalostí. Jedna na účely odvodzovania a druhá na účely vysvetľovania spôsobu získavania dosiahnutých výsledkov.
- ✓ **Bezpečnostný znalostný systém**. Tento spôsob zabezpečenia je založený na tom, že znalostný systém v sebe obsahuje samostatný malý znalostný systém, ktorý zisťuje, či nedochádza k porušeniu integrity systému a k jeho použitiu neoprávneným používateľom. Ďalej kontroluje konzistentnosť bázy znalostí a taktiež sleduje reakcie používateľa. Systém sám sa suspenduje, keď detekuje nezvyčajné reakcie používateľa (nezvyčajné poradie, pokus o zmenu BZ, viacnásobné chybné zadávanie hesla) alebo nezvyčajné okolnosti (doba prihlásenia po polnoci).

70. Ako je možné vyhodnocovať znalostné systémy?

Znalostný systém sa opakovane vyhodnocuje, resp. testuje v procese tvorby znalostného systému a bázy znalostí (Guida-Tasso, 1994). Vyhodnocovanie v sebe zahŕňa tri rozdielne procesy:

- ✓ **Verifikáciu**, teda kontrolu vnútornej konzistentnosti a úplnosti.
- ✓ **Validáciu**, ktorá stanovuje či znalostný systém uspokojivo realizuje ciele pre ktoré bol navrhnutý.
- ✓ **Prijatie používateľom**, ktorá hodnotí prijateľnosť vytvoreného systému pre konečného používateľa.

Testovanie znalostného systému je jedna z najdôležitejších fáz vývoja. Nejde iba o testovanie už hotového prototypu, ale aj o testovanie v každej fáze (cykle) návrhu. Kedykoľvek je prototyp modifikovaný, musí byť znovu vyhodnotený. Nasleduje podrobnejší popis všetkých troch typov testov znalostných systémov.

VERIFIKÁCIA

Znalosti v znalostnom systéme sú reprezentované systémom produkčných pravidiel. To znamená, že sa narába s logickými implikáciami, logickými výrazmi, logickými axiómami a pod. Preto dominantnými testovacími metódami sú logické metódy. V rámci verifikácie sa testuje formálna správnosť, nie významová správnosť. Testuje sa postupne konzistentnosť bázy znalostí, jej kompletnosť a navrhnuté pravidlá.

- ✓ **Konzistentnosť bázy znalostí** predstavuje nemožnosť toho, aby pravidlá tvrdili niečo rozporné, teda navzájom kontradikčné pravidlá odvodené z tých istých platných dát. Príkladom kontradikčných pravidiel je dvojica pravidiel, z ktorých jedno potvrdí a iné vyvráti platnosť faktu z tých istých podmienok.
- ✓ **Kompletnosť bázy znalostí** je jej schopnosť spracovať každú situáciu, ktorá sa môže vyskytnúť v problémovej oblasti. Ak sa vyskytne nespracovateľná situácia, potom je potrebné doplniť ďalšie pravidlá.
- ✓ **Test na navrhnuté pravidlá.** Pri tomto teste sa vyhľadávajú pravidlá, ktoré už sú zahrnuté iným pravidlom. Tak napríklad, z dvoch nasledovných pravidiel je to druhé nadbytočné, pretože je už zahrnuté v prvom pravidle:

$$b \rightarrow c$$
$$a \& b \rightarrow c.$$

Jedným z existujúcich prostriedkov pre logické testovanie bázy znalostí je **COVER**, ktorý vykonáva nasledovné kontroly bázy znalostí:

- 1) vyhľadáva konfliktné, redundantné a zahrnuté množiny individuálnych pravidiel,
- 2) indikuje konfliktné, redundantné a zahrnuté množiny zreťazených pravidiel,
- 3) vyhľadáva cyklické reťazenie pravidiel,
- 4) nachádza neužitočné pravidlá, ktoré nikdy neboli volané počas reťazenia pravidiel,
- 5) indikuje umŕtvujúce pravidlá, ktoré majú záver a ten sa nenachádza v predpokladoch ani jedného pravidla,
- 6) vyhľadáva nesplniteľné podmienky, ktoré testujú prvok voči hodnote, ktorú prvok nemôže nikdy dosiahnuť
- 7) indikuje chýbajúce pravidlá, ktoré odpovedajú situácii, pre ktorú neexistuje žiadne pravidlo v báze znalostí
- 8) odhaľuje bežné syntaktické chyby

VALIDÁCIA

V rámci validácie sa skúma, či systém rieši úlohu, na ktorú bol určený a ako dobre ju rieši. Skúma sa správnosť z významového hľadiska, čo je náročná úloha. Existuje niekoľko rôznych prístupov k validácii znalostných systémov:

- 1) **Empirický prístup** je dominantný. Spočíva v spúšťaní prototypu na testovacích príkladoch a v ohodnocovaní výsledkov z hľadiska prijateľnosti, presnosti a správnosti.
- 2) **Klasická metóda** je inšpirovaná Turingovým testom. Predpokladajme, že máme sadu riešení od experta a sadu riešení od znalostného systému. Ak tretí subjekt, nezávislý expert, nedokáže odlíšiť riešenia experta od riešení znalostného systému, potom má systém rovnakú výkonnosť ako expert, teda je validný. V reálnych podmienkach je znalostný systém možné testovať voči štandardu. Ak neexistuje, môžeme použiť mienky niekoľkých expertov, ak sa dokážu dohodnúť na tom, čo je správne. Na vyhodnotenie možno použiť logaritmické, kvadratické, lineárne a iné miery. Napríklad, nech Va je miera kvality práce znalostného systému, St_i je riešenie štandardu ES_i a je riešenie

znalostného systému. Potom:
$$Va = \frac{\sum_{i=1}^n (St_i - ES_i)^2}{n}$$
. Uvažujme nasledovný príklad. Znalostný systém predpovedal na tri dni dopredu, že bude pršať na 30%, 60% a 90%. Štatisticky sa preukázalo, že skutočne pršalo prvý a posledný deň, čo ilustruje nasledovná tabuľka a po nej výpočet validačnej miery $Va = 0,287$.

Expertný systém	Štatistika
0,3	1
0,6	0
0,9	1

$$Va = \frac{1}{3} \left[(1 - 0,3)^2 + (0 - 0,6)^2 + (1 - 0,9)^2 \right] = 0,287$$

- 3) **Detektor detekuje** či nejaká hypotéza nastala, alebo nie. Predpokladajme, že máme dve distribúcie riešení: P – distribúcia pozitívnych hypotéz (platia) a N – distribúcia negatívnych hypotéz (neplatia). Potom miera D , je pravdepodobnosť toho, že dve riešenia po jednom z každej distribúcie, budú v obrátenom poradí, t.j., že váha x_p z distribúcie P je nižšia ako váha x_n z distribúcie N . Definícia D je:
$$D = p(x_p < x_n / x_p \in P \ \& \ x_n \in N).$$

PRIJATIE POŽÍVATEĽOM

Prijatie používateľom (User Acceptance) predstavuje použiteľnosť znalostného systému pre konečného používateľa a jeho ochotu používať znalostný systém. Vyhodnocujú sa nasledovné faktory:

- ✓ **Lahkosť učenia** je psychologickým faktorom. Základné koncepty musia byť rýchlo a ľahko pochopiteľné. Ovládanie systému musí byť konzistentné.

- ✓ **Riadenie** je tiež psychologický faktor. Používateľ musí mať dojem, že dialóg je riadený ním. Systém by nemal reagovať nepredvídateľným spôsobom. Používateľ by mal mať možnosť návratu k včasnejším štádiám dialógu.
- ✓ **Stupeň úsilia**, ktorý musí používateľ vynaložiť, aby mohol systém použiť. Je potrebné navrhovať systém tak, aby sa vyhlo každému mentálnemu, alebo fyzickému úsiliu používateľa, ktoré nie je nevyhnutné.
- ✓ **Rýchlosť**. Systém má držať krok s užívateľom. Výsledky musia byť v reálnom čase.
- ✓ **Vstup a výstup** informácií má mať formát, ktorý rešpektuje zvyklosti používateľa.
- ✓ **Odolnosť a korekcia chýb**. Náhodné chyby používateľa by mohol systém korigovať. Pri zadávaní údajov je vhodnejšia možnosť výberu alternatívy.

71. Načo je model používateľa v znalostných systémoch?

Model používateľa môže skvalitniť prácu znalostného systému, pretože systém akoby vedel, čo od neho používateľ očakáva, pretože v priebehu minulých konzultácií získal o používateľovi a jeho spôsobe práce zo systému množstvo informácií. Je to podobné, ako keď medzi sebou komunikujú ľudia. Ľudia získavajú a používajú aj znalosti o svojich partneroch v komunikácii.

Model používateľa je zdrojom znalosti pre znalostný systém o používateľovi vo všetkých aspektoch relevantných pre komunikáciu systému s používateľom. Model používateľa je vytváraný inkrementálne. Jednotlivé položky modelu sú ukladané, modifikované a mazané tak, aby bola zabezpečená konzistentnosť modelu.

Model používateľa môže byť:

- ✓ osobný model používateľa ako individua
- ✓ generalizovaný model typického používateľa
- ✓ model človeka, ktorý práve interaguje so systémom.

Model používateľa môže mať dve formy:

- ✓ **Implicitná** forma modelu obsahuje predpoklady návrhára systému o používateľovi,
- ✓ **Explicitná** forma modelu. Informácie o používateľovi sú uložené v samostatnej časti systému, sú kódované v reprezentáčnom jazyku. Táto forma podporuje abstrakciu (triedy používateľov) a podporuje viacnásobné použitie.

Model používateľa sa rôznymi spôsobmi používa na zlepšenie interakcie s používateľom:

- ✓ rozpoznaním a interpretáciou chovania používateľa, jeho cieľov a plánov,
- ✓ poskytnutím pomoci a rady používateľovi, vyhodnotením závažnosti informácie a vhodnosti jej poskytnutia používateľovi
- ✓ získaním informácií od používateľa
- ✓ poskytovaním informácií používateľovi.

Model používateľa ovplyvňuje správnosť, účinnosť, efektívnosť a akceptovateľnosť znalostného systému. Obsahom modelu používateľa sú spravidla:

- ✓ ciele a plány
- ✓ schopnosti používateľa, či už fyzické alebo mentálne
- ✓ preferencie a predsudky
- ✓ znalosti.

Znalostné systémy delíme aj podľa toho, kto nesie zodpovednosť za dialóg. Za dialóg môže niesť zodpovednosť:

- 1) **Používateľ**. V tomto prípade je tvorba modelu používateľa málo pracná. Predpokladá sa, že používateľ používa systém vždy správnym spôsobom.
- 2) **Systém**. V tomto prípade je tvorba modelu používateľa zložitá. Kompetencia používateľa je obmedzená na zodpovedanie otázok.
- 3) **Zdieľaná**. V tomto prípade je model používateľa najkomplexnejší. Systém musí rozpoznať kedy používateľ chce zmeniť tému alebo zameranie interakcie.

Použitie modelu používateľa má význam iba v doménach, kde interakcia s používateľom je dostatočná na to, aby sa prejavil jej účinok na kvalite práce systému. Pri limitovanej interakcii sú výhodnejšie iné techniky, menej náročné na interpretáciu.

72. V čom spočíva riadenie typu tabuľa v znalostných systémoch?

Často sa stáva, že báza znalostí je veľmi rozsiahla. V takom prípade sa môže rozdeliť na menšie časti spolupracujúce pomocou „tabule“ (Návrat a kol., 2002). Prílišná rozsiahlosť bázy znalostí môže spôsobiť rôzne problémy, ako napríklad vysoké nároky na operačnú pamäť, neprehľadnosť bázy znalostí znemožňujúcu jej jednoduchú modifikáciu, nedostatočnú efektívnosť riadiaceho mechanizmu, ktorému nie je umožnený globálny pohľad na riešený problém. Možným riešením tohto problému je rozdelenie bázy znalostí na niekoľko menších báz. Z toho vyplýva potreba vhodnej riadiacej štruktúry na riadenie procesu využívania viac zdrojov znalostí.

Myšlienka „tabule“ vychádza zo základnej predstavy o skupine odborníkov, sediacich pred tabuľou a riešiacich konfliktnú situáciu. Tabuľa umožňuje vzájomnú komunikáciu jednotlivých zdrojov znalostí, predstavujúcich relatívne nezávislé celky. Úlohou tabule je predovšetkým dynamicky určovať poradie využívaných zdrojov znalostí na základe globálneho pohľadu na riešený problém. Riadenie typu tabuľa zahŕňa v sebe také úlohy, ako: predčasné ukončenie činnosti neperspektívnych zdrojov, aktivizácia najslabšieho zdroja znalostí, prerušovanie zdrojov a opätovné vyvolávanie zdrojov.

Rozdelenie celistvej bázy vychádza z myšlienky, že každý zložitý systém je možné rozdeliť na subsystemy, medzi ktorými existujú interakcie, ale ktoré sú relatívne samostatnými jednotkami. Pri riešení musíme tieto interakcie stále využívať. Typ rozdelenia znalostí do jednotlivých zdrojov je závislý na konkrétnej problémovej oblasti:

- ✓ **Hierarchické** rozdelenie je rozdelenie do skupín s rôznou úrovňou abstrakcie.
- ✓ **Heterarchické** rozdelenie znalostí je rozdelenie do skupín podľa ich vzájomného vzťahu, bez ohľadu na stupeň abstrakcie.
- ✓ **Prelínané** rozdelenie predstavuje kombináciu týchto dvoch prístupov. Používa sa vo väčšine praktických prípadov.

Je potrebné definovať, akým spôsobom sa bude uskutočňovať predávanie informácií medzi skupinami. Predstavme si, že bázu znalostí reprezentovanú grafom, rozstrihneme na viacero častí. Každé pravidlo, ktoré reprezentuje prestrihnutú hranu bude zahrnuté len v jednej čiastkovej báze znalostí. Uzol, ktorý zabezpečuje styk dvoch dielčích zdrojov, teda styčný uzol, bude zahrnutý v oboch čiastkových bázach znalostí. Zmeny stavu styčného uzla sa sledujú z úrovne tabule, ktorá zabezpečuje prenos dielčej informácie z jednej dielčej bázy do druhej.

Snahou znalostného inžiniera je rozdeliť bázu znalostí tak, aby bol minimálny počet hrán inferenčnej siete, cez ktoré je vedený rez. Tým splní požiadavku relatívnej samostatnosti jednotlivých dielčích zdrojov znalostí a zamedzí častému odvolávaniu na ďalšie dielčie bázy znalostí.

Existujú rôzne problémy, ktoré je potrebné riešiť v rámci riadenia v systéme s viacerými zdrojmi znalostí:

- ✓ Aké informácie a akým spôsobom prenášať medzi zdrojmi znalostí?
- ✓ Ako a kedy zasahovať do činnosti zdrojov znalostí?
- ✓ Kedy prejsť k vyšetrovaniu ďalšieho zdroja?

- ✓ Či sa vracat' k už použitému zdroju?
- ✓ Aké informácie zahrnúť do tabule?

Prenášajú sa informácie, ktoré majú priamy vzťah k daným zdrojom znalostí získaných z bázy dát ako aj vytvorených predošlými zdrojmi. Prenášajú sa cez globálnu dátovú štruktúru – tabuľu, ku ktorej majú prístup všetky zdroje. Zdroj, ktorý ukončí vyšetrowanie, zapíše výsledky na tabuľu. Iný aktivovaný zdroj ich môže prevziať. Zásah do činnosti zdroja je možný a niekedy žiadúci. Napríklad, keď nastala situácia, že ďalšie vyšetrowanie hypotéz zdroja už nemá zmysel. vtedy je vhodné ukončiť prácu zdroja a prejsť na vyšetrowanie ďalšieho. To sa uskutočňuje pomocou démona, ktorý môže mať formu pravidla, v ktorom podmienková časť predstavuje splnenie niektorých uzlov a záverová časť, resp. akcia predstavuje prechod na nový zdroj. Tieto pravidlá – démoni – sa zapisujú na tabuľu. Prechod na iný zdroj je potom preddefinovaný, alebo závislý na okamžitom stave riešenia. Pomocou démonov je možné vyjadriť že za určitého stavu riešenia bude použitie jedného zdroja znalostí znamenať väčší prínos, ako použitie ostatných zdrojov. Ukončenie činnosti nemusí byť vždy násilné. Zdroj môže byť k riešeniu úlohy využitý úplne, a až potom nasleduje prechod na iný zdroj. Vracat' sa k už použitému zdroju znalostí, alebo nie? Človek to robí a tým si upresňuje riešenie. Interakcie medzi zdrojmi sú obojsmerné, preto je navracanie možné. Tak, ako je možný prechod od zdroja A ku zdroju B prenosom relevantných informácií, tak je možný návrat od zdroja B ku zdroju A po upresnení hypotéz zdrojom B. V tabuli môžu byť zahrnuté rôzne informácie: medzi výsledky, hypotézy jednotlivých zdrojov, dôležité informácie pre prenos medzi zdrojmi, informácie styčných uzlov a pod. Démony ovplyvňujú prácu jednotlivých zdrojov znalostí.

73. Ako funguje riadenie „tabuľa“ vo FEL-EXPERTE?

Tabuľa vo FEL-EXPERTE pozostáva z dátovej a riadiacej časti. **Dátová časť** tabule zachytáva úplnú informáciu o uzloch, sledovaných z úrovne tabule: o styčných uzloch, o cieľových uzloch dielčích báz a o iných uzloch využívaných z úrovne tabule. **Riadiaca časť** obsahuje tabule, zásobníky A, B, F a súbor démonov. V priebehu riešenia sa vytvára aktuálny zásobník úloh, ktoré majú byť prednostne riešené. A-zásobník obsahuje dielčie ciele, t.j. niektoré uzly, na ktorých vyšetrenie sa vyskytne požiadavka v priebehu konzultácie. Môže dochádzať k zmenám ich priority. B-zásobník obsahuje expertom určené poradie vyvolávania dielčích báz znalostí. Priorita bázy znalostí sa môže v priebehu konzultácie meniť. B-zásobník má nižšiu prioritu a preto je využívaný iba keď A-zásobník je prázdny. F-zásobník má najvyššiu prioritu. Zaznamenáva tie ciele, ktoré musia byť prednostne vyšetrené, a ktoré sa nenachádzajú v práve využívanej báze znalostí. V rámci F-zásobníka nie je rozdiel v prioritě cieľov. Musia byť vyšetrené všetky ciele. Zásobníky A a B sa využívajú pri základnom plánovaní činnosti. Zásobník F riadi spracovanie mimo poradia.

Démony vo FEL-EXPERTE majú formu pravidiel so situačnou a akčnou časťou. **Situačná časť** predstavuje logickú kombináciu elementárnych podmienok, v ktorých je pravdepodobnosť uzla porovnávaná voči prahu. **Akčná časť** používa nasledovné typy akcií:

- ✓ zamaskovanie uzla, t.j. jeho vylúčenie z ďalšieho vyšetřovania,
- ✓ ukončenie vyšetřovania aktuálneho cieľa a začatie vyšetřovania cieľa s vyššou prioritou v danej báze,
- ✓ prednostné vyšetřenie uzla z inej bázy,
- ✓ vylúčenie určenej bázy znalostí z ďalšieho vyšetřovania,
- ✓ ukončenie vyšetřovania aktuálnej bázy znalostí a zabezpečenie prechodu na ďalšiu bázu znalostí podľa B-zásobníka,
- ✓ reinitializácia bázy znalostí,
- ✓ reaktivácia démonov,
- ✓ vloženie uzla do A-zásobníka,
- ✓ okamžitý prechod na určenú bázu znalostí,
- ✓ zmena priority v B-zásobníku a pod.

Stratégia činnosti tabule vo FEL-EXPERTE je nasledovná:

- 1) V každom okamihu je vyšetřovaná jedna z cieľových hypotéz, ktorá bola v zásobníku F; alebo ktorá bola v zásobníku A, ak je F prázdne; alebo ktorá mala najvyššiu prioritu v danej báze znalostí, teda bola v B zásobníku, ak A aj F sú prázdne.
- 2) Po každej odpovedi používateľa prebehne aktualizácia dátovej časti tabule a kontrola démonov.
- 3) Po každej odpovedi používateľa smie byť použitý iba jeden démon. Navyiac, každý démon smie byť použitý iba raz. Po použití je deaktivovaný a v ďalšom sa už neuvažuje. Avšak môže byť reaktivovaný pomocou iného démona.

74. Aké sú aplikačné možnosti znalostných systémov?

Znalostné systémy je možné aplikovať vo všetkých oblastiach, ktoré sa vyznačujú kognitívnou povahou riešenia, t.j. riešenie je uskutočniteľné úvahou. Znalostné systémy sa môžu používať v prakticky každej oblasti, ktorá spĺňa dve podmienky: po prvé riešenie problémov tejto oblasti vyhovuje reprezentácia formou zreteľných produkčných pravidiel, po druhé sú k dispozícii expertné znalosti predstavujúce spôsob uvažovania pri riešení problémov ľudským riešiteľom. Najčastejšie pertraktované oblasti aplikácie znalostných systémov sú:

- ✓ **Medicína**, kde ide spravidla o určovanie diagnózy na základe príznakov a ťažkostí pacienta.
- ✓ **Technika**, kde sa spravidla diagnostikujú poruchy technických zariadení na základe ich vonkajších prejavov.
- ✓ **Finančnictvo**. Vo sfére ekonomiky sa pôsobnosť znalostných systémov sústreďuje na diagnostiku podozrivých operácií, na hodnotenie bonity klienta, na diagnostiku chýb v prevodoch, na riadenie burzy a pod.
- ✓ **Informatika**, kde je možné navrhovať počítačové pracovisko na základe požiadaviek klienta.
- ✓ **Automobilový priemysel** môže využiť znalostné systémy napríklad pre navrhovanie prevodovky alebo navrhovanie tvaru niektorých súčiastok. Toto sú úlohy konštrukčného typu a vyžadujú plánovací znalostný systém.
- ✓ **Chémia**, kde je možné analyzovať chemické štruktúry na základe hmotového spektrogramu.
- ✓ **Farmaceutický priemysel**, kde je možné pomocou znalostných systémov šetriť prostriedky za lieky.
- ✓ **Multiagentové systémy**. Znalostné systémy sa môžu použiť na odhad stratégie protivníka pri hraní robotického futbalu z rozostavenia protihráčov na hracej ploche. Je vytipovaná množina možných situácií – rozostavení a sú formulované pravidlá v tvare: ak dané rozostavenie, potom akcia. Pod akciou môžeme rozumieť buchnutie do lopty, pohrávanie sa s loptou pred súperom, a pod.

75. Ktoré sú najznámejšie zahraničné znalostné systémy?

MYCIN (Shortliffe, 1976), (Buchanan-Shortliffe, 1984) diagnostikuje a určuje terapiu infekčných ochorení krvi. Bol vyvíjaný v rokoch 1972-1976 v Stanford Research Institute a dosiahol úroveň ľudského experta.

EMYCYN je prázdny znalostný systém odvodený od MYCINU.

PROSPECTOR (Duda-Gasching, 1979) je znalostný systém na účely geologického výskumu na odhaľovanie ložísk nerastných surovín. Bol vyvíjaný v rokoch 1975-1979 v Stanford Research Institute a dosiahol vysoký ekonomický efekt.

KAS je prázdny znalostný systém odvodený od PROSPEKTORU.

DENDRAL (Lindsay at all, 1980), (Alty-Coombs, 1984) je plánovací znalostný systém odvodzujúci štruktúry chemických látok na základe údajov hmotového spektromera. Bol vyvíjaný na Stanfordskej univerzite. Dosahuje lepšie výsledky ako špičkoví experti.

R1 (Mc Dermott, 1983) vytvára konfigurácie počítačov VAX. Bol vyvíjaný na Carnegie Mellon University. Išlo o objednávku firmy DEC.

HEARSAY II. je znalostný systém zameraný na porozumenie ľudskej reči. Dokáže komunikovať na úrovni 10 ročného dieťaťa. Vznikol v 2. polovici 70. rokov.

HEARSAY III. je prázdny znalostný systém odvodený od HEARSAY II.

QTC (Quick Turnaround Cell) je plánovací znalostný systém zameraný na návrh a vizualizáciu návrhu tvaru súčiastky.

CHIRICO (Multidatabase Appropriateness Review System) vyberá dáta na požiadanie klienta – lekára z databáz, ktoré obsahujú klinické, diagnostické, terapeutické a demografické informácie o každom pacientovi. Tieto dáta sú spracované farmaceutickým znalostným systémom a výsledok generovaný v reálnom čase je k dispozícii na účel šetrenia prostriedkov za lieky, lekárske služby a pod.

AREX (Automobile Reliability EXpert System) je znalostný systém na spracovanie dát o spoľahlivosti automobilov.

Ďalšie zahraničné komerčné znalostné systémy sú:

- ✓ EXSYS – výrobca EXSYS
- ✓ G2 – výrobca GenSym
- ✓ KappaPC – výrobca IntelliCorp.
- ✓ Level_5 – výrobca Level_5
- ✓ M4 – výrobca Teknowledge
- ✓ NexpertObject – výrobca NeuronData
- ✓ XpertRule – výrobca Attar
- ✓ Clips – výrobca NASA.

76. Ktoré sú najznámejšie domáce znalostné systémy?

EQUANT (Hájek a kol., 1987) bol vyvinutý v Matematickom ústave ČSAV. Slúži k základnému výskumu v oblasti práce s neurčitosťou.

SAK (Ivánek, 1984) bol vyvíjaný na VŠE súc inšpirovaný EQUANTOM.

CODEX2 (Popper a kol., 1983) bol vyvinutý vo Výskumnom ústave lekárskej bioniky v Bratislave. Je orientovaný na diagnostiku chorôb. Obsahuje niekoľko rôznych báz znalostí.

FEL-EXPERT (Mařík-Zdráhal, 1987) bol vyvinutý na EF ČVUT v Prahe. Je to diagnostický znalostný systém, ktorý má množstvo báz znalostí, ktoré slúžia na:

- ✓ diferenciálnu diagnostiku genetických ochorení sprevádzaných myopatiou
- ✓ detekcia mužov postihnutých syndrómom fragilného X-chromozómu
- ✓ klasifikácia dlhodobých psycho-fyziologických stavov na základe záznamu EEG
- ✓ klasifikácia organických látok podľa ich toxicity
- ✓ technická diagnostika napájania rádiolokátora
- ✓ plánovanie kusovej výroby

MIFELEX je mikropočítačová varianta systému FEL-EXPERT

PROLEX II (Mach, 1990), (Csontó-Mach, 1988) je členom radu prázdnych prologovsky orientovaných expertných systémov, vyvíjaných na Katedre Kybernetiky a Umelej Inteligencie EF VŠT v Košiciach. K systému bola vyvinutá aj medicínska báza znalostí, zameraná na diagnostikovanie ochorení tráviaceho traktu.

TEAM je znalostný systém vyvinutý VŠE v Prahe. Obsahuje znalosti experta – psychológa Špirka a je určený na hodnotenie a riadenie práce riešiteľských tímov.

77. Ktoré sú najznámejšie ekonomické aplikácie?

Pri ekonomických aplikáciách sa spravidla hodnotí bonita klienta, realizuje sa investičné doporučenie, podpora auditu a podobne. Komerčne úspešné znalostné systémy z tejto oblasti sú držané v tajnosti, keďže ide o peniaze. Spoločnosti vyvíjajúce znalostné systémy sa nechvália ani so svojimi neúspechmi, aby sa konkurencia nedozvedela, kade cesta nevedie. Niektoré zo systémov, ktoré budú nasledovať sú znalostné systémy, iné sú hybridné systémy, ktorých súčasťou je aj znalostný systém. Podrobnejší popis v ďalšom uvedených systémov je možné nájsť v (Kelemen a kol., 1999).

FAST je systém na hodnotenie bonity klienta, na základe čoho dostane, resp. nedostane pôžičku. Využíva znalostný systém EXSYS.

LendingAdvisor bol vyvinutý spoločnosťou Syntelligence. Vyhodnocuje žiadosti o komerčné pôžičky. Rozpoznáva potenciálne riziká pôžičky. Naviac volí vhodnú štruktúru pôžičky a vie odôvodniť svoje rozhodnutia.

PLANET vyvinula firma Price Waterhouse, ktorá vyvinula aj štandardy spôsobu uskutočňovania auditu v celosvetovom merítku. Používa sa na podporu auditorov.

EXPERTAX vyvinula firma Coopers and Lybrand. Pomocník auditorov a daňových poradcov pri zisťovaní informácií, potrebných pre určenie výšky dane.

INVEST radí klientom banky pri nákupe a predaji akcií.

Nyse Focus Review Systém hľadá finančné problémy na základe automatickej analýzy správ z Newyorskej burzy.

Market Mind podporuje finančné operácie na Newyorskej burze. V reálnom čase monitoruje údaje z burzy a dáva doporučenia, kedy obchodovať. Je napojený na burzový transakčný systém a uskutočňuje príkazy na prevedenie operácií na burze.

CARMA teda Computer Assisted Real Estate Market Analyst slúži na analýzu trhu a využíva prázdny znalostný systém EXSYS, ako aj tabuľkový kalkulátor LOTUS 1-2-3.

FINCEN teda Financial Crimes Enforcement Network používa ministerstvo financií USA na preverovanie transakcií nad 10 000\$ za účelom odhalenia eventuality nelegálnych aktivít. Využíva znalostný systém NEXPERT OBJECT.

78. Referencie:

(Alty-Coombs, 1984)

Alty, J.L., Coombs, M.J.: Expert Systems. Concepts and Examples. NNC Publications, 1984.

(Berka, 1994)

Berka, P.: Tvorba znalostních systémů. Praha, Vysoká škola ekonomická, 1994.

(Bieliková a kol., 1992)

Bieliková, M., Frič, P., Frištacký, T., Galbavý, M., Kapustník, I., Molnár, L., Návrat, P., Vojtek, V.: Supporting the Knowledge Based Systems Development. Kybernetika a Informatika, Vol. 5, No. 2-3, 179-184, 1992.

(Buchanan-Shortliffe, 1984)

Buchanan, B.G., Shortliffe, E.H.: Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison-Wesley, Reading, Mass. 1984.

(Csontó, 1990)

Csontó, J.: Umelá Inteligencia. Edičné stredisko VŠT, Košice, 1990, 196 s, ISBN 80-7099-054-6.

(Csontó-Mach, 1988)

Csontó, J., Mach, M.: Aplikácie Jazyka Prolog, JZD Slušovice, 1988.

(Driankov at all, 1993)

Driankov, D., Hellendoorn, H., Reinfrank, M.: An Introduction to Fuzzy Control. Springer-Verlag, Berlin, 1993.

(Duda-Gasching, 1979)

Duda, R.O., Gasching, J.E.: Model Design in the Prospector Consultant System for Mineral Exploration. In: Expert Systems in the Micro Electronic Age, Edinburgh University Press, 1979.

(Guida-Tasso, 1994)

Guida, G., Tasso, C.: Design and Development of Knowledge-Based Systems. From Life Cycle to Methodology. John Wiley&Sons, Trowbridge, GB, 1994, 476 ps, ISBN 0-471-92808-9.

(Hájek, 1985)

Hájek, P.: Combining functions for certainty degrees in consulting systems. Int. J. Man - Machine Studies, vol. 22, 1985, s. 59-76.

(Hájek, 1987)

Hájek, P.: Dempster – Shaferova teorie evidence a expertní systémy. In *Metody umělé inteligence a expertní systémy III*. Praha, ČSVTS FEL ČVUT, 1987, s. 54-61.

(Hájek a kol., 1987)

Hájek, P., Hájková, M., Havránek, T.: Expertní systém EQUANT. In *Expertné systémy*. ČSVTS, Bratislava, 1987, 132-138.

(Ivánek, 1984)

Ivánek, J.: Realizace konzultačního systému SAK-OPTIMALI pro výběr matematické metody rozhodování. In *Expertní systémy: principy, realizace, využití*, Sborník ČSVTS FEL ČVUT, Praha, 1984, 160-172..

(Janota, 2000)

Janota, A.: Práca so systémom EXSYS. Učebná pomôcka. Katedra informačných a zabezpečovacích systémov, Fakulta elektrotechnická, Žilinská univerzita, Žilina, 2000, 45s.

(Kelemen a kol., 1999)

Kelemen, J., Kubík, A., Lenharčík, I., Mikulecký, P.: Tvorba expertních systému v prostředí CLIPS. Praha, Grada Publishing, 1999, 247 s. ISBN 80-7169-501-7.

(Klir-Yuan, 1995)

Klir, G.J., Yuan, B.: *Fuzzy Sets and Fuzzy Logic – Theory and Applications*. Prentice-Hall, Englewood Cliffs, NJ 1995.

(Lindsay at all, 1980)

Lindsay, R., Buchanan, B.G., Feigenbaum, E.A., Lederberg, J.: *Application of Artificial Intelligence for Organic Chemistry – the DENDRAL Project*. McGraw-Hill, New York, 1980.

(Mach, 1990)

Mach, M.: *Prolex II*. TNS Computer, SWS Software Slušovice, 1990, 138 s.

(Mach, 1997)

Mach, M.: *Získavanie znalostí pre znalostné systémy*. Košice, Elfa s.r.o., 1997, 104 s. ISBN 80-88786-67-3.

(Mařík a kol., 1997)

Mařík, V., Štěpánková, O., Lažanský, J. a kol.: *Umělá inteligence I, II*. Academia, Praha, 1997.

(Mařík-Zdráhal, 1987)

Mařík, V., Zdráhal, Z.: Expertní systémy řady FEL-EXPERT. In *Expertné systémy*. ČSVTS, Bratislava, 1987, 100-110.

(McDermott, 1983)

McDermott, J.: Extracting Knowledge from Expert Systems in IJCAI83, 1983, 100-107.

(Michalski-Stepp, 1983)

Michalski, R.S., Stepp, R.E.: A Theory and Methodology of Inductive Learning. In: Michalski R.S., Carbonnell J.G., Mitchell T.M. (eds.): Machine Learning: An Artificial Intelligence Approach, Morgan Kaufmann, 1983.

(Mikulecký a kol., 1998)

Mikulecký, P., Hynek, J., Lenharčík, I.: Znalostní technológie II, Expertní systémy. Gaudeamus, Hradec Králové, 1998.

(Moses, 1975)

Moses, J.: MACSYMA Primer. MIT Computer Science Laboratory, Cambridge, Mass, 1975.

(Návrat a kol., 2002)

Návrat, P., Bielíková, M., Beňušková, L., Kapustník, I., Unger, M.: Umelá inteligencia. Vydavateľstvo STU, Bratislava, 2002.

(Payne-McArthur, 1990)

Payne, E.C., McArthur, R.C.: Developing Expert Systems. A Knowledge Engineer's Handbook for Rules and Objects. John Wiley&Sons, USA, 1990, 401 ps, ISBN 0-471-51413-6.

(Petrušová, 1989)

Petrušová, K.: Intenzionálny prístup k spracovaniu neurčitosti v expertných systémoch. In: Zborník prác VŠT Košice, Zväzok III., ALFA, Bratislava, 1989, 337-353.

(Popper a kol., 1983)

Popper, M., Bartos, P., Gyarfás, F., Guresova, M., Tekusova, M.: Diagnostický expertný systém CODEX-2, Výskumná správa, Výskumný ústav lekárskej bioniky, Bratislava, 1983.

(Popper-Kelemen, 1989)

Popper, M., Kelemen, J.: Expertné systémy. Alfa, Bratislava, 1989.

(Quinlan, 1990)

Quinlan, J.R.: Decision trees and decision making. IEEE Transactions Systems, Man and Cybernetics 20, 2, 1990, pp. 339-346.

(Quinlan, 1993)

Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann, 1993.

(Russell-Norvig, 1995)

Russell, S., Norvig, P.: Artificial Intelligence. A Modern Approach. Prentice-Hall, A Simon&Schuster Company, New Jersey, 1995, 932 ps, ISBN 0-13-360124-2.

(Shortliffe, 1976)

Shortliffe, E.H.: Computer-Based Medical Consultation – MYCIN. American Elsevier, New York, 1976.

(Tien-Chien Chang, 1990)

Tien-Chien Chang: Expert Process Planning for Manufacturing. Addison-Wesley Publishing Company, USA, 1990, 283 ps, ISBN 0-201-18297-1.

(Zadeh, 1965)

Zadeh, L.A.: Fuzzy sets. Information and Control 8, 1965, 338-353.

(Zadeh, 1973)

Zadeh, L.A.: Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. IEEE Trans. on Systems, Man and Cybernetics 1, 1973, 28-44.

NÁZOV: Znalostné systémy v otázkach a odpovediach

AUTOR: Ing. Kristína Machová, CSc.

POČET STRÁN: 108

POČET VÝTLAČKOV: 50 kusov

VYDANIE: Prvé

TLAČ: elfa, s.r.o, Košice, 2005

ISBN: 80-8086-018-1

Kristína Machová

ZNALOSTNÉ SYSTÉMY
V OTÁZKACH
A ODPOVEDIACH

Košice
2005

elfa