

1. Pojem: intelligence, UI, počítačová intelligence → + jaké komponenty

- a) intelligence - schopnost člověka abstraktně a rozumně přemýšlet a odvozovat účelná jednání
- b) UI - schopnost napodobovat inteligenci člověka tím, že programy vykonávají činnosti spojené s člověkem - učení, uvažování, sebezdokonalování
- c) Poč. intelligence - obor UI; aplikace algoritmů biologického světa do informatiky

Komponenty - fuzzy logika; neuronové sítě, evoluční algoritmy

multiválná logika - výroky se ohodnocují mírou pravděpodobnosti z $\langle 0; 1 \rangle$ (jejich výkonnost)

2. Informatická definice UI

- chápe se jako snaha člověka o vytvoření počítačových programů, které by se schopnosti rozhodovat se v předem neočekávaných situacích blížily uvažování a chování člověka a čímž jej napodobovali

3. Specifikujte jednotlivé problemové oblasti UI

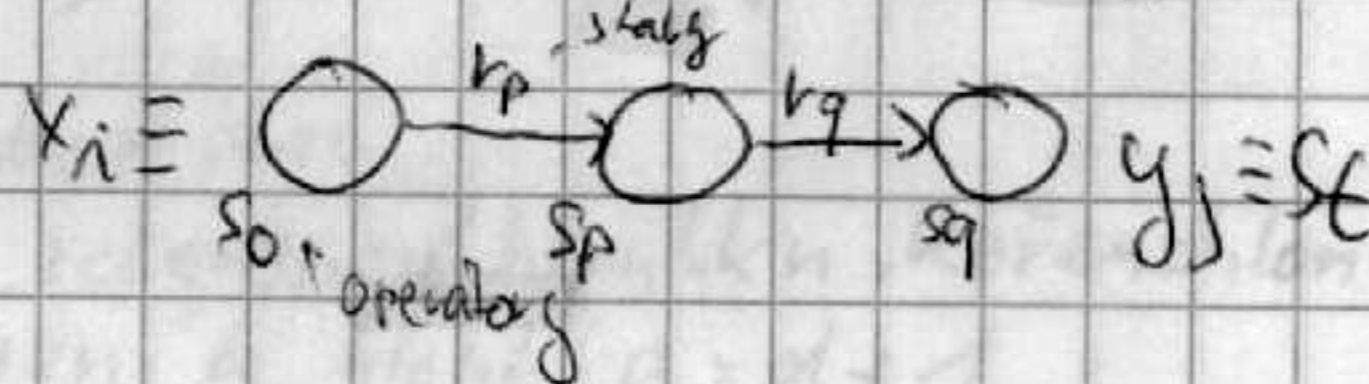
- 1. řešení úloh, herní strategie
- 2. automatické dokazování - provádění důkazů na základě znalostí, které lze formalizovat a v jednodušší formě uchovávat v paměti
- 3. inteligentní přístup k databázi - nalezení dat s co nejmenší odezvou - systém si dokáže vsdodukovat odpověď
- 4. Reprezentace znalosti a znalostní systémy - fakta, znalosti, které si uchováva, používá - výsledek je vhodnější stavem diagnózy
- 5. Kombinatorické a rozhodovací úlohy - plánování a rozhodování činnosti
- 6. Vnímání prostředí - analýza scénů; využití rozpoznávání
- 7. Porozumění přirozenému jazyku - porozumění pokynům člověka
- 8. Robotika - autonomní a cílově orientované instrukce s reálným prostředím
- 9. Automatické programování

4. Data X informace

Data - co přístroje zachytí z prostředí ve formě dat - fakta - bez souvislosti
Informace - data v definovaných souvislostech (např. míra nezaměstnanosti)

5. Úloha

- zobrazení množiny výchozích stavů do cílových $X \rightarrow Y$
- grafem jako posloupnost stavů



6. Obecná definice úlohy, roje řešení

řešení - postup, kdy přivedem úlohy z výchozího do cílového stavu

7. Elementární operátor - operace vedoucí k transformaci jednoho mezi stavu úlohy do jiného

Kompoziční operátor - množina elementárních operátorů

8. Typy úloh a jednoznačnost řešení

úloha (X, Y, R) $X \xrightarrow{R} Y$

a) abduktivní řešení - $(? Y R)$ - 1 exaktní řešení

b) deduktivní - $(X ? R)$ - 1 exaktní řešení

c) indukivní - $(X, Y, ?)$ - řešení je hypotéza, kterou musíme zpětně ověřit dedukcí

9. Vytváření a prohledávání stromu řešení úloh

- metoda pokusů a omylem; hledá se takový kompoziční operátor R , který uschová množinám stavů X a Y
- procedura - Deterministické - předvídané
 - Náhodné
 - Heuristické - kusné řešení

10. Metoda stavového prostoru - zabezpečení úlohy jako posloupnost nezávislých a elem. operací

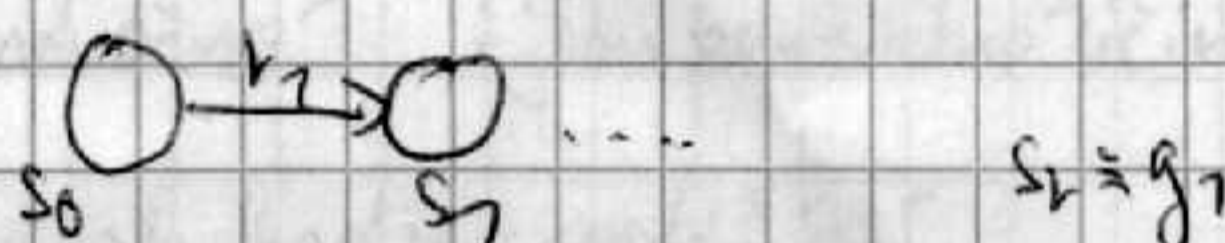
- předpoklady - konečná množina stavů $S = \{s_i\}$
 - jeden uschozí stav $s_0 \in S$
 - konečná množina cílových stavů: $G = \{g_j\} \subseteq S$
 - konečná množina elem. operátorů: $R = \{r_k\}$ ze stavu s_p do s_q

Stavový prostor - definován (S, R) a konkrétní řešení úlohy je $(s_0, G, R_{k_{0j}})$

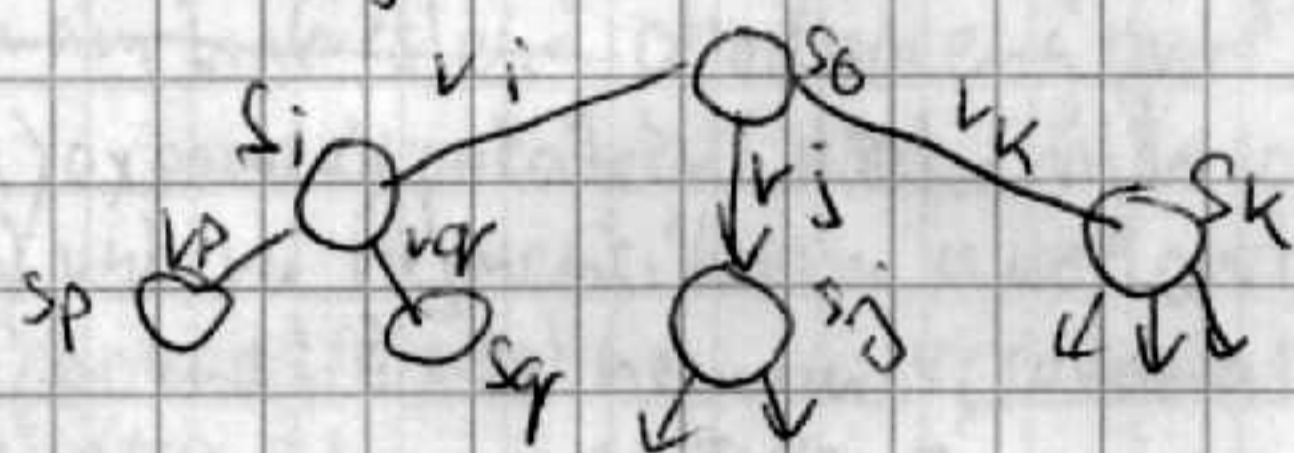
11. Obecná metoda reprezentace úlohy ve stavovém prostoru

Stavový prostor \rightarrow dvojice (S, R) a konkrétní řešení úlohy trojice $(s_0, G, R_{k_{0j}})$

Rěšení: $R_{k_{0j}} = r_1 \dots r_{n-1} r_n$ takový, že $s_1 \leftarrow r_1(s_0)$
 $s_2 \leftarrow r_2(s_1) = r_2(r_1(s_0))$



Reprezentujeme ~~stavením~~ stromovým grafem



12. Expanze uzlu - nalezení všech bezprostředních následovníků daného uzlu

tzn: daný uzel aplikujeme všechny možné elementární operátory

13. Strom řešení úlohy (obecná teorie) - orientovaný graf

- graf má jen jeden kořen - uschozí stav
- u každého uzlu definujeme rekurzivně hloubku. kořen = hloubka 0
pokud hloubka = d ; následovník hloubka = $d+1$
- list je uzel bez následovníka a reprezentuje cílový stav úlohy
- nelze aplikovat elementární operátor a ~~ne~~ neperspektivní stav
- orientovaná hrana reprezentuje přechod úlohy z daného stavu do nového

14. Produkční systém = počítačový program, složený z sady de pravidel chování

- obsahuje:
 - data báze úlohy obsahující fakta
 - báze znalostí obsahující produkční pravidla a podmínka \rightarrow akce
 - řídící mechanismus - úkoly \rightarrow 1. volba pravidla pro aplikaci
2. vybrat fakta z databáze a dosadit do podmínky pravidla
3. ukončení řešení při splnění podmínky

15. Cílová podmínka = podmínka ztavení produkčního systému

- explicitní - odvozená z množiny cílů
- implicitní - nejde-li na obsah databáze aplikovat žádné další produkční pravidlo

16. Neodvolatelné řídicí strategie

- takové, že výběr operátorem už nelze později změnit

Např. hlavolam „8“

	(4)	(7)	(3)	(2)	(1)
2 8 3	2 8 3	2 3	2 3	1 2 3	
→ 6 4 → 7	4 → 1 8 4 →	1 8 4 →	1 8 4 →	8 4	
→ 5	7 6 5	7 6 5	7 6 5	7 6 5	

Např. novolezecký algoritmus

17. Pokusné řídicí strategie

- strategie, které se v případě potřeby dokážou vrátit k určitému uzlu a aplikovat jiné pravidlo

- a) backtracking - stavový, operátorový b) hledání v grafech řešení úlohy -
- neinformované - slepé
- informované - zpravidla heuristické

18. Stavový - uchovává v paměti předchozí stavy - náročný na paměť
Operátorový - uchová v paměti zásobník operací (málo paměti, hodně času)

19. Metody hledání v grafech

- paměť si doposud vygenerovanou část stromu
- základem je expanze uzlu a hodnotující funkce
- paměťově náročný - jsou použitelné účinné globální heuristiky = hodnotující funkce

20. Řešení úlohy obchodní cestující

- pro
- mám nakreslený ohodnocený graf a postupuji:
 - expanduji počáteční uzel - najdu sousedy a každý ohodnotím
 - vyberu nejmenší hranu a uzel opět expanduji - vracím se
 - pokud je uzel nemůže expandovat ani najít cíl, rekurzivně se vrátím
 - algoritmus se zastaví, pokud jsem prošel všechny uzly

21. Vyhrazení stromu řešení úlohy

- expanduji uzel
- vyberu hranu s nejmenším ohodnocením
- a) i b) se opakují, dokud je co expandovat. Pokud dojdeme k listu, který není cílovým stavem, rekurzivně se vrátíme a vybereme další hranu.
- pokud při vrácení dojdeme ke stejnému výsledku jako dříve, zrušíme pointer na řešení, dokud toho jsem se dostal pohranák s větším ohodnocením

22. Slepé strategie hledání řešení úlohy

- 1) do hloubky - každý podstrom řešení expanduji až do listu, pak se popř. rekurzivně vrátím - ohodnocení = hloubka uzlu
- 2) do šířky - expanduji uzel, vyberu potomka s nejlepším ohodnocením a expanduji dál - tímto se najde nejkratší cesta k cílovému uzlu - ohodnocení uzlu = hloubka uzlu

23. Informované (cílené) strategie hledání řešení úlohy

- strategie, které hledají takovou heuristiku (hodnotící funkci), která by našla řešení úlohy v relativně malém počtu vygenerovaných stavů

→ volím heuristickou fci: $f(n_i) = d(n_i) + w(n_i)$

d ... délka cesty z kořene uzlu n

w ... celková diference (nesoulady ve stavu n od cílového)

- většinou narazíme na protichůdné požadavky - např.: nalezení nejkratší cesty když chceme min. nárok na vyhledávací strom

24. Vysvětlete fci A a A^* algoritmu

- modifikace základního algoritmu hledání v grafu

- optimální cesta = ta s minimální cenou \Rightarrow definují ji ohodnocující fce:

$$f^*(n_i) = g^*(n_i) + h^*(n_i) - \text{skut. cena z uzlu } n_i \text{ do cílového}$$

\uparrow
skutečná cena cesty z n_i do cíle

- protože ale ~~používáme~~ neznáme cestu z konkrétního uzlu n_i do cíle, používáme předpis:

$$f'(n_i) = g'(n_i) + h'(n_i)$$

\uparrow
odhad optima

, kde jsou použity odhady cen cest. Pokud platí $0 \leq h'(n_i) \leq h^*(n_i) \Rightarrow$

\Rightarrow jedná se o A^* algoritmus = lichý, nezáporný dolní odhad fce $h^*(n_i)$

25. Více informovaný algoritmus

- s výše heuristickou fci $h'(n_i)$ vyžaduje přesnější heuristickou informaci

26. Způsob rozkladu úlohy na podúlohy

„základní“

Pokud není elementární, jde o nalezení kompozičního operátoru, který odstraní prví ... m -tou diferencí úlohy (nesouhlas v dílčích složkách ušlechtilého a cílového stavu)

Tím dojde k rozkladu úlohy na m -podúloh

27. Konjunktivně disjunktivní graf pro prezentaci rozkladu úlohy na podúlohy

= AND/OR graf - orientovaný acyklický graf

- 2 typy uzlů - AND \wedge - potřeba vyřešit obě dílčí podúlohy

- OR \vee - jen jednu podúlohu

28. Algoritmy hraní jednoduchých her

= algoritmy prohledávání stavového prostoru reprezentovaného stromem řešení úlohy

a) prohledávání celého stavového prostoru do určité hloubky \rightarrow nejčastější

b) prohledávání jen dobře vypadajících částí stavového prostoru

c) cílově orientované algoritmy - cílem jednoho hráče je maximalizovat ohodnocující fci, cílem druhého minimalizovat

29.

- Minimax - projde tahy a oboduje pozice a vybere nejlepší pozici \rightarrow šachy

- ~~Minimax~~ ~~prohledávání~~

- ohodnocení symbolizuje ppst, s jakou dosáhneme vítězství

- musí být vygenerovaný celý strom a postupuje se od nejhlubšího uzlu

30.

- Alpha-beta prohledávání - vylepšení Minimaxu

- při vstupu do stavového prostoru se dříve nerozvíjí větve, které mají ohodnocení horší než aktuální nejlepší - na dané úrovni si pamatujeme max a min

31. Druhy evolučních algoritmů

- Genetické - teorie přirozeného výběru
 - zvládají adaptaci a změny se podmínky učení
- Umělé neuronové sítě - napodobují činnost mozku
 - tvořené z neuronů a synapsí - stimulace chování neuronové soustavy

32. Typy operací v genetických algoritmech

- reprodukce - kvalitní jedinci mají častější reprodukci a žijí déle
- křížení - zabráňuje vytvoření stejných jedinců - prohažují si vlastnosti
- mutace - také zabráňuje vytvoření st. jedinců, ale lepší adaptace v měnícím se prostředí - mění častěji

33. Implementace ES a GA

- jedinec řešení (chromozom) je nejčastěji reprezentován nějakou lineární posloupností symbolů (bitová řetězec), každý řetězec je ohodnocen fitness funkcí, která mu přiřadí kvalitu (kladné číslo)
- pomocí rulety (kvažinahodně) se rozhodne, které chromozomy se budou reprodukovat
- pokud jsou chromozomy ^{skoro} ~~stejně~~ ^{hodně} překříženy, provede se mutace ^{→ volí se stejní jedinci}

34. Použití genetického programování

- pro NP problémy (úplně); pro jednoduché nemá smysl
- uplatnění - vozůvek prac strojů v továrnách, teorie her, řízení robotů, návrhy vzpěrných systémů; simulace úloh umělého života

35. Stochastické optimalizační algoritmy

- používají se pro mnohoparametrické fce s divokým průběhem
- mnohokrát opakovaná náhodná volba počátečního řešení úlohy
- Hill Climbing, GA, ES, Simulované žitání, Tabu search

36. Simulované žitání

- problém obchodního cestujícího
- máme systém popsaný n-rozměrným stavem $x = (x_1, \dots, x_n)$, funkce $f(x)$ každému přiřadí ohodnocení y . úlohou je pak najít takové x , ve kterém fce nabývá globálního minima
- jestli je nový stav přijat (x'), rozhoduje se na základě ppsti, kdy ^{je} vygenerované náhodné číslo a pokud je menší než $P(x' \rightarrow x)$, pak je nový stav akceptován

$$P(\text{horší hodnota}) = \frac{1}{e^{-\frac{\Delta f}{T}}} \quad P(x' \rightarrow x) = 1 \quad f(x') \leq f(x)$$

37. Tabu search - zakázané prohledávání

- vylepšení Hill Climbing
- směr největšího spádu se neodvozuje z gradientu, ale prohledáváním okolí
- v náhodně vygenerovaném bodu z náhodně vyg. okolí najde minimum
- to se použije jako střed pro další minimalizaci. Pustí se několikrát a nejmenší min. je výsledek
- má krátkodobou paměť (fronta) která uchováva transformace k předchozím iteracím →
- ty jsou zakázané, aby se algoritmus nezacyklil

- problém obch. cestujícího

49. Algoritmus K-means

- rozdělovací metoda - dělení objektů podle kritéria, podobnostní funkce (vzdálenost)
- shluk je zde těžištěm objektů

- Alg.:
- 1) zadání počtu k shluků a množiny objektů
 - 2) volba úchozích k středů shluků
 - 3) přiřazení všech objektů ke středům shluků - s min. vzdáleností
 - 4) výpočet nového středů - těžiště objektů => 5. back to 2.
- koniec = nezměněná poloha objektů

50. Základní typy dat

- intervalové proměnné - pozice, váha, výška, jas, datum
 - hutnost normalizace [0, 1] a nastavení stejné váhy proměnným
- binární proměnné - 1 - přítomnost vlastnosti > objekt složen z primitiv 100
 - 0 - nepřítomnost vlastnosti

51. Nejběžnější metriky na měření vzdálenosti

a) eukleidovská vzdálenost - $(x_1, x_2); (y_1, y_2) \Rightarrow D_e = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$

b) manhattan vzdálenost



- stejná u všech cest

c) kosinová vzdálenost

- vzdálenost dvou vektorů
je úhel, který svírají

52+53. Evaluační metriky pro klasifikaci a rozpoznávání

- vyhodnocení přesnosti klasifikace nebo rozpoznávání
- udává se v % nebo jako desetinné číslo

Dělení: 1-1 (klasifikace)

- chybovost: $ER = E/ALL$
počet chyb / počet všech vzorků

- přesnost: $ACC = OK/ALL = 1 - ERR$
počet správně klas. vzorků

1-h (rozpoznávání)

- přesnost: $ACC = (N - D - S - I) / N$ -> počet všech jednotek
počet chyb / počet vzorků

- přesnost, úplnost, F1-míra: správně odhadnuté pozitivní případy

odhad / veřil	1	0	
1	TP	FP	False Positive
0	TN	FN	True Negative

$$F\text{-mess} = 2 \cdot (TP / (TP + FP) \cdot TP / (TP + FN)) / (TP / (TP + FP) + TP / (TP + FN))$$

54. Matice záměn - konfuze matice

- obsahuje hodnoty přesnosti klasifikátoru
- klasifikace dokumentů, rozpoznávání dialog. akcí

	a	b	c
a	88	4	12
b	5	65	5
c	17	25	33

naměřené velikiny
spektrální koeficienty
histogramy

55. Vlastnosti příznakových metod rozpoznávání



56. klasifikátor s diskriminační funkcí

- pro každou třídu definujeme takovou fci, aby $g_s(x) > g_t(x) \mid \forall t \in \{1, R\}, s \neq t$

$w_p = \max(g_s(x), s=1..R)$; $g_s(x) - g_t(x) = 0$ - rovnice vodorovných hranic



57. Klasifikátor na principu kritéria min. vzdálenosti

- v úlohách s oddělitelnými množinami obrazů
- ve fázi trénování se vytvoří Etalony (vzorové obrazy) $\rightarrow e_s$ \rightarrow pouze známých tříd
- neznámé třídy se klasifikují pravidlem $w_p = \|e_p - x\| = \min \|e_s - x\| \quad s=1..R$

58. Bayesův klasifikátor - kvit. min. chyby

- v úlohách s neodělitelnými (prolínajícími) třídami obrazů
- příslušnost ke třídě lze určit jen na základě ppsti
- lze nastavit tak, aby chyby byly min.

ppst příslušnosti třídy s identifikátorem w_s : $P(w_s/x) = \frac{p(x/w_s) \cdot P(w_s)}{p(x)}$; $p(x) = \sum_{i=1}^R p(x/w_i) \cdot P(w_i)$

- rozhodovací pravidlo $w_p = \max_{s=1..R} (P(w_s/x))$

59. Kритérium min. chyby - apost. ppst, ztrátová fce

- ztrátová fce = $J(q) = \int \sum (\lambda(d(x, q) | w_s) \cdot p(x/w_s) \cdot P(w_s)) dx$

60. Strukturální popis rozpoznávaných objektů, strukturální obrazy

- Obraz se rozloží na: primitiva, vlastnosti primitiv, relace primitiv (funkce, časové, prost.)
- z těchto primitiv se pak vytvoří popis (řetězec symbolů, graf, relační struktura)

61. Princip strukturálního rozpoznávání objektů


- rozhodnutí, jestli je slovo (popis objektu) slovem jazyka z tříd. (gramatiky)
- z toho se vytvoří strukturální popis
- jednoduchá reprezentace pro složité objekty

62. Úloha strukturálního rozpoznávání objektů

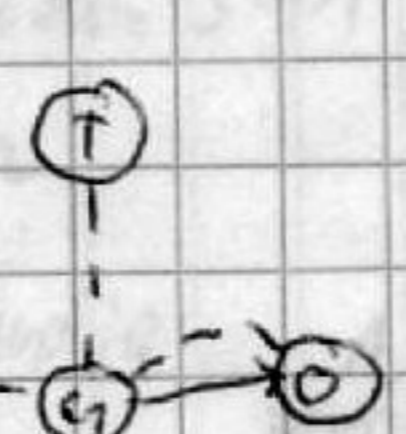
- 1) Trénování: a) vytvořím primitiva a relace b) z primitiv strukturální popis c) určím gramatiku z popisu
- 2) Rozpoznání: a) z objektu primitiva a relace b) strukturální popis c) na základě gramatiky rozhodnu, kam patří d) přiřadím obraz ke třídě

63. Postup vytváření strukturálních obrazů

a) primitiva $M = (T, O, C_1, C_2)$

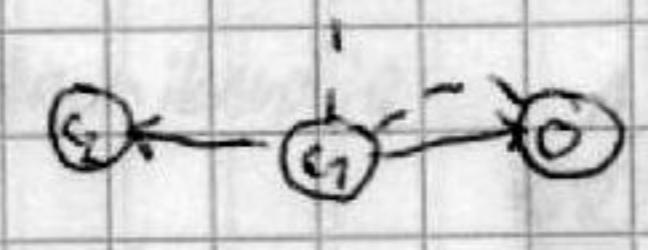


b) užití relace: T - trojúhelník, O - obdélník, C - čtverec

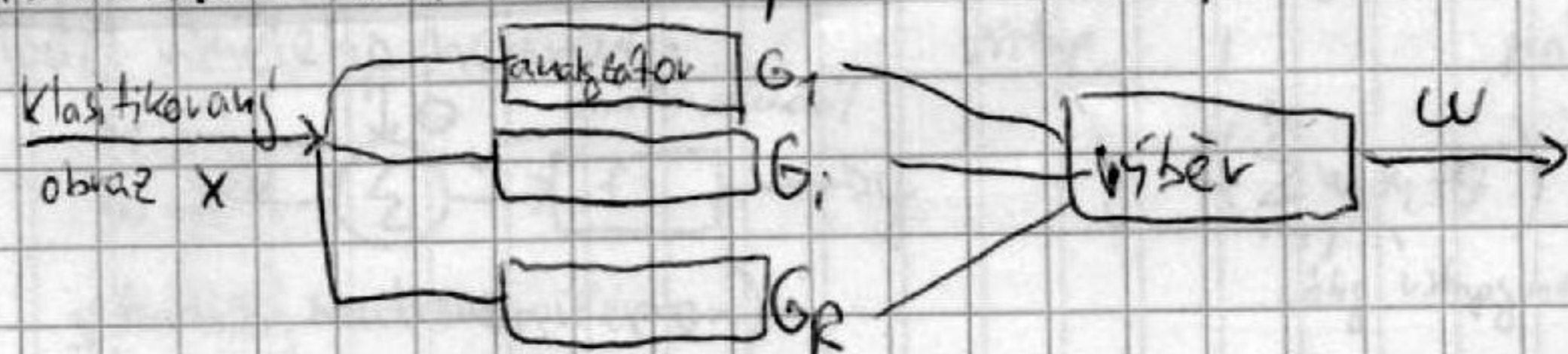


c) binární relace: dotyk se - - - - je uchvátno - - - -

d) semantický vektor



64. Struktura klasifikátoru pro str. rozpoznávání



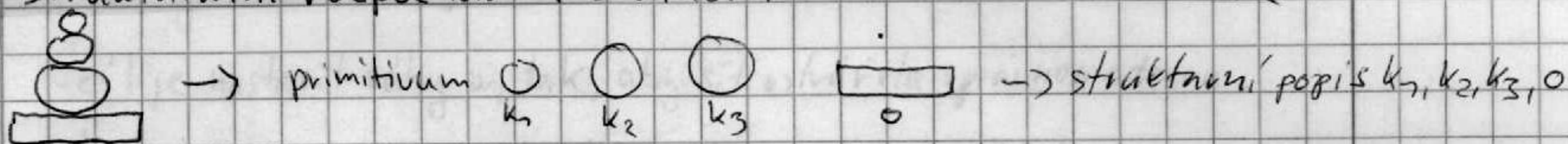
65. Extrakce primitiv Freemanovým kódem



66. Gramatika

$\begin{pmatrix} e & f \\ C & \end{pmatrix}$ $S_1 \rightarrow e X_1$
 $X_1 \rightarrow f X_2$

67. Strukturální rozpoznávání s využitím sémantické informace



logická gramatika:

$$G = (V_N, V_T, S, R)$$

$$V_N = \{S, A, B, C\}$$

$$V_T = \{k_1, k_2, k_3, 0\}$$

$$R = \{S \rightarrow k_1, A \rightarrow k_2 B, B \rightarrow k_3 C, C \rightarrow 0\}$$

atributová gramatika: vektor v obsahuje poloměry koule

$$k_6, k_8, k_{10}, 0$$

$$G_a = (V_N, V_T, S, R_a, P_a, v)$$

$$V_N = \{S, A\} \quad V_T = \{k, 0\}$$

$R_a:$ $S \rightarrow kA$ $A \rightarrow kA$ $A \rightarrow 0$	$C_a:$ $par \geq v_1$	$P_a:$ $v_1 = 0$ $v_2 = par$
---	--------------------------	------------------------------------

68. Proč je důležitý výběr příznaků v klasifikaci (rozpoznávání)?

- určuje cestu měření, výpočtu a použití příznaků (nevhodně zvolené → snížení přesnosti) ACC
- důležitá je vhodná znalost $P(x|c)$ → přesnost vs. rychlost

69. Základní metody pro výběr příznaků

- náhodný výběr - shovadlová metoda; postupně ubírání/přidávání příznaků; ověření
- frekvence dokumentu - jak často se term t_i vyskytuje v dokumentu d_i ; (klasifikátorem)
 - normalizuje se dělkou dokumentu
- inverzní frekvence dokumentu - častá slova (např. "je") jsou nedůležitá pro klasifikaci
- vzájemná informace -
 - náhodné proměnné X, Y ; $MI \rightarrow$ informace X a Y
 - znalost proměnné $X \rightarrow$ snížení nejistoty Y

70. Algoritmus k-nejbližších sousedů



- použijeme lokální hranice pro klasifikaci
- 1-NN: třída dle nejbliž. souseda k -NN: max. počet nejbliž. sousedů

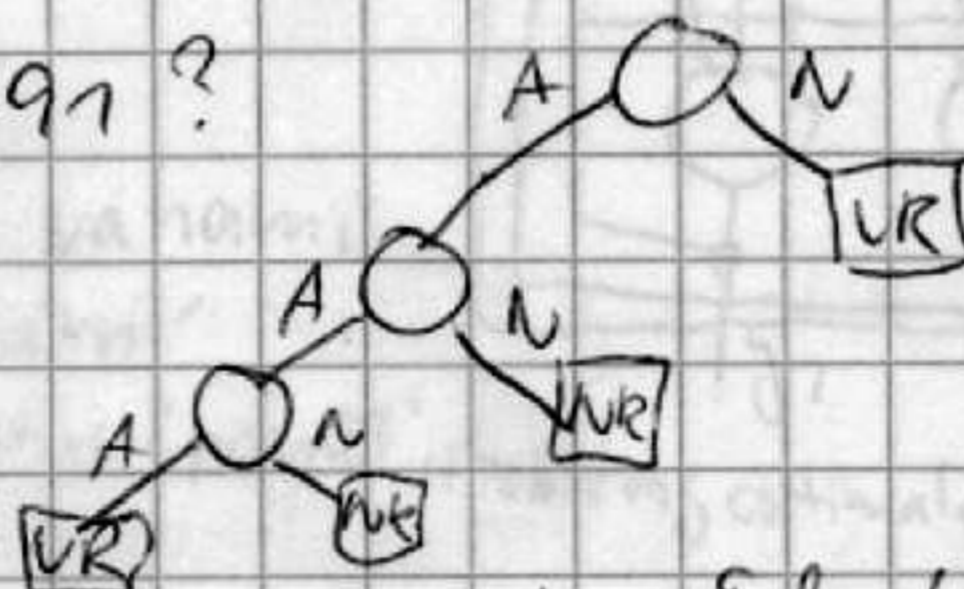
71. Klasifikační a regresní stromy - CART

- popis vzájemných stavů mezi pozorovanými veličinami pomocí stromů

Systolický tlak během 2 hodin > 91?

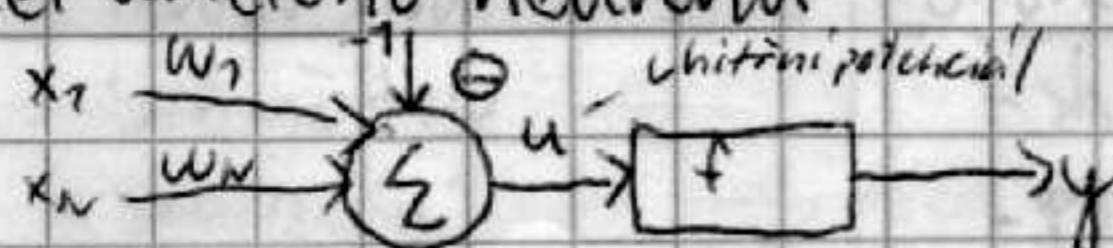
Pacient starší 62, 5 let?

Je přítomen sinus tachykardie



- trénování - nastavení kritérií uzelů → testování - průchod stromem

72. Model umělého neuronu



$$y = f\left(\sum_{i=1}^N w_i x_i - \theta\right)$$

aktivace f
 výstup y
 práh neuronu θ
 váhy vstupů neuronu w_i

- jednoduchý neuron (perceptron)

73. Neuronová aktivace funkce

- převádí vstup neuronu na výstup (např. sigmoidální, hyperbolická, znaménková, heavisidova)



74. Způsoby učení umělých neuronových sítí

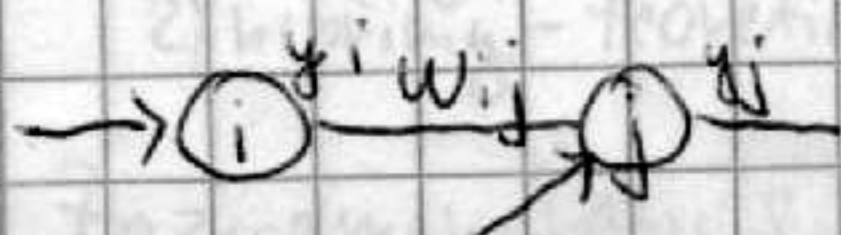
- cíl je nastavit váhy w_i tak, aby síť vstřížela správnou odezvu

a) s učitelem - neuronová síť se učí srovnáváním aktuálního vstupu s výstupem a nastavuje váhy synapsí tak, aby se snížil rozdíl mezi oběma výstupy

b) ~~učitelem~~ bez učitele - váhy jsou nastaveny tak, aby byl stejný výstup u podobných vektorech

75. Hebbův model učení umělých neuronových sítí

- bez učitele; síť a spojení neuronů i s j , pokud jsou oba aktivní \Rightarrow je uměna součinná výstupů



76. Typy umělých neuronových sítí

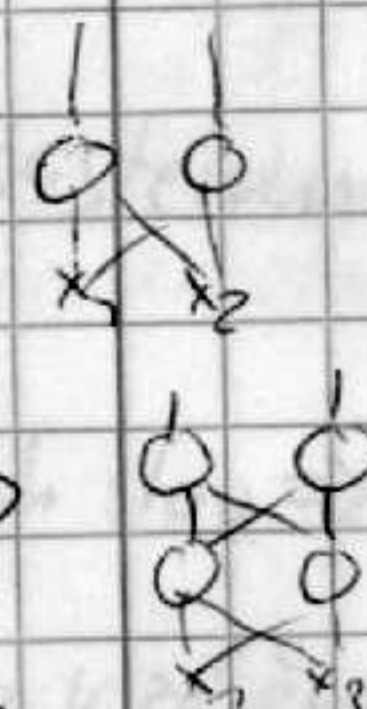
pevné váhy: LAM, Hopfield, BAM

s učitelem: Perceptron, MLP, TDNN

bez učitele: ART 1, 2, Kohonenova síť, ~~neocognitron~~

jednovrstvá
vícevrstvá
rekurentní

úvazí se z výstupu



77. Jednoduchý perceptron

- jednoduchý neuron (jednovrstvá síť s dopředným šířením); učení s učitelem

- použití: pro lineárně separovatelné obrazy

akt. fce - lineární



Trénování:

vah prahu

1) náhodná inicializace w_i a θ 2) vstupní vektor a definice výstupu

3) výpočet odezvy (výstupní hodnota) $y = f(\sum w_i \cdot x_i - \theta)$

4) přizpůsobení vah \rightarrow výstup $d(t)$ je +1 pro třídu A -1 pro B

78. Vícevrstvý perceptron

- vícevrstvá síť s dopředným šířením; učení s učitelem; nelineární síť

- použití: klasifikace obrazů, aproximace fce, predikce čas. řad; řešení

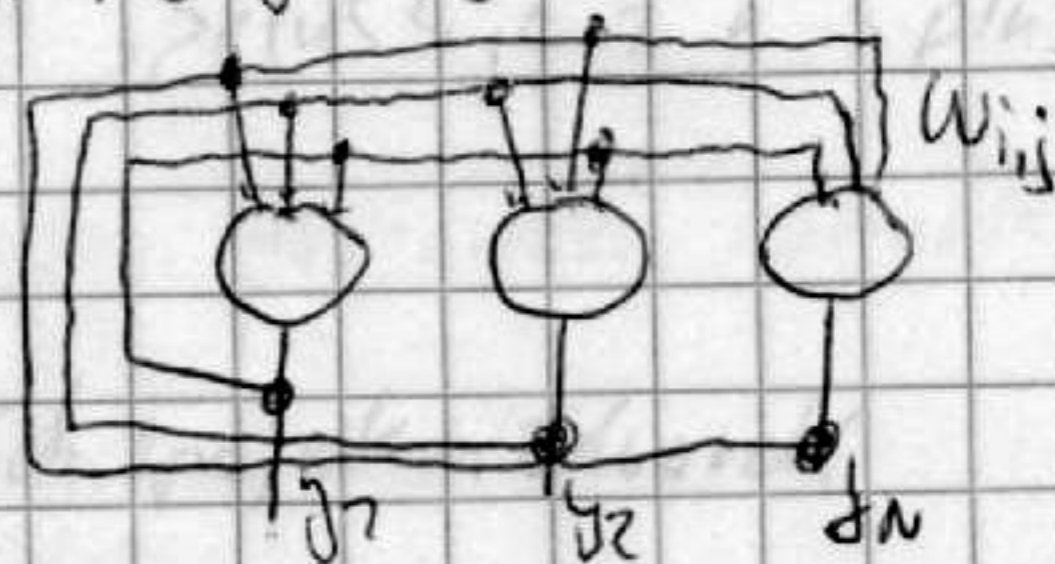
- aktivní neuron. fce: tgh, sigmoidální fce (nelineární, spojité a spojitě diferencovatelné fce)

79. Hopfieldova síť

- jednovrstvá rekurentní síť s pevnými vahami

- vstupní vektory binární nebo bipolární

- použití: rekonstrukce obrazů, autoasociační paměť, problémy optimalizace



80. Vyšší typy neuronových sítí

ART - dvouvrstvá rek. síť

- bez učitele

- shlukování

- rozpoznávání znaků

- porovnávací vrstva

- rozpoznávací vrstva

Kohonenova síť - jednovrstvá s dopředným šířením; bez učitele

- shlukování, analýza dat, sémantické mapy

- porovnání mezi jednotkami; vztěžený neuron přesí svou pozici

81. Neocogniton

- více vrstvá hierarchická nervová síť
- použití: rozpoznávání rukopisných znaků
- je schopný rozpoznat posunutí, natočení obrazy

82. Základní úloha logiky

- nalezení metod správného usuzování \rightarrow přecházení k pravdivším poznatkům k novým

83. Základní poznávání skutečnosti logickými systémy

- 1) přímý - empirický
- 2) nepřímý - teoretický - odvozování nových poznatků ze získaných

84. Zaznamenávání zjištěných poznatků při identifikaci logického systému

- zaznamenáváme vhodným jazykem (syntax = symbol; semantika = význam)
- ze symbolů podle syntakt. pravidel vyjádříme slova. ze správně sestavených slov \Rightarrow formule

85. Dedukce - proces usuzování, kde z premis (předpokladů) docházíme k závěru

Důkaz - konečná posloupnost formulí vedoucích k závěru

86. Důkaz platnosti logických formulí

- a) formule beru jako axiomy (nezvratitelné tvrzení)
- b) pravdivost dokážu posloupností pravidel správného usuzování (z premis)

87. Axiom - nezvratelné tvrzení Premisa - předpoklad; pravidlo správného usuzování

Konkluze - to, co vyplývá z výroku Teorem - formule, kterou dokážeme z axiomů

88. Úplná indukce - musíme projít všechny vzorky

Neúplná - vezmeme jen část vstupních tvrzení: veřejné mínění

89. Elementární výrok - nelze rozdělit na jednodušší výroky „3 plus 4 je 7“

Obecný výrok - tvrzení s hodnotou true/false „3 plus 4 je 7 a 3 plus 3 je 6“

90. Jednoduchý výrok = elementární

Složený - pospojování s jednoduchých, výrokových spojek a závorek

91. Jaké pravdivostní hodnoty může nabývat výrok? \Rightarrow pravda x nepravda

92. Definice výrokové formule

- korektně vytvořený výraz pomocí symbolů výrokové logiky

93. Tautologie - výrok pravdivý bez ohledu na hodnoty jednotlivých členů Kontradikce - výrok nepravdivý \neg \neg

94. Vyhodnocování logických výroků

- pravdivostní tabulka, demorganovy vzorce

95. Logické vyplývání

$A \models B$ A...premise B...konkluze

- formule B vyplývá z A, pokud je B pravdivé vědy, když je A

96. Pravidlo odloučení = modus ponens

- jestliže A a $A \models B$ jsou výrokové tautologie, pak i B je výrokovou tautologií
- odvozování dedukcí

97. Věta o dedukci

- jestli věta B dokazatelná z A_1, \dots, A_n ; pak věta $A_n \models B$ je dokazatelná z A_1, \dots, A_{n-1}

98. Formule formálně dokazatelná - pokud lze získat formuli z dedukce axiomů

Formule logicky vyplývající =

99. Co patří do predikátové logiky 1. řádu

- symboly zabírající výrokové logiky; symboly pro popis objektů a vlastností; kvantifikátory $\forall \exists$

100. Predikátová logika 1. řádu

- "někdo jezdí do práce autem, někdo veřejnou dopravou, někdo chodí pěšky."

$$(\exists x \text{ Jezdi autem}(x)) \wedge (\exists x \text{ Jezdi HD}(x)) \wedge (\exists x \text{ Chodi Pěšky}(x))$$

101. Term = konstanta nebo proměnná dosazovaná do tce $f(t_1, t_2, \dots)$

Atomická formule = výraz $P(x_1, x_2)$; P - m-místný predikátový symbol

Formule - všechny atomické

- formule A, B spojeny výrazem $\wedge \vee \Rightarrow \Leftrightarrow$

Uzavřená F. - taková, kde se nevyskytují volné proměnné $\forall x \exists y (x+y = xy)$

102. klauzule = disjunkce literálů $L_1 \vee L_2 \vee L_3 \vee \dots$

\rightarrow výroková proměnná či její negace (atomická formule)

103. Jak se vyhodnocují kvantifikované formule

- odstraní se kvantifikátory (\forall, \exists) = (převodem do prenexní normální formy)

104. logicky pravdivá = tautologie, pravdivá ve všech interpretacích

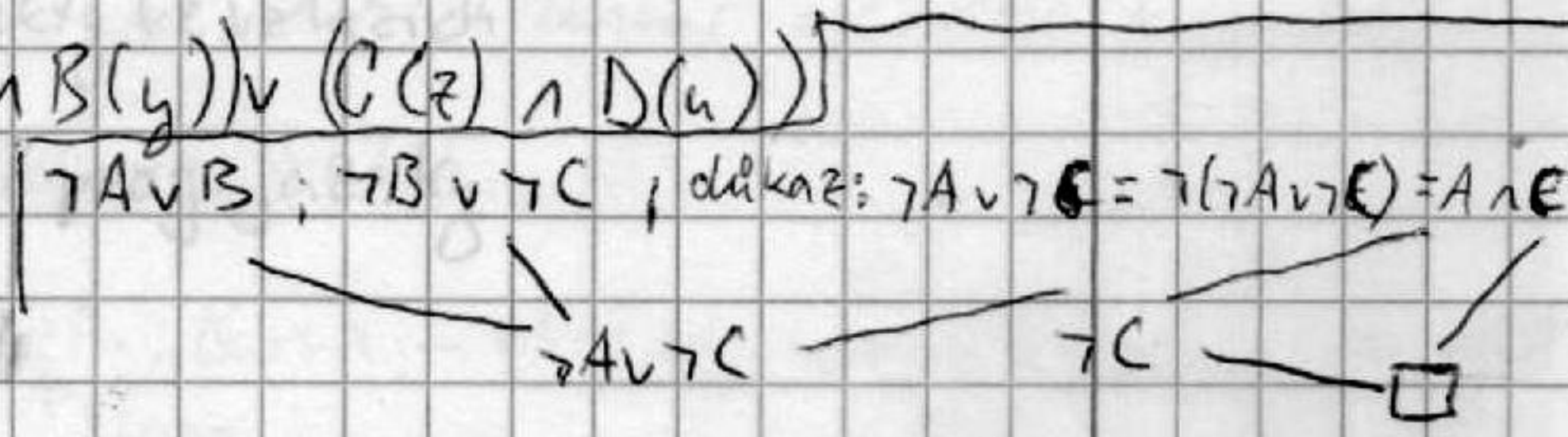
nepravdivá = alespoň jedna interpretace je nepravdivá

splnitelná = alespoň 1 interpretace TRUE

nesplnitelná = kontradikce = všechny interpretace FALSE

105. Disjunktivní normální forma = $(A(x) \wedge B(y)) \vee (C(z) \wedge D(u))$

konjunktivní \Rightarrow opačná



106. Rezolční metoda ve výrokové logice

- když není formule klauzuli, nahradí se ekvivalentem formulí, které jsou klauzuli

107. Rezolční metoda v predi. logice 1. řádu

- stejně, volíme substituce za volné proměnné ve formulích

108. Substituce - dosazení ekvivalentní formule
Instance - výsledný tvar substituce
Základní instance - neobsahuje proměnné
Unifikátor - když se literály po substituci rovnají

109. Rezolventa klauzulí - definice

- disjunkce rodičovských klauzulí s vrhnutím komplementárních literálů
- prostě vsuneš literály, které se doplňují $\Rightarrow A + \neg A = \square$

110. Prenexní normální forma

- pokud formule neobsahuje žádné kvantifikátory

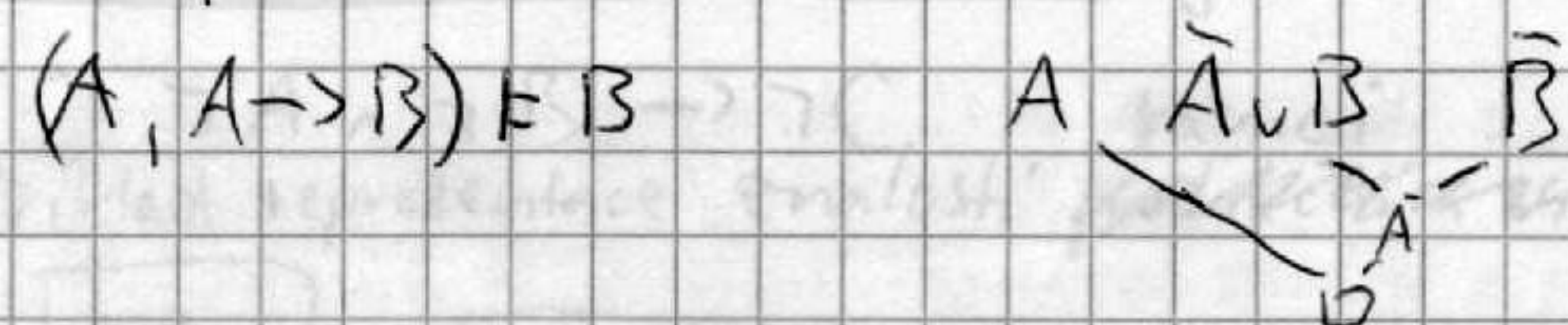
- $(Q_1 x_1) M^1$ matice bez kvantif.
→ prefix $\forall(x) \exists(x)$

- pomocí ekvivalentních úprav se dá každá formule převést do PNF

111. Převést do PNF: $\exists x P(x) \rightarrow \exists x Q(x) \rightarrow \exists x (P(x) \rightarrow Q(x)) \rightarrow \exists x (\neg P(x) \vee Q(x))$

112. Rezolventa - složení dvou rodičovských klauzulí obsahující komplementární liter.

113. Modus ponens rezoluční metodou



114. Co je znalost - postup jak daná data interpretovat do souvislosti

- složky: prvky dat, vlastnosti prvků, Relace a operace mezi nimi

115. = 114.

116. Jaké druhy znalosti rozlišujeme:

- deklarativní - konstatování skutečnosti: znášeng' čístic zvs' si spotřeba
- procedurální - popis, jak dosáhnout cíle: pokud zvs' ena' spotřeba, over čístic'
 ← comandelat

117. Formální systémy reprezentace znalosti

- logické systémy
- produkční systémy - programy založene' na pravidlech chování
- vámce - OOP (vnitřní struktura uzelů)
- Semantické sítě - realita jako objekty ve vztazích
- procedurální systémy a speciální prog. jazyky

118-119. viz 116

120. Předmět reprezentace - to co chce reprezentovat

Prostředek reprezentace - prostředkem čtvím čeho

Reprezentant - vztahový prostředek jak vyjádřit reprezentaci

121. Jaký typ datové struktury jsou reprezentovány znalosti?

- tabulka, graf, strom, halda, spoják

122. Jaká musejí být data a znalosti?

- přehledná, úplná, jednoznačná

123. Fakta - to, co víme Znalost - to, co víme a používáme

124. Odvozování nových faktů dopředným zřetěžením

- je to usuzování řízené daty → získáme data a na jejich základě uvažujeme

- např.: zeptá se jakou mám chuť, co je v lednici a podle toho doporučí recept

125. Odvozování zpětným zřetěžením

- usuzování řízené cílem - dokazuje platnost cíle hledáním dat

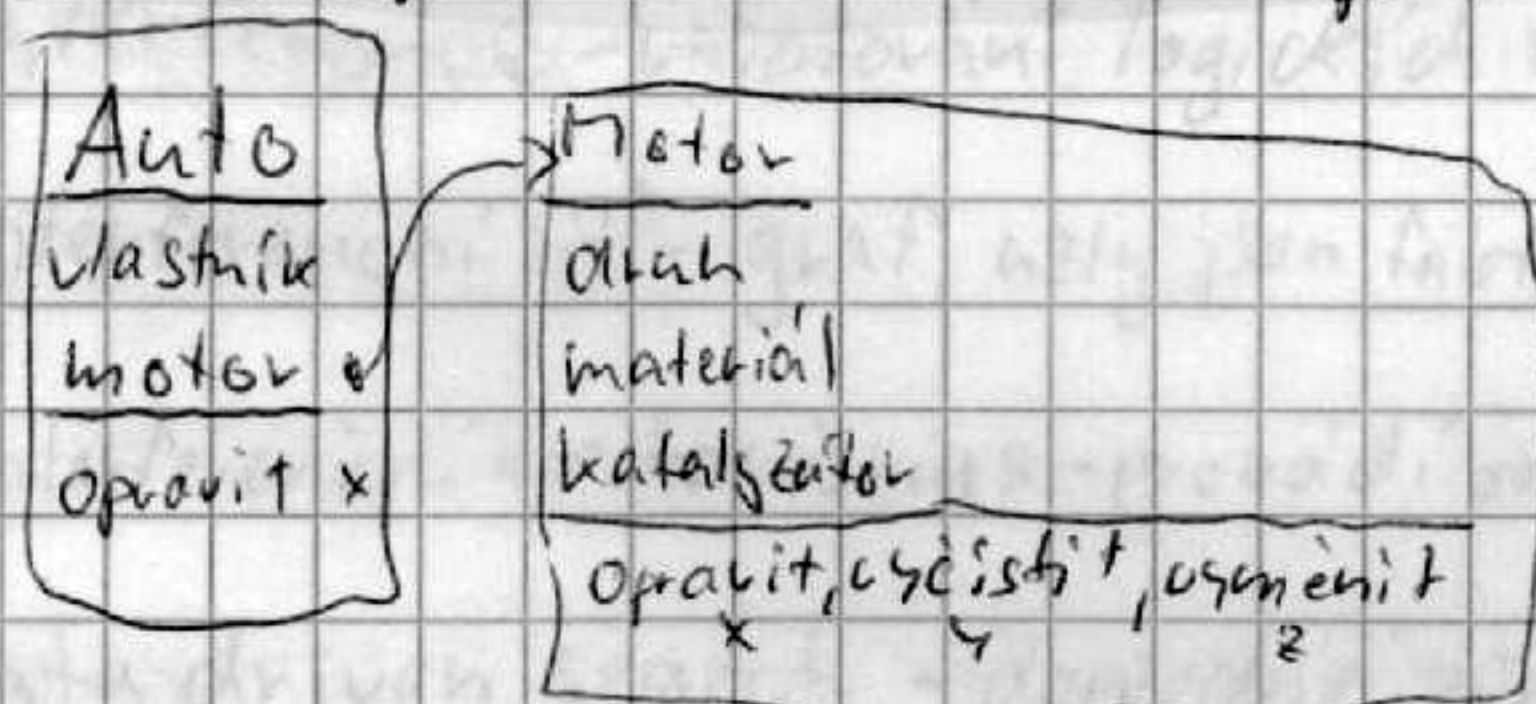
- např.: chci kungpao. Expertní systém na základě stav zásob řekne, jestli je to možné

126. Příklad logické kalkul predikátorovou logikou

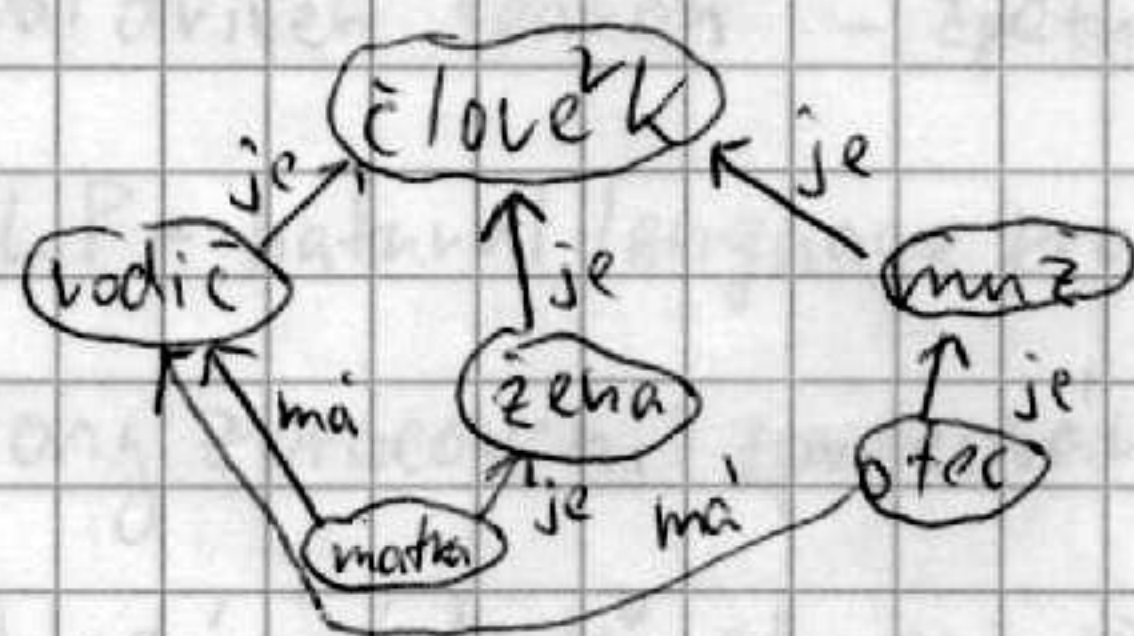
A - motor se točí B - žárovky svítí C - lze nastartovat

$\neg A \wedge \neg B \rightarrow \neg C$ ^{rámcí}

127. Příklad reprezentace znalosti ~~produkčního systému~~



128. Příklad sémantické sítě



129. Příklad procedurální reprezentace znalosti

```
if (a > b) print a
else if (b > a) print b
else print "něco"
```

130. Rámec - struktura dat pro reprezentaci stereotypních situací

- každý rámec má informaci jak ho používat, co očekávat a co dělat když ne

131. Z čeho se skládá reprezentace uzelu rámcové sítě

- název uzlu: Majitel; - vlastnost uzlu: Barva; - akce, když měním a tribut: Notankovat
↑ stopa

132. Z čeho se skládá sémantická síť

- uzly (objekty, koncepce, události) a hrany (metody), trojicí relace mezi uzly

133. Stopa - rámec; atribut rámce - vlastnost uzlu

↑ např. "je", "má část"

134. Rozdíly znalostní a expertní systém

- znalostní je obecnější, ~~expertní~~ vyžaduje znalost experta a vysvětlovací subsystém

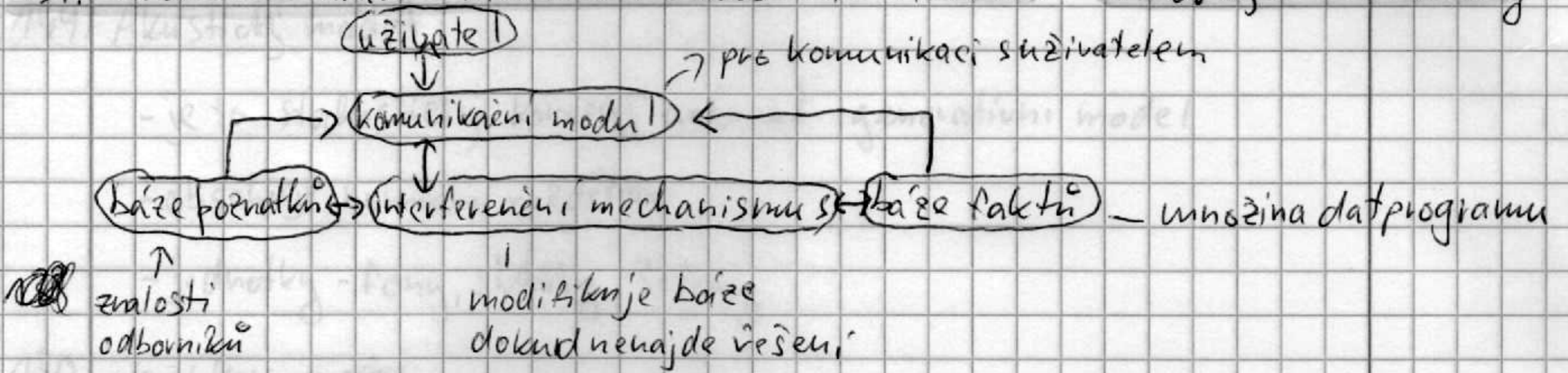
135. Kdy je používán znalostní systém?

- pro řešení úloh vyžadují experta v oboru; diagnostika základy, plánování, rozhodování

136. Jak rozdělujeme znalostní systémy

- diagnostické: prochází data a snaží se určit, která z hypotéz se hodí na daný problém
- plánovací - znám cíl a poč. stav, musíme nalézt posloupnost kroků

137. Nakreslete hrubou strukturu znalostního systému + 138. Vysvětlit moduly



139. K čemu se používá vysvětlovací subsystém

- proč znalostní/expertní systém se tak rozhodl

140. Interference - vyvozování logických závěrů z pravdivých premis

Interferenční síť - graf, uzel je fakt, hrany pravidla

Interferenční mechanismus - provádí odvození z báze

141. Data driven search - dopředné zřetězení - od ~~listu~~ od listu k řešení

Goal driven search - zpětné zřetězení - od cíle k listu

142. NLP - natural language processing = zpracování přirozeného jazyka

143. Úlohy zpracování přirozeného jazyka

- Vyhledávání textů; strojový překlad, oprava pravopisu, třídění emailů
odpovídání na otázky; podpora rozpoznávání

144. Typy systémů komunikujících přirozeným jazykem

- systém typu dotaz - odpověď
- konzultační systémy
- dialogové informační systémy
- znalostní a expertní systémy
- řečové portály

145. Dialogový systém - dokáže s uživatelem komunikovat v jeho jazyce

146. Co je třeba provést při analýze sdělení v přirozeném jazyce?

- snímání, reprezentace, předzpracování signálu
- segmentace a rozpoznání slov
- syntaxická analýza
- semantická analýza
- interpretace sdělení

147. Kroky analýzy promluvy přirozeného jazyka

- a) signálová analýza b) akusticko-fonetická c) lingvistická
- d) extrakce významu a jeho interpretace

148. Vysvětlete a popište úlohu rozpoznávání přirozeného jazyka

- a) zpracování akustického signálu - nahrávání, digitalizace, předzpracování, parametrizace
- b) vytvoření akustického modelu $P(O|W)$
- c) vytvoření jazykového modelu $P(W)$
- d) nalezení nejčastější posloupnosti slov

149. Akustický model

- je to stochastický konečný automat; generativní model
- obsahuje normální rozdělení
- jednotky - fony, slabiky, 3-fony

150. Jazykový model

- určuje pravděpodobnost posloupnosti n slov
- předpovídat další na základě předchozích
- korektura pravopisu; doplnění diakritiky

151. N-gramový jazykový model

- pravděpodobnostní přístup $P(W) = P(w_1 \dots w_n)$
- úspčet z relativní frekvence sekvencí slov na korpusech

př.: Veronika je v kině
Jana v zahradě

testovací sekvence: veronika je v zahradě $P(W) = ?$

$$P(\text{veronika}) = P(\text{kině}) = P(\text{jana}) = P(\text{zahradě}) = \frac{1}{8} = 0,125$$

$$P(\text{je}) = P(v) = \frac{2}{8} = 0,25 \quad P(W) = \left(\frac{1}{8}\right)^2 \cdot \left(\frac{1}{4}\right)^2$$

152. Úloha rozpoznávání entit

- určení významu slov a spojení, které mají předem daný význam
- jména, města, státy
- aby se nestal slojový překlad: Justin Bieber \neq just in beiber