

## Reprezentace a vyvozování znalostí

Aleš Horák

E-mail: [hales@fi.muni.cz](mailto:hales@fi.muni.cz)

<http://nlp.fi.muni.cz/uui/>

Obsah:

- Reprezentace a vyvozování znalostí
- Logika – rezoluční pravidlo
- Extralogické informace – třídy, sémantické sítě, rámce
- Pravidlové systémy
- Nejistota a pravděpodobnost

## Reprezentace a vyvozování znalostí

### REPREZENTACE A VYVOZOVÁNÍ ZNALOSTÍ

otázka:

*Jak zapíšeme znalosti o problému/doméně?*

*Když je zapíšeme, můžeme z nich mechanicky odvodit nová fakta?*

- **reprezentace znalostí** (*knowledge representation*) – hledá způsob vyjádření znalostí počítačově zpracovatelnou formou (za účelem odvozování)
- **vyvozování znalostí** (*reasoning*) – zpracovává znalosti uložené v **bázi znalostí** (*knowledge base, KB*) a provádí **odvození** (*inference*) nových závěrů:
  - odpovědi na dotazy
  - zjištění faktů, které vyplývají z faktů a pravidel v KB
  - odvodit akci, která vyplývá z dodaných znalostí, ...

### REPREZENTACE ZNALOSTÍ

proč je potřeba speciální **reprezentace znalostí**?

vnímání lidí × vnímání počítačů

#### člověk

- když dostane novou věc (třeba pomeranč) – **prozkoumá** a **zapamatuje** si ho (a třeba sni)
- během tohoto procesu člověk zjistí a uloží všechny základní vlastnosti
- později, když se **zmíní** daná věc, vyhledají se a připomenou uložené informace

#### počítač

- musí se spolehnout na informace od lidí
- jednodušší informace – přímé *programování*
- složité informace – zadané v **symbolickém jazyce**

### VOLBA REPREZENTACE ZNALOSTÍ

kteřá **reprezentace znalostí** je **nejlepší**?

Pro řešení skutečně obtížných problémů musíme používat několik různých reprezentací. Každý konkrétní typ datových struktur má totiž své klady a zápory a žádný se sám o sobě nezdá adekvátní pro všechny funkce zahrnuté v tom, čemu říkáme “selský rozum” (*common sense*).

– Marvin Minsky

## LOGIKA – REZOLUČNÍ PRAVIDLO HISTORIE LOGICKÉHO VYVOZOVÁNÍ

450 př.n.l.	stoikové	výroková logika, inference (pravděpodobně)
322 př.n.l.	Aristoteles	inferenční pravidla, kvantifikátory
1565	Cardano	teorie pravděpodobnosti (výroková logika + nejistota)
1847	Boole	výroková logika (znovu)
1879	Frege	predikátová logika 1. řádu
1922	Wittgenstein	důkaz pomocí pravdivostních tabulek
1930	Gödel	$\exists$ úplný algoritmus pro PL1
1930	Herbrand	úplný algoritmus pro PL1 (redukce na výroky)
1931	Gödel	$\neg\exists$ úplný algoritmus pro aritmetiku
1960	Davis/Putnam	“praktický” algoritmus pro výrokovou logiku
1965	Robinson	“praktický” algoritmus pro PL1 – rezoluce

## PŘEDPOKLAD UZAVŘENÉHO SVĚTA

2 užitečné předpoklady:

→ **předpoklad uzavřeného světa** (*closed world assumption*)

- cokoliv o čem **nevíme**, že je **pravda** → bereme za dané, že je to **nepravda**
- využitý např. v Prologu (negace jako neúspěch)

→ **předpoklad jednoznačných pojmenování** (*unique names assumption*)

- různá jména označují různé objekty

## LOGIKA – REZOLUČNÍ PRAVIDLO

vyvozování nových znalostí = hledání **důkazu**

algoritmus konstrukce důkazu:

- dopředné a zpětné řetězení – neúplné pro PL1 (úplné pro Hornovy klauzule)
- rezoluce – úplná pro důkaz sporem
- logické programování – SLD rezoluce

## REZOLUCE V PL1

vyvozování v PL1 je pouze **částečně rozhodnutelné**:

- může najít důkaz  $\alpha$ , když  $KB \models \alpha$
- nemůže vždy dokázat, že  $KB \not\models \alpha$   
viz *problém zastavení* – důkazová procedura nemusí skončit

**rezoluce je důkaz sporem:**

pro důkaz  $KB \models \alpha$  ukážeme, že  $KB \wedge \neg\alpha$  je nespílitelné

rezoluce používá  $KB, \neg\alpha$  v **konjunktivní normální formě** (CNF), např.:

$$\begin{aligned} (P \vee Q) \Rightarrow (Q \Leftrightarrow R) &\equiv && (\neg P \vee \neg Q \vee R) \\ &&& \wedge (\neg P \vee Q \vee \neg R) \\ &&& \wedge (\neg Q \vee R) \end{aligned}$$

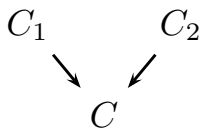
## KONJUNKTIVNÍ NORMÁLNÍ FORMA (CNF)

Algoritmus pro převod každé PL1 klauzule do CNF:

1. převedeme implikace na disjunkce:  $P \Rightarrow Q \rightarrow \neg P \vee Q$
2. přesuneme  $\neg$  dovnitř k literálům:  $\neg \forall x P \rightarrow \exists x \neg P$
3. přejmenujeme proměnné:  $\forall x P \vee \exists x Q \rightarrow \forall x P \vee \exists y Q$
4. přesuneme kvantifikátory doleva:  $\forall x P \vee \exists y Q \rightarrow \forall x \exists y P \vee Q$
5. eliminujeme  $\exists$  pomocí Skolemizace:  $\exists x P(x) \rightarrow P(c_1)$ ,  
 $\forall x P(x) \Rightarrow \exists y Q(y) \rightarrow \forall x P(x) \Rightarrow Q(f(x))$
6. zahodíme univerzální kvantifikátory
7. roznásobíme  $\wedge$  pomocí  $\vee$ :  $(P \wedge Q) \vee R \rightarrow (P \vee R) \wedge (Q \vee R)$

## REZOLUČNÍ PRAVIDLO

algoritmus je založen na opakované aplikaci **rezolučního pravidla** – ze dvou klauzulí odvoď novou klauzuli



→ klauzule:  $C_1 = P_1 \vee P_2 \vee \dots \vee P_n$

a  $C_2 = \neg P_1 \vee Q_1 \vee Q_2 \vee \dots \vee Q_m$

→ výsledek:  $C = P_2 \vee P_3 \vee \dots \vee P_n \vee Q_1 \vee Q_2 \vee \dots \vee Q_m$

→ vyruší se opačné literály  $P_1$  a  $\neg P_1$

postup **rezolučního důkazu tvrzení  $F$** :

- začneme s  $\neg F$
- rezolvujeme s klauzulí z KB (která obsahuje  $F$ )
- opakujeme až do odvození **prázdné klauzule**  $\square$
- když se to podaří  $\rightarrow$  došli jsme ke sporu (pro  $\neg F$ )  $\rightarrow$  **musí platit  $F$**

## REZOLUCE – PŘÍKLAD

→ pravidla

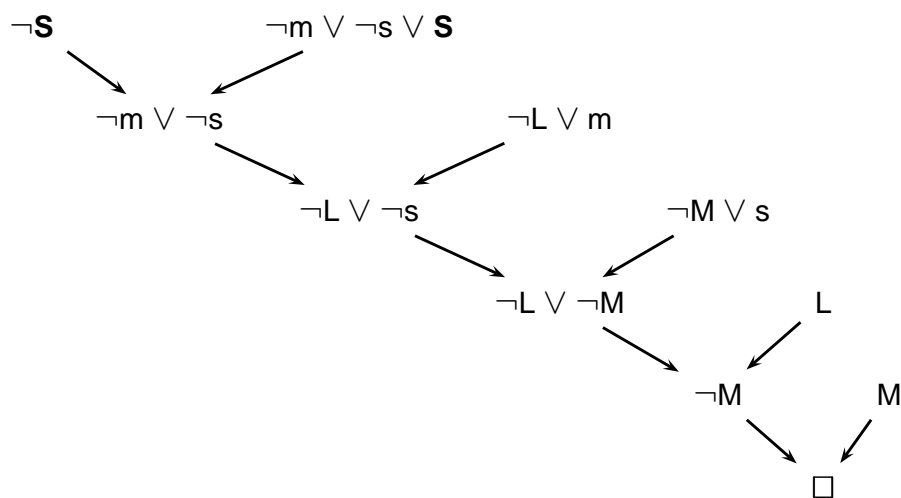
- mráz  $\wedge$  srážky  $\Rightarrow$  sněží  
 $\neg$ mráz  $\vee \neg$ srážky  $\vee$  sněží
- Leden  $\Rightarrow$  mráz  
 $\neg$ Leden  $\vee$  mráz
- mraky  $\Rightarrow$  srážky  
 $\neg$ mraky  $\vee$  srážky

→ fakta – Leden, mraky

→ dotaz (co se má dokázat) – sněží?

## DŮKAZ TVRZENÍ “SNĚŽÍ”

S – sněží, s – srážky, m – mráz, L – Leden, M – mraky



$\neg m \vee \neg s \vee S$ $\neg L \vee m$ $\neg M \vee s$ L, M
---

## EXTRALOGICKÉ INFORMACE

### co jsme dosud ignorovali:

- objekty reálného světa mají mezi sebou **vztahy**
  - třídy/kategorie, podtřídy  $\times$  nadtřídy
  - hierarchie vztahů části/celku
  - dědění vlastností v hierarchiích
- stav světa se může **měnit** v čase
  - explicitní reprezentace času
  - nemonotónní uvažování (pravdivost se může měnit v čase)
- ne každá informace je “černobílá”
  - nejistota
  - statistika, fuzzy logika

## TŘÍDY OBJEKTŮ

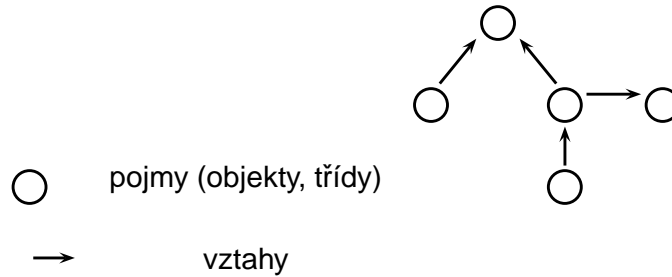
- “Chci si koupit fotbalový míč.”
  - *Chci si koupit FM27341* – špatně
  - *Chci si koupit objekt, který je prvkem třídy fotbalových míčů* – správně
- objekty jsou organizovány do **hierarchie tříd**
  - $FM27341 \in \text{fotbalové\_míče}$
  - $\text{fotbalové\_míče} \subset \text{míče}$
- fakta (objekty)  $\times$  pravidla (třídy)
  - Všechny míče jsou kulaté.
  - Všechny fotbalové míče mají X cm v průměru.
  - FM27341 je červenomodrobílý.
  - FM27341 je fotbalový míč.
  - (Proto: FM27341 je kulatý a má X cm v průměru.)

## SÉMANTICKÉ SÍTĚ

**sémantické sítě** – reprezentace faktových znalostí (pojmy + vztahy)

→ vznikly kolem roku 1960 pro reprezentaci významu anglických slov

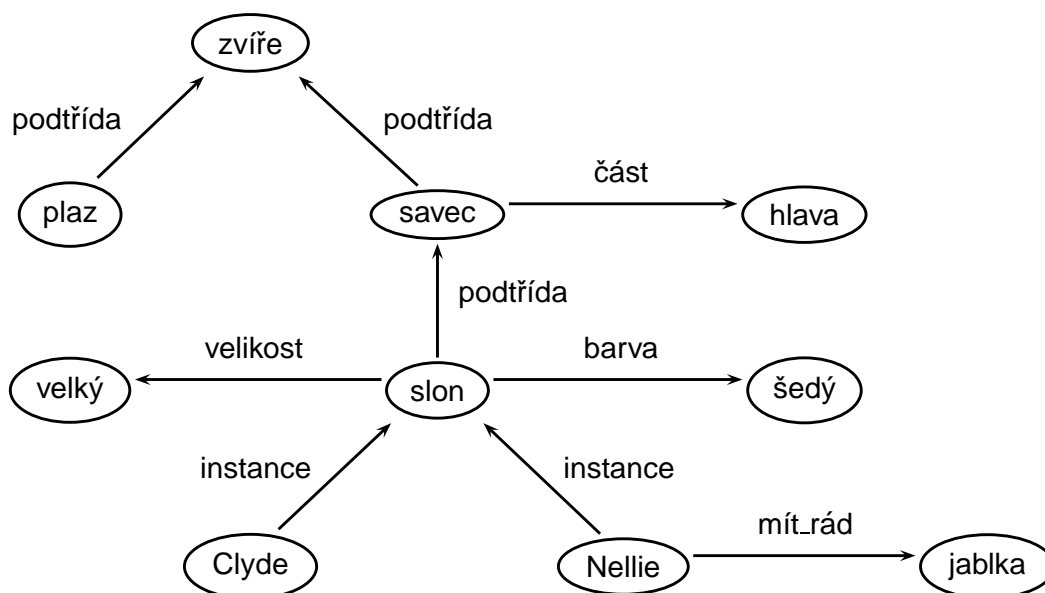
→ znalosti jsou uloženy ve formě grafu



→ nejdůležitější vztahy:

- **podtřída** (*subclass*) – vztah mezi třídami
  - **instance** – vztah mezi konkrétním objektem a jeho rodičovskou třídou
- jiné vztahy – část (*has-part*), barva, ...

## SÉMANTICKÉ SÍTĚ – PŘÍKLAD





## DĚDIČNOST V SÉMANTICKÝCH SÍTÍCH

- pojem sémantické sítě *předchází* OOP
- **dědičnost:**
  - jestliže určitá vlastnost platí pro třídu → platí i pro všechny její podtřídy
  - jestliže určitá vlastnost platí pro třídu → platí i pro všechny prvky této třídy
- určení hodnoty vlastnosti – rekurzivní algoritmus
- potřeba specifikovat i výjimky – mechanismus **vzorů** a **výjimek** (*defaults and exceptions*)
  - vzor – hodnota vlastnosti u třídy nebo podtřídy, platí ta, co je blíže objektu
  - výjimka – u konkrétního objektu, odlišná od vzoru

## DĚDIČNOST VZTAHŮ ČÁST/CELEK

- “krávy mají 4 nohy.”
  - každá noha je částí krávy
- “Na poli je (konkrétní) kráva.”
  - všechny části krávy jsou taky na poli
- “Ta kráva (na poli) je hnědá (celá).”
  - všechny části té krávy jsou hnědé
- “Ta kráva je šťastná.”
  - ~~všechny části té krávy jsou šťastné~~ – neplatí
- lekce: některé vlastnosti jsou děděny částmi, některé nejsou  
explicitně se to vyjadřuje pomocí pravidel jako
$$\text{part-of}(x, y) \wedge \text{location}(y, z) \Rightarrow \text{location}(x, z)$$

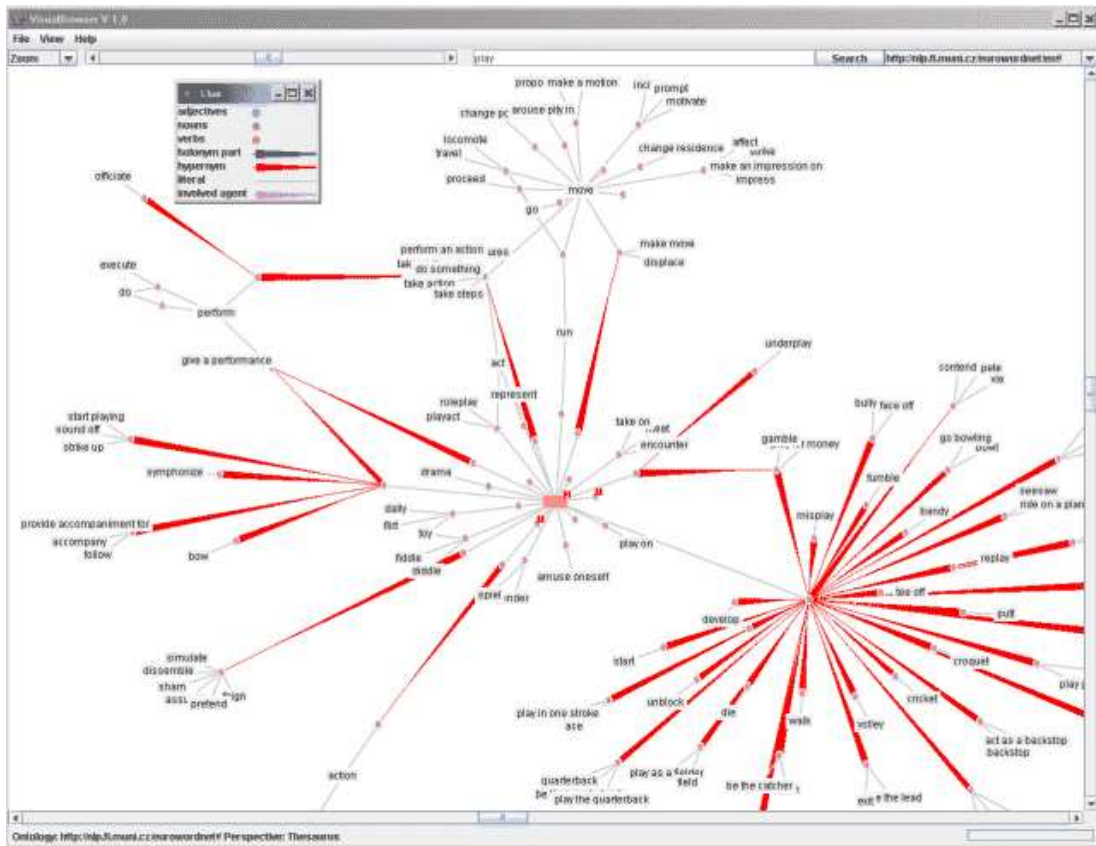
## VZORY A VÝJIMKY – PŘÍKLAD

- “všichni ptáci mají křídla.”
- “všichni **ptáci** umí létat.”
- “ptáci se zlomenými křídly jsou ptáci, ale neumí létat.”
- “**tučnáci** jsou ptáci, ale neumí létat.”
- “**kouzelní tučnáci** jsou tučnáci, kteří umí létat.”
- kdo umí létat:
  - “Penelope je pták.” ⇒ “Penelope **umí** létat.”
  - “Penelope je tučnák.” ⇒ “Penelope **neumí** létat.”
  - “Penelope je kouzelný tučnák.” ⇒ “Penelope **umí** létat.”
- všimněte si, že víra v hodnotu vlastnosti objektu se může měnit s příchodem nových informací o klasifikaci objektu

## APLIKACE SÉMANTICKÝCH SÍTÍ

(Princeton) **WordNet** – <http://wordnet.princeton.edu/>

- sémantická síť 100.000 (anglických) pojmů, zachycuje:
  - synonyma, antonyma (významově stejná/opačná)
  - hyperonyma, hyponyma (podtřídy)
  - odvozenost a další jazykové vztahy
- tvoří se **národní wordnety** (navázané na anglický WN)  
český wordnet – cca 30.000 pojmů
- nástroj na editaci národních wordnetů – DEBVisDic/VisDic, vyvinutý na FI MU –  
<http://deb.fi.muni.cz/>
- VisualBrowser – <http://nlp.fi.muni.cz/projekty/visualbrowser/>  
nástroj na vizualizaci (sémantických) sítí, vznikl jako DP na FI MU



## RÁMCE

**Rámce** (*frames*):

- varianta sémantických sítí
- velice populární pro reprezentaci znalostí v expertních systémech
- všechny informace relevantní pro daný pojem se ukládají do univerzálních struktur – **rámců**
- stejně jako sémantické sítě, rámce podporují dědičnost
- OO programovací jazyky vycházejí z teorie rámců

## RÁMCE – PŘÍKLAD

rámec obsahuje **objekty**, *sloty* a hodnoty slotů

příklady rámců:

**savec:**

*podtřída:* zvíře

*část:* hlava

*\*má\_kožich:* ano

**slon:**

*podtřída:* savec

*\*barva:* šedá

*\*velikost:* velký

**Nellie:**

*instance:* slon

*mít\_rád:* jablka

'\*' označuje **vzorové hodnoty**, které mohou měnit hodnoty u podtříd a instancí

## SÉMANTICKÉ SÍTĚ × RÁMCE

sémantické sítě	rámce
uzly	objekty
spoje	sloty
uzel na druhém konci spoje	hodnota slotu

**deskripční logika** – logický systém, který manipuluje přímo s rámci

## Pravidlové systémy

### PRAVIDLOVÉ SYSTÉMY

- snaha zachytit **produkčními pravidly** znalosti, které má expert
- obecná forma pravidel
  - IF        podmínka
  - THEN    akce
- podmínky – booleovské výrazy, dotazy na hodnoty **proměnných**
- akce – nastavení hodnot proměnných, příznaků, ...
- důležité vlastnosti:
  - znalosti mohou být strukturovány do modulů
  - systém může být snadno rozšířen přidáním nových pravidel beze změny zbytku systému

## PRAVIDLOVÁ BÁZE ZNALOSTÍ – PŘÍKLAD

pravidla pro **oblékání**:

- pravidlo 1 IF X je seriózní  
AND X bydlí ve městě  
THEN X by měl nosit sako
- pravidlo 2 IF X je akademik  
AND X je společensky aktivní  
AND X je seriózní  
THEN X by měl nosit sako a kravatu
- pravidlo 3 IF X bydlí ve městě  
AND X je akademik  
THEN X by měl nosit kravatu
- pravidlo 4 IF X je podnikatel  
AND X je společensky aktivní  
AND X je seriózní  
THEN X by měl nosit sako, ale ne kravatu

**společenská** pravidla:

- pravidlo 5 IF X je podnikatel  
AND X je ženatý  
THEN X je společensky aktivní
- pravidlo 6 IF X je akademik  
AND X je ženatý  
THEN X je seriózní

**profesní** pravidla:

- pravidlo 7 IF X učí na univerzitě  
OR X učí na vysoké škole  
THEN X je akademik
- pravidlo 8 IF X vlastní firmu  
OR X je OSVČ  
THEN X je podnikatel

## EXPERTNÍ SYSTÉMY

- aplikace pravidlových systémů
- zaměřeny na specifické oblasti – medicínská diagnóza, návrh konfigurace počítače, expertíza pro těžbu ropy, ...
- snaha zachytit **znalosti experta** pomocí pravidel  
ale znalosti experta zahrnují – postupy, strategie, odhady, ...
- expertní systém musí pracovat s procedurami, nejistými znalostmi, různými formami vstupu
- vhodné oblasti pro nasazení expertního systému:
  - **diagnóza** – hledání řešení podle symptomů
  - **návrh konfigurace** – složení prvků splňujících podmínky
  - **plánování** – posloupnost akcí splňujících podmínky
  - **monitorování** – porovnání chování s očekávaným chováním, reakce na změny
  - **řízení** – ovládání složitých komplexů
  - **předpovědi** – projekce pravděpodobných závěrů z daných skutečností
  - **instruktáž** – inteligentní vyučování a zkoušení studentů

### NEJISTOTA

definujme akci  $A_t$  jako “Vyrazit na letiště  $t$  hodin před odletem letadla.”

jak najít odpověď na otázku “Dostanu se akcí  $A_t$  na letiště včas k odletu letadla?”

problémy:

1. částečná pozorovatelnost (stav vozovky, záměry ostatních řidičů, ...)
2. nejistota výsledků akcí (píchnutí kola, ...)
3. obrovská složitost modelování a předpovědi dopravní situace

čistě logický přístup tedy:

- riskuje chybu – “ $A_5$  mě tam dostane včas.”
- vede k závěrům, které jsou příliš slabé pro rozhodování: “ $A_5$  mě tam dostane včas, pokud nebude na dálnici nehoda a pokud nebude pršet a jestli nepíchnu kolo a jestli nebude fronta na odbavovacích přepážkách a jestli nebudou problémy při kontrole zavazadel ...”

### METODY PRO PRÁCI S NEJISTOTOU

#### defaultní/nemonotónní logika

Předpokládejme, že nepíchnu cestou kolo.

Předpokládejme, že  $A_5$  bude OK, pokud se nenajde protipříklad.

#### logická pravidla s faktory nejistoty (zastaralé)

$A_5 \mapsto_{0.3}$  dostat se na letiště včas.

zalévání  $\mapsto_{0.99}$  mokrý trávník

mokrý trávník  $\mapsto_{0.7}$  déšť

#### pravděpodobnost (míra předpokladu, že hodnota bude true)

Vzhledem k dostupným informacím,  $A_3$  mě tam dostane včas s pravděpodobností 0.05.

poznámka: fuzzy logika se zabývá **mírou pravdivosti**, NE **pravděpodobností**

## PRAVDĚPODOBNOST

Pravděpodobnost **sumarizuje** nejistotu pocházející z

- **lenosti** – nepodařilo se vypočítat všechny výjimky, podmínky, ...
- **neznalosti** – nedostatek relevantních údajů, počátečních podmínek, ...

**subjektivní** × **Bayesovská** pravděpodobnost:

- pravděpodobnostní vztah mezi tvrzením a jeho pravdivosti vzhledem k podmínkám:

$$P(A_4 | \text{žádné hlášené nehody}) = 0.5$$

nejedná se o vyjádření **pravděpodobnostní tendence** (ale může se získat ze znalostí podobných případů v minulosti)

- pravděpodobnost tvrzení se může měnit s novými (vstupními) podmínkami:

$$P(A_4 | \text{žádné hlášené nehody, je 4:00 ráno}) = 0.63$$

**Bayesovská síť** – acyklický orientovaný graf, uzly obsahují **tabulky podmíněných pravděpodobností** rodičů, umožňuje efektivní přesné nebo přibližné (Monte Carlo) **vyvozování**. Nejčastěji používaný aparát pro vyvozování z nejistých znalostí.

## VYVOZOVÁNÍ Z NEJISTÝCH ZNALOSTÍ

→ použití **náhodných proměnných** (*random variables*) – funkce, která vzorkům přiřazuje hodnoty → vrací výsledky měření sledovaného jevu

**distribuce pravděpodobností** náhodné proměnné = (vektor) pravděpodobnost(i), že daná náhodná proměnná bude mít určitou konkrétní hodnotu

např.: náhodná proměnná *Odd* vyjadřující, že výsledek hodu kostkou bude lichý

náhodná proměnná *Weather* vyjadřující, jaké bude počasí (slunce, déšť, mraky, sníh)

$$Odd(1) = true \quad Weather(21.11.2005) = \text{déšť}$$

distribuce pravděpodobností proměnných *Odd* a *Weather*

$$P(Odd = true) = 1/6 + 1/6 + 1/6 = 1/2$$

$$P(Odd) = \langle 1/2, 1/2 \rangle$$

$$P(Weather) = \langle 0.72, 0.1, 0.08, 0.1 \rangle$$

→ pravidla pro výpočet pravděpodobnosti logicky souvisejících událostí

$$P(a \vee b) = P(a) + P(b) - P(a \wedge b)$$



## BAYESOVSKÉ PRAVIDLO PRO VYVOZOVÁNÍ

pravidlo pro **podmíněnou pravděpodobnost** –  $P(a|b) = \frac{P(a \wedge b)}{P(b)}$  if  $P(b) \neq 0$

z toho lze odvodit **Bayesovské pravidlo** pro určení **diagnostické** pravděpodobnosti ze znalosti **příčinné** pravděpodobnosti:

$$P(\text{Příčina}|\text{Následek}) = \frac{P(\text{Následek}|\text{Příčina})P(\text{Příčina})}{P(\text{Následek})}$$

např. *ZMB* zánět mozkových blan, *ZK* ztuhlý krk:

$$P(zmb|zk) = \frac{P(zk|zmb)P(zmb)}{P(zk)} = \frac{0.8 \times 0.0001}{0.1} = 0.0008$$

vyvozování =

1. rozdělení akce na **atomické události**
2. zjištění pravděpodobností atomických událostí
3. výpočet/odvození pravděpodobností pomocí **složených distribucí pravděpodobností** (*joint probability distribution*)