

# 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
- Pravidlové systémy
- Nejistota a pravděpodobnost

# 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**?

# 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 sní)
- během tohoto procesu člověk **zjistí** a **uloží** všechny základní vlastnosti předmětu
- 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

která reprezentace znalostí je nejlepší?

# Volba reprezentace znalostí

která **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

# Obsah

- 1 Repräsentace a vyvozování znalostí
  - Repräsentace znalostí
- 2 Logika – rezoluční pravidlo
  - Předpoklad uzavřeného světa
  - Logika – rezoluční pravidlo
  - Rezoluční pravidlo
- 3 Extralogické informace
  - Třídy objektů
  - Ontologie
  - Sémantické sítě
  - Rámce
- 4 Pravidlové systémy
  - Pravidlová báze znalostí
  - Expertní systémy
- 5 Nejistota a pravděpodobnost
  - Nejistota
  - Pravděpodobnost

# 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  $KB \models \alpha$   
viz *problém zastavení* – důkazová procedura nemusí skončit  
nejde použít pro **generování**, pouze pro **vyvrácení**

**rezoluce** je **důkaz sporem**:

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

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

$$(P \vee Q) \Rightarrow (Q \Leftrightarrow R) \quad \equiv \quad \begin{array}{l} (\neg P \vee \neg Q \vee R) \\ \wedge (\neg P \vee Q \vee \neg R) \\ \wedge (\neg Q \vee R) \end{array}$$

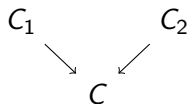
# Konjunktivní normální forma (CNF)

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

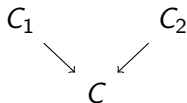
algoritmus je založen na opakované aplikaci **rezolučního pravidla** – ze dvou klauzulí odvodí novou klauzuli



- 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 (rezolventa):  
 $C = P_2 \vee P_3 \vee \dots \vee P_n \vee Q_1 \vee Q_2 \vee \dots \vee Q_m$
- vyruší se opačné literály  $P_1$  a  $\neg P_1$

# Rezoluční pravidlo

algoritmus je založen na opakované aplikaci **rezolučního pravidla** – ze dvou klauzulí odvodí 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 (rezolventa):  
 $C = P_2 \vee P_3 \vee \dots \vee P_n \vee Q_1 \vee Q_2 \vee \dots \vee Q_m$
- 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  $F$**

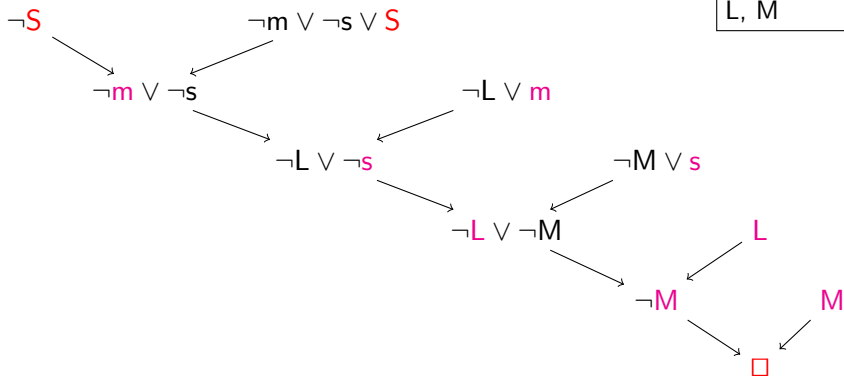
# Rezoluce – příklad

- pravidla
  - $\text{mráz} \wedge \text{srážky} \Rightarrow \text{sněží}$   
 $\neg \text{mráz} \vee \neg \text{srážky} \vee \text{sněží}$
  - $\text{Leden} \Rightarrow \text{mráz}$   
 $\neg \text{Leden} \vee \text{mráz}$
  - $\text{mraky} \Rightarrow \text{srážky}$   
 $\neg \text{mraky} \vee \text{srážky}$
- 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





# Obsah

- 1 Reprezentace a vyvozování znalostí
  - Reprezentace znalostí
- 2 Logika – rezoluční pravidlo
  - Předpoklad uzavřeného světa
  - Logika – rezoluční pravidlo
  - Rezoluční pravidlo
- 3 Extralogické informace
  - Třídy objektů
  - Ontologie
  - Sémantické sítě
  - Rámce
- 4 Pravidlové systémy
  - Pravidlová báze znalostí
  - Expertní systémy
- 5 Nejistota a pravděpodobnost
  - Nejistota
  - Pravděpodobnost

# Extralogické informace – třídy, sémantické sítě, rámce

## 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.*)

# Ontologie

- **ontologie** ve filozofii – nauka o **existenci** a typech existencí
- **ontologie** v informatice – **formální popis znalostí**, pojmy a vztahy mezi pojmy, hierarchie
- ontologie **obecné** (*upper level*) × **doménové**
- různé dostupné obecné, žádná standardem (zatím):  
(Open)Cyc, SUMO/MILO, Dublin Core, DOLCE, ...  
<http://www.ontologyportal.org>

# Ontologie

- **ontologie** ve filozofii – nauka o **existenci** a typech existencí
- **ontologie** v informatice – **formální popis znalostí**, pojmy a vztahy mezi pojmy, hierarchie
- ontologie **obecné** (*upper level*) × **doménové**
- různé dostupné obecné, žádná standardem (zatím):  
(Open)Cyc, SUMO/MILO, Dublin Core, DOLCE, ...  
<http://www.ontologyportal.org>

```
(=>
  (and
    (instance ?KILL Killing)
    (patient ?KILL ?OBJ))
  (exists (?DEATH)
    (and
      (instance ?DEATH Death)
      (experiencer ?DEATH ?OBJ)
      (causes ?KILL ?DEATH))))
```

*Pokud nějaký proces (?KILL) je instancí zabíjení (Killing) a nějaký agent (?OBJ) je předmětem toho procesu*

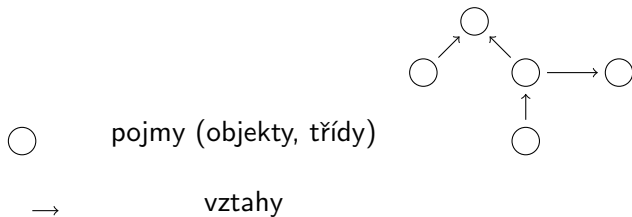
=>

*pak existuje jiný proces (?DEATH) takový, že tento jiný proces je instancí smrti (Death) a agent ?OBJ se účastní tohoto jiného procesu ?DEATH a původní proces ?KILL je příčinou tohoto jiného procesu ?DEATH*

# Sémantické sítě

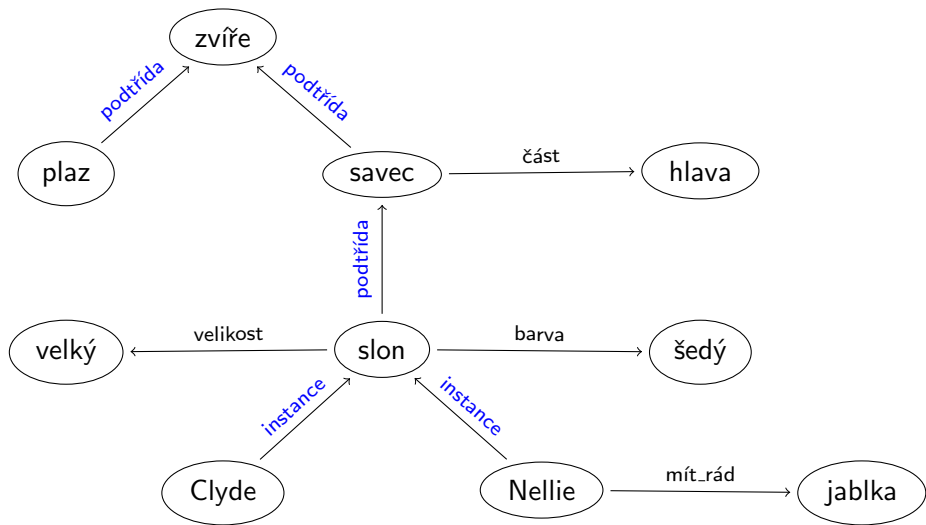
**sémantické sítě** – reprezentace faktových znalostí (pojmy + vztahy)

- vznikly kolem roku 1960 pro reprezentaci významu anglických slov
- znalosti jsou uloženy ve formě grafu



- nejdůležitější vztahy – **taxonomie**:
  - **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

# 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
$$part-of(x, y) \wedge location(y, z) \Rightarrow location(x, z)$$

# Vzory a výjimky – příklad

- “Všichni ptáci mají křídla.”
- “Všichni **ptáci** umí létat.”
- “Ptáci se zlomenými křídly jsou ptáci, ale neumí létat.”
- “**Tučňáci** jsou ptáci, ale neumí létat.”
- “**Kouzelní tučňáci** jsou tučňáci, kteří umí létat.”
- kdo umí létat:
  - “Penelope je pták.”
  - “Penelope je tučňák.”
  - “Penelope je kouzelný tučňák.”

# Vzory a výjimky – příklad

- “Všichni ptáci mají křídla.”
  - “Všichni **ptáci** umí létat.”
  - “Ptáci se zlomenými křídly jsou ptáci, ale neumí létat.”
  - “**Tučňáci** jsou ptáci, ale neumí létat.”
  - “**Kouzelní tučňáci** jsou tučňáci, kteří umí létat.”
  - kdo umí létat:
    - “Penelope je pták.”
    - “Penelope je tučňák.”
    - “Penelope je kouzelný tučňák.”
- ⇒ “Penelope **umí** létat.”

# Vzory a výjimky – příklad

- “Všichni ptáci mají křídla.”
- “Všichni **ptáci** umí létat.”
- “Ptáci se zlomenými křídly jsou ptáci, ale neumí létat.”
- “**Tučňáci** jsou ptáci, ale neumí létat.”
- “**Kouzelní tučňáci** jsou tučňáci, kteří umí létat.”
- kdo umí létat:
  - “Penelope je pták.”  $\Rightarrow$  “Penelope **umí** létat.”
  - “Penelope je tučňák.”  $\Rightarrow$  “Penelope **neumí** létat.”
  - “Penelope je kouzelný tučňák.”

# Vzory a výjimky – příklad

- “Všichni ptáci mají křídla.”
- “Všichni **ptáci** umí létat.”
- “Ptáci se zlomenými křídly jsou ptáci, ale neumí létat.”
- “**Tučňáci** jsou ptáci, ale neumí létat.”
- “**Kouzelní tučňáci** jsou tučňáci, kteří umí létat.”
- kdo umí létat:
  - “Penelope je pták.”  $\Rightarrow$  “Penelope **umí** létat.”
  - “Penelope je tučňák.”  $\Rightarrow$  “Penelope **neumí** létat.”
  - “Penelope je kouzelný tučňák.”  $\Rightarrow$  “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 150.000 (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





The screenshot displays the DEBVisDic application interface. At the top, the main menu includes 'User', 'Settings', 'Tools', 'Windows', and 'Help'. The 'Tools' menu is open, showing options: 'Dictionary - SSJČ', 'dictionary - SSČ', 'Morph. analyzer ajka', and 'Google'. Below this, several wordnet windows are visible:

- English Wordnet:** Search for 'dog'. Results include: [n] andiron:1, fire-dog:1, dog:7, dog; [n] frump:1, dog:2; [n] cad:1, boulder:1, blackguard:1, dog:4, hound:1; [n] dog:1, domestic dog:1, Canis familiaris:1; [n] frank:2.
- Greek Wordnet:** Search for 'οὐδὲ ἰκό'. Results include: [n] περι τοὺ ἰκό:1; [n] περι τοὺ ἰκό:0.
- Czech Wordnet:** Search for 'pes'. Results include: [n] zakopaný pes:1; [n] policejní pes:1; [n] hlíďač:4, hlíďací pes:1; [n] pes:1; [n] slepecký pes:1, vodící pes:1.
- Russian Wordnet:** Search for 'журнал'. Results include: [n] журнал:1.

The Russian window also shows a detailed view for 'журнал' with POS: n, ID: RUS-1234560515, and Synonyms: книга:1. A context menu is open over this window, listing actions like 'Show in Czech Wordnet', 'Take key from Czech Wordnet', 'AutoLookUp in', 'Copy entry to Czech Wordnet', and 'Import IDs from file'. The bottom of the Russian window shows 'Number of entries: 1'.

The Czech window shows XML-like markup for the entry 'pes':

```

-<SYNONYM>
<LITERAL lnote="" sense="1">pes</LITERAL>
<WORD>pes</WORD>
</SYNONYM>
<ILR type="hypernym">ENG20-020005
<ILR type="holo_member">ENG20-020006
<ILR type="holo_member">ENG20-075
<STAMP>xcapek1 2003/06/25</STAMP>
<BCS>3</BCS>
<RILR type="hypernym">ENG20-02002
<RII.R tvne="hynernym">ENG20-02027

```

The bottom of the Czech window shows 'Number of entries: 12'.

# 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:

<i>podtřída:</i>	zvíře
<i>část:</i>	hlava
* <i>má_kožich:</i>	ano

slon:

<i>podtřída:</i>	savec
* <i>barva:</i>	šedá
* <i>velikost:</i>	velký

Nellie:

<i>instance:</i>	slon
<i>mít_rád:</i>	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

# Obsah

- 1 **Reprezentace a vyvozování znalostí**
  - Reprezentace znalostí
- 2 **Logika – rezoluční pravidlo**
  - Předpoklad uzavřeného světa
  - Logika – rezoluční pravidlo
  - Rezoluční pravidlo
- 3 **Extralogické informace**
  - Třídy objektů
  - Ontologie
  - Sémantické sítě
  - Rámce
- 4 **Pravidlové systémy**
  - Pravidlová báze znalostí
  - Expertní systémy
- 5 **Nejistota a pravděpodobnost**
  - Nejistota
  - Pravděpodobnost

# 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 chování, reakce na změny
  - **řízení** – ovládání složitého komplexu
  - **předpovědi** – projekce pravěpodobných závěrů z daných skutečností
  - **instruktáž** – inteligentní vyučování a zkoušení studentů



# Obsah

- 1 Reprezentace a vyvozování znalostí
  - Reprezentace znalostí
- 2 Logika – rezoluční pravidlo
  - Předpoklad uzavřeného světa
  - Logika – rezoluční pravidlo
  - Rezoluční pravidlo
- 3 Extralogické informace
  - Třídy objektů
  - Ontologie
  - Sémantické sítě
  - Rámce
- 4 Pravidlové systémy
  - Pravidlová báze znalostí
  - Expertní systémy
- 5 **Nejistota a pravděpodobnost**
  - **Nejistota**
  - **Pravděpodobnost**

# 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 přšet 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)

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

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 vývozování

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:

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

**vývozování** =

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