

Reprezentace znalostí

8. října 2024

Znalostní systémy a znalostní inženýrství

znalostní versus expertní systémy

- ▶ obecně definovány jako **systémy zpracovávající symboly**

Znalost je (v počítačové terminologii) považována za nejvyšší formu organizace strukturovaných dat – je definována

- ▶ **prvky dat** (resp. jejich reprezentanty)
- ▶ **vlastnostmi** prvků dat
- ▶ **relacemi** mezi prvky dat
- ▶ **operacemi** (akcemi) nad prvky dat

Základ: reprezentace znalostí, resp. poznatků

3. Reprezentace znalostí

Úlohy řešené v oblasti znalostních systémů:

- ▶ návrh způsobu (systému) reprezentace faktů a znalostí
 - ▶ návrh báze faktů
 - ▶ návrh báze znalostí
 - ▶ návrh způsobu komunikace systému s uživatelem
 - ▶ vyřešení komunikace systému s okolím
- popřípadě
- ▶ implementace inferenčního mechanismu
 - ▶ návrh formátu zpráv a jejich uložení

. . .

3. Reprezentace znalostí

Reprezentace znalostí (poznatků)

▶ **deklarativní –**

Zanesený čistič vzduchu způsobí zvýšení spotřeby pohonných hmot.

▶ **procedurální –**

Má-li automobil zvýšenou spotřebu pohonných hmot, ověř, zda nemá zanesený čistič vzduchu.

▶ **tacitní –** nevyslovené znalosti, které nelze popsat (vyjádřit) jazykem.

Požadavky na systém reprezentace znalostí (poznatků):

- ▶ vyjádřitelnost poznatků
- ▶ použitelnost reprezentace
- ▶ logická adekvátnost reprezentačního systému
- ▶ heuristická síla, expresivita a přesnost reprezentace
- ▶ začlenitelnost do kontextu již reprezentovaných poznatků
- ▶ jednoduchost (pohodlnost) vyjádření (zápisu)

3. Reprezentace znalostí

Způsoby reprezentace znalostí:

- ▶ logické systémy
- ▶ produkční systémy (založené na pravidlech)
- ▶ rámce
- ▶ sémantické sítě
- ▶ procedurální systémy a speciální programovací jazyky
- ▶ objekty (objektová reprezentace) a ontologie

Příklady (některé):

Jestliže $P_1 \& P_2 \& \dots \& P_m$, pak $Q_1 \& Q_2 \& \dots \& Q_n$. $\Rightarrow$

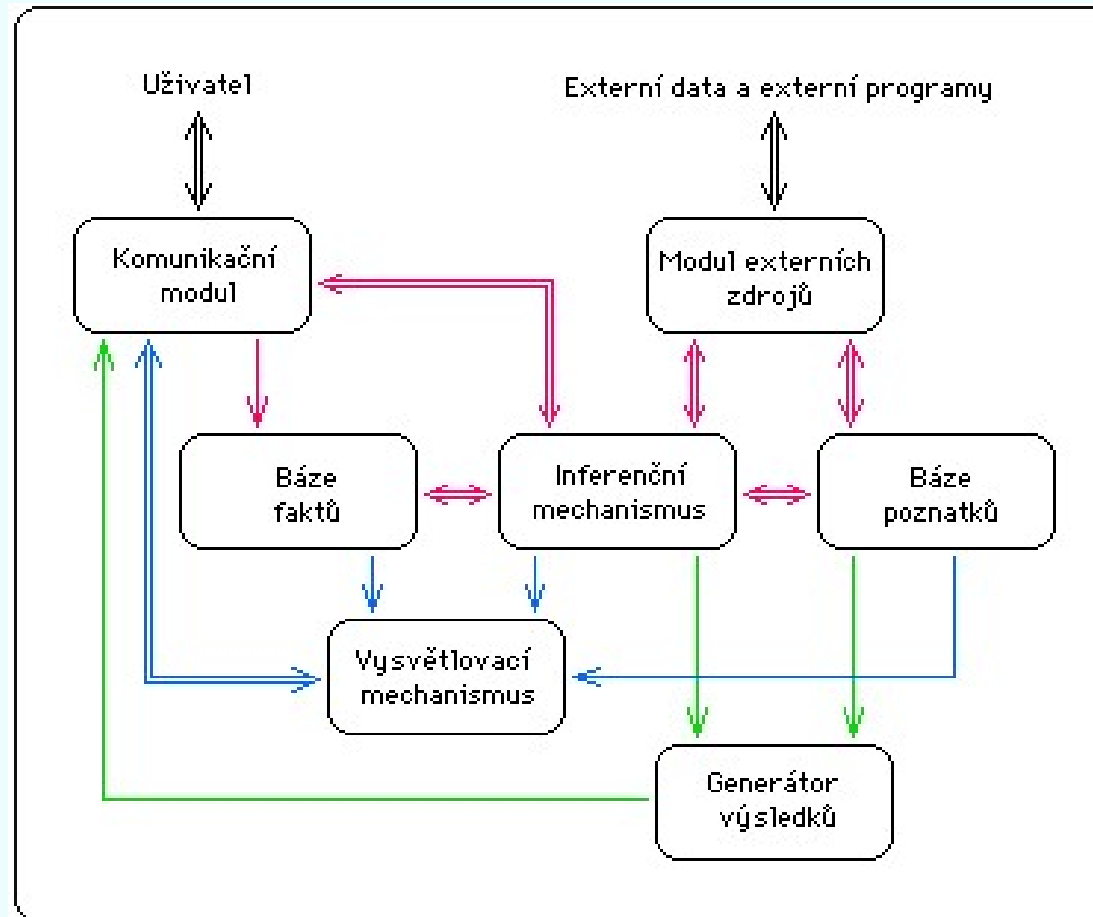
$$P_1 \wedge P_2 \wedge \dots \wedge P_m \rightarrow Q_1 \wedge Q_2 \wedge \dots \wedge Q_n$$

Jestliže auto má zvýšenou spotřebu, pak ověř, zda není zanesený čistič vzduchu.

Jestliže je Petr synem Marie a Marie a Karel jsou jeho rodiče, pak Karel je otcem Petra. $\Rightarrow$ otec(karel,petr) :- syn(petr,marie), manželé(karel,marie).

3. Reprezentace znalostí

Obecná architektura znalostního systému:



3. Reprezentace znalostí

Báze znalostí (poznatků)

Jedná se o datovou strukturu, do které jsou ukládány znalosti (algoritmy pro řešení problémů) z konkrétní specifické oblasti.

Ukládají se nejen poznatky exaktně dokázané, ale i znalosti, které získal expert během své praxe (ovšem tyto poznatky jsou obvykle zatíženy značnou neurčitostí) – nazýváme je **heuristické** informace.

Při vytváření báze znalostí je kladen důraz na:

- **modularitu** znalostí – zajišťuje přehlednost pro snadné úpravy či rozšíření stávající báze
- **strukturalizaci** znalostí – slouží pro rychlé vyhledávání znalostí

3. Reprezentace znalostí

Báze faktů (dat)

Báze faktů (dat) se využívá pro ukládání konkrétních dat k danému problému. Tyto údaje jsou **inferenčním modulem** (využívajícím bázi znalostí) různě **modifikovány** tak, aby znalostní systém mohl poskytnout odpovídající závěr nebo provést určitou akci.

Data jsou získávána např.:

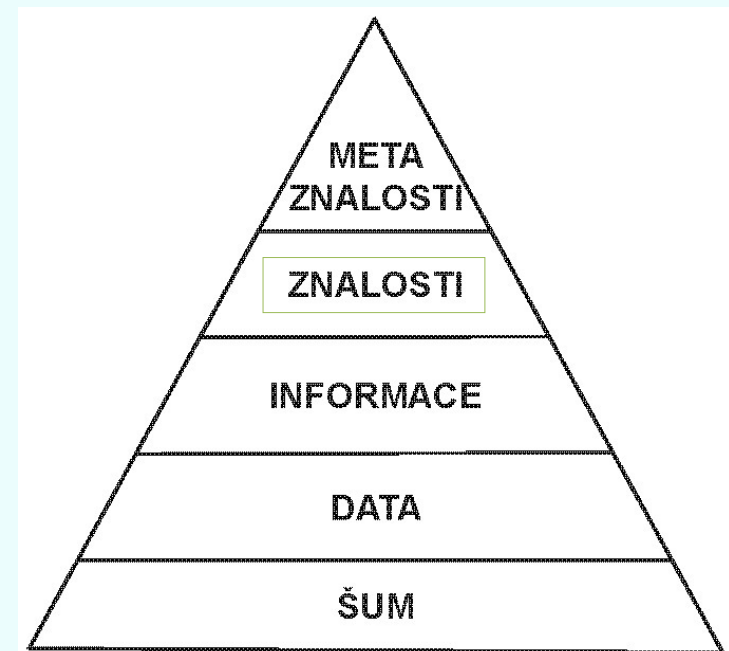
- z měřicích přístrojů
- od uživatele...

3. Reprezentace znalostí

Co je znalost ?

Pojem **znalost** zahrnuje nejen teoretické vědomosti člověka z dané domény, ale také jeho dlouhodobé zkušenosti z praxe.

V počítačové terminologii je **znalost** považována za nejvyšší formu organizace strukturovaných dat.



3. Reprezentace znalostí

Získávání (akvizice) znalostí

- Proces získávání znalostí (od expertů, z odborných textů, dat...) a jejich reprezentace
- Jedná se o klíčovou (a časově nejvíce náročnou) část implementace znalostních systémů
- Způsoby získávání znalostí:
 - automatizované získávání znalostí (strojové učení)
 - získávání znalostí od expertů (interview, brainstorming)

3. Reprezentace znalostí

Co je reprezentace znalostí ?

Znalost je klíčovým prvkem znalostně orientovaných aplikací, např. znalostních a expertních systémů. Na tvůrcích znalostního systému leží úloha volby či návrhu vhodného způsobu zviditelnění (zachycení, reprezentace) znalostí, tzv. **reprezentační schéma**. Takové schéma si lze představit jako **soubor pravidel a postupů**, které musejí být dodržovány pro zachycení znalostí. Schéma musí být pro člověka **dostatečně srozumitelné, univerzální, resp. schopné zachytit širokou škálu znalostí** – od obecnějších po specifitější. A konečně by reprezentační schéma mělo podporovat snadnou modularitu (rozšiřování) znalostí.

3. Reprezentace znalostí

Vlastnosti reprezentace znalostí

Na reprezentaci znalostí jsou kladeny různé požadavky:

- **Modulárnost**, kde je reprezentace znalostí realizována systémem jednoduchých a vzájemně nezávislých zdrojů znalostí, které umožňují proces inference nových faktů. Nové znalosti jsou začleňovány přidáním dalších jednoduchých a nezávislých modulů.
- **Sémantické sdružování znalostí**, u kterého je požadováno vzájemné sdružování znalostí souvisejících s určitým faktem.
- **Sdružování objektů a jejich zařazení do tříd**, kde jsou objekty a třídy uspořádány do hierarchií. Objekty a třídy níže postavené v hierarchii (speciálnějši) mohou některé vlastnosti získávat tzv. **děděním** od objektů a tříd hierarchicky výše postavených.

3. Reprezentace znalostí

Specifikace typů reprezentace znalostí

Sousloví „**procedurální reprezentace znalostí**“ už samo napovídá, že hlavním prvkem této reprezentace je procedura. Tu si lze představit jako určitý výkonný programový kód (menší prográmeček), který řeší otázky typu „**jak?**“, nikoliv „**co?**“. Pojem procedura je znám z programovacích jazyků jako Java, C, Pascal; ve spojení s takovou formou reprezentace ale můžeme hovořit i o tzv. pravidlech (produkcích), které se vyskytují např. v produkčních systémech, resp. znalostních nebo expertních, a slouží k přenosu znalostí expertů do počítačové podoby. Pravidlo (produkce) má odlišnou podobu od procedur používaných třeba v Javě. Obsahuje různé podmínky a akce, které jsou proveditelné na základě splnění těchto podmínek. Budeme-li chtít získat určitou znalost reprezentovanou procedurálně, pak musíme proceduru nebo pravidlo provést. Řešenou úlohou v případě procedurální reprezentace může být např. zjištění většího čísla ze dvou možných. Následující obrázky demonstrují způsob zjištění většího čísla ze dvou možných procedurou zapsanou v jazyce Java a pravidel zapsaných v prostředí Clips (prostředí pro tvorbu znalostních aplikací).

3. Reprezentace znalostí

Př.:

```
/*
 * Procedurální reprezentace znalostí - procedura Main
 * a zjištění většího čísla ze dvou možných
 */
package cisla;

/**
 * @author Martina Musáková
 */

public class Cisla {

    public static void main(String[] args) {
        int a = 2;
        int b = 6;

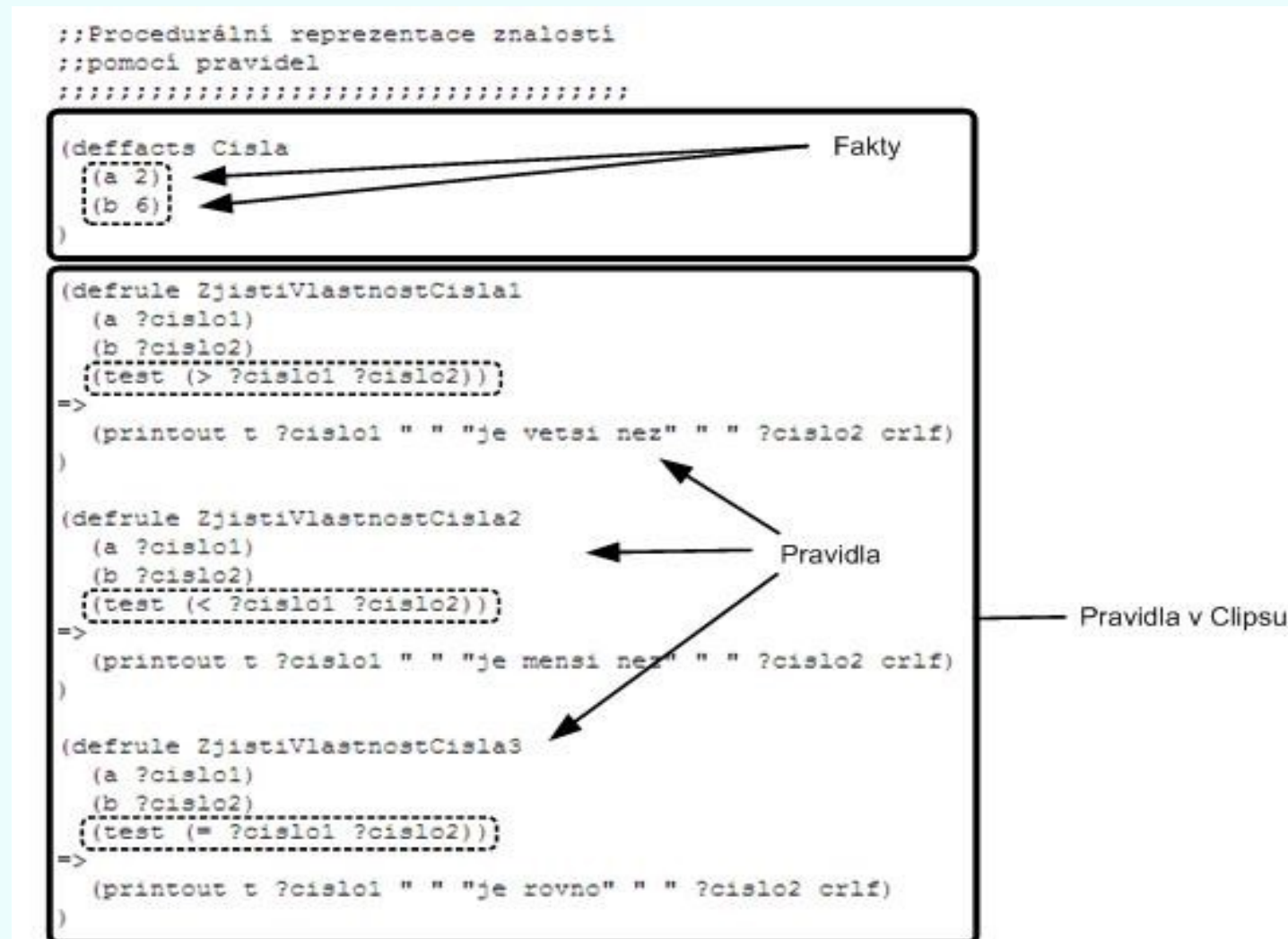
        if (a > b)
            System.out.println(a + " je větší než " + b);
        else {
            if (a == b)
                System.out.println(a + " je rovno " + b);
            else
                System.out.println(a + " je menší než " + b);
        }
    }
}
```

Třída Cisla

Procedura main

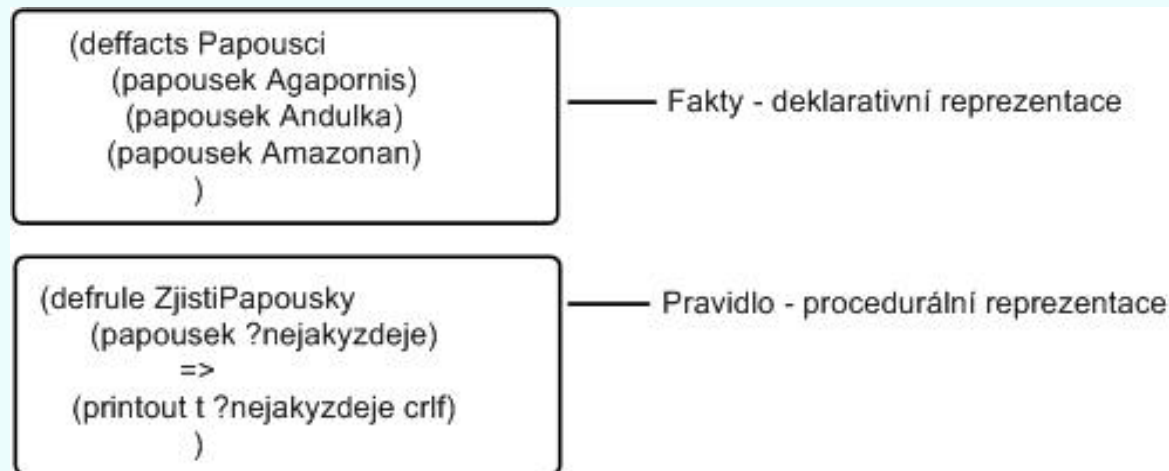
Procedura v Javě

3. Reprezentace znalostí



3. Reprezentace znalostí

Deklarativní způsob reprezentace znalostí naopak zodpovídá otázku „**Co** má být řešeno?“, nikoliv „Jak to má být řešeno?“. Neptáme se proto například na způsob výpočtu největšího čísla, ale spíše na to, jak je největší číslo definováno. Znalostní (expertní) systém má svoji procedurální i deklarativní reprezentaci znalostí – fakta obsažená v bázi dat (faktů). Tato fakta jsou používána pravidly např. pro vyvozování dalších souvislostí a znalostí. Následující obrázek ukazuje procedurální a deklarativní složku znalostní aplikace:



3. Reprezentace znalostí

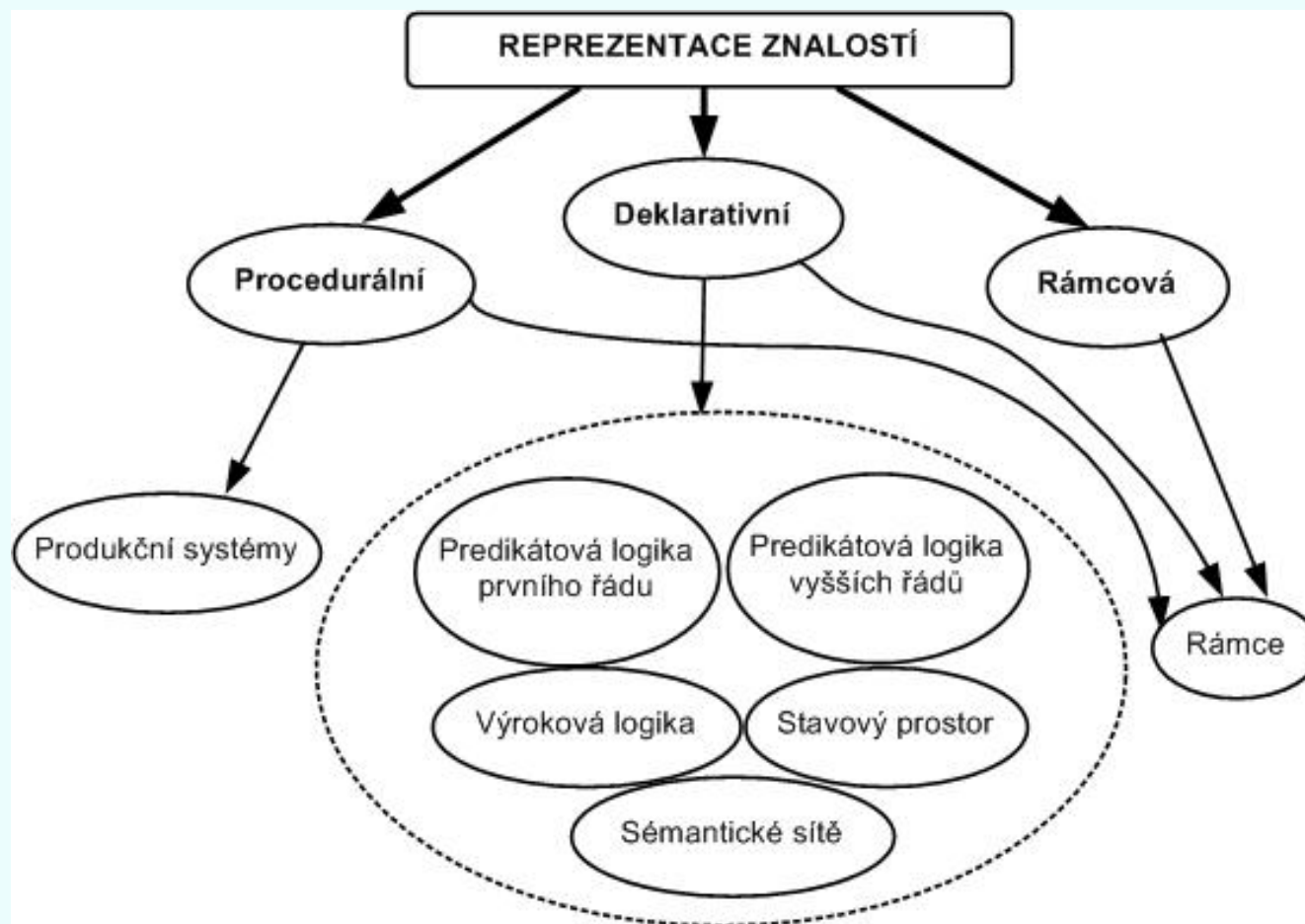
Poznámka: Typickými představiteli deklarativního způsobu reprezentace znalostí jsou logická schémata (výroková logika, predikátová logika prvního a vyšších řádů), sémantické sítě a stavový prostor.

Rámcová schémata jsou kombinací procedurálního a deklarativního přístupu. Deklarativnost spočívá ve způsobu zachycování informací o objektech reality pomocí slotů (vlastnosti rámce) a facet (hodnoty vlastností rámce). Procedurálnost je obsažena v procedurách, které mohou být součástí rámcové struktury. Příklad rámce:

Jméno rámce: Andulka vlnkovaná
položky:
Is-a: Andulka
Barva: modrá
Pohlaví: samec
DožíváSe: 6
...

3. Reprezentace znalostí

Souvislosti výše uvedených (klasických) reprezentací znalostí



3. Reprezentace znalostí

Reprezentace znalostí podrobněji

Reprezentace znalostí je schopnost jazyka (vyjadřovací síla) popisovat – reprezentovat jistou (omezenou) třídu znalostí. Základní klasifikací těchto tříd je způsob poznání, prostřednictvím kterého byly znalosti získány. Přirozené poznání používá jako způsob poznání vágnost a získané znalosti jsou inherentně vágní (Bertrand Russell). Umělé poznání exaktních věd užívá diskrétní způsob Newtonův, získané znalosti jsou popsány exaktně.

Dva typy znalostí, mezi nimiž je „propast“:

Existuje velký rozdíl mezi znalostmi získanými přirozeným poznáním a znalostmi získanými poznáním metodou exaktních věd. V prvním případě se na svět díváme „filtrem“ vágnosti, v druhém případě „filtrem“ „dírkovaným“; dírkami „vidíme“ atributy (veličiny a parametry) – elementární manifestace reálného světa a vztahy mezi nimi a nic jiného. Newton „digitalizoval“ přirozený vágní pohled člověka na reálný svět.

3. Reprezentace znalostí

Inherentně vágní znalosti získané přirozeným poznáním lze sdělovat (reprezentovat, popsat) jen a jen neformálním jazykem, nejčastěji přirozeným. Znalosti získané umělým poznáním lze reprezentovat umělým formálním jazykem (matematika, logika, programovací jazyky). Charakteristickým rysem neformálních jazyků je vágní, subjektivní a emocionálně zabarvené přiřazení významu dané jazykové konstrukci, kterému se říká konotace. Sémantický diferenciál konotace je vždy nenulový. Formální jazyky naproti tomu musejí mít sémantický diferenciál nulový, je to jejich základní ustavující podmínka. Konotace je v tomto případě exaktní a v matematice (logice) se jí říká interpretace. Formální jazyky proto nelze použít pro reprezentaci znalostí získaných přirozeným poznáním a reprezentovaných původně přirozeným jazykem.

Reprezentaci znalostí z pohledu umělé inteligence tedy můžeme chápat jako proces získávání, zaznamenávání a ukládání znalostí do strojem srozumitelné podoby a jejich využívání v rámci znalostního systému.

3. Reprezentace znalostí

Co je to formálně **znalost** ?

Znalost můžeme vyjádřit jako schopnost člověka nebo jakéhokoli jiného inteligentního systému uchovávat, komunikovat a zpracovávat informace do systematicky a hierarchicky uspořádaných znalostních struktur.

Znalost je charakterizována schopností abstrakce a generalizace dat a informací.

Znalost je vnitřní náhled, porozumění a praktické know-how, které všichni ovládáme. Je to základní zdroj, který nám umožňuje chovat se inteligentně.

Znalosti jsou spojeny s pojmy data a informace, respektive znalosti vycházející z informací a ty zase vycházejí z dat.

3. Reprezentace znalostí

Za data lze pokládat např. všechny znakové řetězce, které vstupují do výpočetního procesu. Taková data většinou mají nějaký význam, jsou tedy nějak interpretována.

Interpretací dat se rozumí smysluplné přiřazení významu (sémantiky) datům. Informaci tvoří data spolu se svou interpretací, pojem informace je tedy neoddělitelný od významu dat, která jsou jejími nositeli. A znalost je potom informace, která je použitelná a začlenitelná respektive odvoditelná v souvislosti s jinými informacemi.

Přechod od dat ke znalostem je doprovázen dvěma úrovněmi přiřazování metadat. Na první z nich je datům přiřazován jejich význam nějakým definičním jazykem, na druhé úrovni se nad nimi definují jejich vzájemné vztahy, souvislosti a možné důsledky.

3. Reprezentace znalostí

Rozlišení dat a znalostí z hlediska budování informačního, resp. znalostního, systému je tento:

- Jestliže se můžeme při sběru materiálu spolehnout na automatický proces sběru informací, pak zpravidla hovoříme o datech. Správnost dat vzhledem k reálnému světu pak může být objektivně potvrzena jejich opakovaným pozorováním a vyhodnocováním.
- Jestliže ale potřebujeme někoho (odborníka, experta), kdo by poskytl postupy, metody, algoritmy atd., potom hovoříme o znalostech. Ty obvykle obsahují abstrakce a generalizace metod, postupů, algoritmů apod. (obecně materiálu) a zpravidla (při získávání na verbální úrovni) jsou méně přesné a těžko objektivně ověřitelné.

3. Reprezentace znalostí

Vyjádření znalostí

Implicitní znalost – jedná se o takovou znalost, která není přímo vyjádřena v určitém zdroji, je primárně skryta, ale je potenciálně vyjádřitelná (sdělitelná). Obvykle je zahrnuta v jednání, způsobu řešení úloh, souboru dat apod. Speciální skupinou implicitních znalostí jsou znalosti tacitní (slovy nesdělitelné). Ve znalostních systémech se jako implicitní označují znalosti rozptýlené v jednotlivých programových instrukcích, které se aplikují podle předem stanoveného algoritmu (viz procedurálně reprezentované znalosti).

Explicitní znalost – je znalost zaznamenaná v určitém jazyce a dostupná přímo v určitém informačním zdroji (např. dokument, záznam v databázi). V počítačových znalostních systémech se jako explicitní označují znalosti uložené v bázi znalostí, jež je oddělena od programu, který ji umožňuje využívat a odvozovat z ní nové znalosti (viz deklarativně reprezentované znalosti).

3. Reprezentace znalostí

Znalostní modely a ontologie

V klasickém pojetí je získávání znalostí (např. pro expertní systémy) založeno na získávání znalostí od expertů. Zpočátku mělo získávání znalostí podobu přebírání znalostí: znalostní inženýr převzal znalosti od experta a přímo je vkládal do expertního systému. Takto vytvářené báze znalostí byly ale obtížně modifikovatelné a přenositelné, protože v nich nebyly rozlišeny statické znalosti, týkající se celé aplikační oblasti, a znalosti vztahující se pouze k řešení dané konkrétní úlohy.

Byl proto na přelomu 80. a 90. let změněn pohled na proces získávání znalostí a začal být chápán jako modelování znalostí, tedy jako tvorba přehledných a opakovaně použitelných modelů dané úlohy. Znalosti jsou zachycovány nezávisle na odvozovacích metodách a jazyku reprezentace znalostí konkrétního expertního systému.

3. Reprezentace znalostí

Výhody tohoto přístupu jsou v zásadě dvojí:

- usnadnění vývoje aplikace – model vede tvůrce systému k lepšímu strukturování řešené úlohy a ke
- sdílení a opakovanému používání – pokud jsou modely založeny na standardizované terminologii, pak jsou srozumitelné nejen tvůrcům aplikace, ale celé komunitě.

Nejnověji se ve znalostním modelování objevuje pojem **ontologie**. Tento pojem je zde (na rozdíl od filosofického pojetí, kde ontologie znamená nauku o „bytí“) chápán jako označení domluvené terminologie pro určitou aplikační oblast, která umožňuje sdílení znalostí z této oblasti.

Ontologie tedy umožňují formalizovat doménové znalosti.

3. Reprezentace znalostí

Reprezentační jazyky

V běžném životě obvykle využíváme k vyjádření svých znalostí přirozeného jazyka. Ten však odráží kromě informační funkce i kulturně historické podmínky svého vývoje a je příliš složitý, nepravidelný a zpravidla nejednoznačný. Proto, aby bylo možné pracovat se znalostmi v podobě vhodné pro počítačovou implementaci, byly zavedeny **umělé – reprezentační jazyky**.

Uměle vytvořené (formální) jazyky jsou **jednoznačné a úsporné** a jejich vhodná volba má značný význam i pro časovou, výpočetní a prostorovou náročnost kladenou na zpracování znalostí. Takový jazyk musí mít:

- **vyjadřovací schopnost** (třída možných světů, o kterých je jazyk schopen vypovídat),
- **schopnost podporovat využívání popsaných znalostí v inferenčním procesu.**

3. Reprezentace znalostí

Logické systémy – predikátová logika

Jako základní nástroj pro reprezentaci znalostí v umělé inteligenci je považován jazyk predikátové logiky prvního řádu. Tento jazyk má přesně definovanou syntax a sémantiku a jeho podstatnou výhodou je to, že kromě schopnosti popsat znalosti ještě poskytuje metody jak odvozovat znalosti další. Přitom zaručuje, že pokud jsou výchozí znalosti pravdivé, jsou pravdivé i znalosti odvozené. Jazyk predikátové logiky obsahuje:

- individuové proměnné $x, y, z, \dots$ (nekonečně mnoho)
- predikátové symboly $P, Q, R, \dots$ (konečně mnoho)
- funkční symboly $f, g, h, \dots$ (konečně mnoho) – např. : $\sin, \ln$, ale také matka (někoho) apod.

3. Reprezentace znalostí

- konstanty $A, B, C, \dots$ (nejvýše konečně mnoho)
- kvantifikátory $\forall, \exists$ a
- logické spojky: pro konjunkci, pro disjunkci, pro implikaci, pro ekvivalenci, pro negaci.

Z individuových proměnných, konstant a funkčních symbolů jsou vytvářeny **termy**. Termy reprezentují všechna univerzální i speciální **jména objektů**, o nichž může být při použití daného jazyka řeč. **Predikátové symboly, kvantifikátory a logické spojky** pak slouží k vytváření **formulí**, což jsou právě takové **jazykové výrazy**, které reprezentují znalosti.

Poznámka: Mezi nimi hrají důležitou roli takové formule, kterým lze přidělit pravdivostní hodnotu a chovají se jako výroky.

3. Reprezentace znalostí

Pravidlové systémy

Pravidla představují nejpoužívanější prostředek pro reprezentaci znalostí ve znalostních systémech. Pravidla jako IF-THEN jsou známa z programovacích jazyků, sémantika pravidel vychází z implikací ve výrokové nebo predikátové logice. Taková pravidla mohou být chápána dvojím způsobem:

- **deklarativně:** jestliže existuje předpoklad, pak platí závěr
- **procedurálně:** jestliže vznikne situace, pak proved' akci

Pravidla z báze znalostí jsou při zpracování úloh znalostním systémem interpretována – je možno je znázornit v podobě AND/OR grafu, uzly grafu představují výroky a orientované hrany představují pravidla. Konjunktivní vazba mezi výroky v předpokladu pravidla je znázorněna obloučkem.

3. Reprezentace znalostí

Deklarativní interpretace pravidel odpovídá diagnostickým znalostním systémům – je-li splněn příslušný předpoklad, systém odvodí daný závěr. Taková interpretace je vlastně speciálním případem procedurální interpretace – provádí se **jediná standardní akce** – odvození závěru.

Procedurální interpretace pravidel je běžná v generativních znalostních systémech – nastala-li příslušná situace, systém provede danou akci.

3. Reprezentace znalostí

Reprezentace rámci

Rámce byly navrženy v polovině 70. let Marvinem Minským z MIT jako možný prostředek pro reprezentaci znalostí. Rámce v původní představě měly umožňovat **reprezentovat stereotypní situace**, práce s rámci měla být založena na **postupném vyplňování stránek**, do kterých se zapisují hodnoty položek (vlastnosti). Přitom se hojně využívalo předdefinovaných hodnot. Rámce dobře **umožňují vyjádřit statické znalosti**, tedy nějakou **hierarchii pojmů** (s použitím položek **a_kind_of**) nebo **dekompozici** (s použitím položek **part_of**). Vazbu mezi rámci lze znázornit grafem. Na rozdíl od sémantických sítí (viz dále) mají uzly grafu – rámce – přesně definovanou vnitřní strukturu. Sestávají z **položek (slots)** sloužících k popisu **vlastností objektu**, položkám jsou **přiřazovány** v průběhu užívání **konkrétní hodnoty** (popis položky se skládá z jména položky a hodnoty). Položky dále dělíme na **fasety (facet)** a jednotlivé hodnoty se stávají jednou z faset.

3. Reprezentace znalostí

Rámce

- datová struktura umožňující reprezentaci stereotypních znalostí o daném objektu (pojmu)
- reprezentují **obecné znalosti** (pravdivé pro většinu případů)
- jsou často využívány při modelovém a případovém usuzování
- rámec je tvořený **jménem** a **atributy**
 - atributy (rubriky, sloty) slouží pro ukládání vlastností
 - při vytváření jsou atributy prázdné => jsou vyplňovány během inicializace
 - využívání předdefinovaných hodnot

3. Reprezentace znalostí

- mohou obsahovat položky (*aktuální hodnoty, výchozí hodnoty, rozsah hodnot...*) a procedury, které jsou aplikovány, pokud nastane příslušný stav
- procedury v rámci jsou:
 - IF-NEEDED – pokud je požadována hodnota atributu, ale daný atribut nemá inicializační hodnotu
 - IF-ADDED – je aplikována při vkládání hodnoty
 - IF-CHANGED, IF-REMOVAL, IF-DELETED...

3. Reprezentace znalostí

Příklad 1:

Příklad rámce

Výrobce	Firma
Model	---
CPU	Intel Core i5...
RAM	---
HDD	1 TB

DĚDIČNOST

DOPLNĚNÍ

ZMĚNA

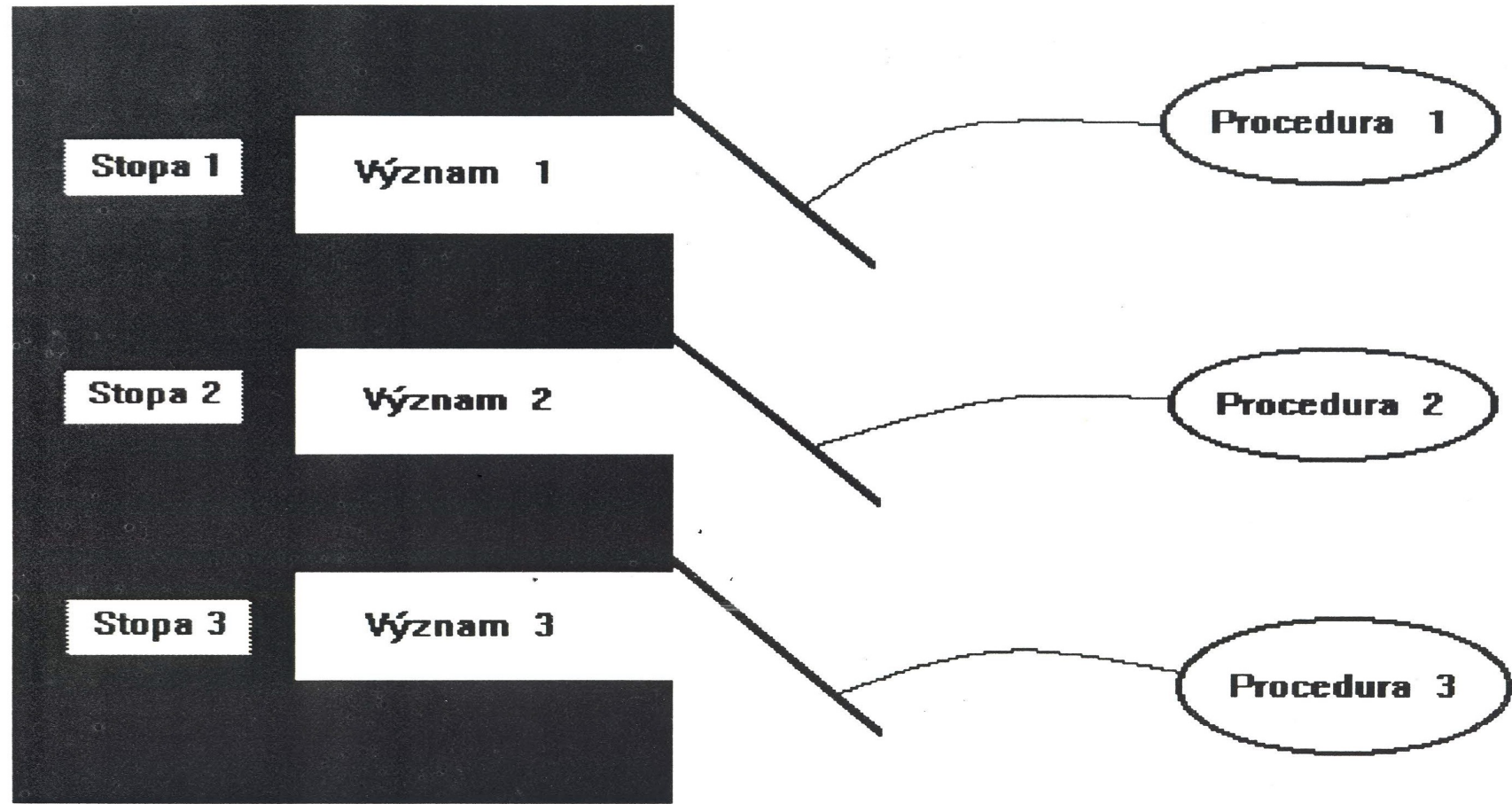
Příklad instance rámce

Výrobce	Lenovo
Model	G510
CPU	Intel Core i5...
RAM	8 GB
HDD	2 TB

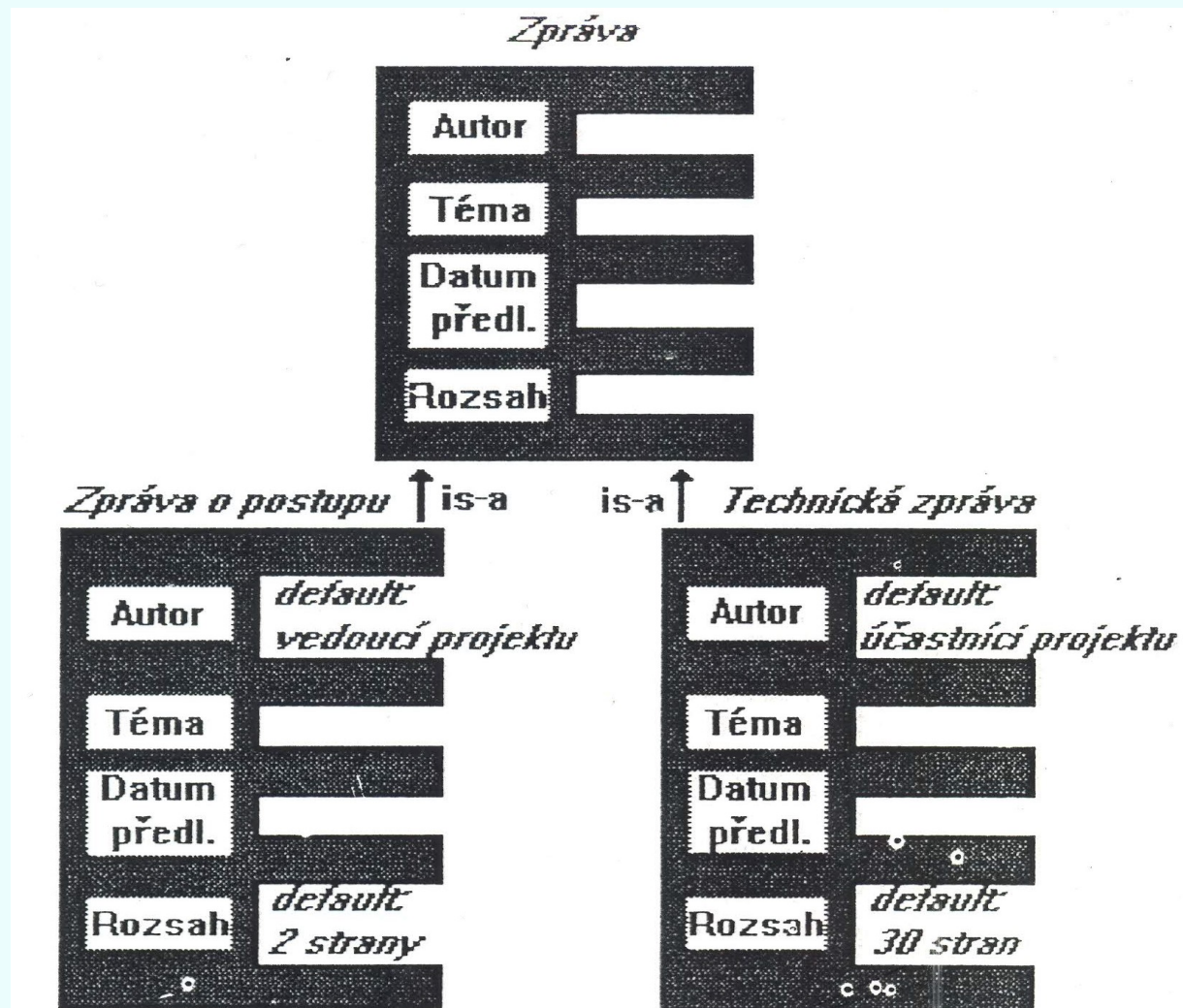
3. Reprezentace znalostí

Příklad 2:

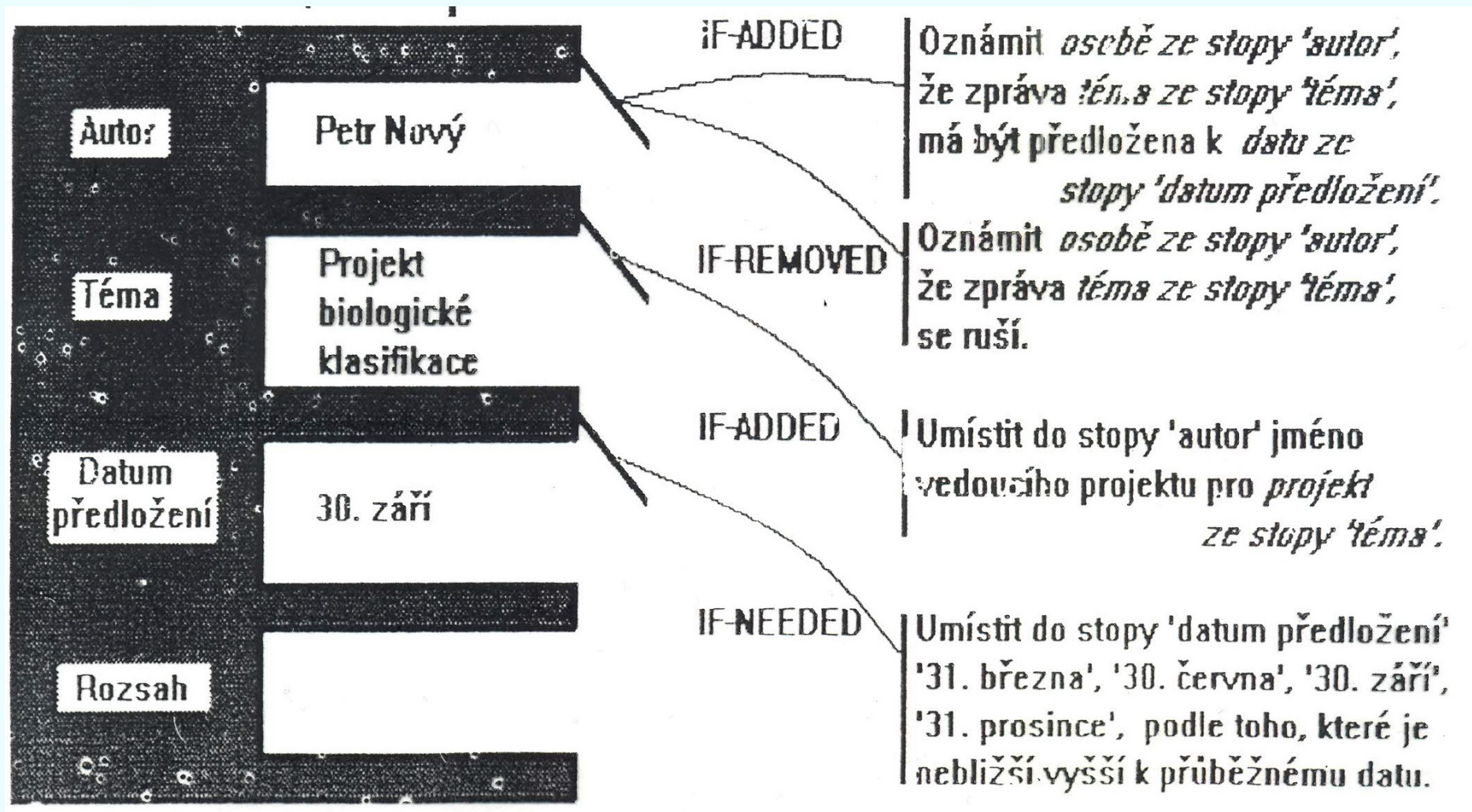
POJEM



3. Reprezentace znalostí



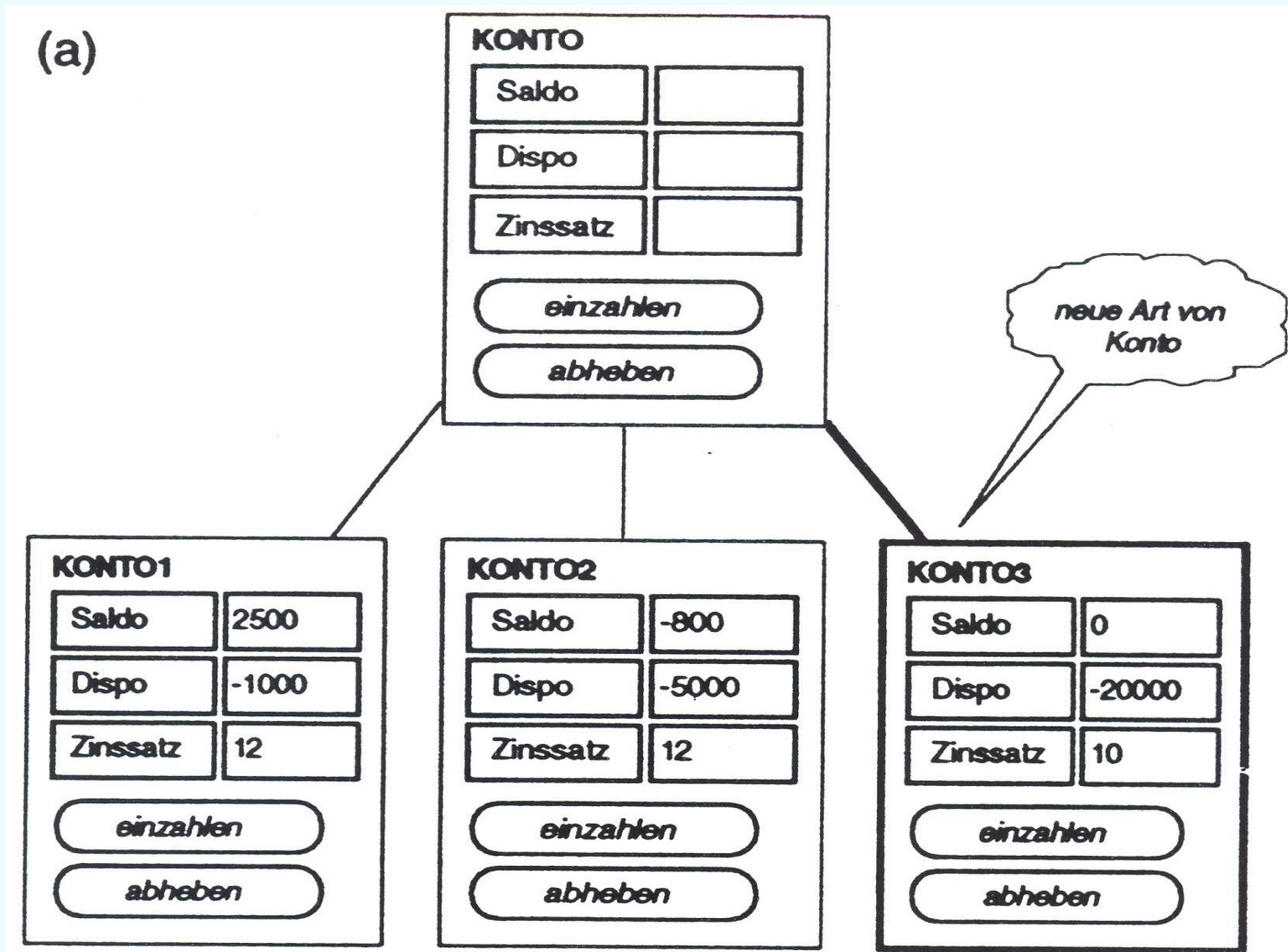
3. Reprezentace znalostí



3. Repräsentation von Wissen

Příklad 3:

(a)



3. Reprezentace znalostí

Pozn.: Rámcová reprezentace pronikla před léty do programovacích jazyků, tam se pro ně používá název **objekty** (viz objektově orientované programování). V případě použití rámců pro reprezentaci znalostí má největší význam **dědičnost** (lze dědit položky i jejich hodnoty; obvyklé je dělení shora dolů) a **zapouzdření** (součástí rámce jsou kromě datových struktur i procedury pro práci s nimi).

Pozn.: Reprezentace se někdy nazývá **rámcová schémata** – představují syntézu deklarativního a procedurálního přístupu k reprezentaci znalostí. Vlastnosti objektů a hodnoty těchto vlastností jsou vyjádřeny deklarativně, podmínky použití rámce jsou reprezentovány procedurálním způsobem.

3. Reprezentace znalostí

Sémantické sítě

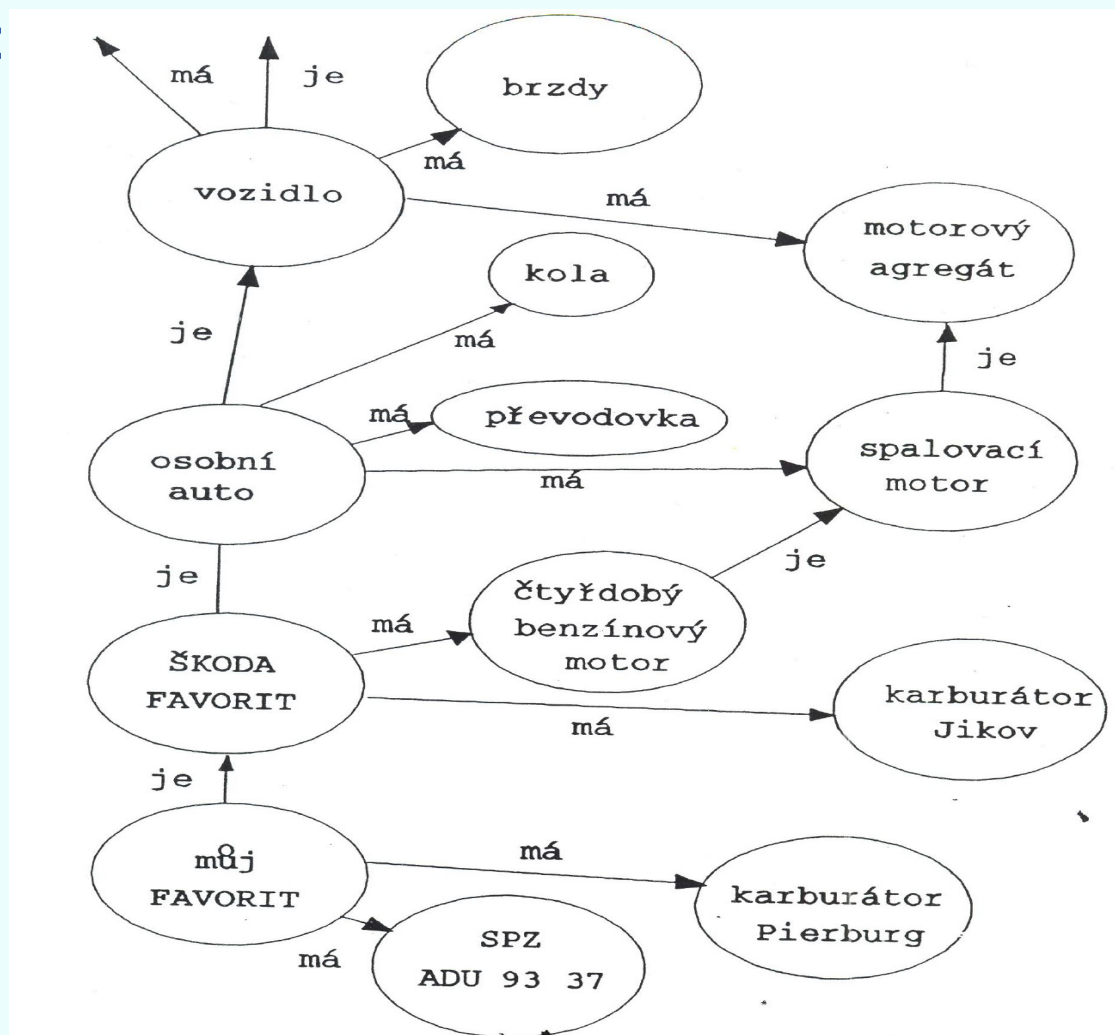
Sémantické sítě byly navrženy v druhé polovině 60. let pro strojové porozumění přirozenému jazyku. Nabízejí reprezentaci znalostí objekty (entitami) a relacemi mezi nimi (binárními nebo víceargumentovými) a umožňují popisovat realitu jako objekty, které jsou navzájem v nějakých vztazích (relacích).

Sémantická síť má přirozenou grafovou reprezentaci – objekty jsou uzly a relace mezi nimi jsou hrany v grafu. **Relace** v sémantických sítích představují základní prostředek pro vyjadřování znalostí. Nejčastěji vyskytujícími se relacemi jsou:

- **is_a** – umožňuje vyjádřit, že nějaký konkrétní objekt (instance) patří do určité třídy objektů
- **a_kind_of** – umožňuje vyjadřovat hierarchii tříd
- **part_of** – umožňuje vyjádřit, že objekt (třída objektů) je tvořen svými částmi

3. Reprezentace znalostí

Příklad 1:



3. Reprezentace znalostí

Sémantická síť reprezentující strukturovaná data příslušející "mému Favoritu" zapsaná příkazy Prologu:

```
je (muj_Favorit, skoda_Favorit).           % (je1)
je (skoda_Favorit, osobni_auto).           % (je2)
je (osobni_auto, vozidlo).                 % (je3)
je (ctyrdoby_benzinovy_motor, spalovaci_motor). % (je4)
je (spalovaci_motor, motorovy_agregat).     % (je5)

ma (muj_Favorit, spz(adu_93_37)).          % (ma1)
ma (muj_Favorit, karburator(pierburg)).    % (ma2)
ma (skoda_Favorit, karburator(jikov)).     % (ma3)
ma (skoda_Favorit, ctyrdoby_benzinovy_motor). % (ma4)
ma (osobni_auto, spalovaci_motor).        % (ma5)
ma (osobni_auto, prevodovka).             % (ma6)
ma (osobni_auto, kola).                   % (ma7)
ma (vozidlo, motorovy_agregat).           % (ma8)
ma (vozidlo, brzdy).                      % (ma9)
```

3. Reprezentace znalostí

Položíme-li potom Prologu následující dotazy, dostaneme odpověď:

```
?- je (skoda_Favorit, osobni_auto).
```

```
yes
```

```
?- ma (skoda_Favorit, okno).
```

```
no
```

```
?- je (osobni_auto, vozidlo), ma (vozidlo, brzdy).
```

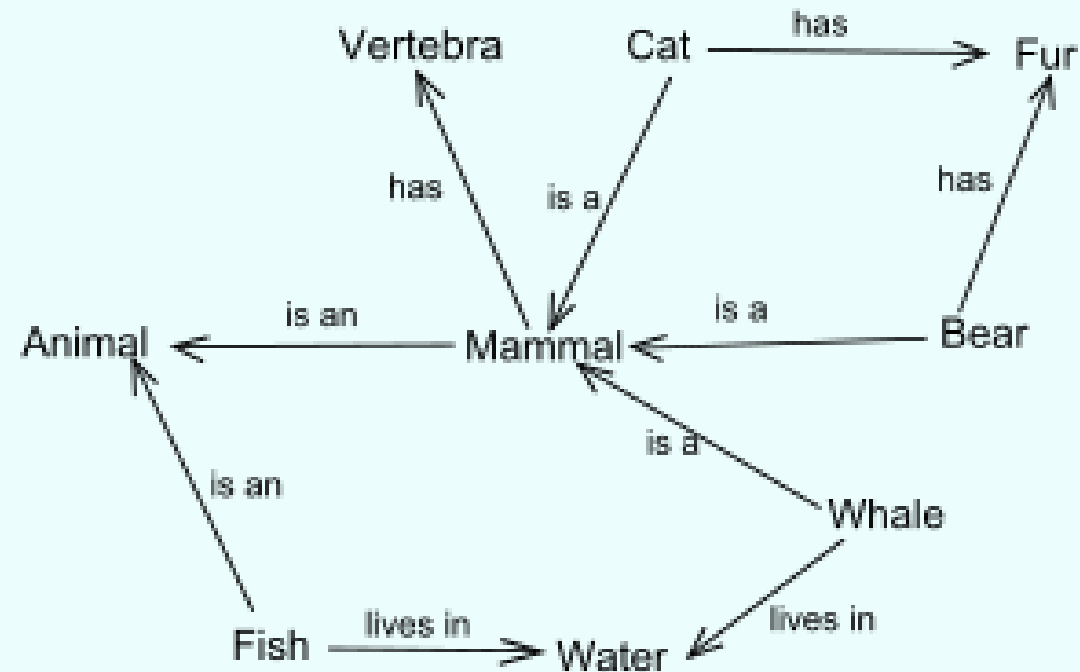
```
yes
```

Pomocí rezoluční metody se pokuste určit, jakou dostanete odpověď na dotaz

```
?- ma (muj_Favorit, motorovy_agregat).
```

3. Reprezentace znalostí

Příklad 3:



3. Reprezentace znalostí

Další způsoby reprezentace

Předchozí čtyři způsoby reprezentace znalostí patří mezi nejčastěji využívané. K dalším způsobům patří například:

- **deskripční logika** – zastřešuje řadu příbuzných logických formalismů, jejichž společným rysem je snaha zachytit strukturu tříd a relací chápaných obdobně, jako v systémech založených na rámcích.
- **konceptuální grafy** – zachycují abstraktní, konceptuální strukturu jazykové podoby vyjádřené znalosti. Mohou být znázorněny graficky či zapsány v podobě jim odpovídajícího lineárního textového kódu. Tvorba konceptuálních grafů vychází z lingvistické znalosti gramatické stavby vět přirozeného jazyka a z pojetí, že věty v přirozeném jazyce vyjadřují vztahy mezi koncepty.

3. Reprezentace znalostí

- **deklarativní schémata** – pracují s vyjádřením znalostí ve formě struktur poznatků (onotologií), případně s množinou specifických, navzájem nezávislých faktů a množinou všeobecných procedur (tvrzení) na manipulování s obsahem množiny specifikovaných faktů. Do této skupiny schémat patří také speciální schémata založená na matematické logice, algebře, relačních strukturách či grafech. Mezi jazyky pro deklarativní reprezentaci znalostí patří RDF (Resource Description Framework), RDFS (RDF Schema), OWL (Web Ontology Language), SKOS, Topic maps, ERA (Entity Relationship Attributes), UML (Unified Modeling Language).
- **procedurální schémata a produkční systémy** fungují na základě sady pravidel chování (tzv. **produkčních pravidel**) užívaných systémem k řešení situací a provádění akcí v souladu s předem stanoveným cílem.

3. Reprezentace znalostí

- metoda stavového prostoru – struktura problému je reprezentována alternativami přípustnými pro daný stav. Jde o formu reprezentace znalostí původně určenou pro řešení úloh (problem solving) a programování her – viz tam.
- fuzzy logika
 - fuzzy logika je přístup k uvažování o fuzzy množinách
 - je základem pro fuzzy znalostní systémy
 - v booleovské teorii množin se pracuje pouze se dvěma hodnotami (0 nebo 1)
 - prvky fuzzy množin mohou nabývat reálné hodnoty z intervalu $<0, 1>$ (jedná se o tzv. stupeň příslušnosti) => klasické množiny jsou speciálním případem fuzzy množin

3. Reprezentace znalostí

- funkce příslušnosti:

$$\mu_A(x): X \rightarrow \langle 0; 1 \rangle$$

- stupeň příslušnosti:
 - 0 – absolutní nepříslušnost
 - 1 – absolutní příslušnost

3. Reprezentace znalostí

Fuzzy množina

- Jde o množinu prvků, které jsou výstupy funkce příslušnosti $\mu_A(x)$ (k množině A)
- Obecný zápis fuzzy množiny:

$$A = (a_1, a_2, \dots, a_n), \text{ kde } a_i = \mu_A(x_i)$$

- Prvky, pro které platí $\mu_A(x) > 0$, jsou tzv. podpory množiny
- Prvky, pro které platí $\mu_A(x) > \alpha$ (kde α je zvolená konstanta), tvoří tzv. α -řez množiny
- Prvky, kde $\mu_A(x) = 1$, vytvářejí tzv. jádro množiny

3. Reprezentace znalostí

Operace s fuzzy množinami

Pro fuzzy množiny **neplatí** zákon vyloučení třetího a zákon protikladu.

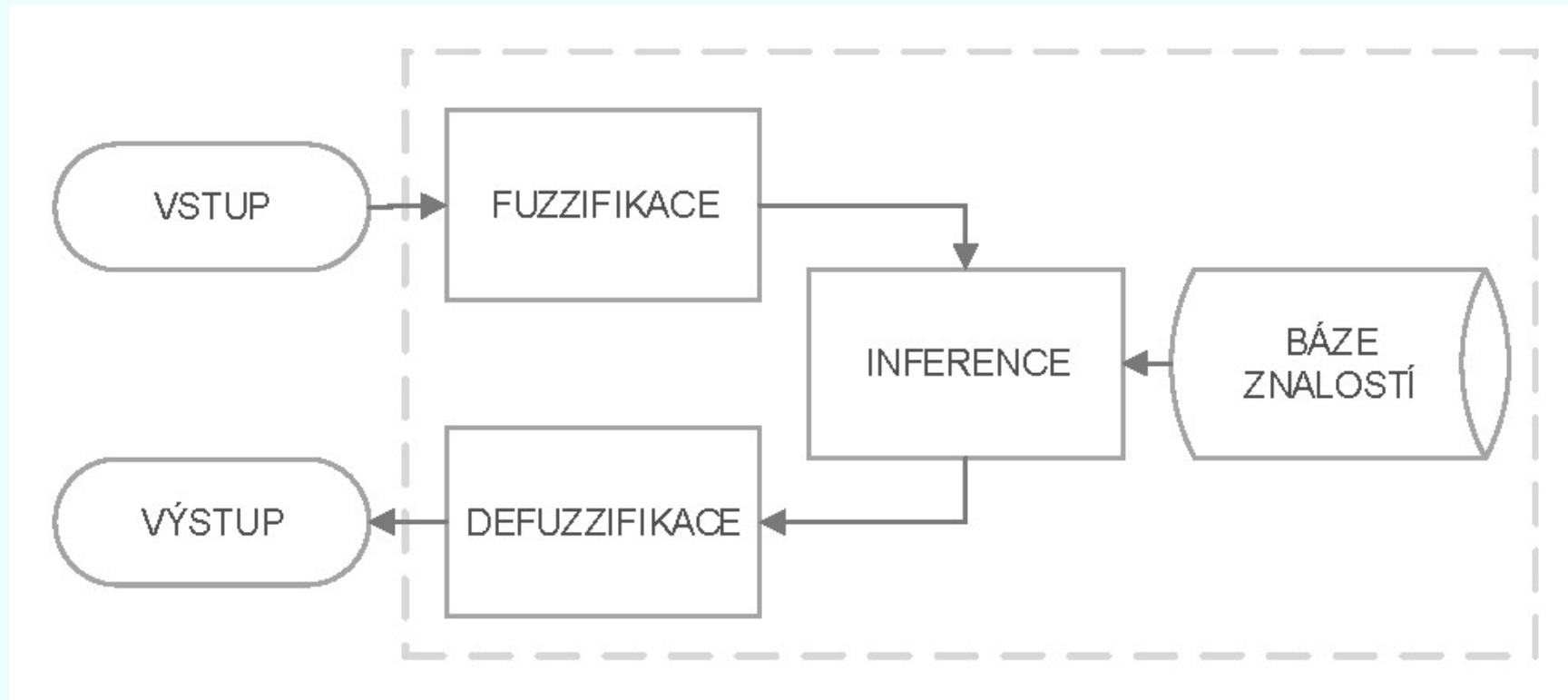
Možné operace:

průnik, sjednocení, doplněk, fuzzy negace, rovnost, inkluze, normalizace, koncentrace, dilatace, intenzifikace...

kartézský součin fuzzy množin je tzv. **fuzzy relace**.

3. Reprezentace znalostí

Fuzzy ZS



3. Reprezentace znalostí

Fuzzifikace

- Převod vstupů do fuzzy množin(y)
- Probíhá ve dvou krocích:
 - 1) normalizace univerza na interval $\langle 0; 1 \rangle$
 - 2) přiřazení stupňů příslušnosti k daným fuzzy množinám (fuzzy množinami musí být pokryté celé univerzum)

Defuzzifikace

- Získání jedné hodnoty z fuzzy množiny
- Existuje mnoho metod – nejčastěji používaná je **fuzzy centroid**

3. Reprezentace znalostí

Závěr

Reprezentace znalostí je stále se rozvíjející obor, který nachází praktické uplatnění ve znalostních a expertních systémech, které dovolují vykonávat rozhodovací operace bez nutnosti lidského zásahu právě na základě reprezentovaných znalostí. Z popsaných prostředků pro reprezentaci znalostí jsou pravidla tím nejpoužívanějším. Jejich výhodou je jednoduchost, srozumitelnost a modularita. Nevýhodou je, že v bázi tvořené souborem pravidel mohou být skryty strategické znalosti o způsobu řešení úlohy (např. pořadí pravidel). Dále pak například využití rámců je výhodné pro zachycení struktury nějakých konceptů nebo pro aplikace, kde se provádí porovnávání mezi daty a hypotézami. Mezi hlavní nevýhodu rámců patří to, že přítomnost nebo nepřítomnost jiných rámců může ovlivnit položky v daném rámci kvůli vzájemné provázanosti rámců (jsou tedy méně modulární).

3. Reprezentace znalostí

Současné nástroje pro tvorbu aplikací mohou nabízet více než jeden prostředek pro reprezentaci znalostí. Jedná se o takzvaná hybridní schémata reprezentace, kde se nejčastěji kombinují pravidla a rámce. Vhodný prostředek pro reprezentaci znalostí pak souvisí i s typem a složitostí aplikace.