

# IV130 Přínosy a rizika inteligentních systémů

**Reprezentace znalosti, kategorie, taxonomie, plánování**

14. dubna 2023

# Znalostní báze

- Budování **znalostní báze**, vytvoření **ontologie** (pojmového modelu reprezentace světa) jako výchozí součást chápání problémové oblasti (domény)
- Ontologické inženýrství zahrnuje metody a metodologie pro budování ontologií, tj. reprezentace, pojmenovávání a definice kategorií, vlastností a relací mezi koncepty, daty a entitami
- Původně (cca 80. léta) motivováno expertními a znalostními systémy (znalostní inženýrství)
- Výchozí hypotéza (Brian Smith, 1985): Jakýkoli mechanicky realizovaný inteligentní proces se bude skládat ze strukturálních složek, které
  - a) my jako vnější pozorovatelé považujeme za vztahový popis znalostí, které proces jako celek zahrnuje,
  - b) nezávisle na tomto vnějším sémantickém přiřazení hrají formální, ale kauzální a zásadní roli při vzniku chování, které znalosti vyjevuje

# Znalostní báze

- Obecně rámec znalostní reprezentace:
  - nahrazuje věci samy entitami, které je reprezentují (umožňují vyvozovat znalosti o světě místo přímého zasahování do světa),
  - je sadou ontologických pohledů, odpovídá na otázku, v jakých termínech uvažujeme o předmětném světě,
  - zahrnuje dílčí teorii vyvozování (základní pojetí inteligentního vyvozování, množinu vyvozování, které v reprezentaci lze provádět, a množinu doporučených vyvozování),
  - poskytuje médium pro pragmaticky efektivní výpočty, prostředí v němž se myšlení realizuje pomocí doporučených vyvozování,
  - je to také médium pro lidské vyjadřování – jazyk, jímž se vyjadřujeme o světě.

# Omezení ontologií

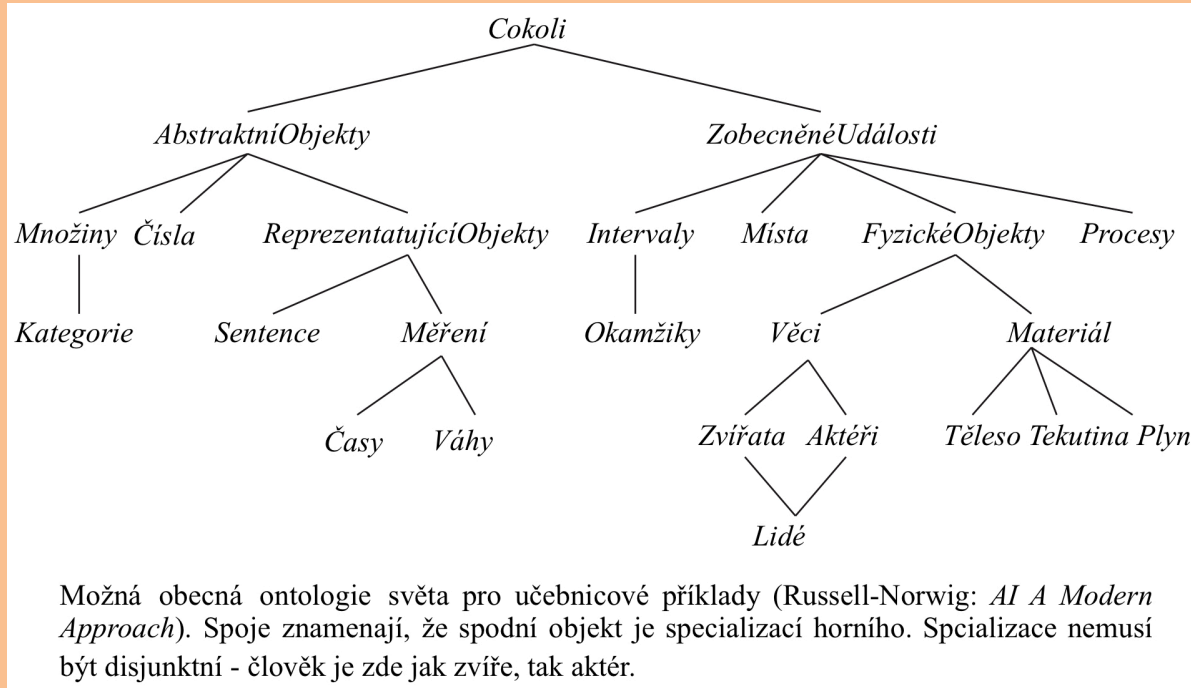
- Reálně se v úspěšných systémech obecná ontologie nepoužívá, používají se specializované ontologie vztažené k řešenému problému
- Každá ontologie je jistou smlouvou – společenskou smlouvou – mezi lidmi s motivací ji sdílet (Tom Gruber, 2004)
- Různé oblasti znalostí se specializovanými ontologiemi vyžadují jejich sjednocení (unifikaci), pokud se mají užívat ve společném systému
- Reálné ontologie vznikly některým z následujících způsobů:
  - Týmem ontologů nebo logiků vytvářejících architekturu ontologie a její axiomy,
  - Importem kategorií, atributů a hodnot z existující databáze nebo databází,
  - Analýzou textových dokumentů a extrahováním informací z nich,
  - Zahrnutím amatérských výpovědí dobrovolníků reprezentujících obecné znalosti.

# Ontologické/znalostní inženýrství

- Pochopení předmětné oblasti (domény) – analýza toho, jak funguje, ve spolupráci s expertem v dané doméně
- Identifikace konceptů potřebných pro řešení problémů, analýza toho, jaké otázky budeme klást a co je potřeba pro nalezení odpovědí
- Návrh ontologie, resp. specificky pro znalosti návrh formální reprezentace objektů a relací a volba vhodného kódování pro počítač
- Možný postup při práci znalostního inženýra:
  - Identifikace úlohy a určení druhu otázek na znalostní bázi
  - Získání potřebných znalostí a analýza sémantiky dat vzhledem k fungování domény
  - Definování slovníku predikátů, funkcí a konstant (přeložením konceptů ze světa do logických termínů)
  - Zakódování obecných informací o doméně (nalezení axiomů, které v doméně platí)
  - Zakódování specifické instance problému v doméně (zakódování informací o současném stavu domény)
  - Formulace otázek, spouštění vyvozovacího mechanismu a získávání odpovědí (využívání obecného vyvozovacího mechanismu nad informacemi ze znalostní báze v doméně)
  - Ladění znalostní báze (ověřování úplnosti axiomů, jejich doplňování, atd.)

# Obecná zastřešující ontologie

- Nelze reprezentovat ze světa úplně vše
- Obecný rámec konceptů v ontologii může vypadat např. takto:



- Kategorie lze reprezentovat pomocí logiky, např.:

- Objekt může být **členem** kategorie:  $MemberOf(BB_9, Basketballs)$
- Kategorie může být **podtřídou** jiné kategorie:  $SubsetOf(Basketballs, Balls)$
- Všichni členové kategorie mají nějakou **vlastnost**:  $MemberOf(x, Basketballs) \Rightarrow Round(x)$
- Všichni členové kategorie jsou **rozpoznatelní** podle nějaké vlastnosti:  $Orange(x) \wedge Round(x) \wedge Diameter(x) = 9.5in \Rightarrow MemberOf(x, Basketballs)$
- Kategorie jako celek může mít nějakou **vlastnost**:  $MemberOf(Dogs, DomesticatedSpecies)$
- Kategorie mohou obsahovat složené objekty  $PartOf(Brno, Czechia)$
- Další relace mohou popisovat vztahy částí složených objektů, atd.

- Aktéři pracují s reálnými objekty, ale
- vyvozování provádějí na úrovni kategorií objektů
- Aktér z vnímaných vlastností odvozuje příslušnost objektu do dané kategorie
- Kategorie je množina svých členů
  - jde o objekt s relacemi být členem a být podmnožinou
- Podtřídy uspořádávají kategorie do **taxonomie** jako hierarchické struktury kategorizace objektů

# Akce a události

- Popis pomocí logiky se nemusí týkat jen statického světa, ale lze je rozšířit na dynamické akce

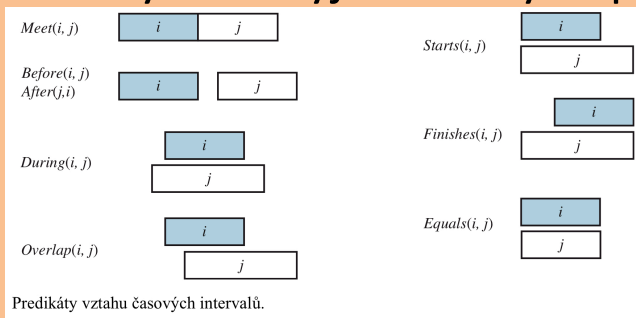
- Například specifikce leteckého spojení:

$MemberOf(E_1, Flyings) \wedge Flyer(E_1, George) \wedge Origin(E_1, Praha) \wedge Destination(E_1, London)$

- nebo účinek události s létáním:

$E = Flyings(a, here, there) \wedge Happens(E, t_1, t_2) \Rightarrow Terminates(E, At(a, here), t_1) \wedge Initiates(E, At(a, there), t_2)$

- se vztahy časů vyjadřovanými pomocí predikátů:



- Role objektů se mohou v čase měnit, odtud jejich začlenění pod události, např. *President(USA)* a tvrzení o prvním prezidentovi:

$T(Equals(Prezident(USA), GeorgeWashington), Begin(AD1790), End(AD1790))$

# Zjemnění používané logiky

- Referenční transparentnost – zachování sémantiky při náhradě podvýrazů jinými podvýrazy se stejným významem – neumožňuje korektní práci s propozičními postoji (slovesa „věří“, „ví“, „zná“, „chce“, „informuje“)
- Řešit to lze pomocí modálních logik a formalizace možných světů (propojených relací přístupnosti), které umožňují i formalizaci tvrzení o mentálních objektech
- Nemonotónní logiky, cirkumskripce a defaultní logiky dovolují formalizovat explicitně nevyjadřované předpoklady a defaultní vyvozování (např. předpoklady s platností vyvozené jen z jejich konzistence)
- Specifické úlohy se řeší pomocí systémů udržujících pravdivost – aktualizace znalostí uložených ve znalostní bázi, ale i vysvětlování závěru, který byl vyvozen



# Celkový kontext reprezentace znalostí

- V 60. letech převládal pohled směrem na „reprezentace problémů“
- V 70. letech převládal přístup „expertních systémů“ (resp. „systémů založených na znalostech“)
- První expertní systém DENDRAL (1970-81) dovedl interpretovat výstup z hmotnostního spektrometru stejně dobře jako chemický expert
- Později vzrostl zájem o formalismy standardizovaných znalostí s výrazným přesahem do filosofie a lingvistiky
- Dílčí úspěchy byly dosaženy např. při předpovídání recidivy propouštěných vězňů nebo úspěchu studentů v programech průpravy (Grove a Meehl, 1996) – v 19 případech z 20 výsledek lepší než lidský expert
- Educational Testing Services užívá program pro známkování milionů esejů v testech GMAT – program se shoduje s lidskými hodnotiteli v 97 procentech případů, což je stejná úspěšnost jako sjhoda dvou lidských hodnotitelů – systém je používán od roku 2001

# Celkový kontext reprezentace znalostí

- Specifické oblasti úspěchu:
- Kvalitativní fyzika (vývody bez nutnosti velkých početních operací – používá se pro hodiny, návrh stěračů a šestinohých kráčejších strojů)
- Prostorové vyvozování – poprvé použito cca 1986, dnes spolus kalkulem propojení používáno pro geografické informační systémy
- Psychologické vyvozování – odvozování psychologie chování pro pozorované aktéry („folk psychology“), je používáno v systémech porozumění přirozenému jazyku, kde je důležité odvození postoje člověka např. pro kontinuitu porozumění textu

# Plány

- Posloupnosti akcí = **plány**
- Úkoly v této oblasti zahrnují úlohy projekční, tj. otázku po situaci, která je důsledkem provedených akcí,
- a úlohy plánovací, kde je otázka po posloupnosti akcí vedoucích k dané situaci.
- **Klasické plánování** je úloha hledání posloupnosti akcí v diskrétním, deterministickém, statickém a plně pozorovatelném prostředí
- Planning Domain Definition Language (PDDL) – stav vyjádřen jako konjunkce základních atomických výrazů popisujících následující stav
- Každá akce je popsána dvojicí axiomů stanovujících použitelnost akce a její efekt, např.
  - $At(p,from) \wedge Plane(p) \wedge Airport(from) \wedge Airport(to) \Rightarrow Fly(p,from,to)$
  - $Fly(p,from,to) \Rightarrow \neg(At(p,from)) \wedge At(p,to)$
- K tomu tzv. problém rámce, přidání velkého počtu axiomů konstatujících, co se akcí nemění, resp. efektivněji pro stavy (fluenty) jako axiom následujícího stavu; obdobně předpoklad uzavřeného světa (CWA) jako předpoklad úplnosti poskytnuté informace, nepravdivosti neuvedených tvrzení

# Plánování

- Plánovací problém s jazykem  $L$  a množinou operátorů  $O$  sestává
  - z plánovací domény nad  $L$  s operátory z  $O$  se stavy jako trojice,
  - ze stavů jako podmnožiny instanciováných atomů nad  $L$ ,
  - z akcí  $A$  jako všech instanciováných operátorů z  $O$  nad  $L$  pro použitelné akce (splněny jejich podmínky a nejedná se o cílový stav) a
  - z přechodové funkce generované účinky akcí s tím, že je vůči ní množina stavů uzavřená,
- počátečního stavu  $a$
- množiny instanciováných literálů splňujících cílovou podmínku.
- Prohledávaný prostor odpovídá stavovému prostoru, kde
  - uzly odpovídají stavům,
  - hrany odpovídají stavovým přechodům pomocí akcí a
  - cílem je najít cestu mezi počátečním stavem a některým koncovým stavem (instanciováným literálem splňujícím koncovou podmínku).

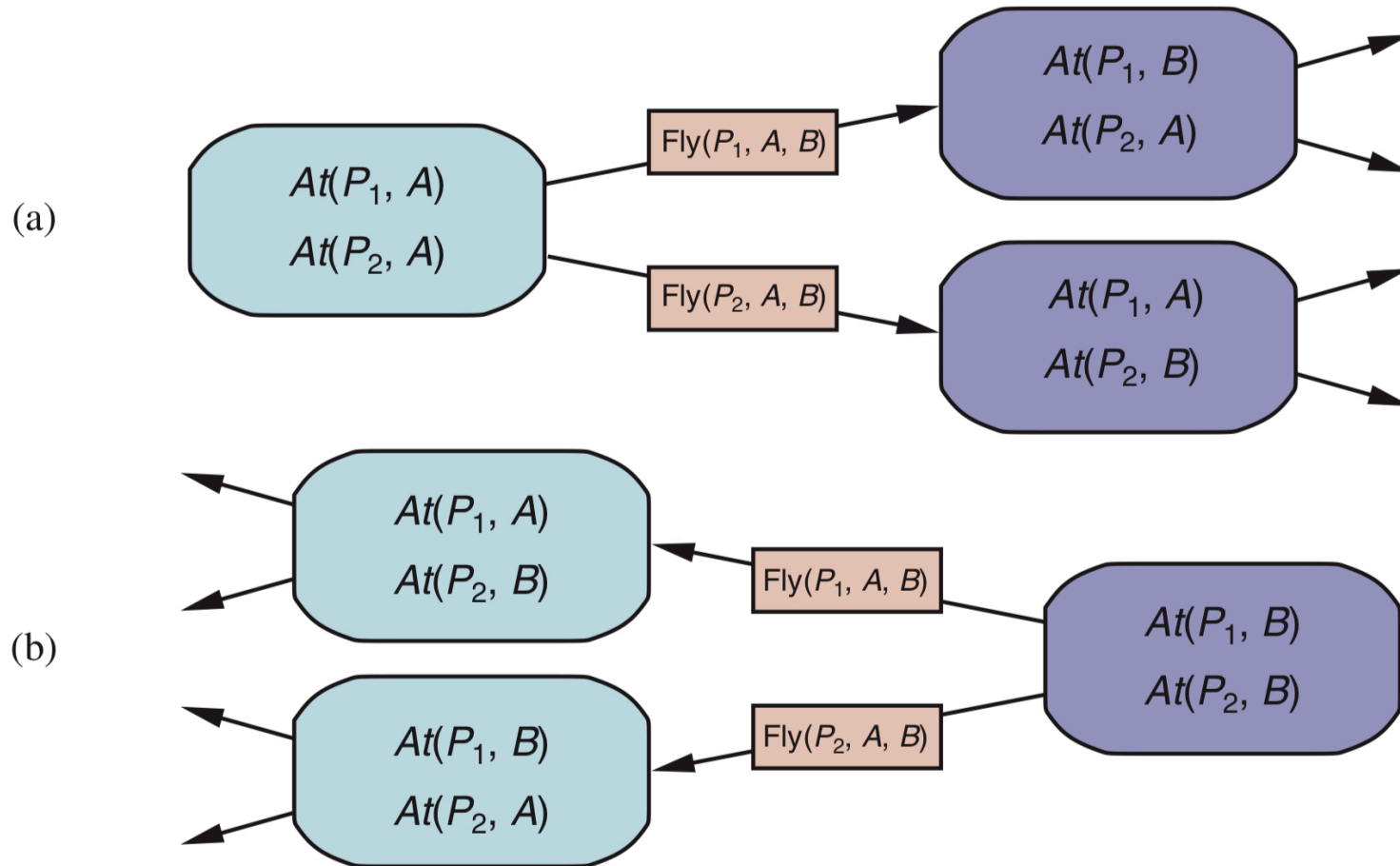
# Dopředné plánování

- Hledání plánu jako nejkratší cesty z počátečního stavu do některého z cílových.
- Jde o korektní metodu (pokud vrátí řešení, jedná se o plán),
- V nedeterministické verzi výběru dalšího uzlu je úplná (najde řešení, pokud existuje).
  
- V deterministické implementaci lze realizovat jako:
  - Prohledávání do šířky
    - korektní a úplné, ale paměťově velmi náročné (nutnost pamatovat si všechny uzly příslušné úrovně v prohledávaném stromu)
  - Prohledávání do hloubky
    - Korektní, ale není úplné kvůli riziku nekonečných cyklů
    - Efektivně implementovatelné vzhledem k paměti, protože jako datová struktura stačí jen struktura prohledávané cesty v zásobníku
    - Korekce kontrolou cyklů, opakovaných výskytů stejného uzlu na cestě, nebo iterativním prohlubováním (je úplné a paměťová náročnost se nahrazuje opakovaným procházením uzlů do limitované hloubky metodou hledání do hloubky)
  
- Algoritmus  $A^*$  s vhodnou heuristikou  $h(x)$ , která v daném uzlu stromu odhaduje celkovou cenu jako  $f(n) = g(n) + h(n)$ ,  
kde  $g(n)$  je aktuální cena uzlu (dosavadní délka cesty) a  
 $h(n)$  je odhad ceny (délky cesty) pro výpočet úplného řešení jako tzv. přípustná heuristika, která celkovou cenu nenahodnocuje (nestanoví ji větší než skutečnou délku cesty, tj. např. přímá cesta z daného uzlu do cíle).
  
- Vysoký stupeň větvení v prohledávaném stromu výrazně komplikuje časové nároky řešení

# Zpětné plánování

- Hledání plánu z množiny (všech) instanciovaných cílových stavů na základě schopnosti určit o uzlu, zda odpovídá cíli, najít pro daný uzel relevantní akce (jichž je tento uzel výsledkem) a jeho podcíle (z nichž těmito akcemi mohl vzniknout)
- Jde o korektní metodu a při detekci cyklů i úplnou metodu při deterministické implementaci
- Obecně menší stupeň větvení, lze jej dále redukovat tzv. liftingem (práci s proměnnými a instanciací nejobecnějším unifikátorem)

# Dopředné a zpětné plánování



Dva přístupy k hledání plánu. (a) Dopředné plánování v podobě prohledávání stavového prostoru, které začíná v počátečním stavu a pokračuje akcemi spojenými s problémem v hledání prvku cílových stavů. (b) Zpětné plánování popisuje stavy, které začíná v cíli a používá inverze akcí k tomu, aby hledal směrem k počátečnímu stavu.

# Problém výpočetní náročnosti

- Plánování kombinuje dvě oblasti AI, *vyhledávání* a *logiku*
- Prakticky se daří implementovat techniky použitelné v netriviálních systémech (miliony stavů, tisíce akcí)
- Kritickým omezením je kombinatorická exploze (výpočetní složitosti příslušných algoritmů): v doméně s  $N$  propozicemi je  $2^N$  stavů
- Výrazně pomáhá rozložitelnost problému – dopředné hledání dokáže heuristicky hledat podmnožiny propozic odpovídající nezávislým podproblémům
- Existují systémy pracující s portfoliem algoritmů, které je zkoušejí vedle sebe.
- Úplné porozumění volby algoritmů však dosud neexistuje.
- Plánování bylo od počátku AI jednou z ústředních aktivit, používalo se na SRI u robota Shakleyho i u GPS (General Problem Solver)
- Dnes existují netriviální aplikace mimo jiné pro plánování optimálního pozorovacího času na satelitních teleskopech, běžně se používá v továrnách nebo obchodech.