Znalosti a jejich reprezentace, základní postupy, výroková logika

Jiří Kléma

Katedra kybernetiky,
FEL, ČVUT v Praze

http://cw.felk.cvut.cz/doku.php/courses/a4b33zui/start

Průchod blokem pro témata znalosti, usuzování a rozhodování

- P1: znalosti a jejich reprezentace
  - motivační příklady, základní postupy, výroková logika,
  - jazyk reprezentace, souvislost s následným usuzováním,
- P2: predikátová logika, situační kalkulus, produkční systémy
  - jak modelovat změnu v klasické logice, problém rámce,
  - další tradiční způsoby reprezentace,
- P3: rozhodování za neurčitosti
  - jednotlivá rozhodnutí agenta v nejistém světě,
  - jsou lidé racionální agenti?
- P4: sekvenční rozhodování za neurčitosti
  - jak optimalizovat kumulovanou odměnu pokud se rozhodnutí ovlivňují?
  - markovské rozhodovací procesy.

Celá čísla – motivační příklad 1

:: Srovnejme dvě odlišné reprezentace čísel – **arabskou** a **římskou**:

- Délka zápisu
  - 1000 vs. M, 1997 vs. MCMXCVII, 100000 vs. MMMM...M,
  - průměrná délka (PD) zápisu,
  - doplněná o odhad redundance (R), protože jeden kód pracuje s 10 symboly a druhý pouze se 7

Soustava	Arabská		Římská	
Interval	PD[znaků]	R[%]	PD[znaků]	R[%]
1..1000	2.89	0.3	6	11.5
1..3000	3.63	2.6	7	5.5

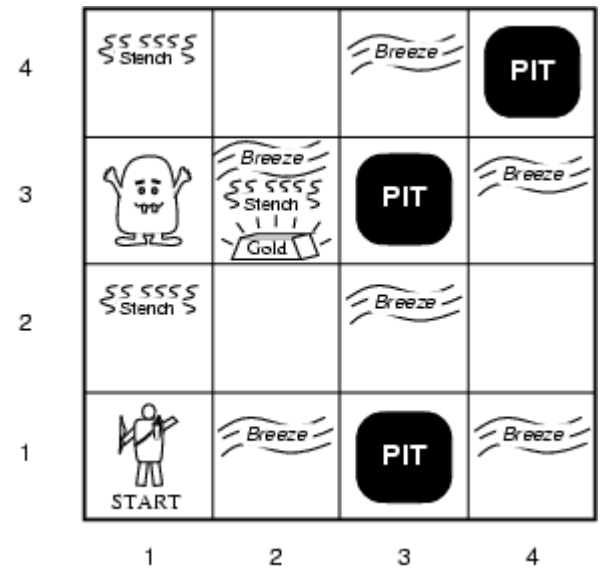
- Složitost aritmetických operací, př. sčítání a dělení
  - arabská čísla jsou poziční (operace pod sebou), existuje symbol pro 0,
  - u římských čísel je aritmetika obtížnější (problémy s notací IV → IIII, nutnost expanze L → XXXXX, apod.).

:: Závěr: reprezentace významně spoluurčuje způsob, efektivitu a srozumitelnost nakládání s čísly.

Wumpusův svět – motivační příklad 2

:: **Hunt the Wumpus** byla jednou z prvních počítačových her. Wumpus je záhadnou příšerou číhající v systému jeskyní. Definice prostředí je:

- Hodnocení: zlato +1000, smrt -1000 (wumpus i šachta), -1 za krok, -10 za použití šípu
- Prostředí: jeskyně sousedící s wumpusem smrdí, jeskyně vedle šachty jsou větrné, jeskyně se zlatem se třpytí,
- Senzory: náraz (pro krok do vnější zdi), zápach, vánek a třpyt (pouze lokálně v dané jeskyni), výkřik (slyšitelný kdekoli v bludišti),
- Akce: otoč se vlevo, otoč se vpravo, jdi vpřed, uchop (sebere zlato, hráč musí být v jeskyni se zlatem), polož (zlato), vystřel (šíp zabije wumpuse pokud hráč stojí čelem k němu – hráč má jediný šíp).



©Russel, Norvig: Artificial Intelligence: A Modern Approach.

:: Cílem je volit akce tak, aby maximalizovaly bodový zisk hráče. Ten začíná i končí na poli [1,1].

Wumpusův svět – intencionální a reaktivní agent

:: Charakteristika úlohy:

- deterministická (výstupy jsou přesně dány), statická (wumpus ani šachty se nepohybují), sekvenčně-diskrétní (jednotkou je krok), prostředí není plně pozorovatelné (lokální senzory),
- uvažujme prostředí 4x4 s jediným wumpusem i pokladem a pstí šachty v každém poli 0.2 (startovací pole je vždy prázdné).

:: Šance intencionálního* (IA) a reaktivního** agenta (RA):

- většina zadání je neřešitelná (zlato je v šachtě, je šachtami obklíčeno, agent se nemůže bezpečně pohnout apod.),
- IA ve 30% případů najde bezpečně zlato a donese je zpět na start, ve zbytku případů musí riskovat nebo se vrátit bez zlata,
- RA bezpečně uspěje pouze ve 20% případů, provede mnohem více akcí.

:: Závěr: vnitřní reprezentace světa a schopnost základní inference jsou nutnými podmínkami úspěšného agenta/hráče. Řešení IA viz. dále.

---

* **Intencionální agent** pracuje s vnitřním modelem světa (tj. pamětí), využívá znalostní bázi a inferenci.

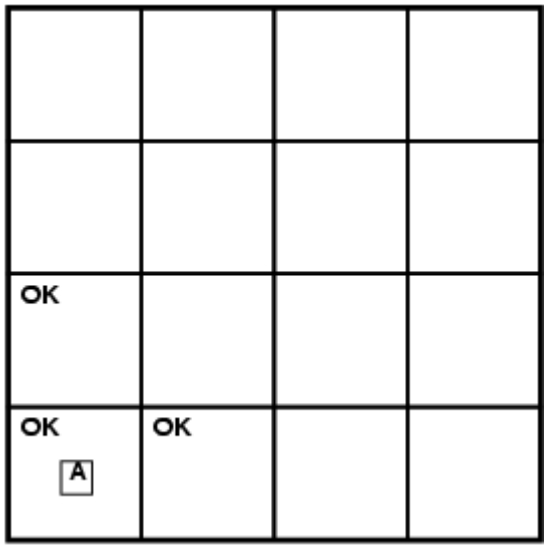
* **Reaktivní agent** nemá vnitřní reprezentaci ani model světa, řídí se pouze aktualním stavem senzorů.

Ve wumpusově světě (1)

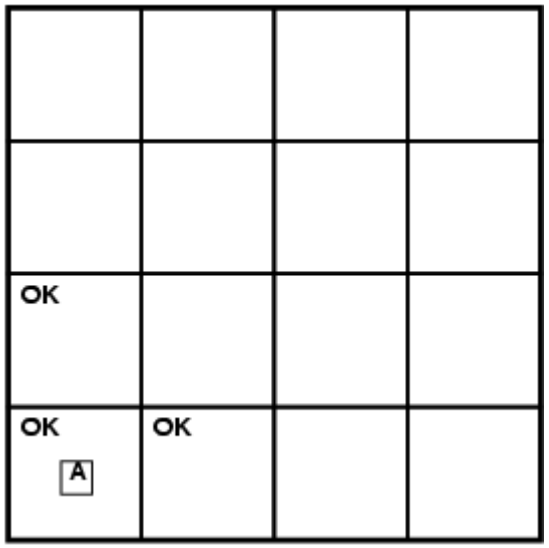
:: Reaktivní agent:

- if Třpyt=ano then Akce=uchop,
- if Náraz=ano then Akce=rand(otoč se vpravo, otoč se vlevo),
- if Zápach=ano or Vánek=ano then Akce=[otoč se vlevo, otoč se vlevo, jdi vpřed, rand(otoč se vpravo, otoč se vlevo, jdi vpřed)].

:: Průchod světem:



RA – vyhodnocení 1



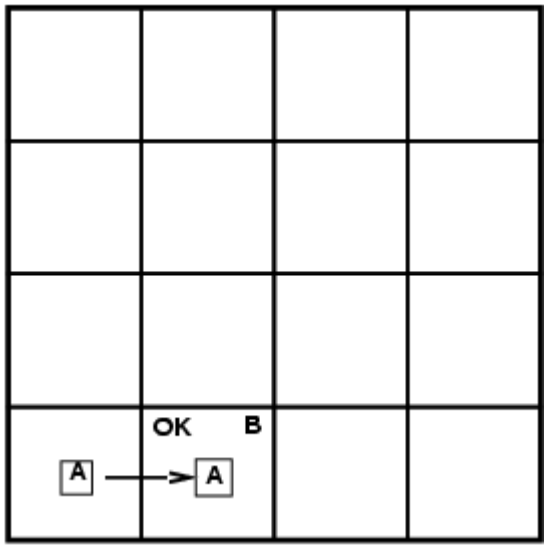
IA – inference 1

Ve wumpusově světě (2)

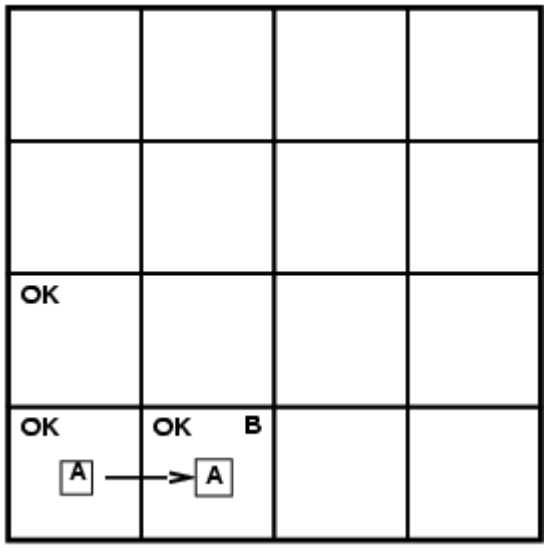
:: Reaktivní agent:

- if Třpyt=ano then Akce=uchop,
- if Náraz=ano then Akce=rand(otoč se vpravo, otoč se vlevo),
- if Zápach=ano or Vánek=ano then Akce=[otoč se vlevo, otoč se vlevo, jdi vpřed, rand(otoč se vpravo, otoč se vlevo, jdi vpřed)].

:: Průchod světem:



RA – krok 1



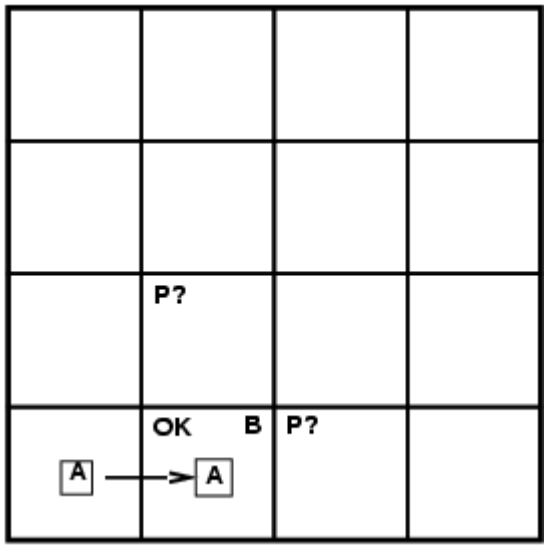
IA – krok 1

Ve wumpusově světě (3)

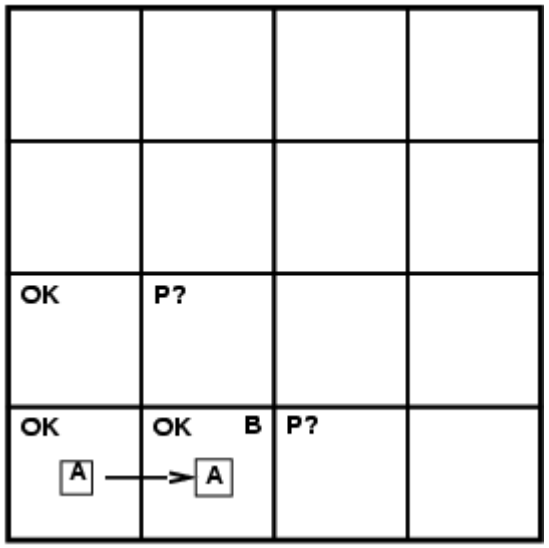
:: Reaktivní agent:

- if Třpyt=ano then Akce=uchop,
- if Náraz=ano then Akce=rand(otoč se vpravo, otoč se vlevo),
- if Zápach=ano or Vánek=ano then Akce=[otoč se vlevo, otoč se vlevo, jdi vpřed, rand(otoč se vpravo, otoč se vlevo, jdi vpřed)].

:: Průchod světem:



RA – vyhodnocení 2



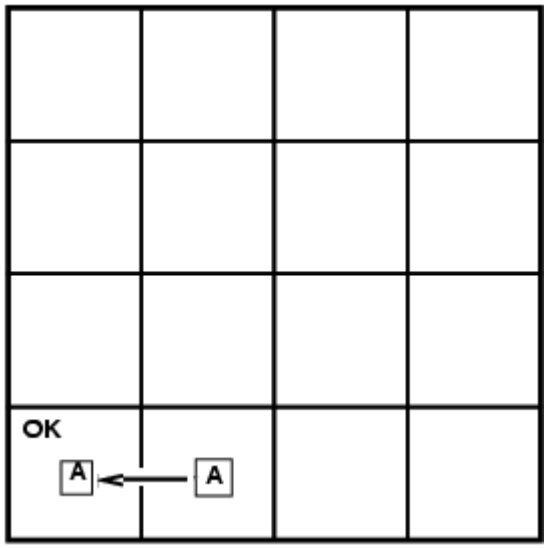
IA – inference 2

Ve wumpusově světě (4)

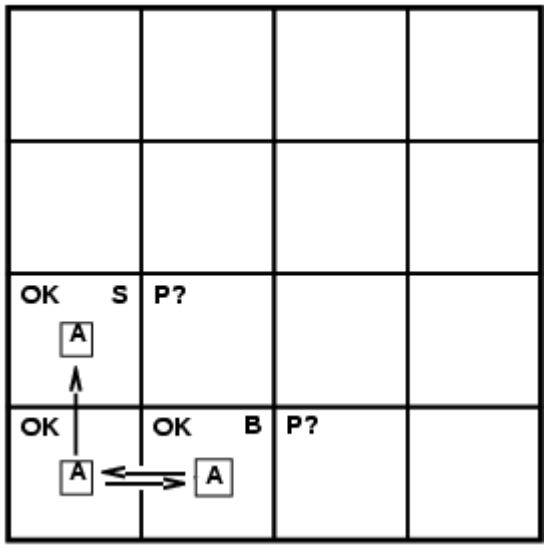
:: Reaktivní agent:

- if Třpyt=ano then Akce=uchop,
- if Náraz=ano then Akce=rand(otoč se vpravo, otoč se vlevo),
- if Zápach=ano or Vánek=ano then Akce=[otoč se vlevo, otoč se vlevo, jdi vpřed, rand(otoč se vpravo, otoč se vlevo, jdi vpřed)].

:: Průchod světem:



RA – krok 2



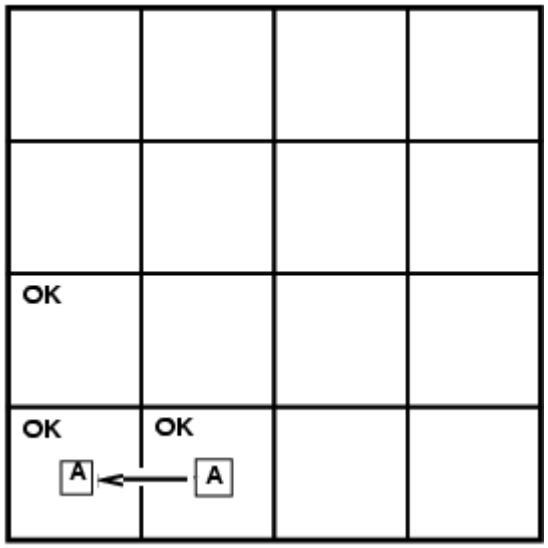
IA – kroky 2 a 3

Ve wumpusově světě (5)

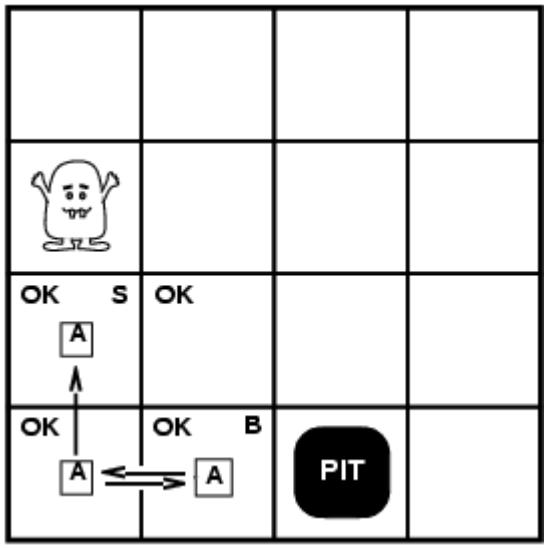
:: Reaktivní agent:

- if Třpyt=ano then Akce=uchop,
- if Náraz=ano then Akce=rand(otoč se vpravo, otoč se vlevo),
- if Zápach=ano or Vánek=ano then Akce=[otoč se vlevo, otoč se vlevo, jdi vpřed, rand(otoč se vpravo, otoč se vlevo, jdi vpřed)].

:: Průchod světem:



RA – vyhodnocení 3



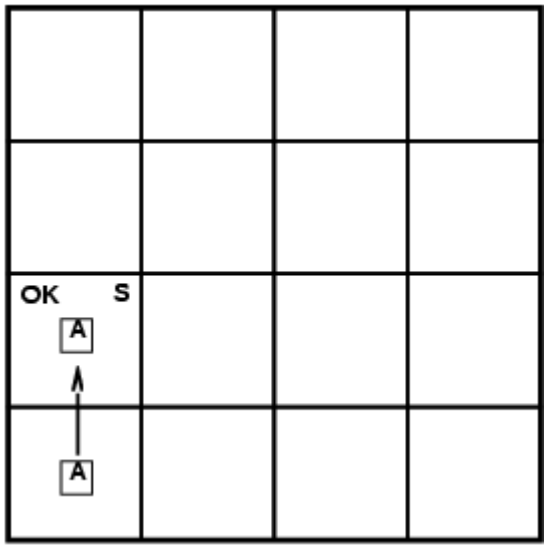
IA – inference 3

Ve wumpusově světě (6)

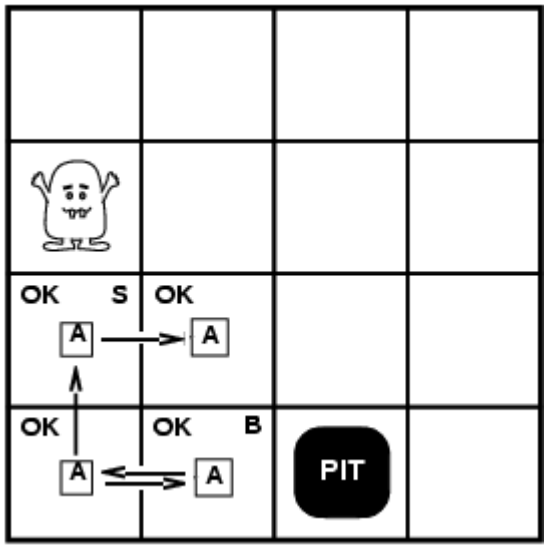
:: Reaktivní agent:

- if Třpyt=ano then Akce=uchop,
- if Náraz=ano then Akce=rand(otoč se vpravo, otoč se vlevo),
- if Zápach=ano or Vánek=ano then Akce=[otoč se vlevo, otoč se vlevo, jdi vpřed, rand(otoč se vpravo, otoč se vlevo, jdi vpřed)].

:: Průchod světem:



RA – krok 3



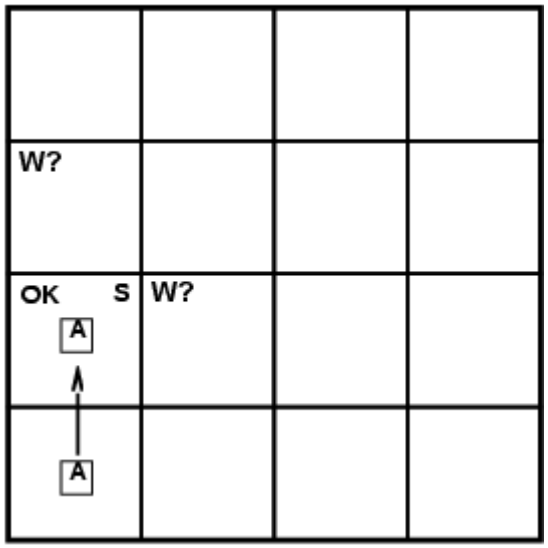
IA – krok 4

Ve wumpusově světě (7)

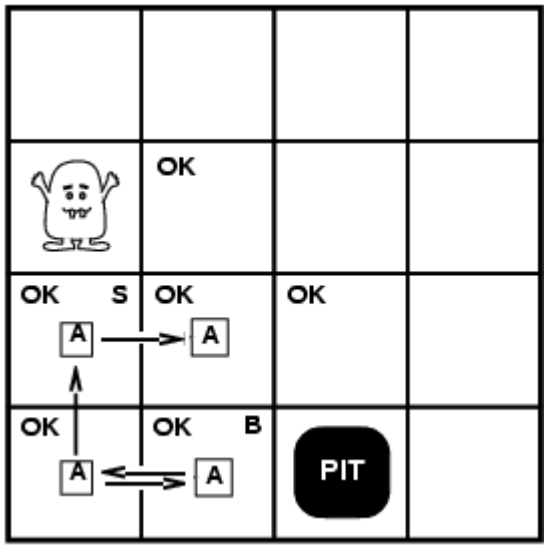
:: Reaktivní agent:

- if Třpyt=ano then Akce=uchop,
- if Náraz=ano then Akce=rand(otoč se vpravo, otoč se vlevo),
- if Zápach=ano or Vánek=ano then Akce=[otoč se vlevo, otoč se vlevo, jdi vpřed, rand(otoč se vpravo, otoč se vlevo, jdi vpřed)].

:: Průchod světem:



RA – vyhodnocení 4



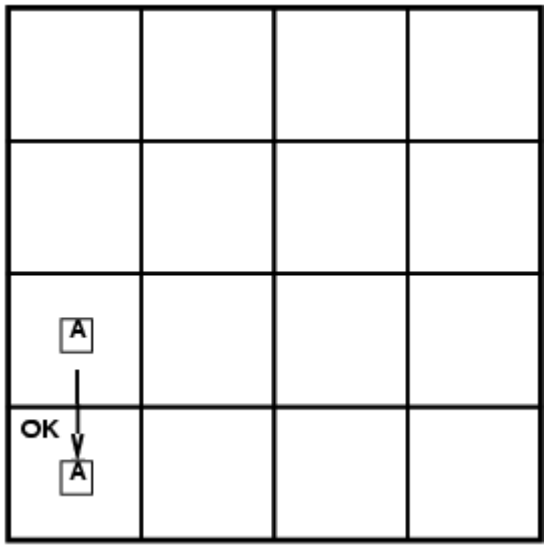
IA – inference 4

Ve wumpusově světě (8)

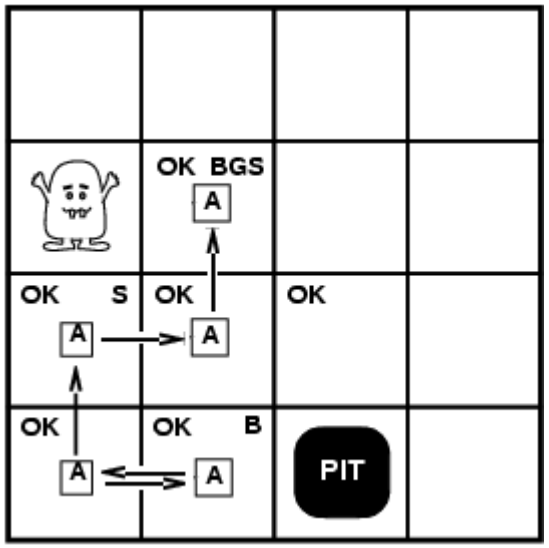
:: Reaktivní agent:

- if Třpyt=ano then Akce=uchop,
- if Náraz=ano then Akce=rand(otoč se vpravo, otoč se vlevo),
- if Zápach=ano or Vánek=ano then Akce=[otoč se vlevo, otoč se vlevo, jdi vpřed, rand(otoč se vpravo, otoč se vlevo, jdi vpřed)].

:: Průchod světem:



RA – krok 4



IA – krok 5

Klade Zuzana vajíčka? – motivační příklad 3

:: Pokud vám řeknu, že:

- (S1) Ptakopysk a ježura jsou jediní savci kladoucí vejce.
- (S2) Pouze ptáci a savci jsou teplokrevní.
- (S3) Zuzana, můj pásovec, je teplokrevná a nemá peří.
- (S4) Každý pták má peří.

:: a zeptám se: (D) Klade Zuzana vejce?

Klade Zuzana vajíčka? – motivační příklad 3

:: Pokud vám řeknu, že:

- (S1) Ptakopysk a ježura jsou jediní savci kladoucí vejce.
- (S2) Pouze ptáci a savci jsou teplokrevní.
- (S3) Zuzana, můj pásovec, je teplokrevná a nemá peří.
- (S4) Každý pták má peří.

:: a zeptám se: (D) Klade Zuzana vejce?

:: Inference v přirozeném jazyce:

- Zuzana nemá peří a proto není pták.
- Zuzana je teplokrevná a není pták, proto musí být savec.
- Zuzana je savec a pásovec, protože není ježura ani ptakopysk nemůže klást vejce.

:: Jak realizovat strojově? Reprezentace pomocí řetězců je problematická z hlediska odvozování nových tvrzení ...

IBM Watson – motivační příklad 4

:: Systém pro zodpovídání otázek kladených v přirozeném jazyce

- ve vědomostní soutěži Jeopardy (Riskuj) v roce 2011 porazil dva historicky nejúspěšnější lidi,
- místo otázek slovní rébusy, soutěžící formuluje otázku definovanou předloženým textem,
- integruje techniky zpracování přirozeného jazyka, vyhledávání informací, reprezentace znalostí a strojového učení.

:: Zřejmě obtížnější problém než u DeepBlue (šachový systém, také IBM)

- řešení otázek v přirozeném jazyce není ohraničený problém.

:: Z pohledu reprezentace znalostí

- heuristický přístup odlišný od "klasického" logického s formálním dokazováním,
- pokrytí a rychlost na úkor přesnosti,
- nejde ale o prosté statistické vyhledávání informací bez strukturované reprezentace (tomu chybí přesnost už příliš),
- hybridní přístup: nestrukturovaný text, ale i relace a strukturovaná a polo-strukturovaná znalost mj. ze sémantického webu.

IBM Watson – motivační příklad 4

:: Příklady rébusů a výsledných otázek

- rébus literární postavy: Pokrývka hlavy toho, kdo oslovuje přítele slovem milý.
- otázka: Co nosí na hlavě Sherlock Holmes?
- rébus americká města: Má největší letiště pojmenované po hrdinovi 2. světové války, druhé největší po bitvě 2. světové války.
- otázka: Co je Chicago? (O'Harovo a Midway)

:: Dovednosti nutné k řešení rébusu

- shopnost rozložit rébus na podúkoly,

  (americké město, hrdinové, bitvy, letiště),
- mít všechny informace k dispozici,
- složit dílčí odpovědi a sestavit otázku.

Znalosti

:: Obecný a často používaný pojem

- epistemologie (teorie poznání) versus znalostní inženýrství,
- různé úrovně abstrakce: data, informace, znalosti.

:: Způsoby chápání znalosti

- pro osobu - odbornost a dovednosti získané učením nebo ze zkušenosti,
  - Petr ví, že Tereza přijde na schůzku. (vztah mezi osobou a popisným tvrzením)
  - Petr věří, že Tereza přijde na schůzku. (existují různé druhy vztahů ...)
  - Petr umí hrát na piano. Petr zná Terezu. (explicitní popisné tvrzení chybí)
- pro obor - veškerá známá fakta a informace,
- spolehlivé porozumění věci se schopností ji vhodně používat.

:: Jaké **kognitivní** schopnosti jsou obvykle třeba k získání znalostí

- vnímání, učení, komunikace, asociace, uvažování.

:: Dále se podíváme na související klíčové pojmy: reprezentaci (blíže) a usuzování (méně).

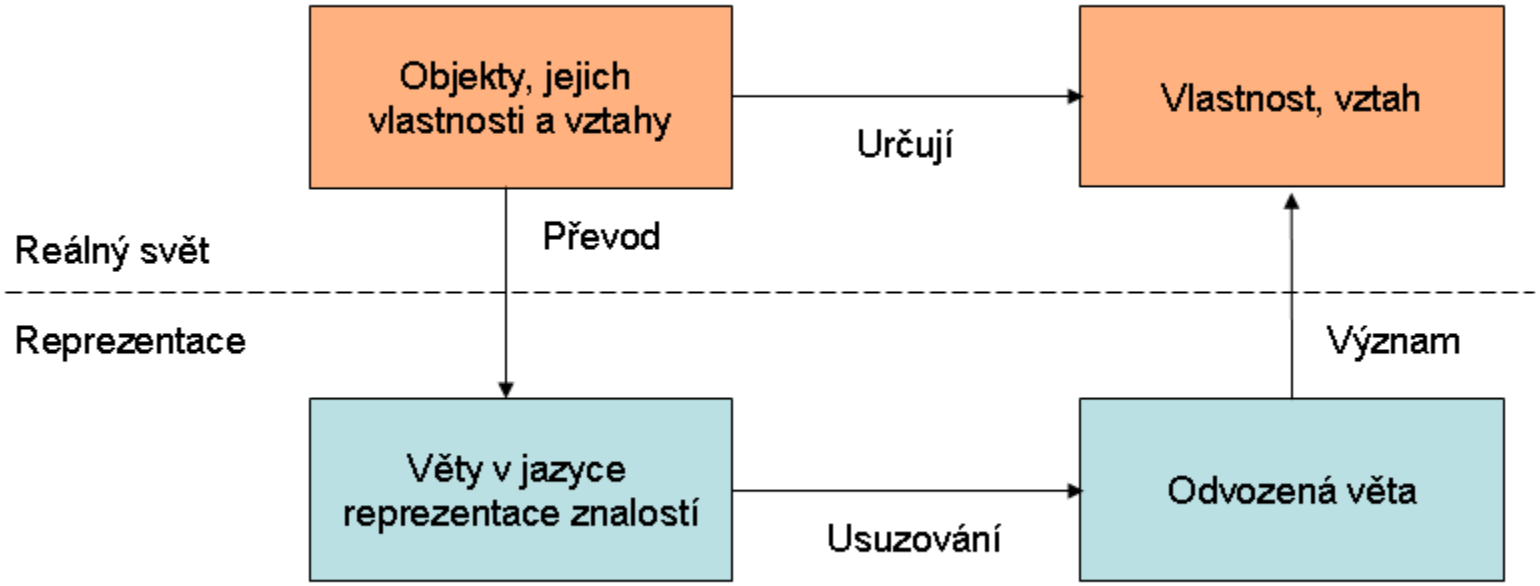
Reprezentace znalostí

:: Co to je? Jakou může hrát roli?

- náhražka, umožňuje rozhodovat o následcích **uvažováním** bez **provádění** skutečných akcí
  - vnitřní substituce vnějších objektů, u reálných objektů nutně nedokonalá,
- prostředek pro efektivní výpočetní realizaci uvažování a myšlení
  - mechanistický pohled – důraz na efektivitu inference jako výpočetního procesu,
- odpověď na otázku jakým způsobem bych měl uvažovat o světě
  - volba reprezentace spoluurčuje způsob myšlení,
  - některé inferenční postupy jsou snadné a jiné naopak,
- prostředek lidského vyjádření
  - jakou formou předávat informace o světě jiným lidem (nebo strojům).

Reprezentace znalostí – základní schéma

- Jazyk pro reprezentaci znalostí musí mít jasně danou **syntaxi** a **sémantiku**,
- je nutné explicitně znát význam vět jazyka v reálném světě.



Jazyk pro reprezentaci znalostí – základní vlastnosti

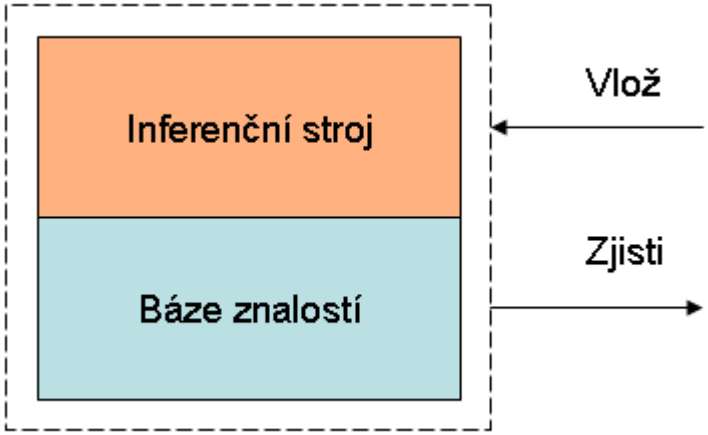
- Přirozenost
  - věty jazyka jsou intuitivně srozumitelné,
  - srovnej pták(zuzana) a ydbk!op pro tvrzení: Zuzana je pták.
- Expresivita
  - požadovanou škálu objektů, jejich vlastností a vztahů lze reprezentovat,
  - např. výroková logika není dost expresivní pro tvrzení/problém:
    - Každý člověk je smrtelný. Sokrates je člověk. Je Sokrates smrtelný?
- Vhodnost pro usuzování
  - **korektnost** usuzování – odvozené věty v reálném světě vždy platí,
  - efektivita usuzování – odvození s rozumnou časovou a paměťovou složitostí,
  - **úplnost** usuzování – lze odvodit všechny věty, které vyplývají z aktuální báze znalostí, zpravidla v rozporu s požadavky na efektivitu.

:: Přirozený jazyk (čeština) je přirozený, expresivní, ale zcela nevhodný pro automatické usuzování. Mj. je nejednoznačný (Prohnal si kulí hlavu.) a kontextově závislý (Je to pěkný párek.)

:: Neexistuje univerzální jazyk, různé jazyky jsou vhodné pro různé problémy. Optimální je volba nejjednodušší dostatečně expresivní reprezentace pro daný problém.

Znalosti – reprezentace a použití

- Báze znalostí
  - množina vět v jazyce pro reprezentaci znalostí,
  - data (znalosti) specifická pro daný problém, jak uživatelská, tak i odvozená,
  - pracovní paměť – popisuje okamžitý stav řešeného problému.
- Inferenční stroj
  - realizuje usuzování – odvozování nových znalostí,
  - problémově nezávislý obsah (obecná odvozovací pravidla, mechanismus řešení konfliktů),
  - řízení báze znalostí.



Znalostní inženýrství

- Vývoj a údržba znalostních systémů
  - tj. počítačových systémů napodobujících lidské řešení problémů založených na znalostní bázi a inferenčním mechanismu.

- Identifikace a konceptualizace problému
  - přehled o pojmech a vztazích,
  - výsledkem je abstraktní model – např. ve formě znalostní **ontologie**,
- Formalizace znalostí a implementace systému
  - volba vhodné reprezentace znalostí a formalizace konceptuálního modelu,
  - datové typy, operační systém, programové prostředí – často standardní,
- Jak znalosti získat?
  - přejímání od expertů $\times$ induktivní odvozování z dat,
  - data – ověřitelná elementární fakta,
  - znalosti – zobecněná tvrzení, návody jak data interpretovat a využívat,
- Testování a údržba systému
  - konzultace výsledků s odborníky – úpravy a rozšiřování báze i změny modelu,
  - aktualizace a opravy znalostní báze – mj. problém konzistence.

Znalostní inženýrství vs klasické programování

- Jaká je analogie mezi oběma typy činností?
  - lze je rozložit na 4 odpovídající si kroky

Znalostní inženýrství	Programování
Volba jazyka reprezentace	Volba programovacího jazyka
Tvorba báze znalostí	Psaní programu
Volba (implementace) inferenčního mechanismu	Volba (implementace) překladače
Odvozování nových znalostí	Spuštění programu

- Znalostní inženýrství využívá **deklarativní** (popisný, konstatující) přístup
  - je méně explicitní – pouze objekty a platnost vztahů mezi nimi,
  - použití i v databázových systémech nebo deklarativním programování (Prolog – viz dále).

Jazyky pro reprezentaci znalostí – přístupy

- Logická schémata
  - deklarativní znalostní báze ve formě logických tvrzení,
  - poté klasická logická inference pro konkrétní fakta reprezentující problém,
  - nejčastějším formalismem je predikátová logika 1. řádu, nástrojem pak jazyk PROLOG.
- Procedurální schémata
  - znalostní báze ve formě instrukcí,
  - typickým příkladem jsou pravidlové produkční systémy – pravidla typu if ... then ...
- Síťové modely
  - znalost je uchována ve formě grafu,
  - objekty či pojmy jsou uzly, vztahy mezi nimi reprezentují hrany,
  - příkladem jsou sémantické sítě.
- Strukturované modely
  - rozšíření síťových modelů, uzly grafu mohou být komplexními strukturami,
  - příkladem jsou skripty, rámce a objekty.

Logika

- Formální jazyk s jasně definovanými:
  - syntaxí – jak se sestaví korektní věta,
  - sémantikou – co tato věta znamená v reálném světě,
  - axiomy a odvozovacími pravidly – nástroje pro usuzování.

:: Nejjednodušší je **výroková logika**

- Syntaxe – elementy jazyka
  - neprázdná množina symbolů, každý symbol je atomickou výrokovou formulí,
  - logické spojky $\Rightarrow$ a $\neg$,
  - pro zkrácení zápisu často doplněné o $\wedge$, $\vee$ a $\Leftrightarrow$ (např. $A \vee B$ pro $\neg A \Rightarrow B$),
  - závorky ().
- Syntaxe – definice korektních výrokových formulí
  - každá atomická výroková formule je též výroková formule,
  - jsou-li A a B výrokové formule, pak i $(\neg A)$ a $(A \Rightarrow B)$ jsou výrokové formule,
  - nic jiného výrokové formule nejsou.

Výroková logika

- Sémantika – význam symbolů, spojek a formulí
  - symboly reprezentují elementární výroky o světě, např. P pro "Petr má rád čokoládu.",
  - každý výrok může být buď pravdivý (True) nebo nepravdivý (False),
  - přiřazení pravdivostní hodnoty všem symbolům nazýváme **interpretací**,
  - interpretaci nazveme **modelem** formule, pokud je formule pro danou interpretaci pravdivá,
  - každé spojce odpovídá pravdivostní tabulka, ta vyjadřuje pravdivost složené formule na základě pravdivosti formulí atomických.
- Axiomy – základní pravdivá tvrzení, nedokazují se
  - (A1) $A \Rightarrow (B \Rightarrow A)$,
  - (A2) $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$,
  - (A3) $(\neg B \Rightarrow \neg A) \Rightarrow (A \Rightarrow B)$.
- Odvozovací pravidlo – pro vytváření odvozených formulí a dokazování pravdivosti
  - základním je Modus Ponens: $A, A \Rightarrow B \models B$,
  - základními pro rozšířenou množinu spojek: vyloučení třetího $\models A \vee \neg A$, zavedení konjunkce $A, B \models A \wedge B$, zavedení disjunkce: $A \models A \vee B$, eliminace disjunkce $\neg A, A \vee B \models B$, atd.

Logický důsledek a logické usuzování

- Logický důsledek
  - $\alpha \models \beta$ – věta β je **logickým důsledkem** věty α,
  - $\alpha \models \beta$ iff všechny modely věty α jsou také modelem věty β,
- Logické usuzování (inference)
  - KB $\vdash_i \alpha$ – α je **odvoditelná** z KB (báze znalostí je také větou!) pomocí procedury i,
  - **korektnost** – i je korektní pokud: KB $\vdash_i \alpha \Rightarrow$ KB $\models \alpha$,
  - **úplnost** – i je úplná pokud: KB $\models \alpha \Rightarrow$ KB $\vdash_i \alpha$,

Výroková logika – příklad

:: Na ostrově poctivců a padouchů lidé buď mluví vždy pravdu – poctivci – nebo vždy lžou – padouši. Potkáme A a ten prohlásí: "Já jsem padouch, ale B ne." Určete povahu A a B. Použijte výrokovou logiku.

Výroková logika – příklad

:: Na ostrově poctivců a padouchů lidé buď mluví vždy pravdu – poctivci – nebo vždy lžou – padouši. Potkáme A a ten prohlásí: "Já jsem padouch, ale B ne." Určete povahu A a B. Použijte výrokovou logiku.

- Formalizace úlohy:
  - atomické výroky – a: "A je padouch." b: "B je padouch.",
  - (F1) $a \Rightarrow \neg(a \wedge \neg b)$,
  - (F2) $\neg a \Rightarrow a \wedge \neg b$,
- Řešení pravdivostní tabulkou (model checking)
  - hledáme model(y) splňující současně (F1) i (F2).

a	b	$\neg a$	$\neg b$	$(a \wedge \neg b)$	F1	F2
F	F	T	T	F	F	T
F	T	T	F	F	F	T
T	F	F	T	T	T	F
T	**T**	F	F	F	**T**	**T**

Výroková logika – příklad (pokr.)

- Deduktivní důkaz – použití libovolných odvozovacích pravidel
  - (F1) $a \Rightarrow \neg(a \wedge \neg b)$,
  - (F2) $\neg a \Rightarrow a \wedge \neg b$,
  - (F3) $\neg a \vee \neg(a \wedge \neg b)$ (převod implikace v (F1)),
  - (F4) $\neg \neg a \vee (a \wedge \neg b)$ (převod implikace v (F2)),
  - (F5) $\neg a \vee \neg a \vee b$ (de Morganův zákon v (F3)),
  - (F6) $(a \vee a) \wedge (a \vee \neg b)$ (zákon dvojí negace a distribuce v (F4)),
  - (F7) $\neg a \vee b$ (idempotence (F5)),
  - (F8) $a \wedge (a \vee \neg b)$ (idempotence (F6)),
  - (F9) a (absorbce (F8)),
  - (F10) $a \wedge b$ (absorbce negace (F7) $\wedge$ (F9)),
  - (F11) b (eliminace konjunkce (F10)).

Výroková logika – příklad (pokr.)

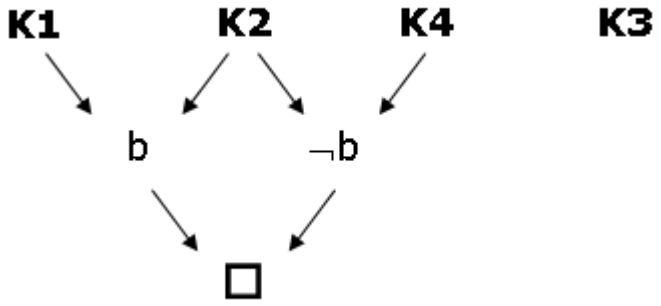
- Axiomatický důkaz – pouze axiomy a modus ponens (MP)
  - (F1) $a \Rightarrow \neg(a \wedge \neg b)$,
  - (F2) $\neg a \Rightarrow a \wedge \neg b$,
  - (F3) $a \Rightarrow \neg a \vee b \Leftrightarrow a \Rightarrow (a \Rightarrow b)$ (F1 pouze s implikacemi),
  - (F4) $\neg a \Rightarrow \neg\neg(a \wedge \neg b) \Leftrightarrow \neg a \Rightarrow \neg(a \Rightarrow b)$ (F2 pouze s implikacemi),
  - (A1) $A \Rightarrow (B \Rightarrow A)$,
  - (A2) $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$,
  - (A3) $(\neg B \Rightarrow \neg A) \Rightarrow (A \Rightarrow B)$,
  - (F5) $(a \Rightarrow (a \Rightarrow b)) \Rightarrow ((a \Rightarrow a) \Rightarrow (a \Rightarrow b))$ (dosazení do (A2) A/a, B/a, C/b),
  - (F6) $(a \Rightarrow a) \Rightarrow (a \Rightarrow b)$ (MP (F3), (F5)),
  - (F7) $(a \Rightarrow b)$ (MP (F6), $(a \Rightarrow a)$, **platnost druhého bez důkazu – zkuste doma!**),
  - (F8) $(\neg a \Rightarrow \neg(a \Rightarrow b)) \Rightarrow ((a \Rightarrow b) \Rightarrow b)$ (dosazení do (A3) $A/(a \Rightarrow b)$, B/a),
  - (F9) $(a \Rightarrow b) \Rightarrow a$ (MP (F4), (F8)),
  - (F10) a (MP (F7), (F9)),
  - (F11) b (MP (F7), (F10)).

Rezoluce

- často potřebujeme **úplný** mechanismus odvozování,
- deduktivní aplikace pravidel?
  - volba axiomu, odvozovacího pravidla či tvrzení závisí na intuici,
  - důkaz se obtížně automatizuje – snaha o úplnost vede ke kombinatorické explozi.
- model checking?
  - korektní a úplné, ale $O(2^n)$, kde n je počet symbolů.
- **rezoluce** je korektní a v kombinaci s úplným prohledávacím algoritmem úplná (úplnost popřením – daný výrok potvrdí či popře, negeneruje výčet platných tvrzení!),
- příčiny "efektivity":
  - úplná normální konjunktivní forma – minimalizuje počet použitelných pravidel
    - literál – atomická formule nebo její negace, klauzule – disjunkce literálů,
    - rezoluční pravidlo: $\mathsf{A} \vee \psi$, $\neg\mathsf{A} \vee \phi \models \psi \vee \phi$,
    - z dvojice klauzulí odvodí jejich rezolventu, A a $\neg$A jsou doplňkové literály.
  - důkaz sporem – lepší řízení průchodu prohl. prostorem, orientuje se na cíle
    - $\mathsf{T} \models \phi$ iff $\mathsf{T} \cup \{\neg\phi\} \models \Box$,
    - spor odpovídá prázdné klauzuli, značení $\Box$.

Rezoluce – příklad (pokr.)

- Rezoluční důkaz – nejprve převod do klauzální formy
  - (F1) $a \Rightarrow \neg(a \wedge \neg b)$,
  - (F2) $\neg a \Rightarrow a \wedge \neg b$,
  - (K1) $\neg a \vee b$ (úprava (F1), viz. deduktivní důkaz),
  - (K2,3) $a \wedge (a \vee \neg b)$ (úprava (F2), viz. deduktivní důkaz).
- Rezoluční důkaz – doplnění o negaci dokazovaného tvrzení
  - (D1) $a \wedge b$,
  - $(\neg D1) \equiv$ (K4) $\neg a \vee \neg b$.
- Strom rezolučního zamítnutí *



* Nelze rezolvovat K1 a K3. Proč?

Rezoluce – strategie prohledávání, složitost

- Jak vybírat klauzule, které mají být rezolvovány?
  - řízeno **strategií** prohledávání,
  - tvoří derivační graf (DG) – listy jsou klauzule, uzly jsou rezolventy,
  - strom rezolučního zamítnutí – podgraf DG jehož kořenem je $\square$.
- Úplné strategie
  - prohledávání do šířky
    - nejprve všechny rezolventy 1. řádu (1 aplikace rezolučního pravidla na výchozí množinu klauzulí), poté všechny rezolventy 2. řádu (alespoň 1 rodič je 1. řádu), atd.,
    - prázdná klauzule nalezena na nejnižší možné hladině, možná kombinatorická exploze,
  - podpůrná množina
    - teorie je bezesporná – spor může vzniknout pouze z negace dokazovaného tvrzení,
    - pouze rezolventy mající alespoň za 1 předka (rodiče, prarodiče, . . .) dokazované tvrzení,
    - počet rezolvent každého řádu roste pomaleji.
  - lineární
    - poslední generovaná rezolventa je nejbližším rodičem,
    - opět omezuje počet rezolvent.

Rezoluce – strategie prohledávání, složitost

- Neúplné strategie
  - jednotková – alespoň jeden z rodičů je klauzule s jediným literálem,
  - vstupní – alespoň jeden z rodičů je klauzule z výchozí množiny,
  - filtrační – vstupní strategie s příbuzností – lze rezolvovat i klauzuli s jejím předkem,
  - kombinované – např. lineárně-vstupní, úplnost pouze pro Hornovy klauzule (viz. dále).
- Omezování množiny rezolvent
  - odstranění tautologií ($P \vee \neg P$),
  - odstranění specializací (důsledků) existující klauzule (P, $P \vee \neg Q$),
  - testování pravdivosti literálů (zejména v predikátové logice: větší(3,4)).
- Složitost rezoluce
  - ačkoli patří do třídy exponenciálních algoritmů, je v řadě případů "mnohem efektivnější" než model checking,
  - efektivně řešit jdou zejména úlohy s malým a velkým poměrem mezi klauzulemi a symboly,
  - alternativou pro výrokovou logiku je Davis-Putnamův algoritmus pro testování splnitelnosti logické formule v CNF.

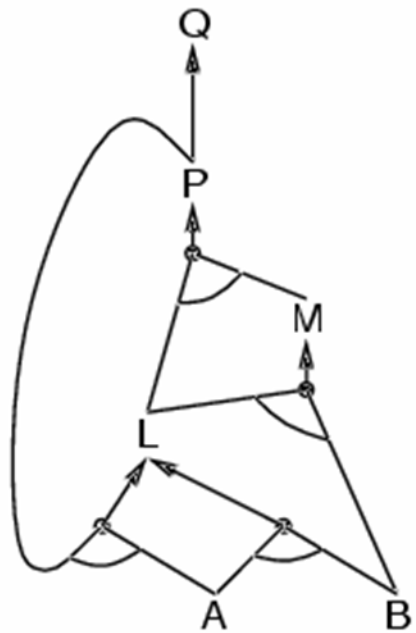
Přímé a zpětné řetězení

- Omezení na Hornovy klauzule s nejvýše jedním pozitivním literálem
  - $(P \vee \neg Q \vee \neg R) \Leftrightarrow (Q \wedge R) \Rightarrow P$, nelze zapsat např. $(\neg P \vee Q \vee R) \Leftrightarrow P \Rightarrow (Q \vee R)$,
  - tj. pouze jednoduchá fakta a implikace s jediným důsledkem.
- Odměnou je odvozování v $O(n)$, kde n je velikost báze znalostí (počet klauzulí)
  - odvozovat lze přímým či zpětným řetězením, oba postupy jsou korektní a úplné,
  - hlavně složitost zpětného řetězení je často sublineární.
- Přímé řetězení
  - udržujeme *agendu* platných (a dosud nepoužitých) symbolů,
  - u každé implikace *počítadlo* nesplněných předpokladů,
  - vybereme symbol z agendy, snížíme počítadla u příslušných implikací,
  - pokud 0, rozšíříme agendu o důsledek implikace,
  - končí vyčerpáním agendy nebo odvozením dokazovaného tvrzení,
  - řízeno daty – postupuje od platných faktů ke splnění cíle nebo vyčerpání možných operací.
- Zpětné řetězení
  - najde předpoklady platnosti dokazovaného tvrzení a ty se postupně snaží ověřit,
  - řízeno cíli – postupuje od dokazovaného tvrzení k jeho předpokladům.

Přímé a zpětné řetězení – příklad

:: dotaz: Q?

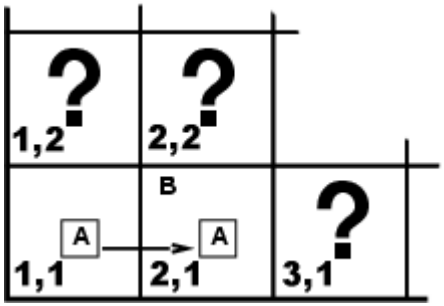
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Báze znalostí a AND-OR graf

Výroková logika ve wumpusově světě

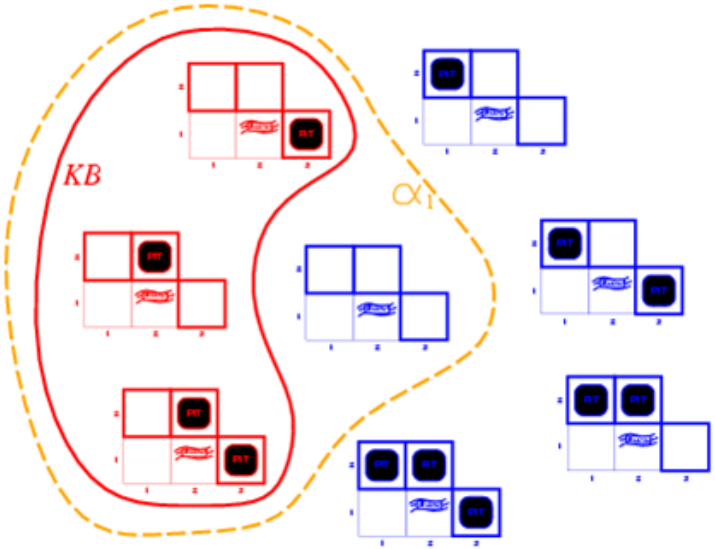
- Zjednodušený příklad
  - pouze šachty, omezený počet pozic,
  - KB vychází z pozorování světa v polích [1,1] a [2,1]
    - $\neg B_{1,1}$ – bez vánku v poli [1,1], $B_{2,1}$ – vánek v [2,1], $\neg P_{1,1}$ – v [1,1] není šachta,
  - KB obsahuje lokální odvozovací pravidla
    - $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$, $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$,
  - cílem je zjistit výskyt šachet v polích [1,2], [2,2] a [3,1].

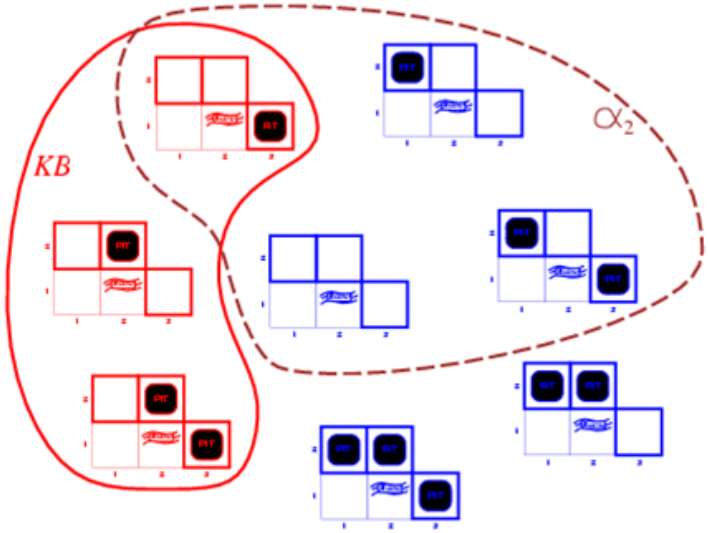


Fragment světa - 3 pole - 8 modelů

- Praktická demonstrace usuzování
  - model checking – věta musí mít všechny modely, které jsou modelem KB,
  - rezoluce – důkaz sporem.

Model checking ve wumpusově světě

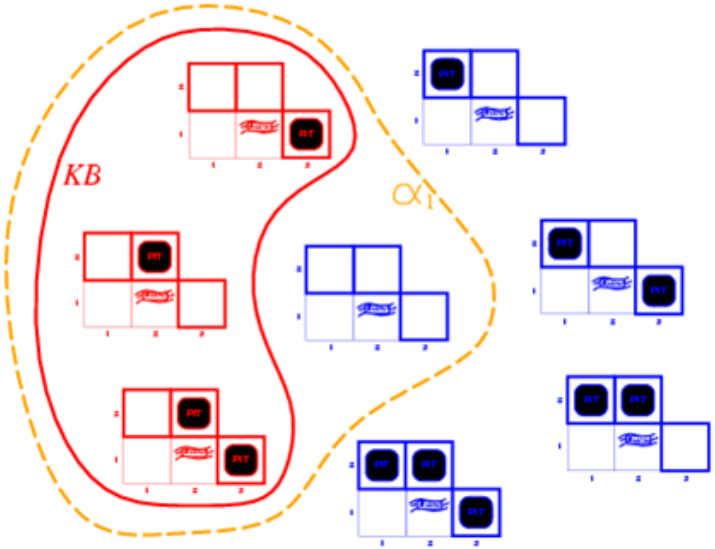


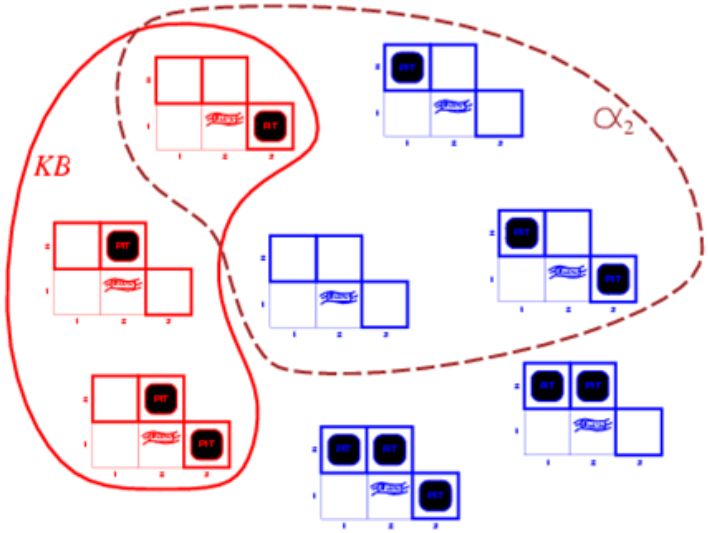


$\alpha_1 \equiv \neg P_{1,2}$, M(KB) $\subseteq$ M(α_1) $\Rightarrow$ KB $\models \alpha_1$

$\alpha_2 \equiv \neg P_{2,2}$, M(KB) $\not\subseteq$ M(α_2) $\Rightarrow$KB $\not\models \alpha_2$

Model checking ve wumpusově světě





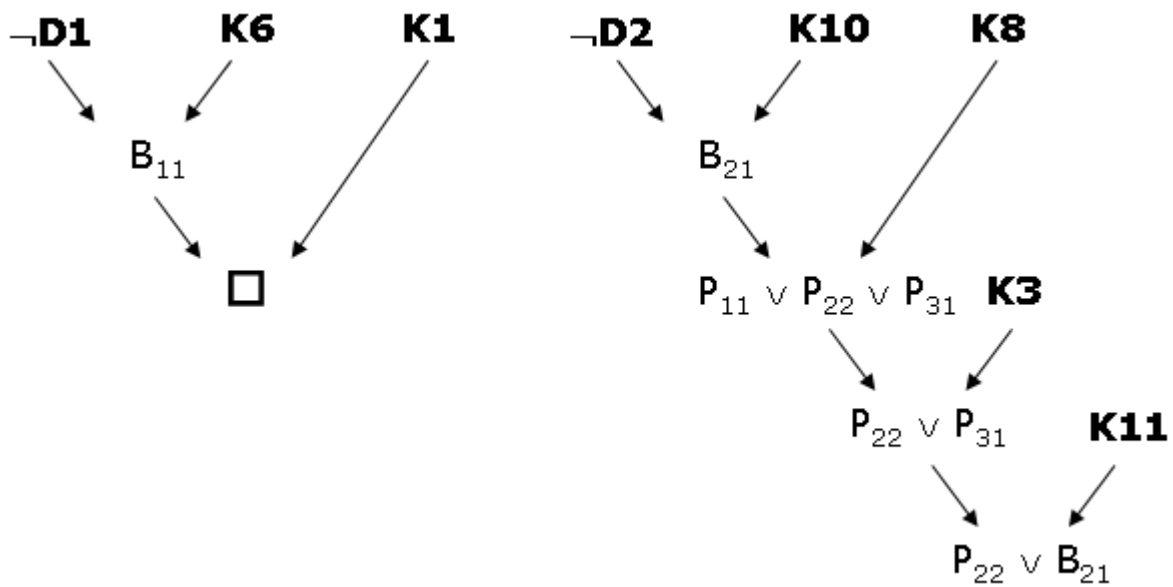
$\alpha_1 \equiv \neg P_{1,2}$, $\mathsf{M(KB)} \subseteq \mathsf{M}(\alpha_1) \Rightarrow \mathsf{KB} \models \alpha_1$

$\alpha_2 \equiv \neg P_{2,2}$, $\mathsf{M(KB)} \nsubseteq \mathsf{M}(\alpha_2) \Rightarrow \mathsf{KB} \not\models \alpha_2$

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	KB	α_1	α_2
F	F	F	F	F	F	F	F	T	T
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
F	T	F	F	F	F	T	**T**	**T**	**T**
F	T	F	F	F	T	F	**T**	**T**	**F**
F	T	F	F	F	T	T	**T**	**T**	**F**
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
T	T	T	T	T	T	T	F	F	F

Rezoluce ve wumpusově světě

- Převod do klauzální formy
  - (F1-5) $\neg B_{1,1}$, $B_{1,2}$, $\neg P_{1,1}$, $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$, $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$,
  - (K1-3) $\neg B_{1,1}$, $B_{1,2}$, $\neg P_{1,1}$,
  - (K5-7) $\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$, $\neg P_{1,2} \vee B_{1,1}$, $\neg P_{2,1} \vee B_{1,1}$ (úpravou (F4)),
  - (K8-11) $\neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}$, $\neg P_{1,1} \vee B_{2,1}$, $\neg P_{2,2} \vee B_{2,1}$, $\neg P_{3,1} \vee B_{2,1}$ (úpravou (F5)).
- Stromy rezolučního zamítnutí pro $D_1 \equiv \neg P_{1,2}$ a $D_2 \equiv \neg P_{2,2}$



Intencionální agent ve wumpusově světě

function **PL-wumpus-agent**(senzory=[zápach,vánek,třpyt]) returns akce

if zápach **then** Vlož(KB,$S_{x,y}$) **else** Vlož(KB,$\neg S_{x,y}$) % aktualizace KB

if vánek **then** Vlož(KB,$B_{x,y}$) **else** Vlož(KB,$\neg B_{x,y}$) % x,y určují aktuální polohu

if třpyt **then** akce $\leftarrow$ uchop

else if plán $<>$ $\{\}$ **then** akce $\leftarrow$ Vyber(plán) % plán je zásobníkem neuskutečněných akcí

else

repeat % hledej bezpečné sousední pole

Sousední-pole(x,y,i,j) % nejprve nenavštívená, posléze navštívená pole

until Zjisti($\neg P_{i,j} \wedge \neg W_{i,j}$) = True

plán $\leftarrow$ Vytvoř-plán(x,y,orientace,i,j) % posloupnost akcí k dosažení pole i,j

akce $\leftarrow$ Vyber(plán) % proveď první akci z plánu

Uprav(orientace, pozice (souřadnice x,y),navštívená pole, akce) % procedurálně, mimo KB

return akce

Shrnutí

- Základní východiska této přednášky
  - znalost lze reprezentovat ve formě různých **symbolů**
    - symboly – obecné a potenciálně složité datové struktury,
    - objekty, koncepty, fakta, pravidla, strategie,
  - inteligentního chování lze dosáhnout **manipulací** se symboly,
  - jde o předpoklady symbolické (tradiční) UI
    - alternativou je konekcionismus – neuronové sítě.
- Inteligentní systém pak vyžaduje
  - **formální jazyk** pro reprezentaci znalostí,
  - **schopnost odvozovat** nové znalosti/závěry.
- Neexistuje univerzální jazyk ani inferenční postup použitelný pro všechny problémy.
- Logika je velmi obecnou formální reprezentací schopnou inference
  - rezoluce je postupem, který umožňuje korektní a úplné logické usuzování.

Shrnutí

- Výroková logika je jednoduchým jazykem
  - – založeným na výrokových symbolech a logických spojkách,
  - – umožňuje přehledné a rozumně efektivní řešení řady problémů,
  - – pro jiné problémy není použitelná z důvodu
    - ∗ efektivity zápisu – viz. prostorové vazby ve wumpusově světě NxN,
    - ∗ konečnosti množiny výrokových symbolů – nemožnost zvládnout neohraničené problémy,
    - ∗ nedostatečné expresivity – viz. smrtelný Sokrates.
- Příště: predikátová logika a další formy reprezentace.

Doporučené doplňky – zdroje přednášky

:: Četba

- Russel, Norvig: **AI: A Modern Approach**, Logical Agents, chapter 7
  - reprezentace z pohledu inteligentního agenta,
  - dostupná v pdf – `http://aima.cs.berkeley.edu/newchap07.pdf`.
- Mařík a kol. **Umělá inteligence 1**
  - kapitola Reprezentace znalostí: základní formáty, logika, sémantické sítě, rámce,
  - kapitola Řešení úloh a dokazování vět: predikátová logika a důkazní prostředky.
- Mařík a kol. **Umělá inteligence 2**
  - kapitola Znalostní inženýrství: praktická, znalostní systémy v konkrétních aplikacích.
- Brachman, Levesque: **Knowledge Representation and Reasoning**
  - kniha, The Morgan Kaufmann Series in Artificial Intelligence.