Logický agent, výroková logika

Aleš Horák

E-mail: hales@fi.muni.cz
http://nlp.fi.muni.cz/uui/

Obsah:

- Logický agent
- Logika
- Výroková logika

Logický agent

znalosti prohledávání stavového prostoru – jen zadané funkce (přechodová funkce, cílový test,. . .)

potřeba obecné formy umožňující kombinace znalostí

$\rightarrow$ znalosti logického agenta

logický agent = agent využívající (formálně zadané) znalosti
(*knowledge-based agent*)

Příklad využití znalostí

Pokud $X + Y = 10$ a $X - Y = 4$, kolik je X?

jak řešit?

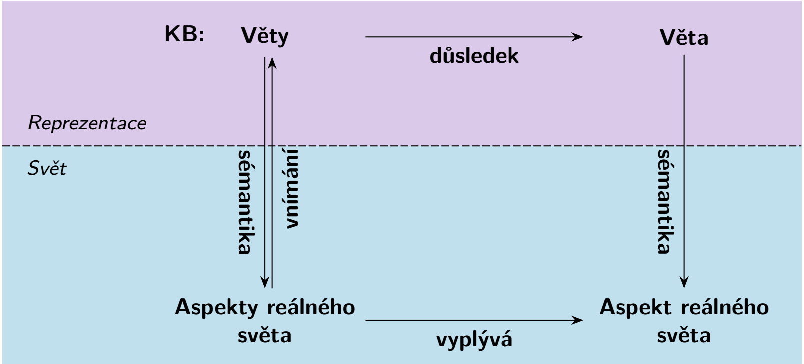
- prohlédavat prostor všech možných hodnot např. do šířky
  cíl = platnost obou omezujících podmínek
- problém s omezujícími podmínkami, proměnné X a Y
- **sečíst obě rovnice a vydělit 2**
  pravidla:
  - **if** $((A = B) \wedge (C = D))$ **then tell**$((A + C) = (B + D))$
  - **if** $(n \cdot X = B)$ **then tell**$(X = B/n)$

  řešení:
  $$\begin{aligned} X + Y + X - Y &= 10 + 4 \\ 2 \cdot X &= 14 \\ X &= 7 \end{aligned}$$

Použili jsme operace zachovávající pravdivost
tj. odvozování založené na logice, **logickou inferenci**

Báze znalostí

báze znalostí (*Knowledge Base*, **KB**) =
množina vět (formulí, tvrzení, *sentences*)
vyjádřených v jazyce reprezentace znalostí



Báze znalostí

2 koncepty:
- – reprezentace znalostí (*knowledge representation*)
- – vyvozování znalostí (*knowledge reasoning*) $\rightarrow$ inference

obecné znalosti – důležité v částečně pozorovatelných prostředích (*partially observable* environments)

flexibilita logického agenta:
- schopnost řešit i nové úkoly
- možnost učení nových znalostí
- úprava stávajících znalostí podle stavu prostředí

logický agent může odpovědět v principu na libovolnou otázku (podle KB) na rozdíl od prohledávacích algoritmů, kde otázka je přímo zakódovaná v zadání

Návrh logického agenta

agent musí umět:
- reprezentovat stavy, akce, ...
- zpracovat nové vstupy z prostředí
- aktualizovat svůj vnitřní popis světa
- odvodit skryté informace o stavu světa
- odvodit vlastní odpovídající akce

přístupy k tvorbě agenta – deklarativní × procedurální (kombinace)

co je potřeba pro **návrh agenta**:

- znalostní hledisko – tvorba agenta → zadání znalostí pozadí, znalostí domény a cílového požadavku
  např. automatické taxi
  - znalost mapy, dopravních pravidel, ...
  - požadavek – dopravit zákazníka na FI MU Brno
- implementační hledisko – jaké datové struktury KB obsahuje + algoritmy, které s nimi manipulují

Komponenty agenta

komponenty logického agenta:

inferenční stroj (inference engine)
báze znalostí (knowledge base)

algoritmy nezávislé na doméně

znalosti o doméně

obsah **báze znalostí**:

- na začátku – tzv. znalosti pozadí (*background knowledge*)
  axiom – tvrzení/pravidlo, které je dáno (ne vyvozeno)
- průběžně doplňované znalosti → metoda **tell(KB, Sentence)**

akce logického agenta:

function KB-AgentAction(*KB*, *ATime*, *Percept*, *Action*) *# vrací akci a nový čas*
  PSentence ← **make_percept_sentence**(*Percept*, *ATime*)
  tell (*KB*, *PSentence*) *# přidáme výsledky pozorování do KB*
  Query ← **make_action_query**(*ATime*),
  Action ← **ask**(*KB*, *Query*) *# zeptáme se na další postup*
  ASentence ← **make_action_sentence**(*Action*, *ATime*)
  tell (*KB*, *ASentence*) *# přidáme informace o akci do KB*
  return *Action*, *ATime* + 1

Popis světa – PEAS

zadání světa rozumného agenta:

- míra výkonnosti (***P****erformance measure*)

  plus body za dosažené (mezi)cíle, pokuty za nežádoucí následky
- prostředí (***E****nvironment*)

  objekty ve světě, se kterými agent musí počítat, a jejich vlastnosti
- akční prvky (***A****ctuators*)

  možné součásti činnosti agenta, jeho akce se skládají z použití těchto prvků
- senzory (***S****ensors*)

  zpětné vazby akcí agenta, podle jejich výstupů se tvoří další akce

např. zmiňované automatické taxi:

míra výkonnosti	doprava na místo, vzdálenost, bezpečnost, bez přestupků, komfort, ...
prostředí	ulice, křižovatky, účastníci provozu, chodci, počasí, ...
akční prvky	řízení, plyn, brzda, houkačka, blinkry, komunikátory, ...
senzory	kamera, tachometr, počítač kilometrů, senzory motoru, GPS, ...

Wumpusova jeskyně

- původně textová adventure hra
  Hunt the Wumpus
  Gregory Yob, 1973
- poměrně oblíbená, šířená i komerčně
- jeden z prvních příkladů her typu survival horror
- 1975 zveřejněný zdrojový kód
- hra vyžaduje logické uvažování

You are in room 3.
Tunnels lead to 2, 4, 12.
Shoot or Move (S-M)? M
Where to? 12

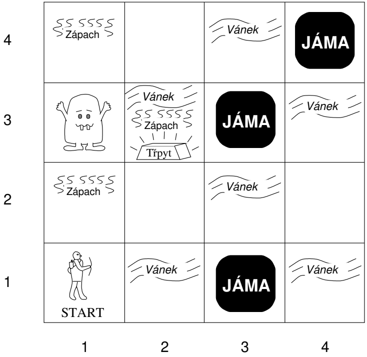
You are in room 12.
I smell a Wumpus.
Tunnels lead to 3, 11, 13.
Shoot or Move (S-M)? S
Room? 13

AHA! You got the mumpus!
HEE HEE HEE The Wumpus'll
get you next time!!
Same setup (Y-N)? Y

Wumpusova jeskyně

PEAS zadání **Wumpusovy jeskyně**:

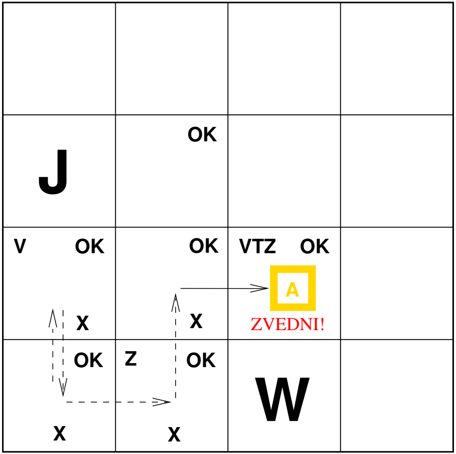
- **P** – míra výkonnosti
  zlato +1000, smrt -1000, -1 za krok, -10 za užití šípu
- **E** – prostředí
  Místnosti vedle Wumpuse zapáchají.
  V místnosti vedle jámy je vánek.
  V místnosti je zlato ⇔ je v ní třpyt.
  Výstřel zabije Wumpuse, pokud jsi obrácený k němu.
  Výstřel vyčerpá jediný šíp, který máš.
  Zvednutím vezmeš zlato ve stejné místnosti.
  Položení odloží zlato v aktuální místnosti.
- **A** – akční prvky
  Otočení vlevo, Otočení vpravo, Krok dopředu, Zvednutí, Položení, Výstřel
- **S** – senzory
  Vánek, Třpyt, Zápach, Náraz do zdi, Chroptění Wumpuse



Vlastnosti problému Wumpusovy jeskyně

pozorovatelné	ne, jen lokální vnímání
deterministické	ano, přesně dané výsledky
episodické	ne, sekvenční na úrovni akcí (složitější)
statické	ano, Wumpus a jámy se nehýbou
diskrétní	ano
více agentů	ne, Wumpus je spíše vlastnost prostředí

Průzkum Wumpusovy jeskyně

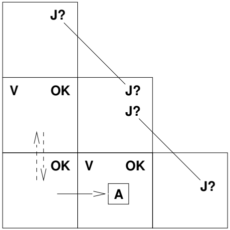


8

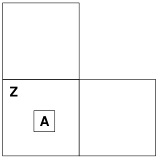
$\boxed{\text{A}}$	=	Agent
V	=	Vánek
T	=	Třpyt
OK	=	bezpečí
J	=	Jáma
Z	=	Zápach
X	=	navštíveno
W	=	Wumpus

Průzkum Wumpusovy jeskyně – problémy

Obtížné situace:



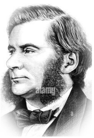
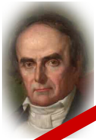
Vánek v $(1,2)$ i v $(2,1)$ ⇒ žádná bezpečná akce
Při předpokladu uniformní distribuce děr
→ díra v $(2,2)$ má pravděpodobnost 0.86, na krajích 0.31



Zápach v $(1,1)$ ⇒ nemůže se pohnout
je možné použít donucovací strategii (*strategy of coercion*):

1. Výstřel jedním ze směrů
2. byl tam Wumpus ⇒ je mrtvý (poznám podle Chroptění) ⇒ bezpečné
3. nebyl tam Wumpus (žádné Chroptění) ⇒ bezpečný směr

Co je Pravda?

- Francis Bacon (1561–1626)
  No pleasure is comparable to the standing upon the vantage-ground of **truth**.
- Thomas H. Huxley (1825–1895)
  Irrationally held **truths** may be more harmful than reasoned errors.
- John Keats (1795–1821)
  Beauty is **truth**, **truth** beauty; that is all ye know on earth, and all ye need to know.
- Blaise Pascal (1623–1662)
  We know the **truth**, not only by the reason, but also by the heart.
- François Rabelais (1490–1553)
  Speak the **truth** and shame the Devil.
- Daniel Webster (1782–1852)
  There is nothing so powerful as **truth**, and often nothing so strange.

Logika

Logika = syntaxe a sémantika formálního jazyka pro reprezentaci znalostí umožňující vyvozování závěrů
Syntaxe definuje všechny *dobře utvořené věty* jazyka
Sémantika definuje "*význam*" vět ⇒ definuje pravdivost vět v jazyce (v závislosti na *možném světě*)

např. jazyk aritmetiky:

- $x + 2 \geq y$ je dobře utvořená věta; $x2 + y >$ není věta
- $x + 2 \geq y$ je pravda ⇔ číslo $x + 2$ není menší než číslo y
- $x + 2 \geq y$ je pravda ve světě, kde $x = 7,\ y = 1$
- $x + 2 \geq y$ je nepravda ve světě, kde $x = 0,\ y = 6$

zápis na papíře v libovolné syntaxi → v KB se jedná o konfiguraci (částí) agenta
vlastní vyvozování → generování a manipulace s těmito konfiguracemi

Historie logiky

náhledy na logiku:

- filozofická logika
  - Thalés z Milétu – geometrické věty a důkazy
  - Aristoteles – první formální systém, princip sporu, princip vyloučení třetího
  - Euklides – axiomy, věty, první axiomatický systém
  - stoikové 3.stol. př.n.l. – základy výrokové logiky
- počátky symbolické logiky (13.–19. století)
  - J. Duns Scotus – z dvou odporujících si tvrzení plyne cokoliv
  - W. Ockham – odlišil tvrzení a odvozovací pravidlo
  - G. W. Leibniz – idea logického kalkulu pro exaktní vědy
  - B. Bolzano – operace odvoditelnosti, kvantifikátory
  - G. Boole – Booleova algebra, fomální logika v moderním slova smyslu

20. století

- matematická logika
  - G. Frege, přelom století – axiomatizace výrokové logiky
  - B. Russell, 1918 – objasnění paradoxu lháře
  - C.S.Lewis, J.Lukasiewicz – neklasické logiky
  - D. Hilbert, W. Ackermann – axiomatizace predikátového počtu
  - úplnost výrokové (Post 1921) a predikátové (Goedel 1930) logiky
  - K. Goedel – neúplnost systémů obsahujících aritmetiku, omezená možnost důkazu bezespornosti
  - A.Church, 1936 – nerozhodnutelnost predikátové logiky
  - A. Turing, 1937 – pojem vyčíslitelnosti, Turingův stroj
- logika v informatice, v AI, výpočtová logika
  - verifikace programů
  - deskriptivní logika
  - znalostní systémy
  - logické programování
  - bayesovské sítě
  - ...

Některé vlastnosti logik

- formální – co je poznané, definované; metody odvozování
- neformální – mentální, co je poznatelné; zdravý selský rozum, komunikace mezi lidmi, heuristické odvozování

Formální logika

- dvouhodnotová – true, false; i vícehodnotová
- extenzionální – pravdivost formule závisí jen na pravdivosti jejich složek
- intenzionální – nejen na pravdivosti složek, také na "okolnostech" (čas, možný svět, …)

Některé vlastnosti logik

Dvouhodnotová extenzionální logika zde

- **výroková**
  Jestliže bude pěkně a nebudu učit, půjdu hrát tenis
  $p \wedge \neg q \Rightarrow r$
- **predikátová**
  - 1. řádu
    Není pravda, že všichni lidé jsou spokojení
    $\neg \forall x : \text{člověk}(x) \Rightarrow \text{spokojený}(x)$
  - 2. řádu
    Existuje vlastnost, kterou mají všichni lidé
    $\exists P \forall x : \text{člověk}(x) \Rightarrow P(x)$

Důsledek

Důsledek (vyplývání, *entailment*) – jedna věc logicky vyplývá z druhé (je jejím důsledkem):

$$KB \models \alpha$$

Z báze znalostí *KB* **vyplývá** věta α $\Leftrightarrow$ α je pravdivá ve všech světech, kde je *KB* pravdivá

např.:

- *KB* obsahuje věty – "Češi vyhráli"
  – "Slováci vyhráli"

  z *KB* pak vyplývá – "Češi vyhráli nebo Slováci vyhráli"
- z $x + y = 4$ vyplývá $4 = x + y$

Důsledek je vztah mezi větami (*syntaxe*), který je založený na *sémantice*.

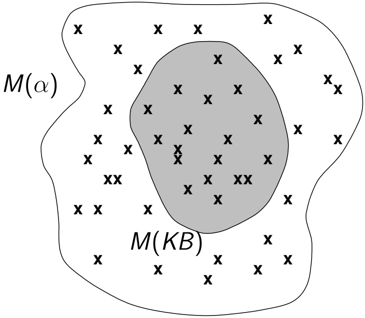
Model

možný svět = **model** … formálně strukturovaný (abstraktní) svět, umožňuje vyhodnocení pravdivosti

říkáme: m je model věty α $\Leftrightarrow$ α je pravdivá v m

$M(\alpha)$ … množina všech modelů věty α

$$\mathbf{KB} \models \alpha \quad \Leftrightarrow \quad M(KB) \subseteq M(\alpha)$$



např.:

$KB =$ "Češi vyhráli" $\wedge$ "Slováci vyhráli"

$\alpha =$ "Češi vyhráli"

Inference

Vyvozování požadovaných důsledků – **inference**

$KB \vdash_i \alpha$ ... věta α může být vyvozena z KB pomocí (procedury) i
(i odvodí α z KB)

všechny možné důsledky KB jsou "kupka sena"; α je jehla
vyplývání = jehla v kupce sena; inference = její nalezení

Bezespornost:	i je bezesporná	$\Leftrightarrow$	$\forall KB \vdash_i \alpha$	$\Rightarrow$	$KB \models \alpha$
Úplnost:	i je úplná	$\Leftrightarrow$	$\forall KB \models \alpha$	$\Rightarrow$	$KB \vdash_i \alpha$

Vztah k **reálnému světu**:

Pokud je KB pravdivá v reálném světě $\Rightarrow \forall$ *věta* α *vyvozená z KB pomocí bezesporné inference je také pravdivá ve skutečném světě*

Jestliže máme sémantiku "pravdivou" v reálném světě $\rightarrow$ můžeme vyvozovat závěry o skutečném světě pomocí logiky

Výroková logika

Výroková logika – nejjednodušší logika, ilustruje základní myšlenky

Syntaxe

- výrokové symboly $P_1, P_2, \ldots$ jsou věty
- negace – S je věta $\Rightarrow$ $\neg S$ je věta
- konjunkce – S_1 a S_2 jsou věty $\Rightarrow$ $S_1 \wedge S_2$ je věta
- disjunkce – S_1 a S_2 jsou věty $\Rightarrow$ $S_1 \vee S_2$ je věta
- implikace – S_1 a S_2 jsou věty $\Rightarrow$ $S_1 \Rightarrow S_2$ je věta
- ekvivalence – S_1 a S_2 jsou věty $\Rightarrow$ $S_1 \Leftrightarrow S_2$ je věta

Úplný systém logických spojek

Existuje minimální dostatečná množina spojek?

- prostřednictvím systému spojek $\{\neg, \wedge, \vee\}$ dokážeme vyjádřit libovolnou spojku
- množina spojek s touto vlastností – **úplný systém logických spojek**
- další úplné systémy např. $\{\neg, \vee\}$, $\{\neg, \wedge\}$, $\{\neg, \Rightarrow\}$
  $(a \Rightarrow b) \equiv (\neg a \vee b)$, $(a \wedge b) \equiv \neg(\neg a \vee \neg b)$
  $(a \vee b) \equiv \neg(\neg a \wedge \neg b)$
- jednoprvkové úplné systémy: Shefferova funkce NAND (negace konjunkce), Nicodova funkce NOR (negace disjunkce)
  např. $(p \vee q) \equiv ((p \mid p) \mid (q \mid q))$

Sémantika výrokové logiky

- každý model musí určit pravdivostní hodnoty výrokových symbolů
  např.: $m_1 = \{P_1 = \mathit{false}, P_2 = \mathit{false}, P_3 = \mathit{true}\}$
- pravidla pro vyhodnocení pravdivosti složených výroků pro model m:

$$\begin{array}{rlclclcl}
\neg S & \text{je } \mathit{true} & \Leftrightarrow & S & \text{je } \mathit{false} & & & \\
S_1 \wedge S_2 & \text{je } \mathit{true} & \Leftrightarrow & S_1 & \text{je } \mathit{true} & \text{a} & S_2 & \text{je } \mathit{true} \\
S_1 \vee S_2 & \text{je } \mathit{true} & \Leftrightarrow & S_1 & \text{je } \mathit{true} & \text{nebo} & S_2 & \text{je } \mathit{true} \\
S_1 \Rightarrow S_2 & \text{je } \mathit{true} & \Leftrightarrow & S_1 & \text{je } \mathit{false} & \text{nebo} & S_2 & \text{je } \mathit{true} \\
\text{tj.} & \text{je } \mathit{false} & \Leftrightarrow & S_1 & \text{je } \mathit{true} & \text{a} & S_2 & \text{je } \mathit{false} \\
S_1 \Leftrightarrow S_2 & \text{je } \mathit{true} & \Leftrightarrow & S_1 \Rightarrow S_2 & \text{je } \mathit{true} & \text{a} & S_2 \Rightarrow S_1 & \text{je } \mathit{true}
\end{array}$$

- rekurzivním procesem vyhodnotíme lib. větu:

$$\neg P_1 \wedge (P_2 \vee P_3) = \mathit{true} \wedge (\mathit{false} \vee \mathit{true}) = \mathit{true} \wedge \mathit{true} = \mathit{true}$$

pravdivostní tabulka:

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

Sémantika výrokové logiky a přirozený jazyk

Rozdíly v chápání logických spojek v jazyce:

- rozdíl OR a XOR – v jazyce spíš XOR

  Viděl jsem na večírku Petra nebo Pavla.
  Korunujeme Jana nebo Richarda?

- implikace – v jazyce chybí část "z neplatné premisy plyne cokoliv," naopak se přidává presupozice, že mezi premisou a závěrem existuje příčinný vztah

  $$P \Rightarrow Q \qquad \Leftrightarrow \qquad \neg P \vee Q$$

  5 je sudá a z toho plyne, že AZ Tower je nejvyšší budova na světě.

  5 je lichá a z toho plyne, že Praha je hlavní město ČR.

Logická ekvivalence

Dva výroky jsou **logicky ekvivalentní** právě tehdy, když jsou pravdivé ve stejných modelech:

$$\alpha \equiv \beta \quad \Leftrightarrow \quad \alpha \models \beta \text{ a } \beta \models \alpha$$

$$\begin{array}{rcll}
(\alpha \land \beta) & \equiv & (\beta \land \alpha) & \text{komutativita } \land \\
(\alpha \lor \beta) & \equiv & (\beta \lor \alpha) & \text{komutativita } \lor \\
((\alpha \land \beta) \land \gamma) & \equiv & (\alpha \land (\beta \land \gamma)) & \text{asociativita } \land \\
((\alpha \lor \beta) \lor \gamma) & \equiv & (\alpha \lor (\beta \lor \gamma)) & \text{asociativita } \lor \\
\neg(\neg\alpha) & \equiv & \alpha & \text{eliminace dvojí negace} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg\beta \Rightarrow \neg\alpha) & \text{kontrapozice} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg\alpha \lor \beta) & \text{eliminace implikace} \\
(\alpha \Leftrightarrow \beta) & \equiv & ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) & \text{eliminace ekvivalence} \\
\neg(\alpha \land \beta) & \equiv & (\neg\alpha \lor \neg\beta) & \text{de Morgan} \\
\neg(\alpha \lor \beta) & \equiv & (\neg\alpha \land \neg\beta) & \text{de Morgan} \\
(\alpha \land (\beta \lor \gamma)) & \equiv & ((\alpha \land \beta) \lor (\alpha \land \gamma)) & \text{distributivita } \land \text{ nad } \lor \\
(\alpha \lor (\beta \land \gamma)) & \equiv & ((\alpha \lor \beta) \land (\alpha \lor \gamma)) & \text{distributivita } \lor \text{ nad } \land
\end{array}$$

Logická ekvivalence – příklad

$$
\begin{aligned}
&(A \wedge (A \Rightarrow B)) \Rightarrow B \\
&\quad \equiv (A \wedge (\neg A \vee B)) \Rightarrow B \\
&\quad \equiv \neg(A \wedge (\neg A \vee B)) \vee B \\
&\quad \equiv (\neg A \vee \neg(\neg A \vee B)) \vee B \\
&\quad \equiv (\neg A \vee (\neg\neg A \wedge \neg B)) \vee B \\
&\quad \equiv (\neg A \vee (A \wedge \neg B)) \vee B \\
&\quad \equiv ((\neg A \vee A) \wedge (\neg A \vee \neg B)) \vee B \\
&\quad \equiv (\mathit{True} \wedge (\neg A \vee \neg B)) \vee B \\
&\quad \equiv (\neg A \vee \neg B) \vee B \\
&\quad \equiv \neg A \vee (\neg B \vee B) \\
&\quad \equiv \neg A \vee \mathit{True} \\
&\quad \equiv \mathit{True}
\end{aligned}
$$

Platnost a splnitelnost

- Výrok je **platný** (*valid*) $\Leftrightarrow$ je pravdivý ve všech modelech
  také *tautologie*, např.: $true$, $A \vee \neg A$, $A \Rightarrow A$,
  $(A \wedge (A \Rightarrow B)) \Rightarrow B$

  Platnost je spojena s vyplýváním pomocí věty o dedukci:
  $KB \models \alpha \quad \Leftrightarrow \quad (KB \Rightarrow \alpha)$ *je platný výrok*

- Výrok je **splnitelný** (*satisfiable*) $\Leftrightarrow$ je pravdivý v některých modelech
  např.: $A \vee B$, C

  Výrok je **nesplnitelný** $\Leftrightarrow$ je nepravdivý ve všech modelech
  také *kontradikce*, např.: $A \wedge \neg A$

Platnost a splnitelnost – problém SAT

problém SAT:

Je daná formule Φ s n výrokovými symboly splnitelná?

problém SAT je NP-uplný

např. hledání řešení problému s omezujícími podmínkami = problém SAT na daných podmínkách

SAT solver – algoritmus zaměřený na efektivní řešení problémů s velkým množstvím symbolů

Splnitelnost je spojena s vyplýváním pomocí důkazu α sporem (*reductio ad absurdum*):

$$KB \models \alpha \quad \Leftrightarrow \quad (KB \wedge \neg\alpha) \text{ je nesplnitelný}$$

Normální forma

Konjunktivní normální forma, *Conjunctive Normal Form*, *CNF*

klauzule – disjunkce symbolů nebo jejich negací (tzv. literálů)

KB v CNF – konjunkce klauzulí

$$
\begin{aligned}
&(\underline{\neg D} \lor \underline{\neg B} \lor \underline{C})\\
&\quad \land (B \lor \neg A \lor \neg C)\\
&\quad \land (\neg C \lor \neg B \lor E)\\
&\quad \land (E \lor \neg D \lor B)\\
&\quad \land (B \lor E \lor \neg C)
\end{aligned}
$$

literály $\neg D$, $\neg B$ a C

pět klauzulí spojených konjunkcí $\land$

každou formuli lze převést do normální formy

Konjunktivní normální forma

příklad $V_{1,1} \quad \Leftrightarrow \quad (J_{1,2} \vee J_{2,1})$:

1. eliminujeme $\Leftrightarrow$: nahradíme $\alpha \Leftrightarrow \beta$ přepisem $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$
   $(V_{1,1} \Rightarrow (J_{1,2} \vee J_{2,1})) \wedge ((J_{1,2} \vee J_{2,1}) \Rightarrow V_{1,1})$
2. eliminujeme $\Rightarrow$: nahradíme $\alpha \Rightarrow \beta$ přepisem $\neg\alpha \vee \beta$
   $(\neg V_{1,1} \vee J_{1,2} \vee J_{2,1}) \wedge (\neg(J_{1,2} \vee J_{2,1}) \vee V_{1,1})$
3. negaci přesuneme k symbolům aplikací pravidel logické ekvivalence (dvojí negace a de Morgan)
   $(\neg V_{1,1} \vee J_{1,2} \vee J_{2,1}) \wedge ((\neg J_{1,2} \wedge \neg J_{2,1}) \vee V_{1,1})$
4. "roznásobíme" závorky pomocí distributivity
   $(\neg V_{1,1} \vee J_{1,2} \vee J_{2,1}) \wedge (\neg J_{1,2} \vee V_{1,1}) \wedge (\neg J_{2,1} \vee V_{1,1})$

Výsledná věta je logicky ekvivalentní původní větě a je v **CNF** nachystaná pro důkaz s využitím rezoluce

Hornovy klauzule

syntakticky vymezená logika Hornových klauzulí – efektivnější vyvozování

Hornova klauzule:

- klauzule, kde nejvýše jeden symbol je pozitivní
  $\neg P_{1,1} \vee \neg \text{Vánek} \vee V_{1,1}$ (odpovídá $P_{1,1} \wedge \text{Vánek} \Rightarrow V_{1,1}$)
- bez pozitivního symbolu – dotaz, cíl
  $\neg P_{1,1} \vee \neg \text{Vánek}$ (odpovídá $P_{1,1} \wedge \text{Vánek} \Rightarrow ?$)
- pouze s jedním pozitivním symbolem – fakt
  Vánek (odpovídá $\mathit{True} \Rightarrow \text{Vánek}$)
- s jedním pozitivním a několika negovanými symboly – pravidlo

vlastnosti logiky Hornových klauzulí:

- snadná interpretace klauzulí, vede k logickému programování
- dokazování pomocí jednoduchých algoritmů dopředného a zpětného řetězení v lineárním čase (k velikosti KB)

Tvrzení pro Wumpusovu jeskyni

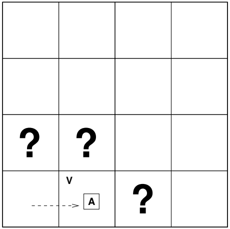
Definujeme výrokové symboly $J_{i,j}$ je pravda $\Leftrightarrow$ Na $[i,j]$ je Jáma.
a $V_{i,j}$ je pravda $\Leftrightarrow$ Na $[i,j]$ je Vánek.

báze znalostí KB:

– pravidlo pro $[1,1]$: R_1: $\neg J_{1,1}$
– pozorování: R_2: $\neg V_{1,1}$, R_3: $V_{2,1}$
– pravidla pro vztah Jámy a Vánku:

"Jámy způsobují Vánek ve vedlejších místnostech"

$$R_4'': V_{1,1} \Leftarrow (J_{1,2} \vee J_{2,1})$$
$$R_5'': V_{2,1} \Leftarrow (J_{1,1} \vee J_{2,2} \vee J_{3,1})$$



"V poli je Vánek právě tehdy, když je ve vedlejším poli Jáma."

$$R_4: V_{1,1} \Leftrightarrow (J_{1,2} \vee J_{2,1})$$
$$R_5: V_{2,1} \Leftrightarrow (J_{1,1} \vee J_{2,2} \vee J_{3,1})$$

– $KB = R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$