

Logika. Výroková logika

Luboš Popelínský

E-mail: popel@fi.muni.cz
<http://nlp.fi.muni.cz/uui/>

Obsah:

- Logika a umělá inteligence
- Logický agent
- Výroková logika
- Splnitelnost
- SAT problém

... místo úvodu

You can think about deep learning as equivalent to ... our visual cortex or auditory cortex. But, of course, true intelligence is a lot more than just that, you have to recombine it into higher-level thinking and symbolic reasoning, a lot of the things classical AI tried to deal with in the 80s. ... We would like to build up to this symbolic level of reasoning - maths, language, and logic. So that's a big part of our work.

Demis Hassabis, CEO and co-founder of Google Deepmind

... místo úvodu

You can think about **deep learning** as equivalent to ... our visual cortex or auditory cortex. But, of course, **true intelligence** is a lot more than just that, you **have to recombine it into higher-level thinking and symbolic reasoning**, a lot of the things classical AI tried to deal with in the 80s. ... We would like to build up to this symbolic level of reasoning - **maths, language, and logic**. So that's a big part of our work.

Demis Hassabis, CEO and co-founder of Google Deepmind

Příklad

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

jak řešit

- prohlédávat prostor všech možných řešení např. do hloubky, kontrolovat platnost obou omezujících podmínek
- CSP (Constraint Satisfaction problem) s proměnnými X a Y ,
- sečíst obě rovnice a vydělit 2.

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

O krok zpět

(zjednodušený) **kognitivní proces**:

data → **UČENÍ** → model

dotaz → **INFERENCE** pomocí modelu → odpověď

O krok zpět

(zjednodušený) kognitivní proces:

data → UČENÍ → model **INDUKCE**

dotaz → **INFERENCE** pomocí modelu → odpověď **DEDUKCE**

INDUKCE odvozuje generalizaci z dat, např. pravidla,

DEDUKCE odvozuje logicky platný závěr na základě těchto pravidel

Zde se soustředíme na **DEDUKCI**, tj. **na INFERENCEI metodami logiky**,

Příklad: Eukleidův axiomatický systém pro geometrii

Logika

dva cíle

- **reprezentovat** znalosti o doméně
- umět **odvozovat** na základě těchto znalostí
- **logika = nástroj pro odvozování důsledků z předpokladů**

Logický agent

- **Znalostní báze** (knowledge base, KB) = množina vět ve formálním jazyce
- Deklarativní přístup k tvorbě agenta
 - **Tell** it what it needs to know (or have it **Learn** the knowledge)
 - Then it can **Ask** itself what to do—answers should follow from the KB
 - pomocí **inferenčního mechanismu**
- **Znalostní systém** může odpovědět v principu na libovolnou otázku (podle KB)
- na rozdíl např. od prohledávacích algoritmů, kde typická otázka je "jak se dostat z A do B"

Logický agent

```
function KB-AGENT(percept) # vrací akci
# globální: KB – báze znalostí; t – číslo, na začátku 0
  tell (KB, make_percept_sentence(percept, t))
  action ← ask(KB, make_action_query(t))
  tell (KB, make_action_sentence(action, t))
  t ← t+1
return action
```

Příklad

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

KB-AGENT

Pravidla do znalostní báze

- if $((A=B) \text{ and } (C=D))$ then add $((A+C)=(B+D))$
- if $(n*X=B)$ then add $(X=B/n)$

Inferenční mechanismus - např. forward chaining (fix point semantics)

Logický agent

Otázky

- jaký jazyk použít
- co musí obsahovat KB
- jaký má být inferenční mechanismus
- existuje pro každou z těchto otázek minimální varianta?
- k čemu je užitečný

nejprve pro výrokovou logiku

Historia Magistra Vitae ... ?

<https://www.britannica.com/topic/history-of-logic>

- filozofická logika
 - Thalés z Milétu - geometrické věty a důkazy
 - Aristoteles - první formální systém, princip sporu, princip vyloučení třetího
 - Euklides - axiomy, větz, 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 - Boolova 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 bezspornosti
 - A.Church, 1936 - nerozhodnutelnost predikátové logiky
 - A. Turing, 1937 - pojem vyčíslitelnosti, Turingův stroj
- logika v informatice, v AI, výpočtová logika
 - verifikace programů
 - binary decision diagrams
 - znalostní systémy
 - logické programování
 - ...

Historia Magistra Vitae ... !

- filozofická logika
 - ...
 - **Euklides - axiomy, věty, první axiomatický systém**
 - ...
- počátky symbolické logiky (13.- 19. století)
 - ...
 - **G. Boole** - Boolova algebra, formální logika v moderním slova smyslu, ale též *he wrote a new book on The Laws of Thought with two parts: Logic + Probability*
 - ...
- matematická logika (konec 19. až polovina 20. stol.)
 - ...
 - **A. Turing**, 1937 - pojem vyčíslitelnosti, Turingův stroj, **kolem 1950 - induktivní inference**
 - ...

Informační zdroje

Logical Agents. In Russell and Norvig. AI. A Modern Approach

Anil Nerode, Richard A. Shore, Logic for Applications. Springer Verlag.
základní kniha, mnohokrát v knihovně FI

Logic and Artificial Intelligence. In: Stanford Encyclopedia of Philosophy
<https://plato.stanford.edu/entries/logic-ai/>

Klasifikace 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
- extensionální - pravdivost formule závisí jen na pravdivosti jejich složek
- intesionální - nejen na psti složek - "možná", "věřím"

Dvouhodnotová extensionální 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 spokojeni*
 $\neg \forall x : (\text{člověk}(x) \Rightarrow \text{spokojený}(x))$
 - 2. řádu *Existuje vlastnost, kterou mají všichni lidé*
 $\exists P \forall x : (\text{člověk}(x) \Rightarrow P(x))$

Výroková logika. Syntax

- **abeceda**

1. výrokové symboly: p, q, r, s, \dots , případně $p_1, p_2, p_3 \dots$
2. symboly pro spojky: $\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow, \dots$
3. pomocné symboly: $(,)$

- **správně utvořená formule** (dále jen formule)

1. každý výrokový symbol je formule (tzv. atomická formule)
2. je-li výraz A formule, pak $\neg(A)$ je formule
3. jsou-li výrazy A, B formule, pak také $(A) \vee (B)$, $(A) \wedge (B)$, $(A) \Rightarrow (B)$, $(A) \Leftrightarrow (B)$, ... jsou formule
4. ... a případně pro další spojky ...
5. nic jiného není formule

- není definována priorita binárních operátorů

- závorková konvence: závorky lze vynechat, pokud to není na újmu jednoznačnosti formule

K logickým spojkám

- **nulární** pravdivostní funkce (nezávislé na žádném argumentu) jsou konstanty odpovídající pravdivostním hodnotám 0, 1
- **unární** (jednoargumentové) spojky: F_1 – unární verum, F_2 – unární projekce p , F_3 – negace p (ozn. $\neg p$); F_4 – unární falsum.
- **binární** – \wedge , \vee , \Rightarrow , \Leftrightarrow , ...
- **ternární** – ternární \wedge , ... , if X then Y else Z
- ...

Úplný systém logických spojek

U každého jazyka nás zajímá nejen jeho expresivita (síla vyjádření).

Existuje minimální dostatečná množina spojek?

- prostřednictvím systému spojek $\{\neg, \wedge, \vee\}$ dokážeme vyjádřit libovolnou spojku.
- množinu spojek s touto vlastností nazýváme **úplným systémem spojek**
- další úplné systémy např. $\{\neg, \wedge\}$, $\{\neg, \vee\}$, $\{\neg, \Rightarrow\}$
- jednoprvkové úplné systémy: Shefferova funkce **NAND** (negace konjunkce), Nicodova funkce **NOR** (negace disjunkce)
Př. $(p \vee q) \Leftrightarrow ((p | p) | (q | q))$

Umíme systematicky hledat úplné systémy spojek?

Jaké vlastnosti musí mít úplné systémy spojek?

https://en.wikipedia.org/wiki/Functional_completeness

Sémantika. Interpretace

- **Pravdivostní ohodnocení (interpretace)** je funkce přiřazující všem atomickým formulím dané úvahy pravdivostní hodnoty (tj. každému výrokovému symbolu přiřadí 0 nebo 1).
- **Valuace** je rozšíření interpretace z atomických na všechny formule dle tabulky pro výrokové spojky (přiřadí 0 nebo 1 např. i $p \wedge \neg q$).
- Interpretace I **splňuje formuli** A (formule je **pravdivá** v I , resp. odpovídající valuace $I'(A) = 1$), pokud
 1. A je výrokový symbol a $I(A) = 1$
 2. A je $\neg B$ a I nespĺňuje B , resp. $I'(B) = 0$
 3. A je tvaru $B \wedge C$ a I splňuje B i C , resp. $I'(B) = I'(C) = 1$
 4. A je $B \vee C$ a I splňuje B nebo C , resp. $I'(B) = 1$ nebo $I'(C) = 1$
 5. A je tvaru $B \Rightarrow C$ a I nespĺňuje B nebo splňuje C , resp. $I'(B) = 0$ nebo $I'(C) = 1$
 6. A je $B \Leftrightarrow C$ a I splňuje B i C nebo I nespĺňuje B i C , resp. $I'(B) = I'(C)$

Splnitelnost. Model formule

- Mějme formuli $\neg p \vee p$; všechny možné interpretace (existují dvě: $I(p) = 1, I_1(p) = 0$) splňují tuto formuli.
Formule, která je splňována každou interpretací, se nazývá **tautologie**.
- Formule $\neg p \wedge p$ není splňována žádnou z možných interpretací; takové formule nazýváme **kontradikce**.
- Formule A je **splnitelná**, je-li splňována alespoň jednou interpretací. Tuto interpretaci označujeme jako **model** formule A .
- Množina formulí \mathbf{T} je splnitelná, pokud existuje interpretace splňující každou formuli z \mathbf{T} . Tuto interpretaci nazýváme **modelem množiny \mathbf{T}** .
- Formule A **logicky vyplývá** (na základě výrok. logiky) z množiny \mathbf{T} , pokud pro každý model I množiny \mathbf{T} I splňuje A . Zapisujeme $\mathbf{T} \models A$.

Logický agent pro výpočet logického důsledku

function TT-ENTAILS?(KB, α) *#returns true or false*

inputs: KB , the knowledge base,

a sentence **in** propositional logic α ,

the query, a sentence **in** propositional logic

symbol \leftarrow a list of the proposition symbols **in** KB and α

return TT-CHECK-ALL($KB, \alpha, symbols, \{\}$)

function TT-CHECK-ALL($KB, \alpha, symbols, model$) returns true **or** false

if EMPTY?(symbols) **then**

if PL-TRUE?($KB, model$) **then return** PL-TRUE?($\alpha, model$)

else return true //when KB is false, always **return** true

else

$P \leftarrow$ FIRST(symbols)

rest \leftarrow REST(symbols)

return (TT-CHECK-ALL($KB, \alpha, rest, model \cup P=true$))

and

TT-CHECK-ALL($KB, \alpha, rest, model \cup \{P=false\}$))

SAT problém

Výroková formule Φ je splnitelná, právě když existuje interpretace (tj. substituce výrokových symbolů), která je modelem, tj. Φ je pravdivá v této interpretaci.

SAT problém: pro danou formuli Φ najít takovou interpretaci nebo vrátit, že neexistuje.

SAT problém je NP-úplný

existují podproblémy, které jsou P-SAT a jsou užitečné?

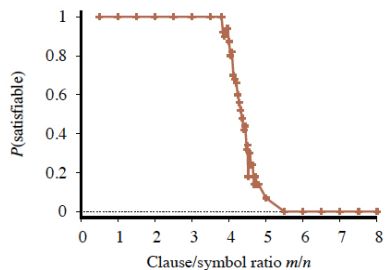
2-SAT. Horn-SAT, monotonní formule

3-SAT?

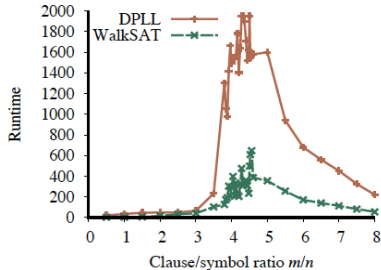
NP-úplný. Důkaz např.

<http://ktiml.mff.cuni.cz/kucerap/NTIN090/NTIN090-poznamky.pdf>

3-SAT



(a)



(b)

Figure 7.19 (a) Graph showing the probability that a random 3-CNF sentence with $n = 50$ symbols is satisfiable, as a function of the clause/symbol ratio m/n . (b) Graph of the median run time (measured in number of iterations) for both DPLL and WALKSAT on random 3-CNF sentences. The most difficult problems have a clause/symbol ratio of about 4.3.

Shrnutí

Víme,

- co je **interpretace** (angl. assignment)
- kdy je formule **pravdivá v interpretaci** (dtto interpretace splňuje formuli)
- co je **model** formule a množiny formulí
- co jsou tautologie a kontradikce
- kdy formule **logicky vyplývá** z množiny formulí/je logickým důsledkem této množiny
- co je **úplný systém logických spojek** (angl. functionally complete)

co je WalkSAT a DPLL?

Pravdivost, dokazatelnost.

Důkazové metody a systémy.

Formální (deduktivní) systémy. Axiomatické systémy