# IB101

*materiály vychází z oficiálních slidů předmětu a externích internetových zdrojů*

*v případě nalezení chyby ji neváhejte opravit a nahrát revizi*

*nemusí obsahovat všechno, co bude u zkoušky*

*vytvořeno 2017*

# Výroková logika

Logika zkoumá problematiku správného usuzování.

Úsudek – z předpokladu plyne závěr

**Výrok**

* Tvrzení nebo jazykový výrok, o jehož (ne)pravdivosti má smysl uvažovat.
* Jednoduchý výrok (bez logických spojek) / složený výrok (obsahuje alespoň 1 log. spojku)

**(N-ární) pravdivostní funkce**

* Funkce přiřazující n-ticím nějakou pravdivostní hodnotu
* Pravdivostní hodnota složeného výroku je jednoznačně dána pravdivostními hodnotami jeho složek

**Nulární pravdivostní funkce**

* Konstanty odpovídající pravdivostním hodnotám 0, 1

**Unární výrokové spojky**

* Verum, projekce, negace, falsum.

**Binární**

* Komutativní (nezáleží na pořadí argumentů)
* P => Q ; inverzní -P => -Q ; konverzní Q => P ; kontrapozitivní -Q => -P
* XOR, NAND, NOR, negace implikace (inhibice)

**Symbolický jazyk výrokové logiky**

* **Abeceda**
* Výrokové symboly (p, q, r, s)
* Symboly pro spojky (negace, konjunkce,…)
* Pomocné symboly (,)
* **Správně utvořená formule** (dále jen formule)
* Každý výrokový symbol je formule (tzv. atomická formule)
* Jeli výraz A formule, pak –(A) je formule
* Jsou-li výrazy A, B formule, pak také (A) V (B), (A) A (B), … jsou formule
* Nic jiného není formule
* **Závorková konvence**
* Závorky lze vynechat, pokud to není na újmu jednoznačnosti formule

**Interpretace** – neboli pravdivostní ohodnocení, je funkce, která přiřazuje atomickým formulím (výrokovým proměnným) pravdivostní hodnotu.

**Model** – je taková sada *interpretací* všech ve formuli či množině formulí zúčastněných výrokových proměnných, při které je ona formule, resp. množina formulí pravdivá.

**Tautologie** – Formuli splňuje každá interpretace (každá interpretace je modelem) |= A

**Kontradikce –** Formule není splněna žádnou z interpretací

**Splnitelnost –** Existujemodel

**Odvoditelnost –** formule A logicky vyplývá z množiny T jestliže pro každý model I množiny T, I splňuje A, zapisujeme T |= A

**Dokazatelnost –** A |- B, B je dokazatelné z A

**Literál –** výroková proměnná nebo její negace / atomická formule nebo její negace

**Klauzule -**  množina literálů chápaná jako jejich disjunkce

**Formule** – konjunkce klauzulí.

# Rezoluce

deduktivní systém pro výrokovou i predikátovou logiku, který může být korektní a úplný

**korektní** -- všechna dokazatelná tvrzení jsou pravdivá, A |- B => A |= B, co dokážeme, v logickém kalkulu platí

**úplný** -- všechna pravdivá tvrzení jsou dokazatelná, A |= B => A |- B, co je odvoditelné, je dokazatelné

**obecná rezoluce** -- můžeme si dělat, co chceme; pro predikátovou logiku je korektní a úplná

**lineární rezoluce** -- vždy rezolvujeme s předchozí rezolventou, můžeme použít dříve odvozené rezolventy; pro predikátovou logiku i výrokový počet je korektní a úplná

**LI-rezoluce** -- začínáme vždy cílovou klauzulí (s všemi negativními literály), právě odvozená klauzule musí být použita v bezprostředně následujícím rezolučním kroku, střední klauzule se nemůže později použít jako klauzule boční; pro výrokový počet je korektní a není úplná (pro hornovy klauzule je korektní a úplná)

**LD-rezoluce** -- používáme pouze Hornovy klauzule, klauzule jsou uspořádané (místo množiny použijeme seznam); právě odvozená klauzule musí být použita v bezprostředně následujícím rezolučním kroku, pro výrokový počet je korektní, ale není úplná

**SLD-rezoluce** -- používáme pouze Hornovy klauzule, klauzule jsou uspořádané (místo množiny použijeme seznam), vybereme jedno pravidlo, podle kterého rezolvujeme, a nadále se ho musíme držet (např. vždy první literály), právě odvozená klauzule musí být použita v bezprostředně následujícím rezolučním kroku, střední klauzule se nemůže později použít jako klauzule boční; pro výrokový počet je korektní, ale není úplná

**T-rezoluce** -- žádná z rodičovských klauzulí není tautologie

# Predikátová logika

Plně přejímá výsledky výrokové logiky

Zabývá se navíc strukturou jednotlivých jednoduchých výroků, na základě této analýzy lze odvodit platnost některých výroků, které ve výrokové logice platné nejsou

**Literál – v** predikátové logice je atomická formule (pozitivní literál), nebo negace atomické formule (negativní literál). Komplementární literály jsou dva literály, z nichž jeden je negací druhého.

**Predikát**

* N-ární relace, vyjadřuje vlastnosti objektů a vztahy mezi nimi
* Unární predikáty vyjadřují vlastnosti
* Binární, ternární,..,n-ární vyjadřují vztahy mezi dvojicemi, trojicemi, n-ticemi objektů
* Nulární predikáty představují původní výroky (bez vypuštěných jmen objektů) ve výrokové logice
* přiřazení syntaktického významu predikátu (interpretace): P(x, y) ⇔ x a y jsou v určitém vztahu

**Konstanty –** nulární fce; reprezentují jména objektů; jedná se o prvky předem specifikované množiny hodnot – domény, začátek abecedy

**Proměnné** – zastupují jména objektů; mohou nabývat libovolných hodnot z dané domény, konec abecedy

* **generalizace** -- náhrada konstanty proměnnou P ├ (∀x)P(x)
* **specializace** -- náhrada proměnné konstantou
* **volná proměnná** -- není pro danou formuli kvantifikována
* **vázaná proměnná** -- je pro danou formuli kvantifikována

pro atomickou formuli je volná každá proměnná, která se v ní vyskytuje

můžeme uvažovat jednotlivé výskyty, např. první výskyt volný, druhý vázaný P(x) => ∀x Q(x)

**Funkce –** reprezentují složená jména objektů (fce f reprezentující sčítání f(2,1) == f(3,0) == f(1,2) == ..)

**Doména –** množina hodnot, kterých mohou nabývat termy, též **univerzum**

**Term –** prvky složené pouze z funkčních symbolů, konstant a proměnných – f(x, g(y, h(h, y), 1), z)

* Nabývají hodnot v rámci dané domény
* Každá proměnná a každá konstanta je term
* Je-li f n-ární funkční symbol a t1,..,tn jsou termy, pak f(t1..tn) je term
* Nic jiného není term
* Uzavřený (základní) term -- term neobsahující proměnnou

**Formule**

* **Atomická formule**
* **Predikáty** – je-li P n-ární predikátový symbol a t1,..,tn termy, pak P(t1,..,tn) je atomická formule
* **Rovnosti** – jsou-li t1 a t2 termy, pak t1 = t2 je atomická formule
* **Formule**
* Každá atomická formule je formule.
* **Negace** – je-li A formule, pak –A je formule
* **Binární spojky** – jsou-li A,B formule, pak (A V B),…, jsou formule
* **Kvantifikátory** – je-li x proměnná a A formule, pak (∀xA) a (∃xA) jsou formule
* Pouze výrazy získané konečně mnoha aplikacemi těchto pravidel jsou formule
* otevřená formule – neobsahuje vázané proměnné = neobsahuje žádný kvantifikátor
* uzavřená formule – neobsahuje volné proměnné = každá proměnná v ní obsažená je kvantifikována

**Sentence –** formule predikátové logiky bez volných proměnných = uzavřená formule

**Teorie –** množina sentencí

**Kvantifikátory –** složené predikáty lze vytvářet i pomocí kvantifikátorů

* **Univerzální** (obecný) ∀xP(x) **-** pro každý prvek x domény platí P(x) – zobecnění konjunkce pro nekonečné domény
* **Existenční** ∃xP(x) – pro některé prvky x domény platí P(x) (existuje alespoň jeden) – zobecnění disjunkce pro nekonečné domény

## Formalizace jazyka predikátové logiky

### Syntax

predikátové logiky je složen z:

* **logických symbolů** – spočetná množina objektových proměnných, výrokové logické spojky, obecné a existenční kvantifikátory
* **speciálních symbolů** – neprázdná množina predikátových symbolů, množina symbolů funkčních a množina symbolů konstantních
* **pomocných symbolů** – závorky, čárka, ( ) , [ ]

### Abeceda

* proměnné (nespecifikovaná jména objektů): x, y, z…
* konstanty (vlastní jména objektů): a, b, c...
* symboly pro spojky: ¬, ⋀, ⋁, ⇒, ⇔
* symboly pro kvantifikátory: ∀, ∃
* n-ární funkční symboly (složená jména objektů): f, g, h…
* n-ární predikátové symboly: P, Q, R…
* pomocné symboly: (,)

### Sémantika a interpretace

pro analýzu sémantiky potřebujeme specifikaci jazyka (říct jakou máme doménu, konstanty, funkční a predikátové symboly)

*interpretace* se někdy nazývá *realizace*

je potřeba chápat, že:

* **D** nám říká, z jaké množiny bereme konstanty a proměnné
* **P(x, y)** nám vrací true nebo false (1, 0), například x > y, x je otcem y
* **f(x, y)** nám vrací nějaké číslo, které je výsledkem operace f, např. x + y, max(x, y)
* interpretací potom chápeme sémantický význam, který takovému D, P nebo f přiřadíme

**formálně**:

Interpretace jazyka predikátové logiky je struktura I složená z:

* libovolné neprázdné množiny D (domény)
* zobrazení I(f) : Dn -> D pro každý n-ární funkční symbol f, n >= 0
* n-ární relace I(P) ⊆ Dn pro každý n-ární predikátový symbol P, n >= 1

Řekneme **například**, že:

* D je množina celých čísel
* I(a) = 0, I(b) = 3 … interpretujeme konstanty (dle domény)
* I(f) = +, I(g) = \* … interpretujeme význam funkčních symbolů = definujeme funkce
* I(P) = >, I(Q) = max … interpretujeme význam predikátových symbolů = definujeme relace

U proměnných provádíme **valuaci**:

* valuace je libovolné zobrazení z množiny všech proměnných do D
* přiřazujeme proměnné x prvek d z množiny D, označíme V[x/d]
* hodnota termu pak závisí na jeho podobě: interpretaci konstant (pokud jsou) a valuaci proměnných (pokud jsou)

## Rezoluce v predikátové logice

platí obdobná pravidla jako u dokazování ve výrokové logice

nutnost převést formule do konjunktivní normální formy

u LD a SLD rezoluce použití hornových formulí

dále provádíme: prenexaci, skolemizaci, standardizaci, substituci, unifikaci

* **prenexová normální forma** -- formule se všemi kvantifikátory na začátku, zachovává ekvivalenci i splnitelnost
* **skolemizace** -- odstranění existenčních kvantifikátorů a nahrazení jimi vázaných proměnných pomocnými Skolemovými funkcemi (nulární funkce jsou konstanty), zachová splnitelnost formulí; (v praxi to znamená, že když nemáš před ∃y žádné ∀x, tak ho nahradíš konstantou, v opačném případě ho nahradíš funkcí toho x)
* **notace** jako u výrokové logiky: klauzule -- množina reprezentující disjunkce literálů, formule -- množina reprezentující konjunkce klauzulí
* **standardizace** proměnných -- je nezbytným předpokladem pro jejich unifikaci, mezi stejně pojmenovanými proměnnými v různých klauzulích není žádná vazba – proměnné přejmenujeme
* **substituce** -- zobrazení z množiny proměnných do množiny termů (tohle/za\_tohle), substituovat lze pouze za volné proměnné
* **přejmenování** -- proměnné, např. x|x1: P(x) > P(x1)
* **unifikace** -- substituce, po jejíž aplikaci na množinu termů dostaneme jeden term, substituovat se může za proměnnou, ale ne za funkci ani ne za konstantu
* **nejobecnější unifikátor** -- substituce, jejíž aplikací na dva či více výrazů získáme identitu; existuje jen jeden
* **teorém** -- formule systému, která je v něm dokazatelná

## Herbrandova věta

Usnadní nám určení, zda je daná množina formulí splnitelná.

Mějme množinu formulí ve Skolemově normální formě S, pak:

* **Herbrandova věta** -- *Buď existuje Herbrandův model S (tj. množiny formulí v snf), nebo existuje konečně mnoho uzavřených instancí prvků S, jejichž konjunkce neplatí.*

K Herbrandovu modelu se dostaneme takto:

* **Herbrandovo universum** U(S): množina všech **uzavřených termů**, které lze vytvořit z konstant a funkčních symbolů z S. tj. konstanty + funkce s konstantami + funkce s funkcemi s konstantami + ...

pokud v základu žádnou konstantu nemáme, můžeme si přidat jednu libovolnou, třeba *a*

* **Herbrandova báze** B(S)­: množina všech **atomických formulí**, které lze vytvořit nad prvky U(S). tj. predikáty s konstantami + predikáty s funkcemi a konstantami + …
* **Herbrandova interpretace**: libovolná podmnožina báze B(S) zahrnující ty aplikace predikátů na prvky univerza, které jsou pravdivé
* **Herbrandův model** M(S): taková Herbrandova **interpretace**, ve které jsou všechny formule z S pravdivé

# Neklasické logiky

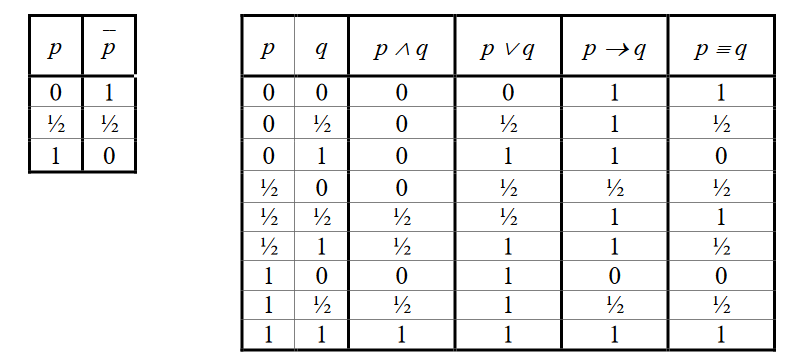
klasické logiky pracují pouze s hodnotami pravda-nepravda (princip dvouhodnotovosti) a nezabývají se praktickým obsahem tvrzení (princip extenzionality)

neklasické logiky porušují jeden nebo oba tyto principy

## Vícehodnotové logiky

### 3-hodnotová Lukasziewiczova logika

* pracuje s hodnotami pravda (1), nepravda (0) a možnost (nevím, 0,5)
* můžeme vytvořit obdobné pravdivostní tabulky jako pro klasickou logiku
* pravdivostní funkce pro implikaci val(p => q) je min(1,1-val(p)+val(q))
* pro snazší zapamatování tabulek je vhodné naučit se konjunkci a disjunkci a implikaci i ekvivalenci si převést na konjunkci-disjunkci



### mlhavá (fuzzy) logika

interval <0;1> čím blíže jsme k 1, tím je vyšší jistota platnosti, pracuje vlastně s pravděpodobností, k určení pravdivostních hodnot složených výroků je několik přístupů

### intuicionistické logiky

(od slova intuice) pravda je to, co lze sestrojit (dokázat), nepravda to, co lze vyvrátit, připouští se, že jsou tvrzení, jejichž pravdivostní hodnotu neznáme

p ⋁ ¬p tedy není vždy tautologie

## Modální logiky

### modální logika s alethickými modalitami

* alethické (aletheia = pravda) modality jsou zde možnost a nutnost
* exitují tvrzení, která jsou buď nutná, nebo možná
* (□ box = nutnost (nebe je modré); ♢ diamond = možnost (člověk má 3 oči))
* možné je všechno to, co je myšlenkově bezesporné, bez ohledu na to, jestli to je nebo není uskutečnitelné

### kripkeho rámec

* jedná se o orientovaný graf, kde každý vrchol je jedním ze světů
* orientace šipek v grafu znamená, že je jeden svět dosažitelný z druhého
* například ♢p → □ p, ale nesmí být □ p → ♢-p
* přistupujeme na existenci světů, v nichž možné tvrzení buď platí nebo neplatí
* **příklad:**
* Pepa je v Praze (w1) a ví, že v Praze platí s (svítí sluníčko).
* Ví, že existuje Brno, ale neví, jestli pro Brno platí s, nebo nikoliv.
* Proto pro něj existují 2 světy Brno (w2, w3), kde jeden svět je, že pro Brno platí s a druhý, že nikoliv.
* Kripkeho rámec by tedy vypadal nějak takto: (w3) ← (w1) → (w2); pro w1 platí s, pro w2 neplatí s, pro w3 platí s.

### temporální logika

výroková temporální logika, temporální modality

* F – bude platit někdy v budoucnosti
* G – bude platit vždy v budoucnosti
* P – platilo někdy v minulosti
* H – platilo vždy v minulosti

**minimální temporální logika** -- nemá omezení

* T – množina časových bodů
* R – relace následnosti na T

# Z odpovědníků

* Mějme nějaký formální systém výrokové logiky a označme T množinu všech teorémů, které v něm lze odvodit. Dále označme V množinu všech správně utvořených formulí výrokové logiky a P množinu všech tautologií.

Platí toto:

Je-li systém **sporný, pak T = V**

Je-li systém **úplný, pak T = P**

Je-li systém **korektní, pak T ⊆ P**

* Výrok je **nulární predikát**.
* Obecná i lineární rezoluce **je korektní a úplná**.
* LI-rezoluce i (S)LD-rezoluce **je korektní ne však úplná pro výrokový počet**.
* Nechť formule A, B obsahují spojky ¬, ⋀, ⋁. Nechť A', B' vzniknou z A, B záměnou spojek ⋀, ⋁.

Pak platí **⊨ A právě když ⊨ ¬A'**. *(„je tautologií“)*

* Pro libovolné výrokové formule A, B, C platí: **{A, B} ⊨ C právě když A ⊨ (B ⇒ C).**
* Mějme výrokové formule A1, A2, A3, B. Pak platí, že A1, A2, A3 ⊨ B **právě když A1, A2 ⊨ A3 ⇒ B**.
* Formule A logicky vyplývá z množiny T, pokud **pro každý model I množiny T I splňuje A**.
* Důkaz se skládá **z nejméně čtyř rezolučních kroků**.
* Rezoluční odvození □ z P ⋃ {G} je zároveň rezolučním **vyvrácením P ⋃ {G}.**
* Každá nesplnitelná množina neprázdných Hornových klauzulí musí obsahovat **alespoň jeden fakt** (aspoň jeden pozitivní literál) **a alespoň jeden cíl** (plně negativní klauzule).
* Prázdná klauzule □ je **vždy nepravdivá.**