



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

TENTO PROJEKT JE SPOLUFINANCOVÁN EVROPSKÝM SOCIÁLNÍM FONDEM A STÁTNÍM ROZPOČTEM ČESKÉ REPUBLIKY.

LOGIKA PRO INFORMATIKY

MARTIN KOTYRBA



PODPORA TERCIÁRNÍHO VZDĚLÁVÁNÍ
STUDENTŮ SE SPECIFICKÝMI
VZDĚLÁVACÍMI POTŘEBAMI
NA OSTRAVSKÉ UNIVERZITĚ V OSTRAVĚ

CZ.1.07/2.2.00/29.0006

OSTRAVA, ČERVENEC 2013

Studijní opora je jedním z výstupu projektu ESF OP VK.

Číslo Prioritní osy:	7.2
Oblast podpory:	7.2.2 – Vysokoškolské vzdělávání
Příjemce:	Ostravská univerzita v Ostravě
Název projektu:	Podpora terciárního vzdělávání studentů se specifickými vzdělávacími potřebami na Ostravské univerzitě v Ostravě
Registrační číslo projektu:	CZ.1.07/2.2.00/29.0006
Délka realizace:	6.2.2012 – 31.1.2015
Řešitel:	PhDr. Mgr. Martin Kaleja, Ph.D.

Tento projekt je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.

Název: Logika pro informatiky
Autor: RNDr. Martin Kotyrba, Ph.D.

Studijní opora k inovovanému předmětu: Logika pro informatiky (KIP/LZUI1)

Jazyková korektura nebyla provedena, za jazykovou stránku odpovídá autor.

Recenzent: doc. RNDr. PaedDr. Hashim Habiballa, PhD., Ph.D.
Ostravská univerzita v Ostravě

© Martin Kotyrba
© Ostravská univerzita v Ostravě
ISBN 978-80-7464-328-6

Obsah:

1 LOGIKA JAKO ZÁKLADNÍ POJEM MYŠLENÍ	9
1.1 Dějiny logiky	9
1.2 Matematická logika	11
2 UMĚLÁ INTELIGENCE A FORMÁLNÍ LOGIKA	12
2.1 Myšlení a jeho modelování	12
2.2 Informace a znalosti	13
3 SYNTAX JAZYKA L VÝROKOVÉ LOGIKY	16
3.1 Pojem výroku a jeho pravdivosti	16
3.2 Expresivita logických spojek	18
3.3 Výroková logika a její gramatika	18
4 SÉMANTIKA JAZYKA L VÝROKOVÉ LOGIKY	22
4.1 Význam jazyka l výrokové logiky	22
4.2 Modely formulí, tautologie, kontradikce a splnitelnost	25
4.3 Quinův algoritmus a rozhodnutelnost	27
4.3.1 Sémantický strom a Quineův algoritmus	27
5 NORMÁLNÍ FORMY VÝROKOVÝCH FORMULÍ	30
5.1 Disjunktivní normální formy formulí	30
5.2 Konjunktivní normální formy formulí	32
6 TABLOVÉ A REZOLUČNÍ NEPŘÍMÉ DŮKAZY VE VÝROKOVÉ LOGICE	34
6.1 Historické podoby sémantických tabel	34
6.1.1 Bethovo sémantické tablo	34
6.1.2 Hintikkovo sémantické tablo	36
6.1.3 Smullyanovo sémantické tablo	37
6.2 Základní myšlenka sémantického tabla	38
6.2.1 Charakteristické rysy sémantického tabla	38
6.2.2 Tablová pravidla	38
6.2.3 Splnitelnost a platnost v rámci sémantického tabla	39
6.3 Sémantické tablo ve výrokové logice	40
6.3.1 Sémantické tablo jako binární strom	40
6.3.2 Splnitelnost a logická platnost	43
6.3.3 Teorie a logický důsledek	43
6.3.4 Přirozená dedukce	44
6.4 Alternativní přístupy tvorby sémantického tabla s diagramatickými přístupy	45
6.4.1 Splnitelnost a platnost	46
6.4.1 Tablo s vlastností binárního stromu	47
7 DEDUKCE VE VÝROKOVÉ LOGICE	49
7.1 Logické důsledky výrokových formulí	49
7.2 Nepřímé důkazy logických důsledků	51
7.3 Logické důsledky a tablová metoda	52
8 SYNTAXE A SÉMANTIKA PREDIKÁTOVÉ LOGIKY	55
8.1 Jazyk predikátové logiky	55
8.2 Syntax jazyka predikátové logiky	56

8.3 Sémantika jazyka L_1	57
9 PREDIKÁTOVÁ LOGIKA A VYUŽITÍ DEDUKCE	60
9.1 Dedukce a sémantické tablo v predikátové logice	60
9.2 Sémantické tablo a logický důsledek	63
10 KLAUZULÁRNÍ FORMA FORMULE V JAZYCE L_1	65
10.1 Algoritmus převodu formule do prenexní normální formy	65
10.2 Klauzulární forma formule, skolemizace	67
11 POROVNÁNÍ ZNÁMÝCH TABLORVÝCH PŘÍSTUPŮ	70
11.1 Vhodné nástroje	70
11.2 Složitost uzlů	71
11.2 Situace v predikátové logice	73
12 ANIMACE A JEJÍ PODPORA VE VÝUCE	75
12.1 Animace pro podporu výuky výrokové logiky	75
12.1.1 Výuková část	76
12.1.2 Vzorové úlohy	77
12.1.3 Procvičovací část	78
12.1.4 Typy příkladů	78
12.1.5 Tlačítka pro řízení animace	79
12.2 Animace a její podpora v predikátové logice	79
12.2.1 Jednotlivá menu	80
12.2.2 Části animace	81
12.2.3 Výuka	82
12.2.4 Příklady	82
13 CVIČEBNICE	84
LITERATURA K DOPORUČENÍ	89
VYSVĚTLIVKY K JEDNOTLIVÝM SYMBOLŮM	88

1 LOGIKA JAKO ZÁKLADNÍ POJEM MYŠLENÍ

V této kapitole se dozvíte:

- Z čeho pojem logiky vychází
- Kdo je považován za hlavního představitele logiky
- Proč a kde se berou historické kameny pro stavbu logiky

Po jejím prostudování byste měli být schopni:

- Znat principy historie logiky
- Rozumět pojmu matematická logika
- Znat nejvýznamnější představitele logiky

Klíčová slova této kapitoly:

Matematická logika, logika, aristoteles.

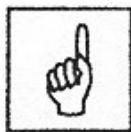
Doba potřebná ke studiu: 0,5 hodina



Průvodce studiem

V této úvodní lekci, si zavedeme pojmy z historie, uvědomíme si proč je důležitá matematická logika, kdo je považován za zakladatele logiky a kam až kořeny logiky sahají.

Logika je vědní disciplína, která se zabývá správností lidského myšlení.



Pojem **logika** má více významů – v češtině se běžně používá ve smyslu *myšlenková cesta, která vedla k daným závěrům*. **Logika** je také formální věda, zkoumající právě onen způsob vyvozování závěrů.

Logika není empirickou vědou o myšlení; studuje objektivní podmínky správnosti, jinak řečeno je to disciplína studující relaci „vyplývání“. Logika také nezkoumá úplně obecně poznání – to je předmětem filosofické disciplíny epistemologie.

Jako mnoho dalších věd vznikla logika coby součást filosofie a částečně takové zařazení stále platí. Logika se výrazně rozvinula i v matematice, a tak je řazena i do matematiky. Některé části logiky mají blíž k filosofii, některé k matematice, proto se někdy rozlišuje matematická logika. Logika má prakticky důležité aplikace v informatice.

1.1 Dějiny logiky

Za zakladatele logiky je považován Aristoteles (384–322 př. n. l.). Založil takzvanou sylogistickou logiku. Princip sylogismu se nejlépe vysvětlí na příkladu:

- Premisa 1: Každý člověk je smrtelný.
- Premisa 2: Sokrates je člověk.
- Závěr: Sokrates je smrtelný.

Aristoteles dále zkoumal modalitu (možnosti), čímž dal základ modální logice. Díla o logice: Organon (organon-nástroj, logika je nástroj vědy) Aristoteles se věnuje obecným termínům. Zabývá se jednoduchými univerzálními výroky (člověk je smrtelný)



Obrázek. 1.1: Dochovaná podoba Aristotela

Stoicko-megarská škola: složité výroky (jestliže prší, pak je mokro)

Středověká scholastika vychází ze sylogismu. Poprvé zavádí prvky temporální logiky, tedy logiky času. Významným představitelem scholastické školy byl William Occam (1290–1349). Usiloval o oddělení filosofie od teologie.

Novověká logika trpí na okraji zájmu ve stínu filosofie, za jejího zakladatele lze stále považovat Aristotela. Více přihlíží ke zkušenostem z okolního světa. Významní představitelé jsou Gottfried Wilhelm Leibniz (1646–1716), Bernard Bolzano (1781–1848), George Boole (1815–1864), Gottlob Frege (1848–1925), Georg Cantor (1845–1912) a Bertrand Russell (1872–1970). Mezi přední české představitelé patří (Alena Lukasová, Marie Duží a jiní).



Bernard Bolzano ve svém díle *Vědosloví* (*Wissenschaftslehre*, 1837) přináší koncept *věty o sobě* – toho, co mají společné věta napsaná (křivka-stopa inkoustu na papíře), věta vyslovená (kmitání akustického prostředí) a věta myšlená (nervový vzruch). Věta o sobě je tedy abstraktní vzhledem ke svým konkrétním realizacím.

Gottlob Frege napsal roku 1892 zřejmě první text moderní sémantiky, nazvaný *O smyslu a významu* (*Über Sinn und Bedeutung*). Podle Fregeho má každé slovo (věta) svůj smysl i význam (teorie A; neznámější); smysl je „způsob danosti“. Výraz buďto označuje přímo svůj význam nebo označuje význam skrze svůj smysl (tzv. teorie B). Modifikací teorie B je, že „slova v nepřímé řeči mají nepřímý význam. Tím odlišujeme obvyklý význam slova od jeho významu nepřímého a jeho obvyklý smysl od jeho smyslu nepřímého“.

1.2 Matematická logika

Matematická logika je vědní disciplína nacházející se na rozhraní mezi logikou a matematikou. Zabývá se zkoumáním, formalizováním a matematizováním zejména těch oblastí logiky, na jejichž základech je postavena matematika. V centru jejího zájmu jsou pojmy jako *důkaz*, *teorie*, *axiomatizace*, *model*, *bezespornost*, *úplnost*, *rozhodnutelnost*.



Základní disciplíny

Současná matematická logika se dělí na tři rozsáhlé disciplíny, které spolu úzce souvisejí. Jsou to teorie důkazu, teorie modelů a teorie aritmetiky.

- **Teorie důkazu** se zabývá vytvářením zkoumáním různých formálních deduktivních systémů jakožto základů pro pojem formálního důkazu. Používá čistě finitní metody nejčastěji aplikované na konečné posloupnosti znaků či slov.
- **Teorie modelů** se zabývá zkoumáním obecného pojmu matematické struktury a platnosti nějakého tvrzení v této struktuře. Zejména se zajímá o pojmy, jako jsou homomorfismus struktur, definovatelnost, axiomatizovatelnost, saturovanost, elementární vnoření. Zcela běžně používá infinitní metody teorie množin a výsledky, kterých lze v teorii modelů dosáhnout, jsou často závislé na přijetí či odmítnutí nějakého dodatečného množinového axiomu (axiom výběru, zobecněná hypotéza kontinua).
- **Teorie aritmetiky** se zabývá zkoumáním formálních aritmetických systémů, jako jsou Robinsonova a Peanova aritmetika a struktury v nich definovatelných množin přirozených čísel. Úzce souvisí s teoretickou informatikou zejména s teorií rekurze a teorií složitosti. Zajímá se také o možnosti „aritmetizace logiky“, tj. vyjádření některých logických pojmů, postupů a tvrzení v řeči přirozených čísel, a o to, jaké důsledky tato aritmetizace přináší (jedním z nich jsou například slavné Gödelovy věty o neúplnosti).

Kontrolní otázky:



1. S čím musí být propojená data, aby tvořila informaci?
2. Jakým nástrojem lze modelovat dedukci?
3. Co je operační sémantika?



Shrnutí:

První kapitola by Vám měla přinést jednoduchý a výstižný pohled do historie pojmu logiky a uvědomit si zásadní definice logiky a matematické logiky. Také by Vám neměli uniknout nejznámější představitelé logiky.

2 UMĚLÁ INTELIGENCE A FORMÁLNÍ LOGIKA

V této kapitole se dozvíte:

- Základní pojmy.
- Definici umělé inteligence
- Pojem formální logiky

Po jejím prostudování byste měli být schopni:

- Definoval znalost
- Rozlišovat procedurální a deklarativní znalosti
- Porovnat pojmům informace, znalost

Klíčová slova této kapitoly:

Umělá inteligence, informace, znalost, data, procedurální, deklarativní.

Doba potřebná ke studiu: 0,5 hodina

Průvodce studiem

V této úvodní lekci, jejíž prostudování by vám mělo trvat zhruba půl hodiny, se dozvíte, jaké místo v teoretické a aplikované informatice zaujímá její oblast, která se nazývá umělou inteligenci. V rámci takto vymezené oblasti se seznámíte s pozicí a rolí formální logiky. Budeme se zabývat úvahami o procesu lidského myšlení a diskutovat možnost jeho modelování pomocí jazyka formální logiky. Naučíte se rozlišit předmětný jazyk, jako nástroj modelování od metajazyka, který slouží k formulaci tvrzení o předmětném jazyce. Budeme též diskutovat pojem znalosti, abyste uměli rozlišit znalosti procedurální od znalostí deklarativních.



2.1 Myšlení a jeho modelování

Každý se bez pochyb nejménou v životě s pojmem „logika“ setkal. Na nejnižší úrovni abstrakce můžeme určitě říci, že s ní máme podobné zkušenosti, ovšem po formální stránce je velice rozsáhlou problematikou zasahující do četného množství oborů. Nás bude tento pojem zajímat z hlediska formalizace lidského myšlení, což je mimo jiné i cíl oboru Umělá inteligence. Zcela neformálně řečeno, logika je vlastně myšlenková cesta, jež vede k určitým závěrům. Ovšem pro účely exaktních věd jako je informatika či matematika je na místě danou definici patřičně formalizovat. Mohli bychom ji tedy chápat jako algoritimizaci lidského myšlení, která má při vší korektnosti dokázat vyvodit platný závěr. Zjednodušeně a striktně, je to věda o správném usuzování. Každého jistě napadne otázka, co ono správné usuzování vlastně znamená či dokonce jak poznáme správné od nesprávného. Logika poskytuje řadu schémat a prostředků pro správné usuzování, ovšem na symbolické úrovni, kde se nezohledňuje, ba dokonce ignoruje, obsah jednotlivých tvrzení.

Znalost je lidský odhad uložený v mysli, získaný pomocí zkušeností a interakcí s okolním prostředím. Znalost je fyzický, mentální nebo elektronický

záznam o vztazích, o kterých věříme, že existují mezi skutečnými či imaginárními entitami, silami, jevy, Znalost je vnitřní náhled, porozumění a praktické know-how, které všichni ovládneme – je to základní zdroj, který nám umožňuje chovat se inteligentně Znalost je informace o světě, která umožňuje expertovi udělat rozhodnutí. Rozhodnutí, které dokáže expert nebo kdokoli kdo s danou znalostí pracuje vytvořit je účelem smyslu jedné z částí vědy, která se nazývá umělá inteligence.

Již dávná filosofové dokázali charakterizovat intelektuální činnosti, kterými se vyznačují myslící bytosti. Těmito činnostmi mají být podle klasiků *analýza, syntéza, indukce, dedukce* a *analogie*. Zmocnit se podstaty těchto činností matematickými, tj. formálními prostředky, aby bylo možno jejich provádění svěřit počítačům, to je problémová oblast, která se nazývá *umělá inteligence*.



Umělá inteligence je věda o vytváření strojů nebo systémů, které budou při řešení určitého úkolu užívat takového postupu, který - kdyby ho dělal člověk - bychom považovali za projev jeho inteligence.

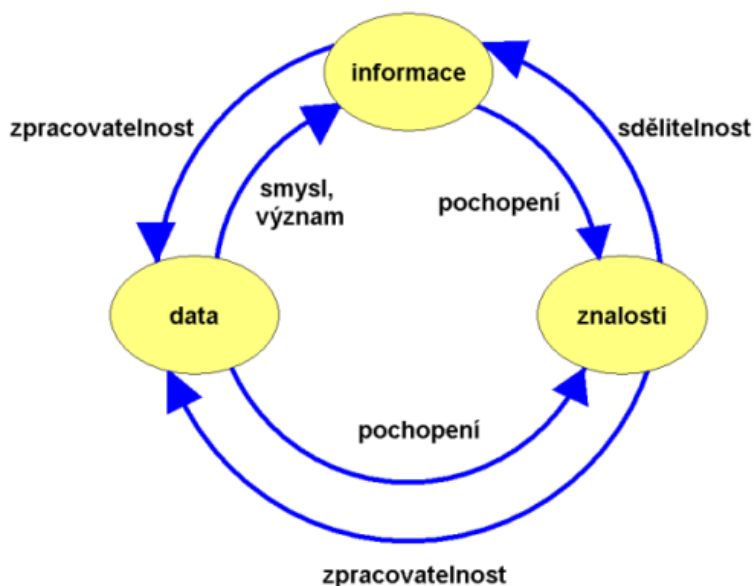
Popsat objekty našeho myšlení tak, aby se pak na nich v rámci uvedených disciplín prováděla analýza nebo aby mohly být klasifikovány na základě analogie, dnes existuje velká spousta nejen matematických disciplín (statistika, relační teorie, mnohorozměrná analýza dat aj.).

Formální logika je prostředek, pomocí kterého můžeme do určité míry úspěšně modelovat nejproblématictější z intelektuálních činností, tj. *dedukci*.

Postupně se můžeme v jisté posloupnosti zajímat o různé typy logik, jako např. o **Výrokovou, Predikátovou, Modální a Fuzzy logiku atd.** Konkrétní problematiku budeme řešit pomocí Výrokové a Predikátové logiky. Výroková logika je mimo jiné základní komponentou pro sestrojování matematických důkazů, ovšem sám o sobě, jako komplexní systém je například v informatice pouze cvičným nástrojem. Jeho nástavba, známá jako Predikátová logika, poskytuje daleko lepší práci s jednotlivými tvrzeními, jelikož umožňuje podstatně hlubší dekompozici výroku a dokáže zavádět podrobnější elementy, a to predikáty. Tento relativně silný odvozovací systém má značné využití například v relačních databázích a v již zmíněné umělé inteligenci.

2.2 Informace a znalosti

Claude Shannon v matematické teorii komunikace vymezil informaci jako statistickou pravděpodobnost určitého signálu či znaku, který je na vstupu určitého systému. Čím je menší pravděpodobnost daného znaku, tím větší má takzvanou informační hodnotu (pokud má nějaký znak pravděpodobnost velkou, je spíše očekáván a jeho výskyt příjemce tolik nepřekvapí). Tím, že systém signál zpracoval, dostal se na nižší úroveň nejistoty (entropie), tedy do stavu s vyšší mírou uspořádanosti.



Obrázek. 2.1: Struktura zpracování informací

Pojem dat není třeba zpřesňovat, neboť jak je známo, všechny znakové řetězce vstupující do výpočetního procesu lze obecně pokládat za data. Do výpočetního procesu však zřídka vstupují data, která nemají žádný význam, která tedy nejsou nějakým způsobem *interpretována*. [1], [2], [3].

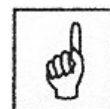
Interpretace dat je smysluplné přiřazení datům jejich významu (*sémantiky*). V informatice je pojem informace jedním ze základních pojmů, proto nestačí převzít jeho poněkud vágní vymezení, na něž jsme zvyklí z přirozeného jazyka. Přesněji vystihuje pojem informace následující definice.

Informaci tvoří data spolu se svou interpretací. Pojem informace je, jak je zřejmé z uvedených definic, neoddělitelný od sémantiky dat, která jsou jejími nositeli.

Znalost je informace, která je použitelná a začlenitelná, resp. odvoditelná v souvislosti s jinými informacemi.

Zpráva a informace jsou dva základní pojmy informatiky a umělé inteligence. Avšak jejich interpretace se liší velmi podstatně v běžném, vědeckém a technickém jazyce. Zpráva a informace mohou být velmi rozdílně interpretované pojmy. Pokud Čech dostane dopis v japonštině, asi mu žádnou informaci nepřinese. Pokud jej dostane Japonec, může informaci v dopise porozumět, protože ovládá jazyk, ve kterém je zpráva zapsána. Podobně v mnoha případech získáváme jen část informace ve zprávě, kterou jsme dostali, protože porozumění zprávě závisí na našich schopnostech rozumět jazyku, kterým je zapsána a na našich předchozích znalostech a zkušenostech.

Můžeme tedy říci, že informace je přenášena zprávou a navíc stejná informace může být přenášena různými zprávami (například zprávami v různých jazycích). Proces pro získávání informace ze zprávy se nazývá interpretace. Interpretaci může provést člověk (subjektivní) nebo je na úrovni technické (počítač) či biologické (testy na zvířatech). Informaci můžeme zachytit v různých podobách.



Pro počáteční období rozvoje informatiky bylo charakteristické, v návaznosti na předcházející etapu postupně stále vyspělejších mechanických pomocníků počítání, její zaměření na výpočetní a výpočetně rozhodovací procedury. Počítače totiž v počátcích svého využívání především realizovaly pracné výpočtové postupy, jako jsou např. výpočetní postupy v účetnictví, statistické nebo vědeckotechnické výpočty. Takový druh modelování lidské intelektuální činnosti vychází ze znalostí určitých pracovních postupů, tedy z určitých *procedurálních znalostí*.



Modelování lidské intelektuální činnosti vychází ze znalosti určitých pracovních postupů, tedy procedurálních znalostí. K modelování procedurálních znalostí slouží abstraktní modely orientované na pracovní postupy – algoritmy.



Deklarativní znalosti o objektech spočívají v konstatování jejich stavů, vlastností nebo vzájemných vztahů. Interpretací deklarativních znalostí jsou vhodně strukturovaná data. Jejich prostřednictvím lze odvozovat další znalosti a provádět i operace související se znalostmi procedurálními.



Data	Znakové řetězce.
Informace	Data spolu s jejich významem (databáze).
Znalosti	Informace vzájemně související (znalostní báze).
Procedurální znalosti	Popisují, jak se realizuje nějaký proces.
Deklarativní znalosti	Konstatují vlastnosti, vztahy a vzájemné souvislosti objektů modelovaného světa.

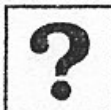
Tabulka 2.1: Příklady důležitých pojmů



Korespondenční úkol 1

Uvažujte, jaké druhy znalostí představují

- návod k použití automatické pračky
- technický popis automatické pračky



Kontrolní otázky:

- S čím musí být propojená data, aby tvořila informaci?
- Jakým nástrojem lze modelovat dedukci?
- Co je operační sémantika?



Shrnutí:

Modelovacím prostředkem lidského myšlení je v umělé inteligenci formální jazyk. Formální jazyk má svoji syntax určenou jeho gramatikou a sémantiku (významovou stránku). Jazykové výrazy reprezentují objekty modelovaného světa a naopak tyto objekty jsou denotáty jazykových výrazů. Znalosti mají procedurální nebo deklarativní charakter.

3 SYNTAX JAZYKA L VÝROKOVÉ LOGIKY

V této kapitole se dozvíte:

- Co je syntaxe výrokové logiky.
- Co ovlivňuje vytvoření formule.
- Jak se vytvořená formule stromově reprezentuje.

Po jejím prostudování byste měli být schopni:

- Co je výrok.
- Rozlišit formuli výrokové logiky.
- Stromově reprezentovat výrokovou formuli.

Klíčová slova této kapitoly:

Syntaxe, výrok, míry složitosti, expresivita.

Doba potřebná ke studiu: 1,5 hodina

Průvodce studiem

*Po zavedení nezbytného matematického aparátu se seznámíte se syntaxí jazyka **L**, a to vymezením množiny symbolů tohoto jazyka a stanovením jeho gramatických pravidel, podle nichž se tvoří formule jazyka. Uvědomíte si souvislost konstrukce dobře utvořené formule jazyka z dané abecedy podle daných pravidel s její reprezentací pomocí ohodnoceného binárního stromu. S ní též bude souviset pojem složitosti konstrukce výrokové formule. Na základě stromové reprezentace výrokové formule se též naučíte stanovit míry složitosti formule, tj. její složitost, její hloubku a její základnu.*



Stejně jako v každém jazyce, je třeba i v jazyce logiky stanovit určitá pravidla, aby byl jazyk srozumitelný a smysluplný. Syntax neboli skladba reprezentačního jazyka každé logiky je dána gramatikou, a tu tvoří abeceda jazyka a soubor pravidel syntaxe. Abeceda je soubor symbolů, které můžeme používat, je spočetný a konečný. Abeceda je tvořena různými symboly a liší se v daných logikách i jazycích. Pravidla syntaxe jsou definována induktivně, to znamená od jednoduchého ke složitějšímu. Pravidla syntaxe jsou tři a jsou to báze, indukce a generalizace.

3.1 Pojem výroku a jeho pravdivosti

Základním termínem logiky je výrok, jehož význam je třeba vysvětlit. Výrok je tvrzení, u něhož můžeme určit pravdivostní hodnotu. Z toho vyplývá, že výrokem nemůžou být například otázky (naopak odpovědi na otázky již výrokem být mohou), zákazy, nařízení apod. Pravdivost výroku se určuje dvěma termíny – pravda a nepravda. Nověji zavedené (respektive zde nověji používané) jsou anglické výrazy true (T) pro pravdu, false (F) pro nepravdu. Možné je také použití čísel 1 (pro true) a 0 (pro false).

Máme-li výrok K , můžeme určit jeho negaci, definicí určenou jako „není pravda, že K “ [9]. V tabulce pravdivostních hodnot to pak vypadá následovně:

A	$\neg A$
1	0
0	1

Tabulka 3.1: – Negace

Lidské myšlení vždy vychází z nějaké množiny tvrzení neboli výroků o uvažovaném světě. *Výroky*, jimiž se zabývá výroková logika, jsou taková tvrzení deklarativního typu, o jejichž pravdivostní hodnotě („pravdivý“/„nepravdivý“) má smysl uvažovat. Mezi nimi lze pak rozlišit výroky *elementární* neboli *atomické*, které již dále nelze rozložit na výroky jednodušší. Množina všech atomických výroků o modelovaném světě tvoří *universum diskursu*, v jehož rámci se modelování pomocí formálních prostředků výrokové logiky pohybuje. Při modelování je proto potřeba mít na zřeteli nejen prostředek, jímž modelování probíhá a kterým je zde výroková logika, ale též přesně vymežit, jaký „svět“ je předmětem tohoto modelování.



Jedním ze základních požadavků praktického využití výrokové logiky je rozpoznání výroků od ne-výroků a s tím souvisící možnost stanovení jejich pravdivostních hodnot. Např. dotaz „ K čemu mi bude učení?“ nebo zvolání „Dost už bylo učení!“ nejsou větami deklarativního typu, proto nepředstavují výroky a nemá smysl uvažovat o jejich pravdivosti. Pro převod skutečností našeho modelovaného světa do tzv. formálně reprezentovaného je potřeba stanovit nový jazyk. Přirozené jazyky jsou nevhodné, jelikož obsahují redundantní informace, které se nesnadno převádějí, proto jsou stanoveny jazyky zjednodušené, nebo-li formální jazyky reprezentace. Pokud převádíme ze světa modelovaného do formálně reprezentovaného, nazývá se tento proces reprezentace (objektu přiřazujeme určitý výraz). Proces opačný se potom nazývá denotace a znamená, že výrazu přiřadíme objekt. Při převodu je důležitým aspektem expresivita daného jazyka.



„Expresivita je míra schopnosti formálního jazyka přiblížit se svým významem jazyku přirozenému.“

Ve výrokové logice je například expresivita mnohem nižší, než v logice predikátové. Dále klauzulární logika převyšuje v expresivitě logiku predikátovou. Zabývat se zde však budeme převážně výrokovou a predikátovou logikou. [1], [2], [3].



Úkol 1

Uvažujte, které z následujících jazykových výrazů jsou výroky. Které z nich, jsou atomické a které komponované?

- Naši sportovci se předčasně vrátili z mistrovství, neboť prohráli rozhodující zápas.

- Škoda, že nedopadli lépe.
- Kdyby naši hokejisté vyhráli s Ruskem a Kanadou, neohrozili by svoji medailovou pozici.

3.2 Expresivita logických spojek

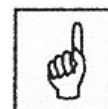
Expresivita formálního jazyka je úměrná tomu, do jaké míry je tento jazyk schopen postihnout způsob, jakým jsou ve výrazech přirozeného jazyka spojeny atomické výroky ve složitější významově soběstačné celky. V případech formálních jazyků používajících logických spojek jsou to právě ony, které tato spojení zprostředkovávají.

Z jednoduchých výroků vznikají složitější pomocí logických spojek.

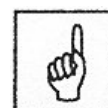
Název	Zápis	Slovní vyjádření
Konjunkce	$(K \& L)$	K a L
Disjunkce	$(K \vee L)$	K nebo L
Implikace	$(K \rightarrow L)$	Jestliže K, tak L
Ekvivalence	$(K \leftrightarrow L)$	K tehdy a jen tehdy, když L

Tabulka 3.2: Logické operace

Logické spojky byly zavedeny tak, aby měly v přirozených jazycích své „protějšky“. Toto vzájemné přiřazení je však značně limitováno. Pět výrokovými spojkami \neg , $\&$, \vee , \rightarrow , \leftrightarrow nelze vyjádřit celou bohatost spojení vět v jazyce přirozeném.



Výrok je tvrzení deklarativního typu, o jehož pravdivostní hodnotě („pravdivý“/„nepravdivý“) má smysl uvažovat. Výroky jsou elementární neboli atomické, které již dále nelze rozložit na výroky jednodušší. [1], [2], [3].

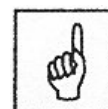


3.3 Výroková logika a její gramatika

Tato kapitola má za cíl přiblížit důležité pojmy a termíny gramatiky výrokové logiky. Celá struktura výrokové logiky vychází ze spočetné množiny symbolů, kterým se říká *abeceda*.

Jazyk **L** výrokové logiky, který zde bude definován, vychází ze základní množiny elementárních symbolů, které tvoří jeho *abecedu* definovanou pro výrokovou logiku takto:

Symboly **abecedy** jazyka **L** výrokové logiky patří výhradně do některé z následujících skupin elementárních symbolů. Abeceda výrokové logiky je tvořena čtyřmi základními typy symbolů[6]:



- 1) symboly pro výrokové proměnné – možno použít jakékoliv malé písmeno abecedy, možno indexovat (dolním indexem)
- 2) symboly logických konstant – T pro true, F pro false
- 3) symboly pro logické spojky – negace \neg , konjunkce $\&$, disjunkce \vee , implikace \rightarrow , ekvivalence \leftrightarrow
- 4) pomocné symboly – závorky ()

Podle počtu operandů se rozlišují mezi základními výrokovými spojkami spojka *unární*, tou je spojka \neg , a spojky *binární*, k nimž patří všechny zbývající čtyři základní spojky $\&$, \vee , \rightarrow , \leftrightarrow . Je to proto, že pomocí spojek binárních můžeme spojovat výroky jednoduché, neboli atomické a vytvářet formule. Logické konstanty true a false bývají někdy též považovány za *nulární* výrokové spojky. K tomu, aby bylo možno definovat syntax formulí jazyka výrokové logiky pomocí indukce a ukázat možnost její stromové reprezentace, je třeba si v následujícím odstavci připomenout některé pojmy teorie množin. Je potřeba pro základní představu definovat pomocí induktivní definice, která se skládá ze tří pravidel gramatická pravidla jazyka L výrokové logiky.



- *Báze*: Každá atomická formule je formulí.
- *Indukce*: Jsou-li X a Y formule, pak $\neg X$, $X\&Y$, $X\vee Y$, $X\rightarrow Y$ a $X\leftrightarrow Y$ jsou formule.
- *Generalizace*: Všechny *dobře utvořené formule* jazyka výrokové logiky jsou výsledkem konečného počtu aplikací základního pravidla a pravidla indukce.

Symboly X , Y uvedené v předcházející induktivní definici gramatiky jazyka výrokové logiky nepatří k symbolům jazyka, nýbrž zde zastupují jeho libovolné formule. Induktivní proces vytváření formule aplikací gramatických pravidel se, jak vyplývá z předcházejícího odstavce, děje v postupných krocích, z nichž v každém rovněž vzniká rovněž dobře utvořená formule. Ty pak tvoří jistou posloupnost zvanou *konstrukce formule*. Konstrukce formule je tím složitější, čím je posloupnost nutných konstrukčních kroků vedoucích k jejímu vytvoření delší.



Zapište posloupnost podformulí představující konstrukci následující formule:

$$((x\&y) \rightarrow \text{false}) \vee \neg y$$

a stanovte složitost této konstrukce.

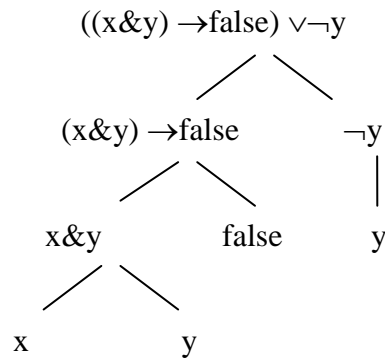
Konstrukci dané formule A tvoří např. (pořadí počátečních prvků posloupnosti označujících výrokové proměnné může být jiné) tato posloupnost jejích podformulí :

$$x, y, \neg y, x\&y, \text{false}, (x\&y) \rightarrow \text{false}, ((x\&y) \rightarrow \text{false}) \vee \neg y.$$

Konstrukce formule má složitost 7, neboť konstrukce formule je v každém případě posloupností o 7 členech.

Dále ukážeme, že se dá libovolná formule stromově reprezentovat pomocí procedury zvané syntaktický formační strom. Protože všechny zavedené výrokové spojky jsou spojkami nejvýše binárními, lze pro

reprezentaci syntaxe výrokových formulí využít vlastností pouze binárních stromů. Konstrukci formule A z předcházejícího příkladu výsleze reprezentovat konečným binárním stromem, jehož kořen je označen návěštím tvaru výsledné formule A a jehož listy nesou jako návěští atomické podformule formule A . Konstrukci formule A z předcházejícího příkladu odpovídá, jak ukazuje schéma níže grafickou formou reprezentovaný syntaktický formační strom.



Syntaktický formační strom formule A jazyka L výrokové logiky je konečný binární strom, jehož všechny uzly jsou označeny návěštím - podformulemi formule A tak, že obecně platí:

Kořen je označen formulí A . Jestliže je uzel označen některým z návěští $X \& Y$, $X \vee Y$, $X \rightarrow Y$, $X \leftrightarrow Y$, kde X , Y jsou formule, pak uzly bezprostředně následující nesou po řadě (zleva doprava) návěští X a Y . Je-li uzel označen podformulí $\neg X$, pak uzel bezprostředně následující nese jako návěští podformule X . Listy jsou označeny atomickými formulemi vyskytujícími se v A .



Ke každé formuli lze sestavit jeho syntaktický formační strom. Jde o způsob, jak zjistit složitost formule. V závislosti na logických spojkách se strom postupně zjednodušuje, vyvábí se další úrovně až do stavu, kdy se budou ve všech formách vyskytovat pouze výrokové proměnné. Pokud je alespoň jeden list ohodnocen jako true, je formule splnitelná. Pokud jsou všechny listy ohodnoceny jako false, je formule nespelnitelná, naopak pokud jsou všechny listy ohodnoceny jako true, je formule logicky platná. U syntaktického stromu formule určujeme hloubku, složitost a základnu formule. Hloubka značí počet úrovní stromu, daná formule má hloubku nula a počítá se postupně až po listy. Složitost je dána počtem spojek („počet uzlů, které nejsou listy“), základna formule je dána počtem proměnných, které jsou ve formuli obsažené. [1], [2], [3].

Korespondenční úkol

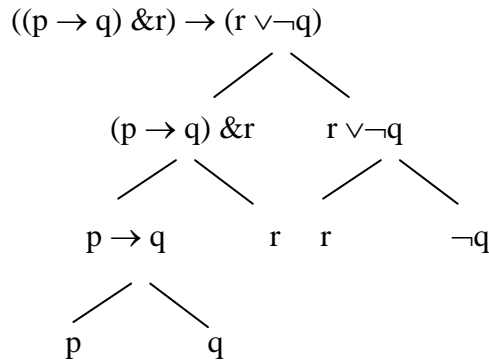
1. Nakreslete syntaktický formační strom formule a stanovte její hloubku, základnu, složitost a složitost konstrukce.

$$(p \vee (q \rightarrow r)) \rightarrow ((p \& s) \rightarrow r)$$

2. K následujícím větám sestavte výrokové formule.
 - a) Prší a mrzne, ale nesněží.



- b) Bud' prší nebo sněží a mrzne.
 c) Mrzne-li, neprší nebo sněží.
 d) Mrzne-li nebo sněží, neprší.
3. Najděte chybu v konstrukci syntaktického formačního stromu formule, opravte ji a z opraveného syntaktického formačního stromu stanovte složitost formule, její hloubku, velikost základny a složitost konstrukce



Kontrolní otázky:

1. Co je to výrok?
2. Jakým nástrojem lze převést formuli VR do stromové reprezentace?
3. Lze jakákoli formule VR vyjádřit pomocí syntaktického stromu?



Shrnutí:

Výrok je tvrzení deklarativního charakteru, o němž lze rozhodnout, zdali je pravdivý nebo nepravdivý, jiná možnost není. Jazyk výrokové logiky je určen jeho gramatikou, tj. abecedou a syntaktickými pravidly tvorby dobře utvořených formulí. Výrokovou formuli reprezentuje její jediný syntaktický formační strom. Charakteristiky složitosti, kterými jsou složitost formule, hloubka formule a základna formule a odpovídají příslušným charakteristikám syntaktického formačního stromu formule. Složitost konstrukce formule je dána počtem podformulí, které je třeba vytvořit, aby podle pravidel syntaxe formule vznikla. Množina všech formulí vytvořených ze spočetné abecedy je spočetná.

4 SÉMANTIKA JAZYKA L VÝROKOVÉ LOGIKY

V této kapitole se dozvíte:

- Co je sémantika výrokové logiky a vše co s ní souvisí.
- Jaké jsou základní rozhodovací algoritmy.
- Co je Quinův algoritmus
- Co je tautologie

Po jejím prostudování byste měli být schopni:

- Definovat model.
- Využívat tabulkovou metodu.
- Znat a umět využívat ekvivalence.
- Rozhodnout u jakékoli formule pomocí rozhodovacího algoritmu.
- Využívat Quinův algoritmus.
- Znat pojmy typu tautologie a kontradikce.

Klíčová slova této kapitoly:

Model, pravdivostní tabulka, ekvivalence, Quinův, algoritmus, tautologie, kontradikce.

Doba potřebná ke studiu: 1,5 hodina

Průvodce studiem

Přiřazovat význam neboli interpretovat prvky jazyka výrokové logiky se naučíte v této lekci, jejíž prostudování by vám mělo trvat zhruba 1,5 h. Seznámíte se v ní s interpretačními pravidly, která umožní na základě určitě stanovené pravdivostní struktury výrokových proměnných vyskytujících se ve formuli stanovit výslednou hodnotu interpretace jednotlivých logických spojení. Podle těchto pravidel budete atomickým a v návaznosti na ně komponovaným formulím jazyka přiřazovat jejich významy - denotáty z dvouprvkové množiny {true, false}. Na základě zavedené sémantiky jazyka L se seznámíte též s pojmy logicky platné (tautologické) formule, splnitelné a nesplnitelné (kontradiktické) formule. Na závěr lekce se budeme věnovat možnostem zjednodušujícího přepisu formulí na ekvivalentní formule, obsahující určité (funkčně úplné) množiny výrokových spojek. Uvidíte, že všechny zde uvedené metody představují jisté zlepšení efektivity rozhodování ve srovnání s rozhodováním interpretačními tabulkami. Seznámíte se s poměrně jednoduchým Quineovým algoritmem, který je založen na myšlence postupných valuací proměnných a částečných interpretací podformulí dané formule jen do takového kroku, od kterého se již valuacemi dalších proměnných výsledky interpretace formule nezmění.



4.1 Význam jazyka l výrokové logiky

Sémantika se týká významové stránky logiky, tudíž určení pravdivostních hodnot – neboli interpretace daných formulí. Interpretovat

formuli můžeme za předpokladu, že jsou ohodnoceny proměnné. Proces hodnocení proměnných, neboli přiřazení daným proměnným jejich významu se nazývá valuace. Sémantika se týká významové stránky logiky, tudíž určení pravdivostních hodnot – neboli interpretace daných formulí.

Interpretovat formuli můžeme za předpokladu, že jsou ohodnoceny proměnné. Proces hodnocení proměnných, neboli přiřazení daným proměnným jejich významu se nazývá valuace. V předcházející lekci byl nezávisle na významu definován čistě formálně konstruovaný jazyk L výrokové logiky. Podobně jako je tomu u sémantiky jazyka přirozeného, kdy jednotlivým jazykovým výrazům náleží jako jejich denotáty pojmy, mají i formule modelově zjednodušeného jazyka výrokové logiky svoji *denotační sémantiku*, spočívající v přiřazování denotátu každé z jeho dobře utvořených formulí. To ve výrokové logice znamená, že při dané pravdivostní struktuře elementárních výroků je každá její dobře utvořená formule jednoznačně interpretována určitou pravdivostní hodnotou.

Jak již vyplývá z vymezení pojmu výroku na základě pojmu pravdivosti či nepravdivosti elementárních tvrzení, atomické formule výrokové logiky, kterými jsou elementární výroky reprezentovány, nabývají vždy pouze některé z logických hodnot true, false. Totéž platí, jak bude dále zpřesněno pro formule komponované. Jinými slovy, *oborem sémantické interpretace* výrokové logiky je dvouprvková množina



$W = \{ \text{true}, \text{false} \}$. - její *universum diskursu*.

Proces přiřazování významu, který se nazývá *interpretací formule*, začíná stanovením určité pravdivostní struktury atomů vyskytujících se ve formuli. V případě logických konstant true, false jsou denotáty konstantně dány příslušnými pravdivostními hodnotami universa diskursu. V případě výrokových proměnných se denotáty stanoví *valuací*, tj. jejich ohodnocením, a to rovněž vždy některou z pravdivostních hodnot universa diskursu. Na základě přiřazení významu elementárním prvkům (atomům) jazyka se pak stanoví pomocí *interpretačních pravidel* její výsledná pravdivostní *hodnota interpretace* komponované formule.[1], [2], [3].



Valuace (ohodnocení) symbolu p označujícího výrokovou proměnnou je funkce $v(p)$ přiřazující proměnné p jednu ze dvou možných pravdivostních hodnot z universa diskursu $W = \{ \text{true}, \text{false} \}$

$v(p) = \text{true/false}$.

Funkční pojetí způsobu určení interpretačních pravidel umožňuje jejich přehledné vyjádření *pravdivostními tabulkami* (tab 4.1). Pravidlu pro interpretaci negace formule K odpovídá funkce jedné proměnné typu $f(K)$, pravidlům pro ostatní logické spojky spojující formule K a L funkce dvou proměnných typu $f(K,L)$.

K	L	$\neg K$	$\neg L$	$K \& L$	$K \vee L$	$K \rightarrow L$	$K \leftrightarrow L$
0	0	1	1	0	0	1	1
0	1	1	0	0	1	1	0
1	0	0	1	0	1	0	0
1	1	0	0	1	1	1	1

Tabulka 4.1: Tabulka pravdivostních hodnot

V tabulce 4.1 jsou uvedena interpretační pravidla pro interpretaci formulí s výrokovými spojkami \neg , $\&$, \vee , \rightarrow , \leftrightarrow přehledným způsobem pomocí interpretačních tabulek, vztahujících se k jednotlivým spojkám. Obdobně přehlednou *tabulkovou metodou* vyhodnocování Booleových funkcí je možno postupovat i při *sémantické analýze formule*. Ta spočívá v interpretacích formule pomocí interpretačních pravidel, přičemž jsou postupně uvažovány a řazeny do *interpretační tabulky* všechny možné valuace jejich výrokových proměnných.

Analyzujte tabulkovou metodou sémantiku formule $((x \& y) \rightarrow \text{false}) \vee \neg y$.



V případě formule $((x \& y) \rightarrow \text{false}) \vee \neg y$ bude potřeba pro dvojici symbolů reprezentujících výrokové proměnné uvažovat $2^2 = 4$ různých variací ohodnocení. Jednotlivé řádky tabulky sémantické analýzy formule představují interpretace dané formule při valuacích dvou výrokových proměnných, uspořádaných v prvních dvou sloupcích tabulky.

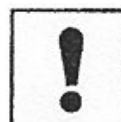
Jak je vidět z tab. 4.2, některé postupně již interpretované podformule je vhodné v tabulce pro přehlednost označit metasymboly.

X a Y jako metasymboly

		X	Y		
x	y	$x \& y$	$X \rightarrow \text{false}$	$\neg y$	$Y \vee \neg y$
0	0	0	1	1	1
0	1	0	1	0	1
1	0	0	1	1	1
1	1	1	0	0	0

Tabulka 4.2: Označení pomocí metasymbolů

Vytvořte si podle pravidel syntaxe nějakou výrokovou formuli F s alespoň třemi výrokovými proměnnými a proveďte její sémantickou analýzu



tabulkovou metodou. Její výsledky budete dále porovnávat s výsledky analýz jinými metodami.

4.2 Modely formulí, tautologie, kontradikce a splnitelnost

Při sémantické analýze formule tabulkovou metodou se v posledním výsledném sloupci objeví jedničky spolu s nulami, pouze jedničky nebo pouze nuly. Je proto vhodné již nyní zavést pojmy, které uvedené tři případy pojmenovávají a víceméně charakterizují.



Vzhledem k tomu, že každou proměnnou výrokové logiky můžeme považovat za atomickou formuli, je na místě zmínit ten fakt, že i na této úrovni uvažujeme její splnitelnost/platnost. Každá atomická formule je tedy splnitelná. Logické konstanty jsou taktéž splnitelné, navíc ještě logicky platné, což je podstatný fakt.

Logicky platné formule neboli tautologie jsou zároveň formulemi splnitelnými (konsistentními). Při dokazování logických platností se často využívá duality formulí, jelikož můžeme konstatovat, že negace tautologie je kontradikce a opačně.



Formule A jazyka výrokové logiky je tautologií právě tehdy, pokud je její negace $\neg A$ nespíitelná. Formule výrokové logiky je splnitelná, pokud existuje aspoň jedna konstelace jejích proměnných taková, že je interpretace dané formule true. Formule je kontradikcí, pokud je interpretována jako false při jakémkoliv ohodnocení jejích proměnných. Formule je tautologií, pokud je interpretována jako true při jakémkoliv ohodnocení jejích proměnných.



Zamyšlení:

tímto zamyšlením by měla výše uvedená definice zpřehlednit.

Je negace splnitelné formule splnitelná?

Je negace logicky platné formule splnitelná?

Je negace splnitelné formule logicky platná?

Dříve, než bude definován pojem modelu formule, je třeba připomenout, že každá valuace výrokových proměnných je přiřazením určité pravdivostní *struktury* elementárním výrokům, které zastoupeny výrokovými proměnnými tvoří součást báze jazyka zkonstruovaného speciálně k reprezentaci určitého problému. Interpretace formule A pravdivostní hodnotou true vychází tedy z určité pravdivostní struktury elementárních výroků o modelovaném světě reprezentovaných výrokovými proměnnými. Ve výrokové logice se v takovém případě hovoří o *modelu* formule A .

V případě, že uvažujeme množinu formulí U namísto formule jediné, se hovoří o modelu množiny formulí U . Model formule je pak modelem jednoprvkové množiny formulí.



Příklad:

Z následující tabulky sémantické analýzy formule $(\neg p \rightarrow r) \rightarrow (q \& (\neg r \rightarrow p))$ stanovte modely této formule:

				X	Y	Z	
	p	q	r	$\neg p \rightarrow r$	$\neg r \rightarrow p$	$q \& Y$	$X \rightarrow Z$
1*	0	0	0	0	0	0	1
2	0	0	1	1	1	0	0
3*	0	1	0	0	0	0	1
4*	0	1	1	1	1	1	1
5	1	0	0	1	1	0	0
6	1	0	1	1	1	0	0
7*	1	1	0	1	1	1	1
8*	1	1	1	1	1	1	1

Tabulka 4.3

Modely určují ty řádky tabulky, v nichž je formule interpretována jako true (zde označené hvězdičkou).

Např. model m_4 odpovídající čtvrtému řádku tabulky má strukturu danou těmito valuacemi výrokových proměnných : $v(p) = \text{false}$, $v(q) = \text{true}$, $v(r) = \text{true}$, což je možno též vyjádřit trojicí $p, \neg q, \neg r$.

„Formule A^D , jejíž interpretační tabulka byla vytvořena z interpretační tabulky formule A vzájemnou záměnou všech jejích pravdivostních hodnot, je formulí duální k formuli A.“[6] Znamená to tedy, že duální formule k formuli K je formule, která má pravdivostní hodnotu přesně opačnou, než právě formule K. Formulí duální značíme horním indexem písmena D – K^D .

Pokud lze formuli K vyvodit z formule L, nebo naopak pokud z dané formule K je vyvozena formule L, jedná se o formule ekvivalentní. V logice zaujímá ekvivalence důležité postavení a je jí věnována velká pozornost. „S využitím ekvivalentních formulí lze v dané formuli přepsat logická spojení určitými spojkami na logická spojení jinými spojkami.“

Komutativnost	$(K \& L)$	$(L \& K)$
	$(K \vee L)$	$(L \vee K)$
Asociativnost	$(K \& L) \& M$	$K \& (L \& M)$

	$(K \vee L) \vee M$	$K \vee (L \vee M)$
Distributivnost	$(K \& L) \vee M$	$(K \vee M) \& (L \vee M)$
	$(K \vee L) \& M$	$(K \& L) \vee (L \& M)$
Idempotence	$(K \& K)$	K
	$(K \vee K)$	K
De Morganova pravidla	$\neg(K \& L)$	$(\neg K \vee \neg L)$
	$\neg(K \vee L)$	$(\neg K \& \neg L)$

Tabulka 4.4– Ekvivalence

4.3 Quineův algoritmus a rozhodnutelnost

V lekci 3. byly zavedeny pojmy splnitelné formule, tautologie a kontradikce. Zde bude soubor základních pojmů doplněn o další pojmy a diskutovány jejich vzájemné souvislosti, které sehrávají významné role při rozhodování splnitelnosti a platnosti formulí.

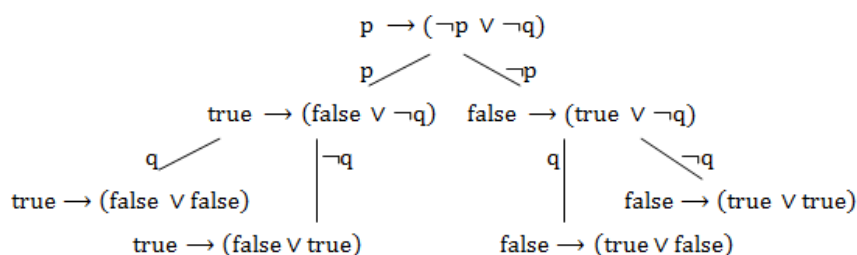
Všechny atomické formule obsahující pouze výrokové proměnné jsou evidentně splnitelné. Atomická formule tvořená logickou konstantou true je tautologie, zatímco false je kontradikce neboli nespjitelná atomická formule. Kontradikce jsou nespjitelné neboli *nekonsistentní* formule. Logicky platné formule (tautologie) jsou zároveň formulemi splnitelnými neboli *konsistentními*.



Formule A jazyka L výrokové logiky je logicky platná (tautologická), právě když je $\neg A$ nespjitelná.

4.3.1 Sémantický strom a Quineův algoritmus

Sémantický strom je jednoduchou a vhodnou procedurou pro zjišťování jak splnitelnosti formule, tak její logické platnosti, jelikož snadno, téměř pouze, pomocí zákonů Booleovy algebry dokážeme sestrojít disjunktivní normální formu formule tak, že zavádíme větvení stromu, kteréž můžeme chápat jako spojku nebo pouze při ohodnocení každé z proměnných hodnotou true či false.



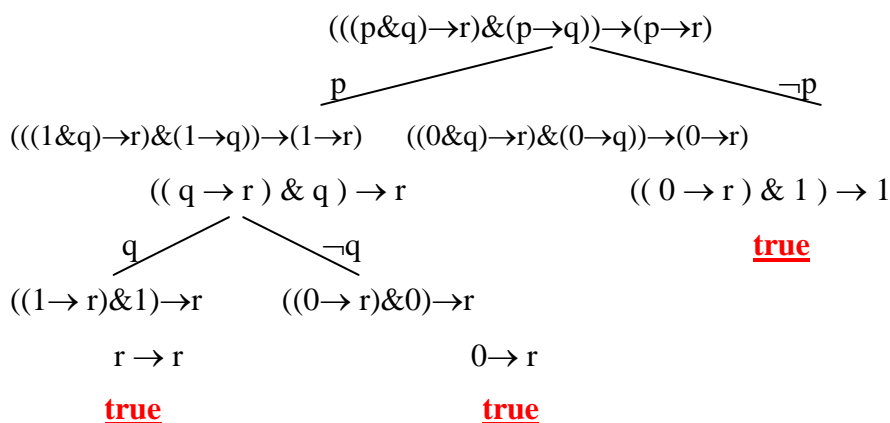
Je vidět, že i jednoduchá formule s dvěma proměnnými obnáší relativně rozvětvený strom. S narůstajícím počtem proměnných se strom může značně zkomplikovat a zpětné dohledávání jejich pravdivostí může být mnohdy nesnadné, což je mírná nevýhoda této procedury a programové řešení může obsahovat spoustu vnořených podmínek či cyklů. Existuje alternativní metoda, jež může podstatně zjednodušit celý výpočet tak, že u „meziformule“, u které bez ohledu na ohodnocení zbytku proměnných, dostaneme stejnou pravdivostní hodnotu, neprovádíme dále větvení, jelikož je to pro nás již redundantní výpočet. Zbytek proměnných může nabývat kteroukoliv z pravdivostních hodnot a při určování disjunktivní normální formu je můžeme buď vypustit, nebo u úplné DNF do ni zahrneme obě ohodnocení. Taková formule může symbolicky vypadat například takto: $(\dots(\dots(\dots)\dots)\dots)\rightarrow\text{true}$.

Při určování logického důsledku, neboli závěru z hypotéz, převedeme toto dedukční schéma do ekvivalentní podoby výrokové formule a kontrolujeme, zda-li nedochází v každé větvi ke sporu. Jestliže tomu tak není, pak je důsledek platný. Mnohdy je vhodné při vyšetřování logického důkazu či obecně platnosti formule užít spíše nepřímého důkazu.

Tento triviální algoritmus rozhodování splnitelnosti formule by, podobně jako je tomu u pravdivostní tabulky, zkoušel všechna možná ohodnocení, tj. prohledával po řadě větve sémantického stromu až do nalezení té, která vede k výsledku true. Při rozhodování platnosti formule by dokonce bylo třeba projít všechny možné větve. Všechny zmíněné nevýhody sémantického stromu eliminuje právě **Quineův algoritmus** částečné interpretace formule (viz následující příklad). Sledování větve sémantického stromu vždy končí tam, kde pokračování v průchodu větvi již nevede ke změně výsledné pravdivostní hodnoty. Některé větve úplného sémantického stromu tak zůstávají nedokončeny a je tedy na místě hovořit o *neúplném sémantickém stromu* formule. [1], [2], [3].

Dokažte pomocí Quineova algoritmu logickou platnost formule

$$(((p \& q) \rightarrow r) \& (p \rightarrow q)) \rightarrow (p \rightarrow r).$$



Formule $(((p \ \& \ q) \rightarrow r) \ \& \ (p \rightarrow q)) \rightarrow (p \rightarrow r)$ obsahuje tři výrokové proměnné, jejichž uspořádání v sémantickém stromě není podstatné. Protože je zde zvoleno pořadí abecední, bude první hladina větvení obsahovat literály p a $\neg p$, druhá hladina literály q a $\neg q$ a třetí hladina literály r a $\neg r$ viz. schéma výše, ale nepsaným pravidlem bývá, že se nejprve ohodnocuje nejčastěji vyskytující se proměnná.



Korespondenční úkol

Zjistěte, zdali existuje model daných formulí

- $A = (a \rightarrow c) \rightarrow ((b \rightarrow d) \rightarrow ((a \vee b) \rightarrow c))$
- $B = (a \rightarrow b) \rightarrow ((b \rightarrow \neg c) \rightarrow \neg a)$
- $C = (\neg a \vee ((b \rightarrow a) \ \& \ c)) \rightarrow \neg b$



Kontrolní otázky:

1. Jaká je negace splnitelné formule?
2. K čemu je valuace?
3. Jak je postup k získání výsledné pravdivostní hodnoty dané formule A pomocí tabulkové metody?
4. Jaký je rozdíl mezi sémantickým stromem a Quinovým algoritmem?
5. Kdy je formule logicky platná?



Shrnutí:

Interpretace výrokové formule A spočívá v přiřazení jedné z pravdivostních hodnot true/false formuli A tímto postupem :

1. *valuace všech výrokových proměnných formule A*
 2. *interpretace formule A postupně z jednotlivých podformulí pomocí interpretačních pravidel v pořadí, jak byly podle pravidel syntaxe vytvářeny.*
- Sémantická analýza výrokové formule spočívá ve stanovení pravdivostních hodnot interpretace formule pro všechny možné valuace proměnných jazyka L. S využitím ekvivalentních formulí lze v dané formuli přepsat logická spojení určitými spojkami na logická spojení jinými spojkami. Množiny spojek, kterými lze přepsat všechny ostatní spojky, jsou funkčně úplnými množinami spojek. Metoda Quinova algoritmu je metoda ohodnoceného sémantického binárního stromu, sledování větve sémantického stromu vždy končí tam, kde pokračování v průchodu větví již nevede ke změně výsledné pravdivostní hodnoty. Každá větev se ohodnocuje pozitivní a negativní částí komplementárního páru.*

5 NORMÁLNÍ FORMY VÝROKOVÝCH FORMULÍ

V této kapitole se dozvíte:

- Co jsou normální formy formulí
- Jaký je rozdíl mezi disjunktivní a konjunktivní formou.

Po jejím prostudování byste měli být schopni:

- Pomocí tabulkové metody určit disjunktivní formu formule
- Pomocí tabulkové metody určit konjunktivní formu formule

Klíčová slova této kapitoly:

Disjunktivní, konjunktivní, úplná normální forma.

Doba potřebná ke studiu: 1 hodina

Průvodce studiem

V této lekci, jejíž prostudování by vám mělo trvat zhruba 1 h, se naučíte pracovat s takovými ekvivalencemi výrokových formulí, které dávají možnost převedení formulí do konjunktivních a disjunktivních normálních forem, na které lze pak snadno aplikovat řadu algoritmů rozhodování platnosti, event. splnitelnosti formulí, uvedených dále v následující lekci.



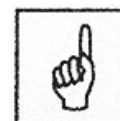
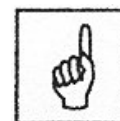
Normální formy slouží k úpravě formulí do určitého standardního tvaru. V tomto tvaru se vyskytují pouze negace, konjunkce a disjunkce (tvořící *funkčně úplnou množinu*). Do takového tvaru lze převést libovolnou formuli. Pravdivostní hodnota formulí se v závislosti na formě nemění, formule jsou stále ekvivalentní.

5.1 Disjunktivní normální formy formulí

Literálem je výroková proměnná p vyskytující se ve formuli nebo její negace $\neg p$. Dvouprvková množina $\{p, \neg p\}$ přitom tvoří **komplementární párliterálů** výrokové proměnné p .

Pro speciální účely je vhodné transformovat výrokové formule do speciálních normálních tvarů. Je-li účelné vystihnout ty valuace výrokových proměnných, které vedou k interpretaci formule jako true, je pro tento účel vhodnou formou ekvivalentní dané formuli její *disjunktivní normální forma*.

Disjunktivní normální forma (DNF) spočívá ve formuli, která je disjunkcí několika konjunktí. Existuje také tzv. úplná normální forma (zde úplná disjunktivní normální forma UDNF), ve které mají členy „shodný rozměr“ [9] – obsahují všechny své proměnné, přičemž ve členu je proměnná zastoupena vždy jen jednou (nevyskytuje se komplementární pár k & $\neg k$).



Výroková formule je v **disjunktivní normální formě** (DNF) formule je-li tvořena disjunkcí konečně mnoha disjunktů, z nichž každý je ve formě konjunktivní klauzule,

$$[(l_1 \wedge l_2 \wedge \dots) \vee (l'_1 \wedge l_2) \vee (l'_1 \wedge l'_2 \wedge \dots) \vee \dots]$$

Je třeba poznamenat, že bez ohledu na ohodnocení výrokových proměnných formule, existuje ke každé formuli její ekvivalentní formule v podobě disjunktivní/konjunktivní normální formy [5].

Při konstrukci tabulkové metody v podstatě vytváříme disjunktivní normální formu formule, jelikož striktně definujeme, jaké konkrétní ohodnocení určité proměnné může figurovat v daném konjunktě. Pro nás je podstatný ten fakt, že ze všech možných ohodnocení výrokových proměnných náležících určitému konjunktě (n-tici proměnných na daném řádku tabulky), vybereme ty, jejichž ohodnocení je true (1), jelikož to znamená, že mohou v dané podobě zastávat určitou podformuli původního tvaru formule svým ohodnocením, kteréž s ní má vlastně ekvivalentní. Pokud je formule kontradikcí, neexistuje vlastně žádné ohodnocení jejích proměnných takové, že by byl kterýkoliv konjunkt interpretován jako true (1). Implicitně z tohoto tvrzení vyplývá, že formule je ekvivalentní s formulí false (0), jež tedy také charakterizuje disjunktivní normální formu formule. Když si na chvíli vypůjčíme z Booleovy algebry zákon o vyloučení třetího členu, a to $p \wedge \neg p = \text{false}$, můžeme na principu této myšlenky postavit spor v konjunktivní klauzuli. Mějme tedy dvojici komplementárních literálů l a l' tak, že: $(l \wedge l' \wedge \dots)$, daná klauzule je zřejmě false. Pokud analogicky zkonstruujeme 0 všechny konjunktory tabulky, z nichž bude každý interpretován jako false, spojíme-li je disjunkcí, pak můžeme hovořit o kontradikci s ekvivalentní disjunktivní normální formou false. Z vlastností binárních spojok \wedge a \vee , Demorganových zákonů a prosté negace snadno zkonstruujeme ekvivalentní konjunktivní normální formu. Pokud tak učiníme, dostaneme tautologii formule. Naše úvahy můžeme formalizovat:

- (1) Formule je kontradikce právě tehdy, jestliže v každé konjunktivní klauzuli disjunktivní normální formy obsahuje komplementární pár literálů.
- (2) Formule je tautologie právě tehdy, jestliže v každé disjunktivní klauzuli konjunktivní normální formy obsahuje komplementární pár literálů.

Sestrojte úplnou disjunktivní normální formu formule

$$a \ \& \ (b \ \vee \ \neg(\neg c \ \rightarrow \ a))$$



			X	Y	
a	b	c	$\neg c \rightarrow a$	$b \vee \neg X$	$a \ \& \ Y$
0	0	0	0	1	0
0	0	1	1	0	0
0	1	0	0	1	0
0	1	1	1	1	0
1	0	0	1	0	0

1	0	1	1	0	0
1	1	0	1	1	1*
1	1	1	1	1	1*

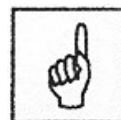
Tabulka5.1: UDNF

Daná formule je ekvivalentní formulí $(a \ \& \ b \ \& \neg c) \vee (a \ \& \ b \ \& \ c)$, která představuje její úplnou disjunktivní normální formu.

5.2 Konjunktivní normální formy formulí

Pro umělou inteligenci mají mimořádný význam formule normalizované do svých konjunktivních normálních forem.

Naopak konjunktivní normální forma (KNF) je konjunkcí několika disjunktí. Taktéž úplná konjunktivní normální forma obsahuje všechny zastoupené proměnné s podmínkou absence komplementárních párů.



Příklad

Sestrojte úplnou konjunktivní normální formu formule

$$(\neg a \rightarrow c) \rightarrow (b \ \& \ (\neg c \rightarrow a)).$$



			X	Y	Z		
a	b	c	$\neg a \rightarrow c$	$\neg c \rightarrow a$	$b \ \& \ Y$	$X \rightarrow Z$	$\neg(X \rightarrow Z)$
0	0	0	0	0	0	1	0
0	0	1	1	1	0	0	1*
0	1	0	0	0	0	1	0
0	1	1	1	1	1	1	0
1	0	0	1	1	0	0	1*
1	0	1	1	1	0	0	1*
1	1	0	1	1	1	1	0
1	1	1	1	1	1	1	0

Tabulka 5.2: UKNF

Podle výše uvedeného postupu je prvním krokem tabulkové metody vytváření úplné konjunktivní normální formy, stejně jako tomu bylo v případě úplné disjunktivní normální formy, její sémantická analýza pomocí pravdivostní tabulky (tab. 4).

Formule

$$(\neg a \ \& \ \neg b \ \& \ c) \vee (a \ \& \ \neg b \ \& \ \neg c) \vee (a \ \& \ \neg b \ \& \ c)$$

je úplnou disjunktivní normální formou negace původní formule. Další negací a s využitím De Morganova pravidla lze pak tuto formuli převést do úplné konjunktivní normální formy formule původní:

$$(a \vee b \vee \neg c) \ \& \ (\neg a \vee b \vee c) \ \& \ (\neg a \vee b \vee \neg c).$$

Kontrolní otázky:

1. Jak vzniká UDNF?
2. Jaký je vztah mezi UDNF a UKNF?
3. Jak vzniká UKNF a jaký je rozdíl s UDNF?



Shrnutí:

Výroková formule je v disjunktivní normální formě, je-li disjunkcí podformulí (disjunktů), z nichž každá je konjunkcí konečně mnoha literálů výrokových proměnných. Výroková formule je v konjunktivní normální formě, resp. V klauzulární formě, je-li konjunkcí podformulí (konjunktů), z nichž každá je disjunkcí konečně mnoha literálů výrokových proměnných. Každou výrokovou formuli lze převést do ekvivalentní disjunktivní (konjunktivní) normální formy.

6 TABLOVÉ A REZOLUČNÍ NEPŘÍMÉ DŮKAZY VE VÝROKOVÉ LOGICE

V této kapitole se dozvíte:

- Jak pracují tablové důkazy.
- K čemu se tablo obecně využívá.
- Co je alfa a beta pravidlo.
- Jaké jsou základní charakteristické vlastnosti.

Po jejím prostudování byste měli být schopni:

- Zjistit charakter výrokové formule pomocí tablových důkazů.
- Znat možnosti historických přehledů tvorby sémantických tabel.
- Aplikovat alfa a beta pravidla.

Klíčová slova této kapitoly:

Alfa, beta, sémantické tablo, Bethovo tablo, Hintikkovo sémantické tablo.

Doba potřebná ke studiu: 1,5 hodiny

Průvodce studiem

V této lekci, jejíž prostudování by vám mělo trvat zhruba 1,5 h, se budete věnovat nejznámějším metodám rozhodování logické platnosti formulí na základě procedury popření. Ověříte si vlastní zkušeností, že sémantické tablo, reprezentující v podstatě postupnou úpravu formule do její disjunktivní normální formy pomocí konečného binárního stromu, skýtá možnost úplné formalizace postupu rozhodování a je proto jedním z nejperspektivnějších algoritmů pro účely automatického rozhodování a odvozování. Rovněž rezoluční algoritmus rozhodování nesplnitelnosti negace formule převedené do její konjunktivní (klauzulární) normální formy skýtá, jak se na příkladech přesvědčíte, možnost úplného odpoutání formálně definovaného postupu od jeho významu a patří proto k základním logickým principům umělé inteligence. Nedílnou součástí je i popis historických technik.



6.1 Historické podoby sémantických tabel

Sémantické tablo je obecně procedura, prostřednictvím které určíme, za pomoci převodu formule do disjunktivní normální formy, zda je formule splnitelná, nebo logicky platná. Tablové důkazy slouží k určení splnitelnosti, nebo zejména logické platnosti dané formule nebo k zjišťování logické dedukce.

6.1.1 Bethovo sémantické tablo

Historie této metody sahá někde do počátku 20. století, kde se o první formu nějaké podoby tabla pokusil belgický matematik Evert Willem Beth [5]. Jeho sémantické tablo sloužilo pro určení tautologie formule, ovšem s

předpokladem, že formule je kontradikce. Tato metoda byla ovšem příliš těžkopádná a složitá.



Nepravdivá	
Pravdivá	Nepravdivá
(1) $(p \Rightarrow q) \wedge (p \Rightarrow \neg q)$	(2)
(3) p	
(4) $(p \Rightarrow q)$	
(5) $(p \Rightarrow \neg q)$	
Pravdivá	Nepravdivá
(6) Q	(7) p
Pravdivá	Nepravdivá
(8)	(10) p

Tabulka 6.1: Bethovo sémantické tablo

Formuli tedy podle implikace rozdělíme na dvě části, pravdivou a nepravdivou. Obě strany implikace samozřejmě musí platit zároveň. Pravdivá část má centrální spojku konjunkci, což znamená, že levá i pravá strana musí

být pravdivá, jelikož jsme v pravdivé větvi. Přeneseme také do pravdivé

větve, kde jej transformujeme na p , protože pokud je $\neg p$ false, musí být p zákonitě true. Formule (5) je v pravdivé větvi, a proto pro její interpretaci true musí platit, že q je true, jelikož p je také true: $(true \Rightarrow q)$. Formule (6) je

pravdivá právě když je pravdivé i $\neg p$, protože p je true. Zde dochází ke sporu u (7) a (9). Jelikož jsme v pravdivé větvi, může být pravdivá i konstelace: $F(p)$

$\Rightarrow T(\neg p)$, ale to není možné, jelikož zde opět nastává spor u (4) a (10).

Je tedy opravdu zřejmé, že takovou metodou bychom se při větší četnosti proměnných a délce formule mohli snadno dostat do bezvýchodné situace. Avšak již zde dokázal Beth najít elementární souvislost mezi sémantickým tablem a Gentzenovou přirozenou dedukcí (viz dále).

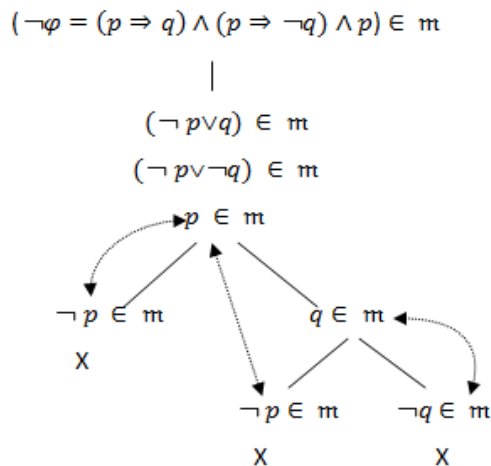
6.1.2 Hintikkovo sémantické tablo

Filozof a logik J. Hintikka se snažil na základě množinového modelu sestrojít sémantické tablo formule s nějakou interpretací, pro kterou sestrojíme množinu formulí, jež bude pravdivá v rámci modelu \mathfrak{M} formule: $m = \{\varphi; \mathfrak{M}(\varphi) = true\}$.

Předpokládejme stejnou formuli jako v prvním případě: $\varphi = (p \Rightarrow q) \wedge (p \Rightarrow \neg q) \Rightarrow \neg p$, tedy

$(\neg \varphi = (p \Rightarrow q) \wedge (p \Rightarrow \neg q) \wedge p) \in m$.

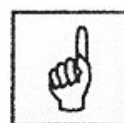
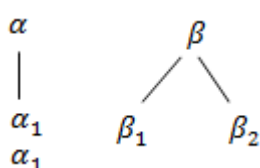
Tato formule tedy patří do množiny m . Centrální spojka \wedge tedy spojuje pravdivé výrazy, a proto:



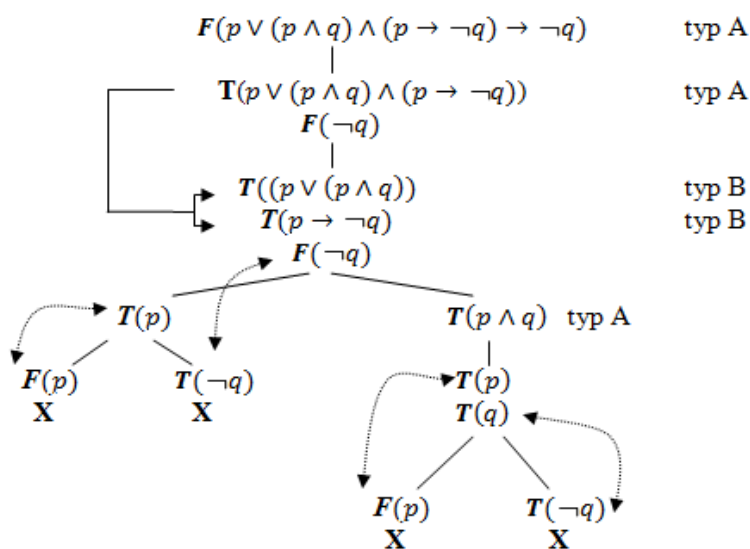
Jak je vidět, Hintikkův model poskytuje přehlednější schéma, jež zjednodušuje práci s podformulemi a snadno dokáže odhalit spor, ovšem i přesto je to pořád značně těžkopádná metoda, u které si uživatel musí kontrolovat podformule náležící modelu.

6.1.3 Smullyanovo sémantické tablo

Smullyan podstatně vylepšil a zjednodušil schéma a pravidla pro konstruování sémantických tabel. Vyšel z předpokladu, že každá formule může být interpretována disjunktivně či konjunktivně. Vytvořil tedy sadu pravidel pro formule typu A a typu B, které charakterizují buď nevětvení formule, kdy se podformule píšou pod sebe, nebo větvení, potom tablo vygeneruje dvě nové větve, ohodnocené dvěma podformulemi.



Tato metoda byla již na podstatně lepší úrovni než předchozí dvě a také konstrukčně velice podobná nynějším koncepcím sémantických tabel, ovšem s rozdílem, že stále se při řešení používala negace formule, a to kvůli získání tautologie formule díky jedinému možnému ohodnocení stavu false u implikace.



Vidíme, že konstrukce tabla začíná být smysluplná a díky exaktně deklarovaným pravidlům pro výstupy formulí typu A i B se situace značně zjednodušuje. Jenže stále se hovoří o tautologii formule, čili nespécifikujeme výstup pro pouze splnitelnou formuli [17].

6.2 Základní myšlenka sémantického tabla

Princip a základní pravidla tvorby sémantického tabla, ať už ve výrokové či predikátové logice, jsou obdobná pro všechny koncepce, ovšem s nepatrnými rozdíly, jež mohou hrát při programových realizacích klíčovou roli. Společnými rysy jsou samozřejmě základní myšlenky principu a užití tabla. Obecně můžeme konstatovat, že sémantické tablo slouží jako procedura v logice k ověřování splnitelnosti a logické platnosti formulí, samozřejmě tedy i k ověřování logického důsledku z množiny formulí, či konsistence teorie, za pomoci převodu formule do její disjunktivní normální formy.



6.2.1 Charakteristické rysy sémantického tabla

Přetváření původní formule do jejího disjunktivního normálního tvaru je proces, který obnáší konečně mnoho operací, přičemž je tento proces chronologický a využívá vhodných ekvivalencí původní formule takových, aby listy daného stromu obsahovaly seznam literálů výrokových proměnných, případně predikátů. Sémantické tablo bývá znázorňováno pomocí binárního stromu a to tak, že původní formule je kořenem stromu, listy stromu obsahují seznamy literálů výrokových proměnných, případně predikátů. Na každý uzel, který není listem, je tedy možné aplikovat operaci, jež dokáže převést formuli daného uzlu na ekvivalentní formuli podle určitých pravidel [17].

6.2.2 Tablová pravidla

Listy sémantického tabla jsou tedy ohodnoceny jakýmsi seznamem atomických formulí, jež platí zároveň. Vyjádření tohoto faktu pomocí již zmíněného seznamu je vlastně metajazyková záležitost. Formálně je nutností zavést pravidla pro předbod formule na tyto seznamy. Ve výrokové logice se jedná o **α -pravidla** a **β -pravidla**, v predikátové platí taktéž plus dvě nová: **γ -pravidlo** a **δ -pravidlo**.

α - pravidla :

α	α_1	α_2
$\neg \neg A$	A	
$A_1 \& A_2$	A_1	A_2
$\neg (A_1 \vee A_2)$	$\neg A_1$	$\neg A_2$
$\neg (A_1 \rightarrow A_2)$	A_1	$\neg A_2$
$\neg (A_1 \leftarrow A_2)$	$\neg A_1$	A_2
$A_1 \leftrightarrow A_2$	$A_1 \rightarrow A_2$	$A_1 \leftarrow A_2$



Tabulka 6.1- tablová alfa pravidla



β - pravidla :

β	β_1	β_2
$B_1 \vee B_2$	B_1	B_2
$\neg (B_1 \& B_2)$	$\neg B_1$	$\neg B_2$
$B_1 \rightarrow B_2$	$\neg B_1$	B_2
$B_1 \leftarrow B_2$	B_1	$\neg B_2$
$\neg (B_1 \leftrightarrow B_2)$	$\neg (B_1 \rightarrow B_2)$	$\neg (B_1 \leftarrow B_2)$

Tabulka 6.2- tablová beta pravidla

Pokud narazíme na formuli **typu α** , sémantické tablo vygeneruje jednu větev jako jejího následovníka a to seznam formulí α_1, α_2 . Pokud narazíme na formuli **typu β** , sémantické tablo vygeneruje dvě větve, z nichž jedna bude ohodnocena formulí β_1 a druhá β_2

6.2.3 Splnitelnost a platnost v rámci sémantického tabla

Sémantické tablo dekomponujeme tak dlouho, až je každý list binárního stromu ohodnocen atomickou formulí, či jejím seznam, jenž charakterizuje konjunktivní klauzuli. Seznamy atomických formulí v každé větvi spojíme disjunkcí a dostaneme disjunktivní normální formu formule. Pokud dochází ke sporu ve větvi (např. $\{p, \neg p, q\}$ či $\{p(x), \neg p(x), q(x)\}$). Tyto větve jsou potom ve výsledné disjunktivní normální formě vynechány. Pokud existují pouze větve, ve kterých dochází ke kontradikčnosti, potom je tablo sporné, hovoříme **onesplnitelné formula** tablo se nazývá **uzavřené (X)**. V opačném případě, což znamená, že existuje aspoň jedna nesporná větev (otevřená), tablo se nazývá **otevřené**, přičemž pokud jsou všechny větve bezesporné, formule je **logicky platná (tautologie)**.



Teorie $\mathbb{T} = (\varphi_1, \varphi_2, \dots, \varphi_n)$ je konzistentní, pokud sémantické tablo $\mathbb{T}(\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n)$ je otevřené.

Teorie $\mathbb{T} = (\varphi_1, \varphi_2, \dots, \varphi_n)$ je nekonzistentní, pokud sémantické tablo $\mathbb{T}(\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n)$ je uzavřené.

Při ověřování logického důsledku z teorie je opět nutností převést dedukční schéma na formuli výrokové či predikátové logiky: $\varphi_1, \varphi_2, \dots, \varphi_n$ převedeme na

$\mathbb{T} = \mathbb{T}(\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n) \rightarrow \omega$. Výhodnější je v některých případech použití nepřímého důkazu negace formule, jež v našem případě transformuje formuli do konjunktivní normální formy a pokud je false, formule je tautologií. Formule φ je logickým důsledkem množiny formulí $T = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$, $\varphi_n \models \varphi$ pouze v případě, že sémantické tablo \mathbb{T} formule $\mathbb{T}(\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n \wedge \neg \varphi)$ je uzavřené; $\varphi_{DNF} \rightarrow false$.

[17].

6.3 Sémantické tablo ve výrokové logice

Charakteristickým rysem této tablové metody je její postupné převádění do DNF takové, že je netřeba si pamatovat předcházející kroky, jelikož na každé úrovni existuje ekvivalentní formule s formulí předcházející. Tuto metodu demonstrujeme pomocí stromové struktury tak, že pozici kořenu obsadíme původní formulí a na každé další úrovni stromu vytváříme ekvivalentní formuli s formulí původní prostřednictvím všech uzlů dané úrovně, jelikož mají mezi sebou disjunktivní vztah. Každý list stromu je seznamem literálů výrokových proměnných. Takovým seznamem jsou vlastně zároveň platící literály, u nichž určíme, jestli jsou sporné či ne.

6.3.1 Sémantické tablo jako binární strom

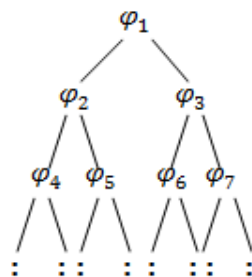
Tím, že vytváříme tablo, konstruujeme de facto syntaktický formační strom od kořene po listy. Takový strom je znázorněn pomocí binárního stromu, který může mít samozřejmě spoustu podob. Pokud vezmeme v úvahu plný binární strom, což by prakticky znamenalo, že se disjunktivní normální forma vytvořila pouze pomocí beta pravidel, obsahoval by strom



uzlů. Pomocí součtu geometrické

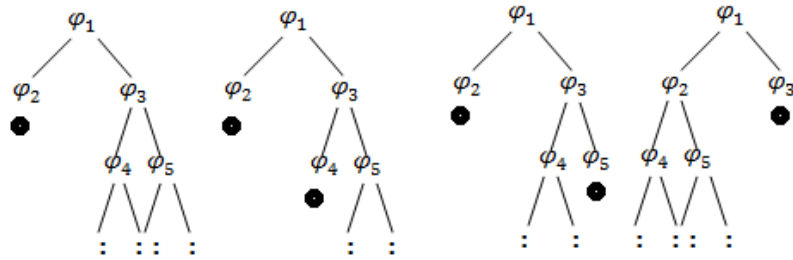
$$2^{n+1} - 1 \quad [3]$$

posloupnosti můžeme tento fakt zapsat jako



Takový plný binární strom by měl formální zápis ekvivalencí jednotlivých úrovní $\varphi_1 \Leftrightarrow \dots \Leftrightarrow (\varphi_{2^i} \vee \dots \vee \varphi_{2^{i+1}-1})$, kde $i = 1, 2, 3, \dots$, přičemž kořen stromu, jakožto základní formuli, charakterizuje nultá úroveň ($i = 0$).

Úskalí takového zápisu je ovšem v kompletnosti binárního stromu. U každého uzlu může nastat více případů. Kromě větvení pomocí beta pravidla, je možno také prodlužovat strom prostřednictvím alfa pravidla a v neposlední řadě může být určitý uzel listem, což znamená, že obsahuje již pouze seznam elementárních literálů či predikátů, ať už sporný či ne. Na diagram níže můžeme vidět různé druhy binárních stromů a některé příklady ohodnocení, což znamená list, nebo alfa pravidlo.



Matematizace je v tomto případě obtížná. Když vezmeme v úvahu pouze beta pravidlo větvení, tak na každé i -té úrovni stromu může být 0 až 2^i

$$2^i$$

dvojic uzlů a 2^i beta uzlů, přičemž základní formule je považována za nultou úroveň ($i = 0$). V celém stromu tedy může být maximálně

$2^{n+1} - 1$ uzlů, kde n je počet úrovní, jelikož beta pravidlo může vygenerovat největší počet větví, a to dvě. Pokud by například již v první úrovni u jedné z větví nastal případ, kdy by byla ohodnocena již výsledným seznamem literálů, složitost stromu by se značně zjednodušila, jelikož je netřeba operovat ve zbývajícím výpočtu se seznamem, který již zůstane konstantní. Konkrétně by se ve stromu s n úrovněmi vyskytovalo $2^{n+1} - 1 - (2^n - 2)$ uzlů. Kdybychom chtěli strom specifikovat pro konkrétní počet listů pro různé úrovně, museli bychom vyjít z úvahy, že existuje základní formule, která se v první úrovni může větvit na dva uzly (bereme v úvahu pouze beta pravidlo). V první úrovni může nastat případ 0, 1 a 2 listů, což se dá charakterizovat předchozí formalizací a parametrem t , který bude vyjadřovat patřičný počet listů:

$$2^{n+1} - 1 - t_1(2^n - 2)$$

Pokud postoupíme o úroveň dále, zde se může vyskytovat už 8 listů, ovšem v závislosti na parametru t_1 z předchozí úrovně. Počet uzlů na druhé úrovni označme jako parametr t_2 . Ovšem je nutné specifikovat situaci tak, že pokud bude $t_1 = 0, t_2 = 1, 2, \dots, 8$.

Jestliže

$$t_1 = 1, t_2 = 1, 2, 3, 4. \text{ Když } t_1 = 2, t_2 = 1, 2.$$

Potom můžeme formalizovat počty uzlů takto:

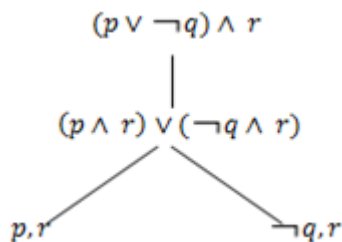
$$2^{n+1} - 1 - t_1(2^n - 2) - t_2(2^{n-1} - 2)$$



Případ lze zobecnit následovně:

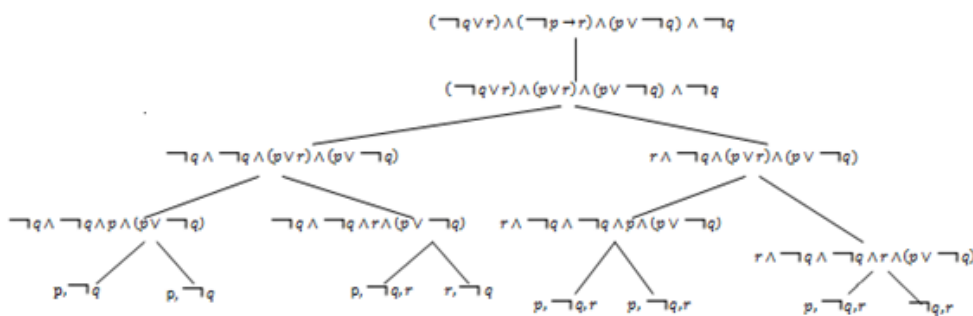
$$\begin{aligned} & \llbracket (2) \rrbracket^{\uparrow(n+1)-1} - \llbracket t_1 \rrbracket (2) \uparrow n - 2 - \llbracket t_2 (2)^{n-1} - 2 \rrbracket - \dots \\ & - \llbracket t_k (2)^{n-k-1} - 2 \rrbracket \end{aligned}$$

Je ovšem nutné patřičně specifikovat konstelaci počtu listů v jednotlivých úrovních, což je náročný výpočet, jež zasahuje mimo definiční obor této práce. Jelikož v každém uzlu může nastat případ ohodnocení trojicí specifikací, a to beta pravidlo, alfa pravidlo, list, formalizace by byla ještě náročnější. Pokud budeme uvažovat z hlediska programového řešení sémantického tabla se vstupem v podobě formule, předdefinovaných pravidel a



kroků sestrojíme sémantické tablo tak, že nejspíš budeme provádět každou změnu na jedné úrovni. Nastíníme korektní tvorby table jednoduché formule. Nějak podobně by mohl náš algoritmus chápat danou formuli. Jde vidět, že mohou existovat i redundantní výpočny, jelikož v případě této formule je opravdu podle zásobníku závorek konjunkce centrální spojkou. Totiž v případě sestavování sémantického tabla těžíme z empirie i expresivity, proto můžeme původní formuli ihned rozložit na dvojici výsledných listů podle spojky disjunkce.

Pokud je zadání deklarováno jako konjunktivní klauzule, může být docela těžkopádné použití této metody, jelikož vygeneruje spousty uzlů díky velkému počtu aplikací zákonů logiky.



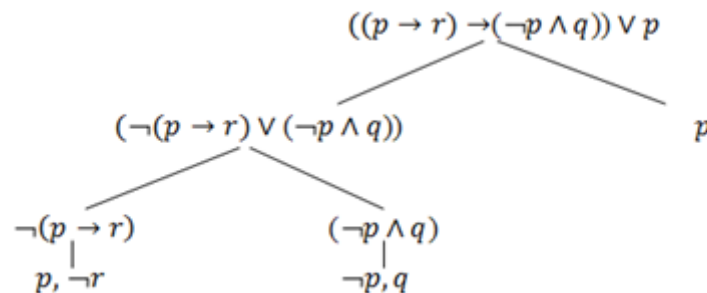
Je samozřejmě docela výhodné vidět, jak se nám formule postupně přetvoří na seznam literalů, jenž je exaktně určen v každém listu, ovšem chybovost zápisu může být v takovém případě poměrně pravděpodobná, stejně jako je přehlednost docela zanedbatelná. V tomto případě máme v zadání “pouze” formuli typu $\varphi = \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4$, přičemž každá z podformulí je ve tvaru beta kromě poslední $\neg q$, ovšem pokud by i tento literál byl formulí typu $\llbracket (\beta)_1 \vee (\beta)_2 \rrbracket$, přibyla by nám další úroveň, jež by obsahovala 16 uzlů a ty by představovaly listy. Takové řešení se může zdát poměrně těžkopádné, pokud si představíme, že s každou přítomnou beta formulí typ $\llbracket (\beta)_1 \vee (\beta)_2 \rrbracket$, kde β_1 i β_2 jsou atomické formule, by nám postupně přibývalo 32,64.... uzlů. Je nejspíš

nevyhnutelné opustit od exaktních pravidel a zkusit seskupovat jednotlivé kroky, které by zjednodušily zápis. Díky ekvivalentním formám jednotlivých formulí je velice pracně dostat se ke kludné formě složité formule. Mějme například formuli: $((p \rightarrow q) \wedge (p \vee (\neg p \vee r))) \wedge (s \vee (q \wedge \neg r))$. Je docela nepředstavitelné vůbec přetvářet takovou formuli podle klasických kroků sémantického tabla. Bylo by nutné učinit nějakou optimalizaci, jelikož v tomto případě samotný algoritmus neučiní tolik kroků jako vedlejší metody úpravy formule [17].

6.3.2 Splnitelnost a logická platnost

Obecně je tablo účinnou metodou k vyšetřování splnitelnosti formule. Přetváření formule pomocí této metody je sice pracné, ovšem všechny listy striktně definují seznamy literálů v každém konjunktivní disjunktivní normální formy. Pomocí této metody je dobře patrná také Hintikkova množina formulí, jelikož sjednocením všech uzlů větve dostaneme množinu splnitelných formulí.

Mějme formuli $\varphi = ((p \rightarrow r) \vee (\neg p \wedge q)) \vee p$. Sestrojme její sémantické tablo a určíme disjunktivní normální formu a Hintikkovu množinu této formule.



$$\varphi_{DNF} = ((p \wedge \neg r) \vee (\neg p \wedge q)) \vee p$$

Hintikkova množina: $\{(\neg p \wedge q), ((p \rightarrow r) \vee (\neg p \wedge q)) \vee p\}$

Jelikož můžeme tvrdit, že seznam literálů v listu je vlastně konjunktem a každý literál je atomickou formulí, není důvod proč každou formuli s centrální spojkou konjunkce nedefinovat jako seznam dvou formulí. Navíc již od začátku můžeme poměrně přesně určit prvky Hintikkovy množiny. Mějme formuli: $\neg(((p \rightarrow q) \wedge (\neg p \vee q) \wedge (\neg q \rightarrow \neg r)) \rightarrow \neg r)$ Sestrojme pomocí tabla její disjunktivní normální formu a Hintikkovu množinu \mathbf{H} :

$$\mathbf{H} = \{p, q, r, p \rightarrow \neg r, p \rightarrow q, \neg q \rightarrow \neg r, \neg(((p \rightarrow q) \wedge (\neg p \vee q) \wedge (\neg q \rightarrow \neg r)) \rightarrow \neg r)\}$$

6.3.3 Teorie a logický důsledek

Teorii můžeme, jak už víme, chápat jako konjunktivní množinu formulí, nebo jako množinu formulí, jež platí zároveň a určují teda nějaké předpoklady. Aby byla platná, musí existovat nějaká konstelace jejích proměnných, nějaký její model. Takový fakt podporuje myšlenku tvorby disjunktivní normální formy, což u



tabla nevyhnutelně nařizuje tvorbu celého stromu, jelikož nestačí nám najít jeden spor. Situace je obdobná jako v předcházejících případech. Například mějme teorii: $T = \{ \}$ a testujme její konsistenci. Tablo již sestrojeno bylo s výsledkem tautologie formule, čili teorie je každopádně konsistentní. Je samozřejmostí, že spor se může vyskytnout kdykoliv, čili například až na úrovni seznamů literálů. Pokud situaci trochu pozměníme:

$T =$

Dokažme, že q vyplývá z teorie T .

Situaci znázorníme, jako:

neboli $\varphi = (\neg q \vee r), (\neg p \rightarrow r), (p \vee \neg q) \rightarrow q$

neboli $\neg\varphi = (\neg q \vee r), (\neg p \rightarrow r), (p \vee \neg q) \wedge \neg q$.

Takovou situaci jsme již řešili, ovšem s tím rozdílem, že nás zajímalo, jestli bude nalezena nesporná disjunktivní forma této formule. Ta nás nyní nebude ani tak zajímat, ovšem potřebujeme zjistit, zdali existuje tautologie formule,

což sporem dokážeme jako nalezení kontradikce. Situace se může podstatně zjednodušit, a to tak, že najdeme list s nesporným seznamem literálů. Tím pádem nemůže kontradikce nastat. V tomto případě jsme přišli v předchozím případě na tautologii u formule $\varphi = (\neg q \vee r), (\neg p \rightarrow r), (p \vee \neg q) \wedge \neg q$. Nyní je ale pro nás situace taková, že

platí $\neg\varphi = (\neg q \vee r), (\neg p \rightarrow r), (p \vee \neg q) \wedge \neg q$, čili jestli je tato negace tautologií, φ je potom kontradikcí.

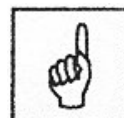
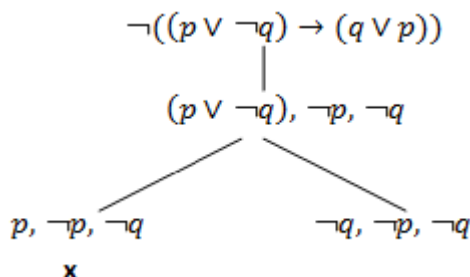
6.3.4 Přirozená dedukce

Jak již víme, pokud je formule logicky platná, její negace je nesporná. Přitom se využívá duality mezi spojkami \wedge, \vee . Aplikujeme negaci formule, znegujeme listy a reverzním Gentzenovským tablem docílíme původní formule. Konstrukčně by mělo být vlastně stejné jako regulární sémantické tablo, ovšem můžou být použity i redundantní úpravy pro lepší orientaci.

Představme si, že vyšetřujeme logický důsledek dané teorie. Požadujeme, aby formule byla kontradikcí, ovšem nestane se tak. Jednu větev nalezneme otevřenou, je možné konstatovat, že logický důsledek nevyplývá z dané teorie, ovšem můžeme se také pokusit modifikovat teorii tak, aby daný logický důsledek byl platným. Optimálním způsobem se může jevit pozměnění jednoho z literal v otevřené větvi tak, aby byl ve sporu s ostatními literály.

Mějme formuli

. Určeme, zda je tautologií.

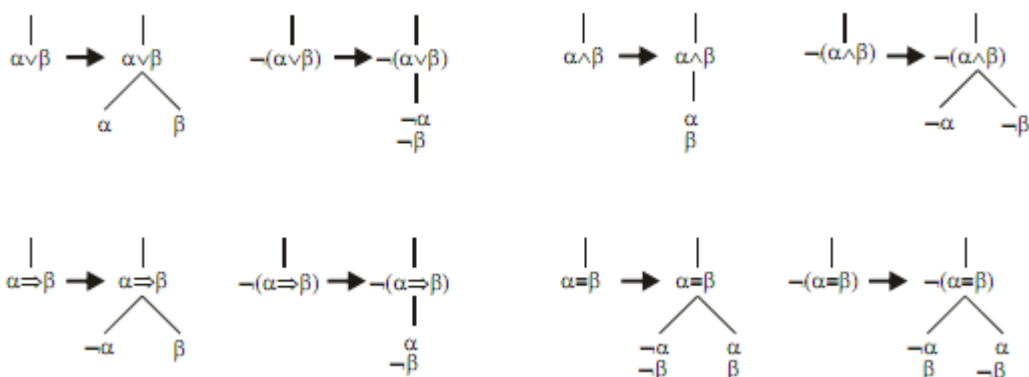
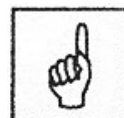


Je vidět, že levá větev je sporná, ovšem pravá ne. Máme zde literal $\neg p$. Abychom získali spor, museli bychom deklarovat i literal p , což by nebylo příliš výhodné vzhledem k tomu, že můžeme ponechat půvosní počet literálů,

víme totiž, že jsme dvakrát dostali $\neg p$, čili jeden z nich změním na p a dostaneme spor, a tedy v tuto chvíli bude formule tautologií. Samozřejmě můžeme listy upravit různým způsobem. Z jedné formule totiž může vyplývat více logických důsledků [17].

6.4 Alternativní přístupy tvorby sémantického tabla s diagramatickými přístupy

Kvasničkova tablová metoda také splňuje poslání obecného paradigmatu sémantických table, a to schématickou stromovou reprezentací splnitelnosti či logické platnosti formule za pomoci jejího převodu do disjunktivní normální formy, charakterizovanu seznamem literál. Tato konstrukce se snaží dosáhnout výsledku co nejrychlejším způsobem pomocí předdefinovaných diagramatických výstupů z alfa či beta formulí a ohodnocováním větví pomocí literal či formulí, jež v každém okamžiku vystupují z těchto formulí:



Obr. Alfa a beta diagramy

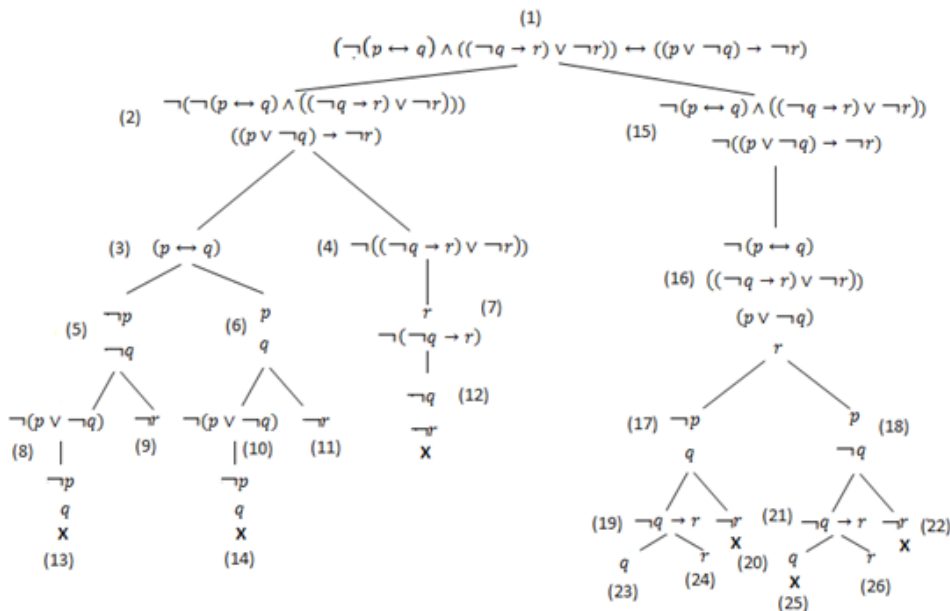
Metoda tedy neuvžívá explicitně De Morganových a jiných logických zákonů, pouze formule rozkládá pomocí již zmíněných předdefinovaných pravidel, zakládajících se na těchto zákonech, což se může v některých případech jevit jako poměrně výhodné.

6.4.1 Splnitelnost a platnost

Díky diagramatickým pravidlům lze touto metodou řešit i velice složité formule.

Mějme formuli: $\varphi: ((p \leftrightarrow q) \wedge ((\neg q \rightarrow r) \vee \neg r)) \leftrightarrow ((p \vee \neg q) \rightarrow \neg r)$.

Vypracujeme sémantické tablo, určíme splnitelnost formule a výslednou DNF.



Popis postupu: Základní formuli (1) rozdělíme podle beta pravidla negace ekvivalence na formule (2) a (16). V uzlu (2) nám vznikne seznam dvou formulí, z nichž vybereme první formuli a rozložíme ji na dvě větve pomocí pravidla negace konjunkce na uzly (3) a (4). Uzel (3) rozdělíme pomocí beta pravidla na dva seznamy literálů (5) a (6). Uzel (4) je negace disjunkce a vygeneruje tedy seznam obnášející dvojici formulí (7), jež je podle alfa pravidla rozložena na dva literály (12) sporné s uzlem (7). Uzly (8) a (9) jsou stejné jako uzly (10) a (11), jsou totiž použity z druhé formule seznamu (2). Literály vystupující podle alfa pravidla negace disjunkce z formule (8) jsou sporné s uzlem (5). Obdobná situace nastane mezi uzly (14) a (6). Uzel (15) je ohodnocen dvojicí alfa formulí, jež vygenerují seznam čtveřice formulí (16). Pro další rozklad vybereme první formuli seznamu, jež je beta pravidlem, a to negací ekvivalence, rozložené na větve (17) a (18). Dvojice uzlů (19), (20) a

(21), (22) jsou již stejné rozložené dvojice formulí vystupující z druhé formule seznamu v uzlu (16), přičemž (20) a (21) vygeneruje spor s atomickou formulí v (16). Literály (21), (24), (25), (26) jsou beta rozklady, z nichž (25) vygeneruje spor s (18).

Formule tedy není logicky platná, jelikož se v tablu vyskytují uzavřené větve, ovšem je splnitelná kvůli přítomnosti otevřených větví. Disjunktivní normální forma formule φ je tedy:

$$\varphi_{DNF} = (\neg p \wedge \neg q \wedge \neg r) \vee (p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge r) \vee (p \wedge \neg q \wedge r).$$

6.4.1 Tablo s vlastností binárního stromu

Podstatný rozdíl mezi touto a předchozí metodou je v používání diagramatického přístupu. Reprezentace binárním stromem není až tak odlišná, jak by se mohlo zdát. Tato metoda je charakterizována prohledáváním stromu do hloubky, jelikož každý konjunkt je charakterizován uzly v cestě od kořene k listu, podobně jako u sémantického stromu. Tímto se uzly značně zjednodušují. Jelikož je smozřejmě počet cest k uzlům stejný jako počet uzlů. Algoritmus se tedy sestává ze dvou částí:

1. **Rychlá konstrukce stromu**, přičemž účelem je dekompozice uzlů a každý literal spojen konjunkcí je zanechán na předchozím uzlu tak, že platí vždy zároveň s následujícími formulemi.

2. **Prohledávání do hloubky**: dalo by se říci, že je nutností formalizovat nějakým způsobem zpětné dohledávání literálů, jelikož při složitých konstrukcích stromu je nasnadě špatné vytvoření disjunktivní normální formy díky menší přehlednosti. Je nutné tedy zvolit algoritmus prohledávání do hloubky.

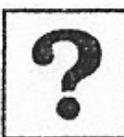
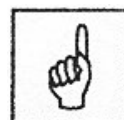
Zatímco u metody prof. Lukasové si postupnými úpravami podle logických zákonů pomalu přetváříme formuli do disjunktivní normální formy, přičemž konstrukce stromu bude vždy stejná, u metody Kvasničky je nutné dávat pozor na ohodnocování alfa formulí či literálů spojených podle alfa pravidla, jelikož nevhodné použití alfa formule může v případě plného binárního stromu místo $2^0 + 2^1 + \dots + 2^i + \dots + 2^n$ vygenerovat uzlů $2^0 + 2^1 + \dots + 2^i + \dots + 2 \cdot 2^n$, protože každý list by by ještě ohodnocen literálem, který platí zároveň se všemi listy. Takový literal by měl být deklarován již na začátku, jelikož s tímto principem, bychom vlastně tuto metodu modifikovaly druhou porovnávanou metodou, což není žádoucí [17].

Dokažte pomocí sémantického tabla logickou platnost formule :

$$\models (a \rightarrow b) \rightarrow ((a \rightarrow \neg b) \rightarrow \neg a)$$

Kontrolní otázky:

1. Jaký je rozdíl mezi přímým a nepřímým důkazem?
2. Proč má smysl využít řádková pravidla a pravidla pro větvení?
3. Pokud naleznou uzavřené tablo formule nepřímým důkazem, jaký je výsledek?
4. Proč větve sémantického tabla uzavře komplementární pár?



Shrnutí:

Rozhodování splnitelnosti výrokové formule pomocí sémantického tabla je založeno na myšlence postupného převodu formule do disjunktivní normální formy pomocí pravidel α a β , která vycházejí z ekvivalencí mezi logickými spojeními. Vyskytne-li se v některém disjunkt, reprezentovaném listem větve tabla, komplementární pár výrokových proměnných, jde o disjunkt nesplnitelný s uzavřenou větví tabla. Jsou-li uzavřeny všechny větve tabla, je formule v jeho kořeni nesplnitelná. Nepřímý tablový důkaz logické platnosti formule spočívá v nalezení uzavřeného sémantického tabla této formule. Rezoluční důkaz probíhá prostřednictvím rezolučního odvozovacího pravidla, aplikovaného na množinu klauzulí vytvořených úpravou formule do klauzulární formy. Nepřímý rezoluční důkaz logické platnosti formule spočívá v úpravě negace formule do klauzulární formy a následných aplikacích rezolučního odvozovacího pravidla, až do odvození sporu – prázdné klauzule.



7 DEDUKCE VE VÝROKOVÉ LOGICE

V této kapitole se dozvíte:

- Co je a jak funguje dedukce ve výrokové logice.
- Jakými metodami lze úlohy dedukce řešit.

Po jejím prostudování byste měli být schopni:

- Určit co jsou hypotézy a závěry a jak spolu souvisejí.
- Vyřešit deduktivní úlohy pomocí tablového důkazu.
- Vyřešit deduktivní úlohy pomocí rezolučního důkazu.

Klíčová slova této kapitoly:

Hypotéza, závěr, dedukce, tablový, rezoluční.

Doba potřebná ke studiu: 1,5 hodina



Průvodce studiem

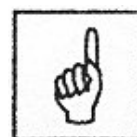
Po prostudování této lekce, které by vám mělo trvat zhruba 1,5 h, si uvědomíte, že i výroková logika je při určité své omezené míře expresivity schopna podchytit deduktivní uvažování vyjádřitelné přirozeným jazykem. Budeme se zde věnovat modelování usuzování neboli dedukci prostředky výrokové logiky z hlediska její sémantiky, tj. významové stránky. Uvidíte, že platnost závěru vyplývajícího z dané množiny předpokladů se vztahuje k určité modelové situaci, charakterizované strukturou pravdivostních hodnot výrokových proměnných zastupujících ve formulích elementární výroky o modelovaném světě. Na řadě příkladů si zde uvědomíte možnosti odvozování, že určitý závěr logicky vyplývá z daných předpokladů. Vedle metody pravdivostních tabulek zde budeme pro prověřování logických důsledků daných předpokladů používat i rezoluční a tablovou metodu vyznačující se tím, že dokazovací postupy jsou v nich stanoveny čistě formálním způsobem.

7.1 Logické důsledky výrokových formulí

Výroková logika by jako model lidského vnímání světa a myšlení o něm neměla valnou cenu, kdyby se v ní nedalo vyjádřit, co z čeho vyplývá. Jde o schopnost, která se v případě skutečného lidského myšlení nazývá schopností *dedukce*. Ve výrokové logice, tak jak byla zavedena v předchozích odstavcích, jsme sémantiku výroků do značné míry odpoutali od jejich skutečného významu tím, že jsme ji redukovali pouze na pravdivostní hodnoty výroků. I když lze problém dedukce formulovat pouze v rámci uvedené modelové zjednodušené sémantiky, hlubšímu porozumění jeho smyslu poslouží krátký návrat k tomu, co bylo řečeno před zavedením hlavních pojmů modelu výrokové logiky. Jazykové útvary reprezentující atomické výroky mají v reálném světě své denotáty tvořící skutečné sémantické universum diskursu výrokové logiky. Pravdivost či nepravdivost uvažovaných výroků v tomto světě může záviset, jak již bylo řečeno, na jeho stavu, resp. na tom, jak je právě

subjektem pojímán. Např. pravdivost výroku o vzájemném protínání se dvou rovnoběžek závisí na tom, zdali jde o výrok Euklidovské nebo jiné geometrie, neboli závisí na tom, k jaké modelové interpretaci se výrok vztahuje. Výroky o modelovaném světě pak klasifikujeme na podmnožinu výroků pravdivých a podmnožinu výroků nepravdivých podle toho, jakou interpretaci máme právě na mysli. Dobře utvořeným formulím jazyka výrokové logiky můžeme přiřazovat různé jeho interpretace přiřazováním jiných světů a s nimi spojených pravdivostních struktur. Ty mohou pocházet už nejen z původně modelovaného reálného světa, ale i jiných světů vytvořených abstrakcí. Správnost závěrů dedukce však musí respektovat právě ty pravdivostní struktury, v nichž platí všechny předpoklady dedukce.[1], [2], [3].

Jak již bylo zmíněno, problém rozhodování, zdali určitá formule A vyplývá z množiny formulí U , se nazývá problém *dedukce* a je jedním ze základních problémů logiky. Ve výrokové logice hovoříme o formuli A , vyplývající z množiny formulí U jako o (tauto)logickém důsledku U . Množina formulí U je v tomto pojetí *speciální množinou předpokladů* (*speciálních axiomů*), na níž je postavena (z jejích logických důsledků) určitá *teorie*.



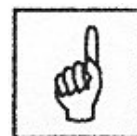
Problém dedukce bývá zpravidla formulován tak, že z množiny *hypotéz* $\{H_1, H_2, \dots, H_n\}$ vyplývá závěr Z . Množina hypotéz $U = \{H_1, H_2, \dots, H_n\}$ tvoří *speciální axiomy* teorie a Z je jejím logickým důsledkem :

$$\{H_1, H_2, \dots, H_n\} \models Z.$$

Nechť H_1, H_2, \dots, H_n jsou výrokové formule. Z je logickým důsledkem množiny formulí $\{H_1, H_2, \dots, H_n\}$, právě když množina formulí

$$\{H_1, H_2, \dots, H_n, \neg Z\}$$

je nespílitelná.



V následujících ukázkách aplikací logického důsledku bude též diskutována otázka splnitelnosti (konsistence) množiny předpokladů a její minimálnosti.[1], [2], [3].

Ukázáno je, že z daných předpokladů vyplývá daný závěr.



Předpoklady: 1. Jan je učitel.

2. Neplatí, že Jan je učitel a zároveň je bohatý.

3. Je-li Jan rockový zpěvák, je bohatý.

Závěr: Jan není rockový zpěvák.

Označíme-li výroky „Jan je učitel.“ - u ,

„Jan je bohatý.“ - b

„Jan je rockový zpěvák.“ - z ,

lze pak množinu předpokladů formálně zapsat jako

$$\{u, \neg(u \ \& \ b), z \rightarrow b\}, \text{ závěr jako } \neg z.$$

Zde jde tedy o důkaz logického důsledku :

$$\{u, \neg(u \ \& \ b), z \rightarrow b\} \models \neg z,$$

resp. o důkaz platnosti implikace

$$(u \& \neg(u \& b) \& (z \rightarrow b)) \rightarrow \neg z.$$

Interpretační tabulka umožňuje zjištění, které z interpretací představují modely dané množiny předpokladů.

Z následující tabulky (tab. 8) je zřejmé, že množina předpokladů má jediný model, a ten je dán ohodnocením výrokových proměnných

$$v(u) = 1, v(b) = 0, v(z) = 0 \text{ (viz řádek označený *)}.$$

	u	b	z	$\neg(u \& b)$	$z \rightarrow b$	$\neg z$
1	0	0	0	1	1	1
2	0	0	1	1	0	0
3	0	1	0	1	1	1
4*	1	0	0	1	1	1
5	0	1	1	1	1	0
6	1	0	1	1	0	0
7	1	1	0	0	1	1
8	1	1	1	0	1	0

Tabulka 7.1-Příklad

To ale znamená, že je pravdivý pouze výrok „Jan je učitel.“, nepravdivé pak jsou výroky „Jan je bohatý.“ a „Jan je rockový zpěvák.“ Formule $\neg z$ je pro tento model rovněž splněna, proto je (v tomto případě triviálním) logickým důsledkem dané množiny předpokladů.

Nyní ještě zbývá otázka, zdali v množině předpokladů

$$\{u, \neg(u \& b), z \rightarrow b\}$$

není některá z formulí logickým důsledkem zbývajících dvou formulí. Z interpretační tabulky lze snadno vyčíst, že tento případ nenastane.[1], [2], [3].

7.2 Nepřímé důkazy logických důsledků



Množinu předpokladů tvoří $\{u, \neg(u \& b), z \rightarrow b\}$, závěrem je formule $\neg z$. Jde tedy o důkaz

$$u, \neg(u \& b), z \rightarrow b \models \neg z.$$

Formule tvořící množinu předpokladů je třeba převést do ekvivalentních klauzulárních tvarů

$$\neg(u \& b) \Leftrightarrow \neg u \vee \neg b,$$

$$z \rightarrow b \Leftrightarrow \neg z \vee b$$

Rezoluční odvozování:

- | | |
|-------------------------|--------------------|
| 1. u | předpoklad |
| 2. $\neg u \vee \neg b$ | předpoklad |
| 3. $\neg z \vee b$ | předpoklad |
| 4. z | popření důsledku |
| 5. $\neg u \vee \neg z$ | rezolventa 2. a 3. |
| 6. $\neg z$ | rezolventa 1. a 5. |
| 7. \square | rezolventa 4. a 6. |

Závěr je tedy logickým důsledkem předpokladů.

Dokažte, že platí závěr: $\neg(a \vee b) \rightarrow c, p \vee c, a \rightarrow b, \neg(b \wedge p) \models c$

Negace důsledku:

$$\neg(a \vee b) \rightarrow c, p \vee c, a \rightarrow b, \neg(b \wedge p), \neg c$$

Eliminace spojky \rightarrow :

$$(a \vee b) \vee c, p \vee c, \neg a \vee b, \neg(b \wedge p), \neg c$$

Aplikace de Morganova pravidla:

$$(a \vee b) \vee c, p \vee c, \neg a \vee b, \neg b \vee \neg p, \neg c$$

- | | |
|-------------------------|-------------------------|
| 1. $(a \vee b) \vee c$ | předpoklad |
| 2. $p \vee c$ | předpoklad |
| 3. $\neg a \vee b$ | předpoklad |
| 4. $\neg b \vee \neg p$ | předpoklad |
| 5. $\neg c$ | popření důsledku |
| 6. $a \vee b$ | rezoluce 1., 5. |
| 7. p | rezoluce 2., 5. |
| 8. $\neg b$ | rezoluce 4., 7. |
| 9. $\neg a$ | rezoluce 3., 8. |
| 10. \square | spor – rezoluce 8., 10. |

Tato prázdná klauzule je nespílitelná, tzn., že rozšířená množina předpokladů je taky nespílitelná. Z toho plyne platnost logického důsledku z dané množiny předpokladů.

7.3 Logické důsledky a tablová metoda

Důkazy, že daná formule je logickým důsledkem dané množiny předpokladů, vycházejí ze stejných úvah, jako tomu bylo u rezoluční metody



popsané v předcházejícím odstavci. Jestliže stejně jako při aplikaci metody rezolučního popření k množině předpokladů dedukce U přidáme negaci formule A , která má být logickým důsledkem U , měla by být takto vytvořená množina $U \cup \{\neg A\}$ nesplnitelná.



Dokažte tablovou metodou platnost pravidla modus ponens: $a, a \rightarrow b \models b$

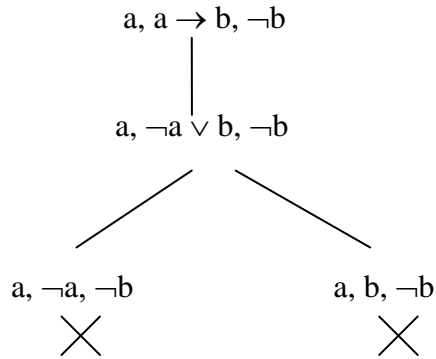
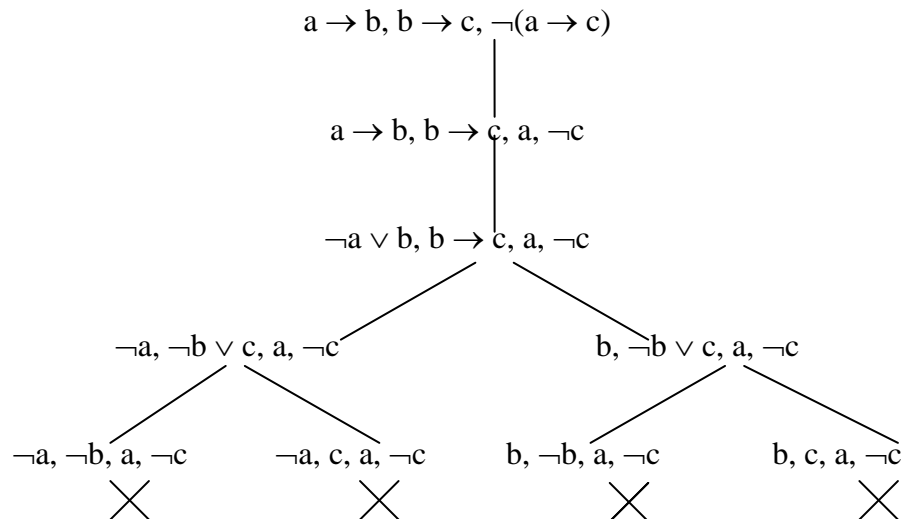


Schéma výše znázorňuje sémantické tablo množiny formulí, jehož obě větve jsou uzavřeny. Proto je množina formulí v kořeni tabla nesplnitelná a formule b je pak logickým důsledkem formulí $a, a \rightarrow b$.



Dokažte tablovou metodou platnost pravidla tranzitivity: $a \rightarrow b, b \rightarrow c \models a \rightarrow c$



Protože se podařilo nalézt tablo, jehož všechny větve jsou uzavřeny, je výchozí množina formulí nesplnitelná.

Formule $a \rightarrow c$ je tedy logickým důsledkem formulí $a \rightarrow b, b \rightarrow c$.

Kontrolní otázky:

1. Lze pomocí rezoluční metody nalézt závěry z předpokladů?
2. Proč řešit dedukci závěru pomocí důkazu sporem?
3. Jaký význam má komplementární pár ve větvi tabla?
4. Jak dokázat logický důsledek?



Shrnutí:

Důkazy, že daná formule je logickým důsledkem dané množiny předpokladů, vycházejí ze stejných úvah, jako tomu bylo u rezoluční metody popsané v předcházejícím odstavci. Jestliže stejně jako při aplikaci metody rezolučního popření k množině předpokladů dedukce U přidáme negaci formule A , která má být logickým důsledkem U , měla by být takto vytvořená množina $U \cup \{-A\}$ nesplnitelná.



8 SYNTAXE A SÉMANTIKA PREDIKÁTOVÉ LOGIKY

V této kapitole se dozvíte:

- Co je predikátová logika.
- Jaká je její abeceda.
- Jaká je expresivita a co je kvantifikátor.
- Jak se sémantika predikátové logiky ovlivňuje výslednou expresivitu.

Po jejím prostudování byste měli být schopni:

- Znat syntaxi predikátové logiky.
- Znat princip kvantifikace.
- Pojmout rozdíl mezi predikátem a jinými prvky abecedy.
- Umět ohodnotit prvky abecedy predikátové logiky.

Klíčová slova této kapitoly:

Predikát, syntaxe, sémantika, kvantifikátor.

Doba potřebná ke studiu: 4 hodiny



Průvodce studiem

V této lekci, jejíž prostudování by vám mělo trvat zhruba 4 h, vám bude predikátová logika prezentována jako nástroj modelování s mnohem vyšší expresivitou, než tomu je v případě výrokové logiky. Seznámíte se s jejími dalšími jazykovými prvky, které se ve výrokové logice nevyskytují, a uvidíte, jak se takto obohacený jazyk blíží svou expresivitou jazykům přirozeným. Seznámíte se zde s pravidly syntaxe dobře utvořených formulí jazyka a naučíte se konstruovat základní stavební prvky formulí: termy, na jejich základě pak formule atomické a z nich formule komponované. Podobně jako ve výrokové logice budete hodnotit složitost formule jazyka predikátové logiky, vycházející ze stromové reprezentace syntaxe predikátové formule. V závěrečných odstavcích lekce se budeme věnovat formulím s kvantifikátory. Dozvíte se, jak zacházet s proměnnými vázanými ve formulích kvantifikátory, a to zejména při substitucích, jimiž se vytvářejí jednotlivé instance formulí. Dále se dozvíte, jakým způsobem se jazyku zkonstruovanému podle formálně daných pravidel přiřadí jeho význam. Budeme zde definovat interpretaci dobře utvořené formule jazyka L_1 jako funkci přiřazující dané formulí pravdivostní hodnotu.

8.1 Jazyk predikátové logiky



Jako příklad si uvedeme větu a přeložíme si ji do predikátové logiky.

„Každý student, který chodí na přednášky, vyniká ve cvičeních.“

Pro predikátovou logiku je situace mírně složitější pro označení, což však pomůže pro lepší a přesnější vyjádření daného výroku se všemi jeho aspekty. Je nezbytné si nejprve určit vzájemné vztahy v dané větě. V první řadě musíme zavést predikáty:

- **chodí**(<kdo>, <kam>) – pro daný příklad –
chodí(student,na_přednášky)
- **vyniká**(<kdo>, <v čem>) – pro daný příklad –
vyniká(student,ve_cvičeních)

Vzhledem k formulaci věty je nutné použít kvantifikátory, které se v predikátové logice vyskytují a mají nemalý význam. Pro tento případ je vhodný kvantifikátor obecný. Výsledný zápis tedy bude:

$\forall x (student(x) \ \& \ (chodí(x, na_přednášky) \rightarrow vyniká(x, ve_cvičeních)))$

Data jsou veškeré znakové řetězce, které vstupují do PC a nacházejí se kolem nás. Informace jsou data s již přiřazeným významem, který dokážeme rozeznat. Znalosti jsou informace, které jsou doprovázeny mechanismy dalších znalostí. Typy znalostí mohou být procedurální (algoritmy procesů – například návod na postup řešení) a deklarativní (svět objektů a jejich vztahů).[1], [2], [3].

8.2 Syntax jazyka predikátové logiky

Základem každého jazyka je množina symbolů, kterými tento jazyk disponuje, neboli jeho abeceda. Stejně jako ve výrokové logice musí jazyk predikátové logiky obsahovat symboly označující proměnné, i když význam proměnných má zde poněkud odlišný charakter. Symboly označující proměnné se totiž, jak již bylo řečeno, nevztahují k elementárním výrokům, ale k prvkům určitých množin tvořících universu diskursu, jehož se modelování predikátovou logikou týká.

To, co mají abecedy jazyků výrokové a predikátové logiky společného (i pokud jde o význam), je pět symbolů logických spojek \neg , $\&$, \vee , \rightarrow , \leftrightarrow , logické konstanty true a false a některé symboly pomocné (závorky). Nově se zde zavádějí symboly pro individuové proměnné a konstanty, funkční a predikátové symboly, kvantifikátory a pomocný symbol čárka. [1], [2], [3].

Abeceda predikátové logiky je tvořena osmi základními druhy symbolů [6]:

- 1) symboly pro individuové proměnné - možno použít jakékoliv malé písmeno abecedy, možno indexovat (dolním indexem)
- 2) symboly pro individuové konstanty – student, matka, pi, 5, 9...
- 3) symboly pro logické konstanty – T pro true, F pro false
- 4) symboly pro logické spojky - negace \neg , konjunkce $\&$, disjunkce \vee , implikace \rightarrow , ekvivalence \leftrightarrow
- 5) funkční symboly – sin, cos, $+(k,l)$
- 6) predikátové symboly – studuje(k,l), zkouší(k,l), počítá(x)
- 7) kvantifikátory - univerzální \forall a existenční \exists
- 8) pomocné symboly – závorky, čárky
- 9)



Syntaktická pravidla mohou být zapsána induktivně, nebo pomocí tzv. Backus – Naurovy definice.[6]

Logické spojky byly dostatečně diskutovány v lekcích týkajících se výrokové logiky. Pro funkční a predikátové symboly je charakteristické, že za nimi mohou následovat seznamy argumentů uvedené vždy v závorce a vzájemně oddělené čárkami. Funkční symbol bez argumentů (nulární funkční symbol) je individuovou konstantou. Nulární predikátový symbol je v podstatě výrokovou proměnnou.



Jedním z nejznámějších základních způsobů vyjádření vztahu mezi objekty je, a to nejen v matematice, funkce přiřazující jednoznačně objektům jedné množiny objekty druhé množiny. Např. totální funkce *matka*(x) přiřazuje každé osobě x daného universa diskursu představujícího nějakou množinu osob (např. členy určité rodiny) její matku.

Funkci *otec* lze definovat např. tabulkou takto:

x	<i>otec</i> (x)
<i>Jan</i>	<i>Ivan</i>
<i>Eva</i>	<i>Petr</i>
<i>Anna</i>	<i>Martin</i>
<i>Petr</i>	<i>Gabriel</i>

Tabulka 8.1- Funkční symboly



Kvantifikátory jsou jazykové prvky predikátové logiky, které umožňují zavést do jazyka možnost formálního vyjádření toho, že n -ární relace platí, resp. neplatí pro všechny nebo pro některé n -tice universa diskursu. Zatímco $\mathbf{p}(x)$ vyjadřuje v predikátové logice, že x má vlastnost \mathbf{p} , $\forall x \mathbf{p}(x)$ znamená, že $\mathbf{p}(x)$ platí pro všechny možné hodnoty x , $\exists x \mathbf{p}(x)$ pak znamená, že existuje taková hodnota, která dosazena za x vede k pravdivému $\mathbf{p}(x)$.

8.3 Sémantika jazyka L_1

Dříve, než bude definována struktura přiřazená jazyku L_1 predikátové logiky a na jejím základě definován způsob interpretace prvků jazyka, bude užitečné shrnout základní fakta týkající se významů jednotlivých nově zaváděných jazykových prvků, která byla v souvislosti se zaváděním těchto prvků jazyka uvedena již v předcházející lekcí.

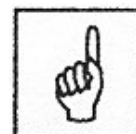
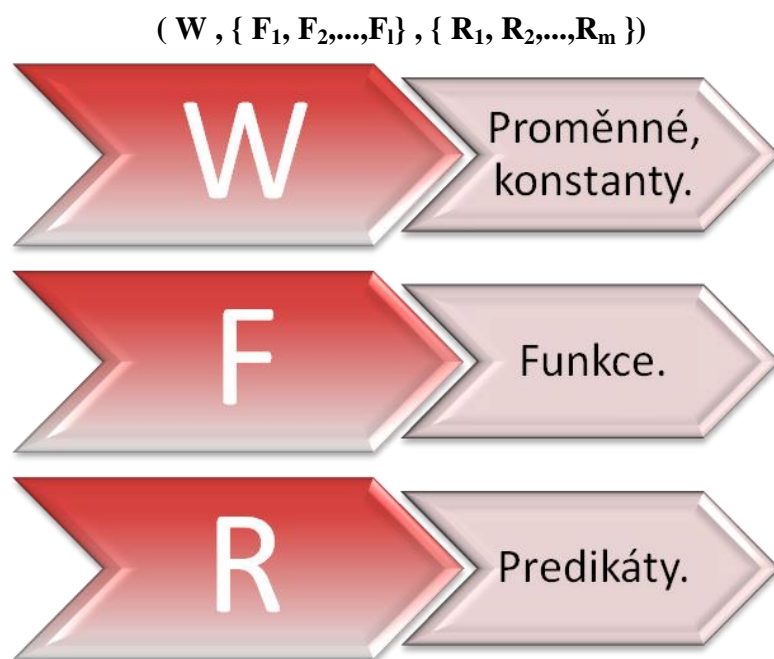
Množina objektů, jichž se význam jazyka týká, je jeho universem diskursu \mathbf{W} . Universum diskursu by mělo přitom být budováno tak, aby individuovým konstantám jazyka L_1 mohly být přiřazeny určité jeho prvky. Přitom každá z konstant je jazykovým výrazem (pojmenováním) svého určitého přiděleného objektu z \mathbf{W} , který tato konstanta v rámci dané interpretace reprezentuje. Prvky universa diskursu \mathbf{W} mají být též ohodnocovány individuové proměnné obsažené ve formulích. Tyto proměnné sehrávají roli parametrů, resp. argumentů predikátů nebo funkcí a vztahují se zpravidla k určitým doménám možných hodnot pro tyto argumenty. Jinak než

jako argumenty funkcí nebo predikátů se individuové proměnné v predikátových formulích nevyskytují. Proto do universa diskursu bezpochyby patří všechny domény argumentů funkcí a predikátů, kterými jazyk disponuje. Alespoň jeden predikátový symbol je „povinným“, minimem jazyka L_1 predikátové logiky. S pojmem predikátu přitom souvisí, co se týká významu, matematický pojem *relace*, konkrétně zde každému n -árnímu predikátovému symbolu odpovídá nějaká n -ární relace. Pravdivostní hodnota predikátu se pak určuje v závislosti na odpovídající *interpretující relaci*. Nulární predikát neobsahuje žádný argument, proto je v podstatě výrokovou proměnnou s možnou pravdivostní hodnotou true/false.[1], [2], [3].

Symbole pro individuové konstanty, funkční a predikátové symboly mají v určitém zavedeném jazyce L_1 predikátové logiky konstantní význam, proto může být v rámci stanovení významu uvažovaného jazyka L_1 jednoznačně stanoven i způsob jejich interpretace, tj. mohou jim být v souladu s následující definicí *struktury* přiřazené jazyku L_1 podle pravidel interpretace též přiřazeny jejich konkrétní denotáty.

Nechť W je neprázdná množina objektů, $\{F_1, F_2, \dots, F_m\}$ je množina funkčních zobrazení, $\{R_1, R_2, \dots, R_n\}$ je množina relací.

Struktura S přiřazená jazyku L_1 predikátové logiky je zobrazení D přiřazující jazykovým výrazům predikátové logiky jejich denotáty z trojice množin



Obrázek 8.1: Proces ohodnocení

Způsob, jakým se formuli predikátové logiky přiřazuje její pravdivostní hodnota, je určen pravidly, která se postupně týkají vyhodnocování (interpretace) termů jazyka, v návaznosti pak interpretace jeho atomických predikátů a z nich konstruovaných formulí.

V případě interpretace formule bude stejně jako ve výrokové logice její výslednou hodnotou jedna z pravdivostních hodnot true / false. Interpretace formule sleduje stejný postup jako její syntax. Interpretují se nejdříve její atomy a potom na jejich základě a pomocí logických spojek formule komponované. Necht' \mathcal{S} je struktura přiřazená jazyku L_1 predikátové logiky, e valuace proměnných jazyka L_1 prvky z universa W a A dobře utvořená formule jazyka L_1 . Indukcí podle složitosti formule A definujeme interpretaci formule A v \mathcal{S} při valuaci proměnných e jako funkční zobrazení

$$I(A[\mathcal{S},e]) = \text{true/false}$$



Kontrolní otázky:

1. Jaká je interpretační struktura predikátové logiky?
2. Jaký je postup a pravidla interpretace?
3. K čemu je valuace a jakým způsobem probíhá?
4. Co je to denotát?



Shrnutí:

V případě interpretace formule bude stejně jako ve výrokové logice její výslednou hodnotou jedna z pravdivostních hodnot true / false. Interpretace formule sleduje stejný postup jako její syntax. Interpretují se nejdříve její atomy a potom na jejich základě a pomocí logických spojek formule komponované. Způsob, jakým se formulí predikátové logiky přiřazuje její pravdivostní hodnota, je určen pravidly, která se postupně týkají vyhodnocování (interpretace) termů jazyka, v návaznosti pak interpretace jeho atomických predikátů a z nich konstruovaných formulí.

9 PREDIKÁTOVÁ LOGIKA A VYUŽITÍ DEDUKCE

V této kapitole se dozvíte:

- Jak pracuje dedukce v predikátové logice.
- Kterými metodami se dedukce řeší.
- Jak pracuje sémantické tablo v predikátové logice.

Po jejím prostudování byste měli být schopni:

- Používat sémantické tablo v predikátové logice
- Schopni řešit deduktivní úlohy v predikátové logice
- Aplikovat sémantické tablo na deduktivní úlohy.

Klíčová slova této kapitoly:

Tablo, alfa, beta, gama, delta, dedukce.

Doba potřebná ke studiu: 2 hodina

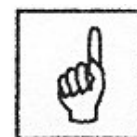
Průvodce studiem

V této lekci, jejíž prostudování by vám mělo trvat zhruba 2 h, se budeme, podobně jako tomu bylo ve výrokové logice, zabývat dedukcí v predikátové logice, tj. pojmem logického důsledku dané množiny předpokladů. Protože tablová metoda sémantické analýzy predikátové formule představuje jednu z efektivních metod prověřování správnosti dedukce, naučíme se, jakým způsobem konstruovat sémantické tablo formulí s kvantifikátory.



9.1 Dedukce a sémantické tablo v predikátové logice

Obdobně, jako tomu bylo ve výrokové logice, bude zde definován pojem modelu množiny formulí predikátové logiky. V předcházející lekci byl model predikátové formule A definován jako taková struktura \mathcal{S} , která je v \mathcal{S} platná (splněna), tj. pro niž je $I(A[\mathcal{S}, e]) = \text{true}$ ve struktuře \mathcal{S} pro všechny valuae e . Nyní zbývá definovat též pojem modelu množiny \mathbf{T} predikátových formulí. [1], [2], [3]. Struktura \mathcal{S} je modelem množiny formulí \mathbf{T} jazyka \mathbf{L}_1 , právě když \mathcal{S} splňuje každou z formulí z \mathbf{T} .



I v predikátové logice můžeme mezi porovnávanými metodami najít několik zásadních rozdílů. Hlavní myšlenkou této metody je deklarování γ a δ pravidel tak, že nejdříve ohodnotíme existenčně kvantifikovanou formuli novou ještě nezavedenou konstantou a poté touto konstantou můžeme ohodnotit i zbylé všeobecně kvantifikované formule. Tím tedy vznikne konkrétní instance formule, o jejíž splnitelnosti můžeme uvažovat. Jestliže existuje více formulí nezávisle existenčně kvantifikovaných, zavádí se do formule více konstant, a to v počtu oněch existenčně kvantifikovaných formulí. V tom případě musíme i těmito konstantami ohodnotit všechny univerzálně kvantifikované formule. Tím pádem se počet výrazů v table značně zvýší. Pomocí ekvivalentních úprav potom formuli transponujeme do ekvivalentní

disjunktivní normální formy. Výhoda této metody spočívá opět v explicitně definovaných uzlech patřičným seznamem atomických formulí.

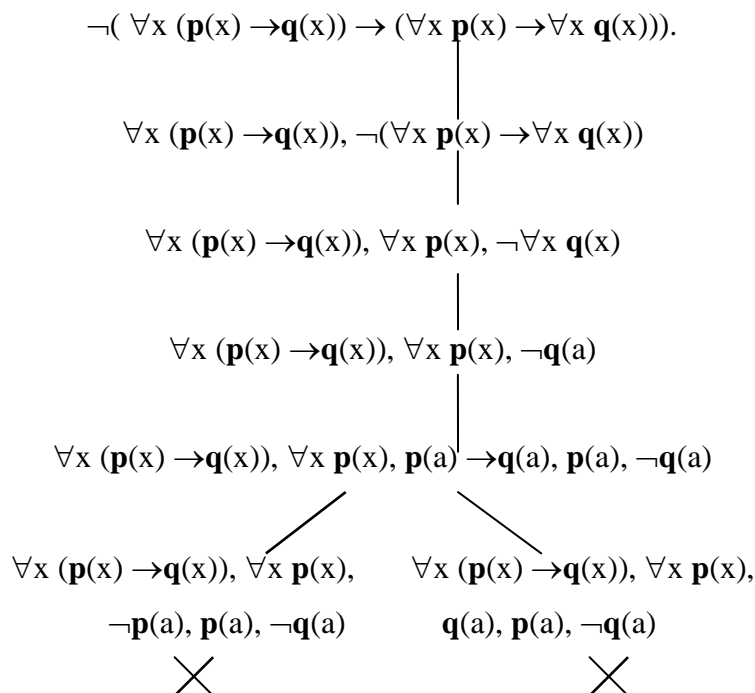
Následující dva příklady budou ukázkou konstrukce konečného sémantického tabla formule, na jehož základě lze buď konstatovat nesplnitelnost formule nebo naopak charakterizovat její model.



Sestrojte sémantické tablo nesplnitelné formule

$$\neg(\forall x (\mathbf{p}(x) \rightarrow \mathbf{q}(x)) \rightarrow (\forall x \mathbf{p}(x) \rightarrow \forall x \mathbf{q}(x))).$$

Sémantické tablo, jehož konstrukce byla právě popsána, je zobrazeno v následujícím schématu.



Odvozovací pravidla sémantického tabla formule predikátové logiky jsou obohacena oproti výrokové logice o další dvě pravidla. Je to proto, že predikátová logika má vyšší expresivitu, obsahuje tudíž více prvků abecedy a proto je i více využívána. Přibývají pravidla gama a delta, která slouží pro odstraňování kvantifikátorů.

α - pravidla:

α	α_1	α_2
$\neg \neg A$	A	
$A_1 \& A_2$	A_1	A_2
$\neg (A_1 \vee A_2)$	$\neg A_1$	$\neg A_2$
$\neg (A_1 \rightarrow A_2)$	A_1	$\neg A_2$
$\neg (A_1 \leftarrow A_2)$	$\neg A_1$	A_2
$A_1 \leftrightarrow A_2$	$A_1 \rightarrow A_2$	$A_1 \leftarrow A_2$

Tabulka 9.1- alfa pravidla

Beta pravidlo je jediné pravidlo, které zaručuje dualitu ve stromě. Tyto pravidla zajišťují větvení, protože všechna ostatní pravidla jsou pouze řádková.

β - pravidla:

β	β_1	β_2
$B_1 \vee B_2$	B_1	B_2
$\neg (B_1 \& B_2)$	$\neg B_1$	$\neg B_2$
$B_1 \rightarrow B_2$	$\neg B_1$	B_2
$B_1 \leftarrow B_2$	B_1	$\neg B_2$
$\neg (B_1 \leftrightarrow B_2)$	$\neg (B_1 \rightarrow B_2)$	$\neg (B_1 \leftarrow B_2)$

Tabulka 9.2- beta pravidla

γ - pravidla pro univerzální formule:

Pokud existuje uzel ohodnocený seznamem formulí:

$$\varphi_1, \varphi_2, \dots, \delta_1, \dots, \delta_k, \dots, \varphi_n$$

pak bezprostředně následující uzel bude ohodnocen seznamem formulí:

$$\varphi_1, \varphi_2, \dots, \delta_1, \dots, \delta_k, \delta_1(a_1), \dots, \delta_k(a_k), \dots, \varphi_n$$

Příčemž konstanty a_1, \dots, a_k jsou nově vygenerované, nevyskytující se ještě ve formuli.

γ	$\gamma(a)$
$\forall x A(x)$	$\forall x A(x), A(a)$
$\neg \exists x A(x)$	$\neg \exists x A(x), \neg A(a)$

Tabulka 9.3- gama pravidla

δ - pravidla pro existenční formule:

Pokud existuje uzel ohodnocený seznamem formulí:

$$\varphi_1, \varphi_2, \dots, \gamma_1, \dots, \gamma_k, \dots, \varphi_n$$

pak bezprostředně následující uzel bude ohodnocen seznamem formulí:

$$\varphi_1, \varphi_2, \dots, \gamma_1, \dots, \gamma_k, \gamma_1(a_1), \dots, \gamma_k(a_k), \dots, \varphi_n$$

Příčemž konstanty a_1, \dots, a_k jsou konstanty již vyskytující se ve formuli.

δ	$\delta(a)$
$\exists x A(x)$	$\exists x A(x), A(a)$
$\neg \forall x A(x)$	$\neg \forall x A(x), \neg A(a)$

Tabulka 9.4- delta pravidla

9.2 Sémantické tablo a logický důsledek

Jednou z metod, které umožňují rozhodovat o tom, zdali je daná formule A logickým důsledkem dané množiny předpokladů T , je metoda využívající sémantické tablo množiny formulí. Jak ukazuje následující příklad, prověřením nesplnitelnosti množiny předpokladů dedukce, rozšířené o negaci jejího předpokládaného logického důsledku, lze prověřit správnost dedukce.



Prověřte pomocí sémantického tabla logický důsledek

$$\forall x (a(x) \rightarrow \neg b(x)), \exists x b(x) \models \exists x \neg a(x).$$

$$\forall x (a(x) \rightarrow \neg b(x)), \exists x b(x), \neg \exists x \neg a(x)$$

$$\forall x (a(x) \rightarrow \neg b(x)), b(a), \forall x a(x), a(a)$$

$$\forall x (a(x) \rightarrow \neg b(x)), a(a) \rightarrow \neg b(a), b(a), \forall x a(x), a(a)$$

$$\forall x (a(x) \rightarrow \neg b(x)), \neg a(a),$$

$$b(a), \forall x a(x), a(a)$$



$$\forall x (a(x) \rightarrow \neg b(x)), \neg b(a),$$

$$b(a), \forall x a(x), a(a)$$



Korespondenční úkol

1. Prověřte pomocí sémantického tabla logický důsledek

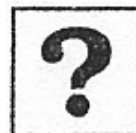
$$\forall x (a(x) \& b(x)), b(a) \rightarrow c(a) \models \exists x c(x)$$

2. Dokažte nepřímo sémantickým tablem logickou platnost formule:

$$\forall x (p(x) \rightarrow q(x)) \rightarrow (\exists x p(x) \rightarrow \exists x q(x))$$

Kontrolní otázky:

1. K čemu slouží pravidla gama a delta?
2. Proč řešit dedukci závěru pomocí důkazu sporem?
3. Jaký význam má komplementární pár ve větvi tabla?
4. Jak dokázat logický důsledek?



Shrnutí:

Důkazy, že daná formule je logickým důsledkem dané množiny předpokladů, vycházejí ze stejných úvah, jako tomu bylo u rezoluční metody popsané v předcházejícím odstavci. Jestliže stejně jako při aplikaci metody rezolučního popření k množině předpokladů dedukce U přidáme negaci formule A , která má být logickým důsledkem U , měla by být takto vytvořená množina $U \cup \{\neg A\}$ nesplnitelná. Je-li v každém modelu (struktuře), ve kterém je splněna formule B , splněna i formule A , pak A se nazývá logický důsledek formule B (označuje se $B \models A$). Je-li v každém modelu množiny formulí T splněna i formule A , pak A se nazývá logický důsledek množiny formulí T (označuje se $T \models A$). Necht' T je množina uzavřených formulí, A je uzavřená formule jazyka L_1 . Formule A je logickým důsledkem množiny formulí T , právě když množina $T \cup \{\neg A\}$ je nesplnitelná. V příštích odstavcích budou ukázány některé efektivnější způsoby prověřování, zdali je daná formule logickým důsledkem dané množiny formulí.



10 KLAUZULÁRNÍ FORMA FORMULE V JAZYCE L_1

V této kapitole se dozvíte:

- Co klauzule v predikátové logice.
- Co je prenexní tvar formule.
- Jak pracuje procedura skolemizace.

Po jejím prostudování byste měli být schopni:

- Převádět formule predikátové logiky na klauzule.
- Řešit deduktivní úlohy v predikátové logice.
- Převádět formule do prenexní formy.

Klíčová slova této kapitoly:

Skolemizace, klauzule, dedukce, prenexní.

Doba potřebná ke studiu: 2 hodina



Průvodce studiem

Pro použití rezolučního odvozování i v predikátové logice, seznámíte se v této lekci, jejíž prostudování by mělo trvat zhruba 2 h, se způsoby transformace predikátových formulí do klauzulární normální formy. Uvědomíte si, že základem normalizace formulí predikátové logiky je převedení formule do prenexní formy, v níž jsou všechny kvantifikátory soustředěny vpředu před formulí a tvoří tak její prefix. Pro následnou transformaci formule do požadované klauzulární podoby se naučíte, jak docílit toho, aby prefix formule univerzálně kvantifikoval všechny vyskytující se proměnné a aby otevřené jádro formule bylo v klauzulárním tvaru, na nějž lze pak aplikovat rezoluční odvozování. Pro nutnou eliminaci existenčních kvantifikátorů zde seznámíte s postupem zvaným skolemizace. Uvědomíte si též, že úprava formule do klauzulárního tvaru obecně nevede k formuli s ní ekvivalentní, nýbrž pouze zachovává její splnitelnost.

10.1 Algoritmus převodu formule do prenexní normální formy



Hlavně existence kvantifikátorů nám opět přidělala problém ve slova smyslu jejich zbavení se při převodu a práci s klauzulemi. Procedura tabla k tomuto odstranění využívá gama a delta pravidla a zde se musíme zbavit kvantifikátorů jednotlivě. Nejprve přejdeme k odstranění univerzální kvantifikace. Prvním krokem převodu predikátové formule do její klauzulární podoby je její převedení do prenexního tvaru, v němž se všechny kvantifikátory soustředí před formulí do *prefixu formule*.

Úpravy, kterými lze převést formuli predikátové logiky do prenexní formy, se též nazývají *prenexní operace*. Jde v podstatě o přepisovací pravidla,

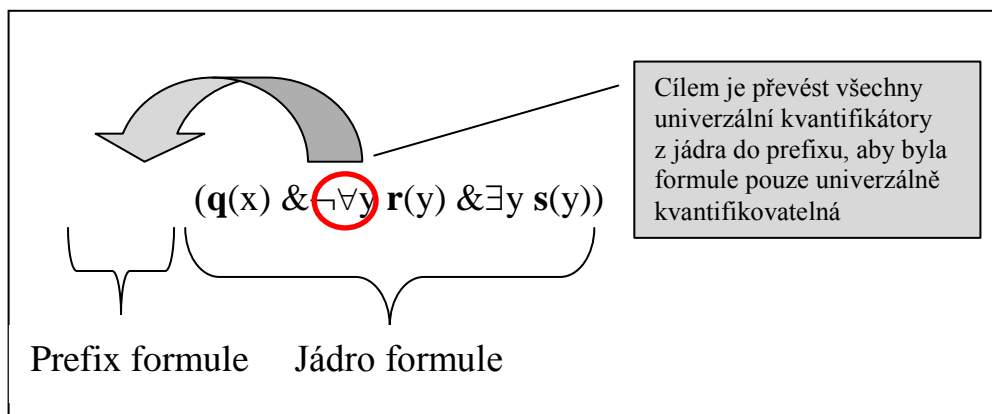
kteřá buď již byla ve formě ekvivalenci dokázána, nebo jejich platnost vyplývá z jejich množinové interpretace. Jak na prenexní operace se dozvíte zde:



1. *Standardizace vázaných proměnných*, spočívající v jejich přejmenování, tak aby se předsunutím symbolu kvantifikace do prefixu formule nezměnily meze jeho působnosti.
2. *Odstranění nadbytečných kvantifikátorů*, ke kterým neexistují příslušné vázané proměnné.
3. *Přesun spojky negace* umístěné před závorkou nebo před kvantifikátorem před atom podle následujících schémat:

$$\begin{aligned} \neg\neg A &\longrightarrow A \\ \neg\forall x A &\longrightarrow \exists x \neg A \\ \neg\exists x A &\longrightarrow \forall x \neg A \\ \neg(A\&B) &\longrightarrow (\neg A\vee\neg B) \\ \neg(A\vee B) &\longrightarrow (\neg A\&\neg B) \end{aligned}$$

4. *Přesun kvantifikátorů* před formuli pomocí přepisovacích pravidel (a) - (e) :



Obrázek 10.1: Smysl prenexního tvaru

Převedení do prenexní normální formy formule

$$\exists x \mathbf{p}(x) \rightarrow (\mathbf{q}(x) \& \neg\forall y \mathbf{r}(y) \& \exists y \mathbf{s}(y))$$



1. Přejmenování proměnné x na w a y na z:

$$\exists w \mathbf{p}(w) \rightarrow (\mathbf{q}(x) \& \neg\forall y \mathbf{r}(y) \& \exists z \mathbf{s}(z))$$

2. Přesun kvantifikátorů před formuli podle operace (a):

$$\exists w \mathbf{p}(w) \rightarrow \exists y \exists z (\mathbf{q}(x) \& \neg\mathbf{r}(y) \& \mathbf{s}(z))$$

3. Přesun kvantifikátoru před formuli podle operace (d):

$$\forall w (\mathbf{p}(w) \rightarrow \exists y \exists z (\mathbf{q}(x) \& \neg\mathbf{r}(y) \& \mathbf{s}(z)))$$

4. Přesun kvantifikátorů před formuli podle operace (e):

$$\forall w \exists y \exists z (\mathbf{p}(w) \rightarrow (\mathbf{q}(x) \& \neg\mathbf{r}(y) \& \mathbf{s}(z)))$$

Jestliže je konečným cílem úpravy formule do normální formy její klauzulární forma, bývá výhodné již během prenexních operací přepisovat spojky \rightarrow a \leftrightarrow spojkami \neg , $\&$ a \vee .



Převedení do prenexní normální formy formule

$$\forall x (\mathbf{p}(x) \& \forall y \exists x (\neg \mathbf{q}(x,y) \rightarrow \forall z \mathbf{r}(a,x,y)))$$

1. náhrada spojky \rightarrow :

$$\forall x (\mathbf{p}(x) \& \forall y \exists x (\mathbf{q}(x,y) \vee \forall z \mathbf{r}(a,x,y)))$$

2. přejmenování proměnných:

$$\forall x (\mathbf{p}(x) \& \forall y \exists u (\mathbf{q}(u,y) \vee \forall z \mathbf{r}(a,u,y)))$$

3. odstranění nadbytečných kvantifikátorů:

$$\forall x (\mathbf{p}(x) \& \forall y \exists u (\mathbf{q}(u,y) \vee \mathbf{r}(a,u,y)))$$

4. přesun kvantifikátorů do prefixu:

$$\forall x \forall y (\mathbf{p}(x) \& \exists u (\mathbf{q}(u,y) \vee \mathbf{r}(a,u,y)))$$

$$\forall x \forall y \exists u (\mathbf{p}(x) \& (\mathbf{q}(u,y) \vee \mathbf{r}(a,u,y)))$$



Převěďte do prenexní normální formy formuli

$$\forall y (\exists x \mathbf{p}(x,y) \rightarrow \exists u \mathbf{r}(y,u)) \rightarrow \forall x \mathbf{s}(x,y)$$

10.2 Klauzulární forma formule, skolemizace

Pro odstranění univerzálních kvantifikací jsme využili převod do prenexního tvaru, kdy je jádro formule uvažováno, jako univerzálně kvantifikovatelné a samotné vázané proměnné už jako vázané brát nemusíme. Pro odstranění existenčního kvantifikátoru slouží procedura skolemizace.



Postup převedení formule do klauzulární formy zahrnuje i její úpravu do Skolemovy formy neobsahující existenčně vázané proměnné, a to postupem zvaným skolemizace. Speciální případ skolemizace byl zaveden již v předcházejícím odstavci při vytváření sémantického tabla formule s existenčním kvantifikátorem. Princip skolemizace spočívá zjednodušeně v následující úvaze : Jestliže např. tvrdíme, že „ke každé hodnotě x existuje jistá hodnota y ...“, pak to ve skutečnosti znamená nějaký funkční vztah mezi x a y tak, že x je zde nezávisle proměnnou a y na ní závisí.

Obecně předpokládejme formuli v prenexní formě tvaru

$$\forall x_1 \dots \forall x_n \exists y,$$



Nechť $\forall x_1 \dots \forall x_n$, $n \geq 0$ a $\exists y_1 \dots \exists y_k$, $k \geq 0$, jsou dvě za sebou následující součásti prefixu formule A v prenexní klauzulární formě. Potom **skolemizace** prefixu s existenčními kvantifikátory $\exists y_1 \dots \exists y_k$ spočívá v odstranění existenčních kvantifikátorů z prefixu.

- pro $n = 0$ v náhradě každého výskytu y_1, \dots, y_k v jádře formule po řadě novými konstantními symboly c_1, \dots, c_k , které se předtím ve formuli nevyskytovaly; nově zavedené konstanty se pak nazývají **Skolemovými konstantami**,
- pro $n > 0, k > 0$ v náhradě každého z výskytů y_1, \dots, y_k postupně termy $f_1(x_1, \dots, x_n), \dots, f_k(x_1, \dots, x_n)$, kde f_1, \dots, f_k jsou n -ární funkční symboly, které se předtím ve formuli A nevyskytovaly; nově zavedené funkční symboly se nazývají **Skolemovými funkcory**. [1]

Pro $k = 0$ je formule uzavřenou formulí v klauzulární formě. Jestliže $n = 0$, tj. $\exists y_1 \dots \exists y_k$ nepředchází žádná část prefixu s univerzálními kvantifikátory, vznikne volbou Skolemových konstant za všechny existenčně vázané proměnné otevřená formule v klauzulární formě. [1], [2], [3].

Postupem zvaným skolemizace se formule převádí do nové podoby, o níž nelze, přinejmenším z důvodu syntaxe, tvrdit, že je formou ekvivalentní původní formuli. Jde pouze o takovou transformaci formule, která jak ukazuje následující věta, zachovává její splnitelnost. Náhrada existenčního kvantifikátoru v prefixu formule A v klauzulární prenexní formě pomocí Skolemova funktoru zachovává splnitelnost této formule.



Úprava formule do klauzulární formy zpravidla navazuje na její úpravu do prenexní formy, spočívají v krocích, které již byly popsány a zdůvodněny. Protože převod formule do její klauzulární formy má za cíl možnost další manipulace pouze s otevřeným jádrem formule, o němž se předpokládá, že všechny jeho proměnné jsou univerzálně vázány, nelze žádnou z proměnných ponechat před úpravou volnou. Formule je proto do klauzulárního tvaru upravována v těchto krocích [1], [2], [3]:

- Změna volných proměnných na vázané proměnné zavedením existenčních kvantifikátorů.
- Transformace formule do ekvivalentní prenexní formy.
- Transformace otevřeného jádra formule do klauzulární formy.
- Provedení skolemizace, která eliminuje existenční kvantifikátory: Odstranění existenčních kvantifikátorů z prefixu formule zavedením nových Skolemových konstant. Převod existenčně vázaných proměnných do funkční podoby skolemizací, v případech, kdy argumenty Skolemovy funkce jsou univerzálně vázané proměnné, které se ve formuli vyskytují vedle uvažované existenčně vázané proměnné.

Skolemizací eliminujte existenční kvantifikátor formule

$$\forall x \exists y \mathbf{p}(x, y)$$

Označíme-li $y = f(x)$, bude mít daná formule tvar:

$$\forall x \mathbf{p}(x, f(x))$$





Převeďte do prenexní klauzulární normální formy formulí

$$\forall x (p(x) \rightarrow q(x)) \rightarrow (\forall x p(x) \rightarrow \forall x q(x))$$

1. přejmenování proměnných:

$$\forall x (p(x) \rightarrow q(x)) \rightarrow (\forall y p(y) \rightarrow \forall z q(z))$$

2. eliminace spojky \rightarrow :

$$\neg \forall x (p(x) \rightarrow q(x)) \vee (\forall y p(y) \rightarrow \forall z q(z))$$

3. eliminace spojky \rightarrow :

$$\neg \forall x (\neg p(x) \vee q(x)) \vee (\neg \forall y p(y) \vee \forall z q(z))$$

4. přepis kvantifikátorů:

$$\exists x \neg (\neg p(x) \vee q(x)) \vee (\exists y \neg p(y) \vee \forall z q(z))$$

5. aplikace de Morganova pravidla:

$$\exists x (p(x) \& \neg q(x)) \vee (\exists y \neg p(y) \vee \forall z q(z))$$

6. převedení do prefixu kvantifikátorů stojících mimo meze jiných kvantifikátorů:

$$\exists x \exists y \forall z (p(x) \& \neg q(x)) \vee \neg p(y) \vee q(z)$$

7. úprava otevřeného jádra distributivním zákonem:

$$\exists x \exists y \forall z (p(x) \vee \neg p(y) \vee q(z)) \& (\neg q(x) \vee \neg p(y) \vee q(z))$$

8. eliminace existenčních kvantifikátorů skolemizací:

$$\forall z (p(a) \vee \neg p(b) \vee q(z)) \& (\neg q(a) \vee \neg p(b) \vee q(z))$$



Korespondenční úkol

Převeďte do prenexní klauzulární normální formy formulí

$$\exists x \forall y p(x,y) \rightarrow \forall x \exists y p(x,y)$$



Kontrolní otázky:

1. Co je to prenexní tvar?
2. Proč se zavádí procedura skolemizace?
3. Co jsou to prenexní operace?
4. Využívají se při převodech aplikace de Morganových pravidel?



Shrnutí:

Postupem zvaným skolemizace se formule převádí do nové podoby, o níž nelze, přinejmenším z důvodu syntaxe, tvrdit, že je formou ekvivalentní původní formuli. Jde pouze o takovou transformaci formule, která jak ukazuje následující věta, zachovává její splnitelnost. Náhrada existenčního kvantifikátoru v prefixu formule A v klauzulární prenexní formě pomocí Skolemova funktoru zachovává splnitelnost této formule. Zavádění nových Skolemových konstant a funktorů ve skutečnosti znamená rozšiřování již zavedeného jazyka predikátové logiky o nové konstantní symboly. Až dosud jsme pro tyto Skolemovy termy nepoužívali speciální znaky abecedy, ani jsme se nezabývali interpretací formulí se Skolemovými termy. To všechno bylo možné proto, že Skolemovy termy byly vytvářeny účelově pouze pro formální manipulaci s formulemi, resp. vytváření speciálních instancí při rozhodování o jejich splnitelnosti. Jinak tomu bude v případě klauzulární logiky, která pracuje pouze s otevřenými jádry formulí v klauzulárním tvaru, z nichž již není možno zpětně usoudit, zdali term vznikl nebo nevznikl skolemizací.

11 POROVNÁNÍ ZNÁMÝCH TABLORVÝCH PŘÍSTUPŮ

V této kapitole se dozvíte:

- Jaké postupy jsou vhodné k řešení vybraných úloh
- Jaké nástroje využívat
- Jaká je obecná náročnost vybraných problémů.

Po jejím prostudování byste měli být schopni:

- Vhodně zařadit postupy tabel.
- Orientovat se ve vybraných technikách.

Klíčová slova této kapitoly:

Diagramatický přístup, tablo, optimalizace.

Doba potřebná ke studiu: 2 hodina

Průvodce studiem

Tato kapitola je věnována porovnání známých typů sémantických tabel, jakožto metod sémantické analýzy logiky i. řádu, z hlediska tvorby pro uživatele. Jelikož rozhodovací algoritmy mají za úkol poskytnout výstup bez zásahu do jakékoliv jazykové interpretace výrazů, můžeme říci, že ať už je schéma metody jakékoliv, vždy musí docílit stejného výsledku, a to i v sebemeně přívětivější formě.



V zásadě nám jde o komparaci dvou popisovaných přístupů konstrukce sémantických tabel v logice I. řádu. Je samozřejmostí, že se nedá postulovat výstup pomocí dvouhodnotové logiky; lepší či horší. Použili jsme spousty příkladů a patřičnou bázi rozhodování, abychom mohli konkretizovat případné výhody a nevýhody či specifikovat pro jaký typ procesu se určitý přístup hodí.

Jelikož rozhodovací algoritmy mají za úkol poskytnout výstup bez zásahu do jakékoliv jazykové interpretace výrazů, můžeme říci, že ať už je schéma metody jakékoliv, vždy musí docílit stejného výsledku, a to i v sebemeně přívětivější formě. Volili jsme bázi rozhodování v rámci základní látky matematické logiky, ovšem také v rámci maximálního využití daného nástroje. Pokud uvážíme, že ve sféře výrokové logiky je tabulková metoda nejvhodnější a nejjednodušejí programově konstruovatelnou metodou důkazu splnitelnosti či platnosti formulí a v predikátové logice dominuje z hlediska tautologičnosti formulí Robinsonův rezoluční princip, je třeba říci, že sémantické tablo jako takové je spíše univerzálním nástrojem pro převod do disjunktivní normální formy, zjišťování splnitelnosti či logické platnosti ba dokonce dedukce, generování teorie a falzifikace tautologií. [17]

11.1 Vhodné nástroje

Je komplikované specifikovat optimální komplexní algoritmus, jenž by dokázal prověřit obě metody tak, aby bylo zřetelné definovat opravdu jejich vhodnost pro konkrétní proces s výsledkem takovým, že by deklaroval



ideálnější variantu. Spousta měřítek může být ovlivněna mírou expresivity a empirie. Samozřejmě je nutné nalézt optimální metriky, ovšem někdy je vhodné i použití slovního komentáře.

Základním hlediskem pro rozhodování je určitě fakt, pro jaký účel daný algoritmus potřebujeme. V podstatě existují základní procesy, jejichž kompozici budeme nazývat **bázi rozhodování**, jejímiž složkami jsou:

- Nalezení disjunktivní normální formy formule
- Obecná splnitelnost a platnost formulí
- Teorie a logický důsledek
- Přirozená dedukce – metoda reverzního sémantického tabla dle Gerharda Gentzena.

Každý z těchto procesů vyžaduje odlišný přístup a každá ze dvou metod může mít vzhledem k těmto složkám různé výhody a nevýhody. Sémantické tablo je obecně univerzální algoritmus v rámci již zmíněné báze rozhodování, čili může přinášet komplexně poměrně pozitivní praktické výstupy. Stejně tak je základním stavebním kamenem výuky logiky, čili je užitečné vzít jej v úvahu i jako pedagogicko-didaktický nástroj a učinit analýzu i z této perspektivy. Je tedy vhodné rozdělit analýzu podle uživatele na:

- Analýza vzhledem k uživateli s očekávaným praktickým využitím
- Analýza z hlediska pedagogicko-didaktického přístupu

Každý algoritmus, který má poskytnout nějaký výsledek, by jej měl poskytnout samozřejmě požadovaně co nej přesněji, s nejmenšími nároky a také co nejrychleji. Zvolíme tedy jako základní kritérium pro požadovaný praktický výstup jako

- Obecná náročnost na konstrukci algoritmu a jeho rychlost.

Deklarujme ještě jedno kritérium, které je také poměrně podstatné vzhledem k samotnému uživateli, ke kterému se vlastně analýza vztahuje, a jeho interakce k algoritmu, a to:

- Přívětivost algoritmu vzhledem k uživateli.

[17]

11.2 Složitost uzlů



Mějme v úvahu jednu větev sémantického tabla od kořene k listu jako posloupnost jejích uzlů a_1, a_2, \dots, a_n tak, že každý prvek a_i bude vyjadřovat parciální součet absolutních četností jednotlivých symbolů obsažených v daném uzlu. Mějme tedy zobrazení $N \rightarrow H : n \mapsto a_n$, kde H je množina součtů absolutních četností v rámci každého uzlu. Pro metodu prof. Lukasové platí vždy, že posloupnost je nerostoucí, jelikož postupně dekomponujeme formuli na seznam literálů. Takhle skutečnost platí pro každou větev sémantického tabla. Složitost uzlů se tedy s přibývajícím úrovní zjednodušuje tím, že každá parciální formule se nachází v jiném uzlu.

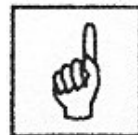
Pojďme najít uzel, u metody prof. Kvasničky, který by obsahoval více symbolů než u metody prof. Lukasové. U prověřování teorie můžeme tuto možnost hned vyloučit, jelikož zde metoda dekompozice formule pomocí distributivního zákona hned zajišťují posloupnosti v rámci každé větve od základní formule, přičemž vždy odstraníme jen jednu složku konkrétní formule

teorie v rámci každého větvení beta pravidlem, čili se následující uzel extrémně nezjednoduší. Metoda prof. Kvasničky ohodnotí každou úroveň stromu jednou formulí teorie. **Neexistuje tedy uzel, který by obsahoval více symbolů u metody prof. Kvasničky.** Součet můžeme obdobně provést jako u větvi. [17]

Proveďme úvahu:

Existuje libovolně složitá formule výrokové logiky. Při nalezení beta pravidla jako centrální spojky se u metody prof. Lukasové použije rozpad na dvě formule, u metody prof. Kvasničky taktéž. Při nalezení alfa pravidla jako centrální spojky je nutné u metody prof. Lukasové najít vnořené beta pravidlo a provést distributivní zákon rozpadu na dvě podformule. U metody prof. Kvasničky najdeme beta pravidlo, rozdělíme obě složky do dvou větví a druhou formuli připojenou tedy pomocí alfa pravidla ohodnotíme obě větve. Čili opět dostáváme menší uzly s menším počtem symbolů. Ekvivalentní úpravy metoda prof. Kvasničky nedefinuje. Neexistuje tedy uzel metody prof. Lukasové, který by obsahoval méně symbolů než u metody prof. Kvasničky. Z čehož vyplývá, že neexistuje ani tablo, pro které by platilo $\mathcal{P}(M_1) < \llbracket \mathcal{P}(M) \rrbracket_2$. \mathcal{P} je tedy počet symbolů v table konkrétní formule.

$$\mathcal{P}(M_1) \geq \llbracket \mathcal{P}(M) \rrbracket_2$$



Pokud narazíme v uzlu na formuli typu: $\varphi_1, \varphi_2, \dots, (a \vee b), \dots, \varphi_n$ a rozhodneme se tedy větvit podle jediné vyjádřené formule $(a \vee b)$, proměnná a se bude vyskytovat již ve všech větvích jedné části stromu a proměnná b ve všech větvích druhé části stromu, rozděleného podle daného beta pravidla. U formule výrokové logiky s velkým počtem proměnných mohou být uzly značně složité a přepisování pomocí distributivního zákona je pak obtížné a zdlouhavé.

Metoda prof. Kvasničky radikálně snižuje složitosti uzlů rozproštěním proměnných do více uzlů tak, že pokud se v daném uzlu vyskytuje seznam literálů, bude jím uzel ohodnocen tak, že každý následný uzel budeme kontrolovat na splnitelnost v rámci tohoto seznamu. Tím se vyhneme složitému rozepisování pomocí distributivního zákona a zmenšíme počet symbolů, jelikož ty redundantní již neopisujeme.

Metoda prof. Lukasové využívá postupné převádění do DNF pomocí posloupností kroků, z nichž u metody prof. Kvasničky jsou některé kroky vynechány, jelikož spadají do předdefinovaných diagramů. Prostřednictvím korektního přístupu metody využívající diagramatických přístupů neexistuje větší počet kroků pro dekompozici formule na listy než u metody prof. Lukasové.

Poznámka:

Situace může být ošemetná pro nezkušené uživatele metody prof. Kvasničky, jelikož například formule tvaru $\varphi_1, \varphi_2, \dots, \varphi_1, \varphi_2, \varphi_3, \dots, \varphi_n$, kde φ_i je libovolná atomická formule a φ_j je libovolná formule, jiná než atomická, může

$$2^n$$

tablo vygenerovat až trojic listů $\varphi_1, \varphi_2, \varphi_3$, což je značně redundantní. Atomickými formulami se ohodnocuje kořen stromu. Pak by například

složitost \mathfrak{C} mohla být $\mathfrak{C}(M_1) < \mathfrak{C}(M_2)$, ovšem v úvahu zde bereme pouze regulérní přístup. [17]

11.2 Situace v predikátové logice



Situace je obdobná i v predikátové logice, i když s nepatrnými rozdíly, které ovšem nejsou kontraproduktivní k výsledku z výrokové logiky, spíše pouze díky charakteristikám predikátové logiky se výsledky prohlubují. Rozdíl je opět v zavedení diagramů pro existenční a univerzální kvantifikátory. Pointa zjednodušení přístupu metody prof. Kvasničky je v tom, že různé existenčně kvantifikované formule jsou ohodnoceny různými od sebe odlišnými konstantami a univerzálně kvantifikované formule jsou chápány jako libovolné prvky v rámci příslušného univerza, přičemž libovolné opravdu doslova. Pokud tedy máme ve všeobecně kvantifikovaných formulích deklarované libovolné individua t_1, t_2, \dots, t_n je možno je volit i takto: $t_1 = t_2 = \dots = t_n$. Tyto rovnosti pak znázorňují jedno a to samé individuum z příslušného univerza. Potom při stanovení určité konstanty a v existenčně kvantifikované formuli ji můžeme položit rovnu například t_1 , čili: $a = t_1$. Další různá individua zavedeny do formule prostřednictvím existenční kvantifikace již v rámci konkrétního predikátu a stejného negovaného predikátu nebudou ve sporu. Metoda prof. Lukasové ovšem svým přístupem vytváří disjunktivní normální formu atomických proměnných se všemi zavedenými konstantami, včetně univerzálně kvantifikovaných formulí a také původních neohodnocených formulí. Můžeme si všimnout, že v příkladu máme formuli $\forall x(p(x) \vee q(x))$. Mimoto jsme deklarovali dvě konstanty a, b . V metodě prof. Lukasové opět řešíme distributivním zákonem, ovšem u metody prof. Kvasničky jen kontrolujeme, zdali nenajdeme spor v rámci dvou různých uzlů.

Jediná podstatná nevýhoda přístupu prof. Kvasničky je ta, že disjunkce seznamů atomických proměnných je zahrnuta opravdu implicitně a je poměrně obtížné ji zjistit. Zvolením libovolných prvků ve formuli tak, že $t_1 = t_2 = \dots = t_n$ v podstatě „šidíme“ řešení nalezení seznamů, ovšem na úkor rychlého nalezení sporu ve formuli. Pokud bychom chtěli opravdu najít výslednou disjunkci atomických proměnných ve formuli, kde figuruje n konkrétních různých konstant, museli bychom učinit postupně $t = a_1, t = a_2, \dots, t = t = a_n$.

Obecnou konstrukční analýzu není třeba rozebírat, jelikož výsledky jsou vesměs stejné, ovšem s generováním dalších formulí v rámci vytváření instancí se počet větví ještě počet větví poměrově zvyšuje. [17]

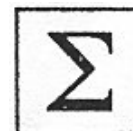
Kontrolní otázky:

1. Co je diagramatický přístup?
2. V čem je diagramatický přístup vhodný?
3. Které operace diagramatický přístup obsahuje?
4. Která technika je obecně využitelnější na komplexní třídu problémů?



Shrnutí:

Pro převod do DNF, či v predikátové logice převod formule na disjunkci seznamu atomických formulí je metoda M_1 bezkonkurenční pro své ekvivalence v rámci úrovní stromů. Zde můžeme o rychlosti tvorby tabla M_2 polemizovat, jelikož, jak již bylo řečeno, dohledávání atomických formulí je proces ne příliš nenáročný na to, aby byl zanedbatelným faktem. Naproti tomu, logická platnost nepotřebuje převod do DNF, z čehož těžší rezoluční algoritmus, proto je mocným nástrojem a nejpoužívanějším nástrojem logického programování. Pokud metodu M_2 zkombinujeme s prohledáváním stromu do hloubky, můžeme docílit rychlého výsledku nalezení sporu, ovšem tuto kombinaci by bylo vhodné volit spíše v programovém řešení, jelikož přívětivost takového přístupu může být poměrně nekomfortní. Ovšem i samotná metoda nabízí rychlé nalezení sporu, jelikož se nezabývá přepisováním formule do každého uzlu. Určování logického důsledku z dané teorie je pro svou převoditelnost do jazyka logiky spojitelné s určováním logické platnosti čili můžeme opět hovořit o silných stránkách metody M_2 v podobě rychlosti vypracování. Pro vytváření reverzního tabla v rámci přirozené dedukce je metoda M_1 uživatelsky daleko přívětivější, jelikož pouze provádí zpětné operace. Pro metodu M_2 jsou opět deklarovány diagramy s patřičnými pravidly a tvorba tabla je opravdu velice nepřehledná a vhodná spíše pro uživatele se zkušenostmi s touto metodou.



12 ANIMACE A JEJÍ PODPORA VE VÝUCE

V této kapitole se dozvíte:

- Které animace k výuce logiky použít.
- Kde dané animace naleznete.
- Co dané animace nabízejí.

Po jejím prostudování byste měli být schopni:

- Pracovat s danými animacemi.
- Orientovat se v instalaci.
- Využít animace k výuce.

Klíčová slova této kapitoly:

Animace, výroková, predikátová.

Doba potřebná ke studiu: 2 hodina



Průvodce studiem

Tato kapitola je věnována využití animací ve výuce logiky a vznikla díky studentům KIP OU, kteří své bakalářské práce zaměřili na oblast logiky a podpory její výuky. Animace jsou pro stromově reprezentující úlohy a zastupují oblast výrokové i predikátové logiky.

Tato kapitola popisuje práci s animacemi, které byly vytvořeny studenty Ostravské univerzity, jako jejich bakalářské práce. Oba typy animací slouží nejen jako teoretická pomůcka, ale i jako praktická ukázka toho, jak jednotlivé metody v obou typech logik pracují. Tyto práce můžete stáhnout na T:kinfo/Kotyrba/lzui1 nebo vyžádat emailem. Aplikace jsou rovněž uloženy na moodle a k jejich spuštění je potřeba software FlashPlayer nebo jiný.

12.1 Animace pro podporu výuky výrokové logiky

Celá interaktivní animace vznikla autorem Jiřím Hromadou, studentem KIP OU, jako bakalářská práce s cílem zkvalitnit studentům výuku logiky za pomoci animací. Animace je zaměřena na úlohy výrokové logiky pomocí stromové reprezentace, jejich řešení a opakování a seznámení se s teorií. Aplikace je vytvořena zejména pro studijní potřeby a obsah je tomu přizpůsoben. Uživatel se v aplikaci orientuje velmi rychle díky nenáročným graficím, která je podpořena jednoznačným textem provázejícím každý postupný krok animace. Interaktivnost animace spočívá v sadě tlačítek, kterými uživatel má možnost řídit zobrazování jednotlivých stránek animace. V aplikaci byla zvolena jednoduchá grafická podoba, aby uživatele neodváděla od pozornosti a tím lépe vnímal obsahovou složku animace. Jednotlivé části aplikace jsou od sebe barevně odlišeny. Základní zvolené barvy jsou červená pro výukovou část a modrá pro část procvičovací. Třetí doplňující barvou je jemný odstín oranžové, kterou je značen rámeček obsahující doprovodný text grafiky.

Veškerý text v animaci je stručný, vystihuje podstatu probírané látky a není psán složitě[5].



Obrázek 12.1: Nabídka výukové animace [5]

12.1.1 Výuková část

V části nazvané výuka je obsah soustředěn na představení a co nejúčinnější vysvětlení problematiky stromové reprezentace úloh výrokové logiky. Při stisknutí tlačítka výuka (obrázek 12.1) se zobrazí rozcestník, kde má uživatel možnost zvolit, kterou z oblastí stromové reprezentace chce zobrazit. Toto výběrové menu je rozděleno do čtyřokruhů[5]:

- Syntaktický formační strom,
- Sémantický strom,
- Quineův algoritmus,
- Sémantické tablo.

Pomocí dvou tlačítek „teorie“ a „vzorové příklady“ si uživatel u každé z těchto čtyřoblastí menu volí, zda chce zobrazit teorii vybrané problematiky, nebo zda se nechá animací provést názorným postupem vytváření stromů různých typů výrokových formulí [5].

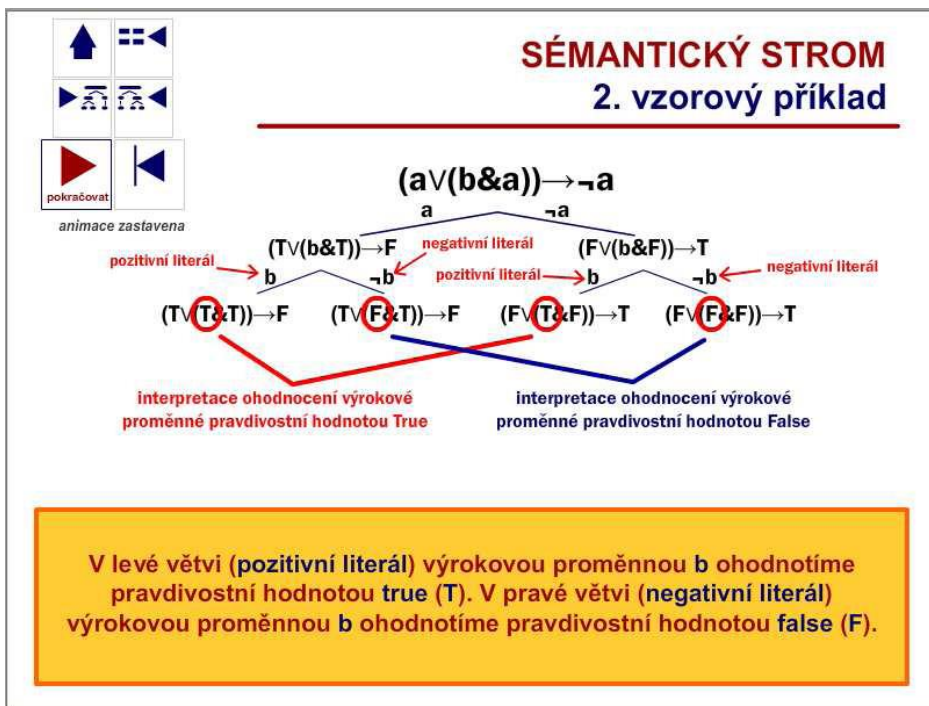


Obrázek 12.2: Výuková část [5]

Stručná teorie v této části animace se týká konkrétně vybraného stromového zobrazení. Text vychází hlavně z [1], [2], [3].

12.1.2 Vzorové úlohy

U každého typu stromové reprezentace byly vytvořeny vlastní vzorové příklady. Syntaktický formační strom obsahuje 5 příkladů různé obtížnosti, na kterých je demonstrován celý proces zpracování řešení zadané úlohy. Sémantické tablo má celkem 4 vzorové příklady, z toho 2 příklady pro ukázkou řešení úlohy přímým dokazováním a 2 pro nepřímé dokazování. Sémantický strom a Quineův algoritmus zahrnují každý 3 vypracované příklady [5].



Obrázek 12.3: Ukázka řešeného příkladu [5]

12.1.3 Procvičovací část

Výběrové menu procvičovací části je strukturou obdobné jako v části výukové a má stejné rozdělení všech okruhů. Každý ze čtyř okruhů má opět dvě tlačítka, tentokrát odkazující na lehčí a těžší příklady. Výjimku tvoří Sémantické tablo, které má tlačítka pro zobrazení příkladů pro přímý důkaz splnitelnosti formule i pro nepřímý důkaz, kterým dokazujeme logickou platnost formule [5].



Obrázek 12.4: Procvičovací část [5]

12.1.4 Typy příkladů

Příklady řešené Syntaktickým formačním stromem jsou rozděleny do dvou sekcí – lehčí a těžší, podle náročnosti a míry složitosti správného určení výsledku. Uživatel má možnost vypracovat 3 lehčí úlohy, které by pro uživatele neměly být tak obtížné na vyřešení, jako další 3 úlohy v těžší části. Stejně jako příklady v Syntaktickém formačním stromu, jsou členěny úlohy řešené Sémantickým stromem a pomocí Quinova algoritmu, tedy na 3 lehčí a 3 těžší. Metoda Sémantického tabla je rozdělena na důkazy přímé a nepřímé. Každý z těchto typů obsahuje 2 testové úlohy [5].

12.1.5 Tlačítka pro řízení animace

K řízení chodu interaktivní animace bylo třeba vytvořit sadu tlačítek. Aby kompaktní vzhled animace nebyl těmito kontrolními prvky narušen, vytvořená grafika je nenáročná a označení funkce tlačítka se zobrazí až při najetí myši. Tabulka ovládacích tlačítek užitých v animaci [5].

GRAFICKÁ PODOBA	GRAFICKÁ PODOBA PŘI NAJETÍ MYŠÍ	POPIS FUNKCE TLAČÍTKA
		zobrazí se úvodní stránka animace
		návrat na výběrové menu částí, ve které je animace postoupena
		zobrazí se příklad bezprostředně následující
		zobrazí se příklad bezprostředně předcházející
		pozastavená animace se spustí
		animace se o jeden krok vrátí zpět a zastaví se

Obrázek 12.5: Použité ovládací tlačítka [5]

Celá animace slouží pro lepší pochopení části výrokové logiky. Jasně demonstruje jednotlivé typy úloh, které se ve výrokové logice řeší. Moc děkuji autorovi, pevně věřím, že tato aplikace bude značným přínosem pro výuku.

12.2 Animace a její podpora v predikátové logice

Celá interaktivní animace vznikla autorem Petrem Veselým, studentem KIP OU, jako bakalářská práce s cílem zkvalitnit studentům výuku logiky za pomoci animací.

12.2.1 Jednotlivá menu

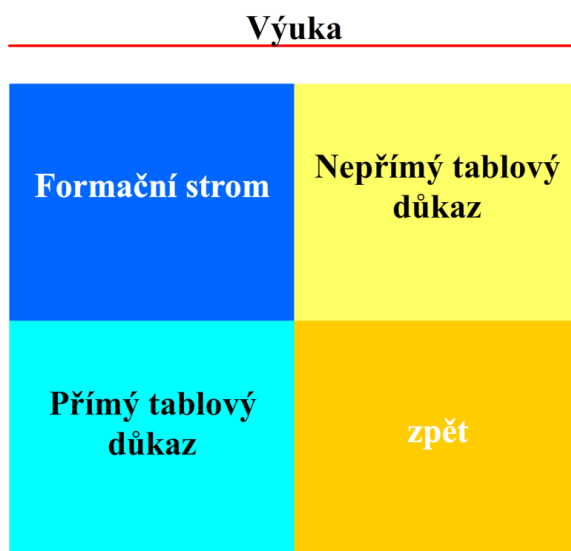
Úvodní menu animace je jednoduché a dává uživateli na výběr pouze ze dvou tlačítek. Prvním je nazváno *Výuka* a druhé *Příklady*, obrázek 12.6 [6].

Stromová reprezentace predikátové logiky



Obrázek 12.6: Úvodní menu aplikace [6]

Pomocí tlačítka *Výuka* se dostaneme do sekce, ve které je představeno, jak se řeší jednotlivá odvětví stromové reprezentace predikátové logiky. Je zde na výběr ze čtyř tlačítek, z jejichž názvu je patrné, které z odvětví predikátové logiky bude blíže představeno, obrázek 12.7[6].

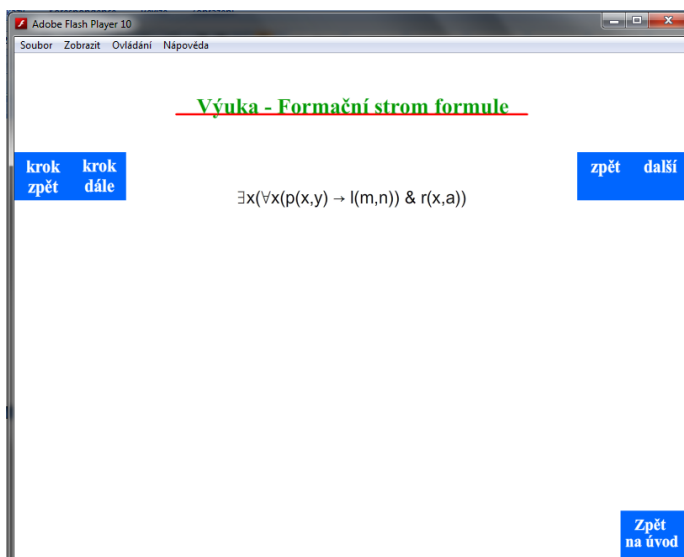


Obrázek 12.7: Menu výuka [6]

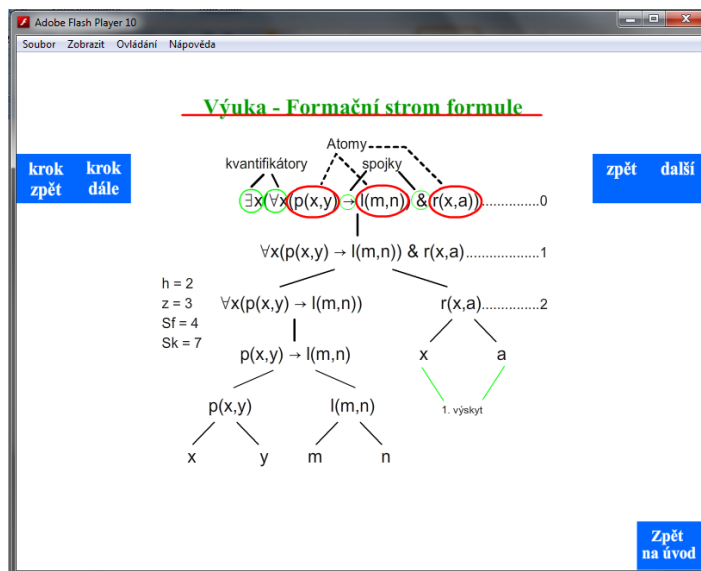
Pomocí čtvrtého tlačítka *zpět* je možno vrátit se zpět na úvodní obrazovku. Obdobně je řešena orientace pomocí druhého tlačítka úvodního menu. A také tlačítek pod menu, obrázek 12.7. Po vybrání některého z tlačítek, jež můžeme na obr. 12.8 nebo 12.9, se dostaneme již k samotným příkladům nebo zpět do úvodního menu [6].

12.2.2 Části animace

Zde jsou přiblíženy jednotlivé volby dvou hlavních částí animace. Orientace ve všech volbách části *Výuka*, je řešena pomocí pěti tlačítek. Pojmenovaných takto: *krok zpět*, *krok dále*, *zpět*, *dále* a *zpět na úvod*. První čtyři zmíněná tlačítka jsou zobrazena po celou dobu proklikávání se vybraným příkladem. Ovšem tlačítko *zpět na úvod* pouze na začátku a na konci příkladu. S úmyslem donutit uživatele se proklikat celým příkladem. Znázorňující obrázky 12.8 a 12.9 na náhodně vybraném výukovém příkladě [6].



Obrázek 12.8: Úvod příkladu [6]



Obrázek 12.9: Konec příkladu [6]

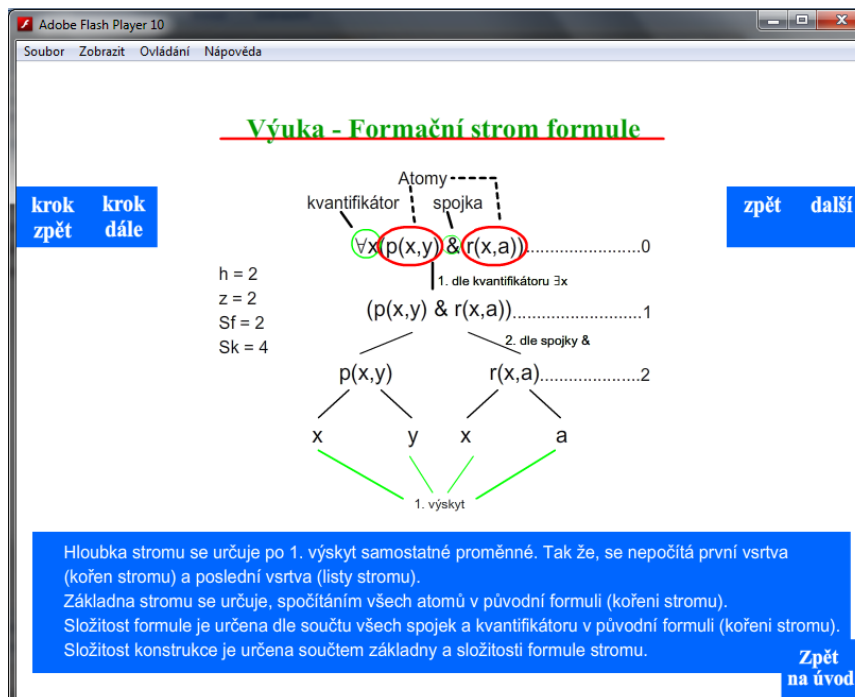
Dále ve všech volbách první části, jsou vždy v prvních dvou příkladech popsány postupy řešení jednotlivých formulí. Pro opakující se text, nejsou úmyslně uvedeny ve všech případech. Ukázky jsou obsahem následujících podkapitol [6].

12.2.3 Výuka

V této kapitole je představen jeden z prvních dvou příkladů animace. Aby byla zřejmá orientace. Řešení, jsou volena tak, aby nebyla překážkou.

- **Formační strom formule**

Na obrázku 12.10 je vidět, jak jsou situovány a voleny umístění jednotlivé částí animace, včetně výše zmíněného vysvětlení postupu řešení a samotného ohodnocování částí stromu.



Obrázek 12.10: Výuka formační strom [6]

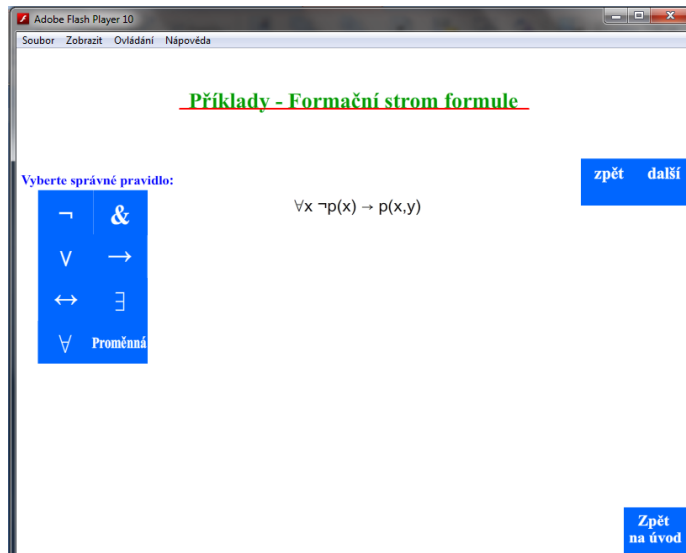
Obrázek 12.10 dále ukazuje, kde jsou situována ovládací tlačítka. Tlačítko *krok zpět* slouží ke krokování právě procházeného příkladu po krocích zpět a tlačítko *krok dále*, pro posun (krokování) vpřed. Další dvě tlačítka, zobrazená vpravo nahoře jsou určena, pro přepínání mezi jednotlivými příklady vybrané části animace. Toto chování, je pro všechny typy příkladů ve výukové části stejné, a proto jsou pro další volby [6].

12.2.4 Příklady

Tato kapitola se zabývá částí *Příklady*. Tato část animací je řešena záměrně rozdílně oproti předešlé z důvodu dosáhnout větší motivace uživatele přemýšlet, při řešení jednotlivých příkladů. Na základě toho co se dověděl, v části *Výuka* [6].

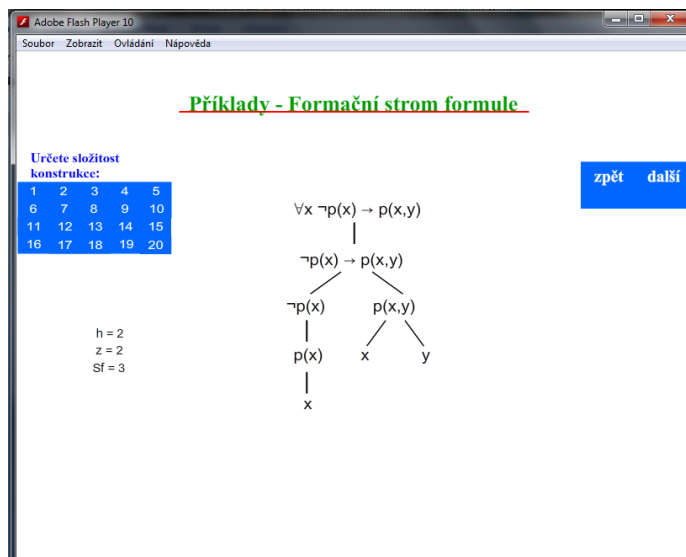
- **Formační strom formule**

Nejprve tedy formační strom formule, jednotlivá tlačítka pro tuto volbu, části *Příklady*, jsou vidět na obrázku 12.11.



Obrázek 12.11: Příklad s možností výběru [6]

Tlačítka spojek, kvantifikátorů a tlačítko *Proměnná*, slouží k samotnému řešení zvoleného příkladu. Při správné volbě tlačítka provedou, k čemu jsou určena. V opačném případě vybrané tlačítko po kliknutí zčerveneá. Toto chování je stejné pro celou část *Příklady*. Tlačítka *zpět*, *další* a *Zpět na úvod* mají stejnou funkci jako v části *Výuka*. Postupným řešením příkladů, se nám zobrazí ještě další sada tlačítek s čísly. Ta slouží k ohodnocování jednotlivých částí stromů, jako jsou pro připomenutí, hloubka, základna, složitost formule a složitost konstrukce, obrázek 12.12[6].



Obrázek 12.12: Formační strom formule [6]

Pro řešení přímých a nepřímých tablových důkazů jsou tlačítka stejná. Umožňují uživateli zjišťovat splnitelnost, nesplnitelnost nebo logickou platnost. Dále rozhodovat se v jednotlivých úrovních sémantického tabla pomocí určitého pravidla. Celá animace slouží pro lepší pochopení části predikátové logiky. Jasně demonstruje jednotlivé typy úloh, které se v predikátové logice řeší. Moc děkuji autorovi, pevně věřím, že tato aplikace bude značným přínosem pro výuku.

13 Cvičebnice

V této kapitole se dozvíte:

- Procvičíte teorii i příklady.
- Dozvíte se správné výsledky.

Po jejím prostudování byste měli být schopni:

- Aplikace jednotlivých příkladů a teorie.

Klíčová slova této kapitoly:

Procvičování teorie a příkladů.

Doba potřebná ke studiu: 1 hodina



Průvodce studiem

V této lekci je pro Vás připraveno množství teoretických příkladů k procvičení vztahující se především k teoretické problematice.

Otázky k testování

Co jsou to informace?

- a) datové řetězce
- b) *datové řetězce s přiřazeným významem*
- c) informace vzájemně začlenitelné a odvoditelné

Jaké rozlišujeme typy znalostí?

- a) *procedurální a deklarativní*
- b) znalostní a procedurální
- c) deklarativní a znalostní

Jak se nazývá smysluplné přiřazení datům jejich významu?

- a) ekvivalence
- b) *interpretace*
- c) idempotence

Negace věty: „Všichni studenti jsou geniální.“ Je:

- a) Žádný student není geniální.
- b) *Alespoň jeden student není geniální.*
- c) Všichni studenti jsou negeniální.

Negace věty: „Maminka vaří a peče.“ Je:

- a) Maminka nevaří a nepeče.
- b) *Maminka buď nevaří, nebo nepeče.*
- c) Maminka vaří, peče a smaží.

Co je to denotace?

- a) pokud objektu přiřadíme určitý výraz
- b) pokud výrazu přiřadíme objekt
- c) pokud objektu přiřadíme jiný objekt

Co je to výrok?

- a) tvrzení, ve kterém lze určit pravdivostní hodnotu
- b) tvrzení, ve kterém nelze určit pravdivostní hodnotu
- c) jakékoliv tvrzení

Která logická spojka se jinak nazývá logický součet?

- a) konjunkce
- b) disjunkce
- c) implikace

Která binární spojka tvoří nejtěsnější vazbu?

- a) konjunkce
- b) disjunkce
- c) ekvivalence

Co neurčujeme u syntaktického stromu formule?

- a) hloubku
- b) složitost
- c) šířku

Negace splnitelné formule je:

- a) logicky platná
- b) nesplnitelná
- c) splnitelná

Je negace logicky platné formule splnitelná?

- a) ano
- b) ne
- c) záleží na dané formuli

Která formule je ekvivalentní k formuli $(k \rightarrow l)$

- a) $k \& l$
- b) $k \leftrightarrow l$
- c) $\neg k \vee l$

Je pravda, že každou výrokovou formuli lze převést na úplnou disjunktivní normální formu?

- a) ano
- b) ne
- c) záleží na dané formuli

Mezi interpretační pravidla výrokové logiky nepatří:

- a) báze
- b) indukce
- c) industrializace
- d) generalizace

Jak funguje rezoluční odvozovací pravidlo (máme $(C \vee k)$ a $(\neg k \vee D)$?

- a) komplementární pár se vynechá a zapiší se zbylé proměnné $(C \vee D)$
- b) zůstane jen komplementární pár $(k \vee \neg k)$
- c) komplementární pár i proměnné se vyruší a vznikne prázdná klauzule

Která formule je ekvivalentní k formuli $\neg(k \& l)$

- a) $\neg k \vee \neg l$
- b) $k \vee l$
- c) $k \& \neg l$

Jak se určuje splnitelnost (log. platnost) pomocí Quineova algoritmu?

- a) postupně se ohodnocuje strom, na jednu stranu true, na druhou false
- b) pomocí alfa a beta pravidel
- c) jen se rozkládá strom na základě logických spojek

Na kolik větví rozdělují formuli beta pravidla?

- a) jednu
- b) dvě
- c) tři

Jak ověřujeme tablovou metodou logickou platnost?

- a) přímým důkazem
- b) nepřímým důkazem
- c) nezáleží na tom

Při všech uzavřených větvích tabla nepřímým důkazem jsme:

- a) došli ke sporu a tím ověřili logickou platnost
- b) došli ke sporu a tím ověřili nesplnitelnost formule
- c) se zacyklili a formule je tudíž nesplnitelná

Co je to formule v klauzulární formě?

- a) je to formule v konjunktivní normální formě
- b) je to formule v disjunktivní normální formě
- c) je to formule, kterou ještě musíme upravit

Co je to bazový term?

- a) term obsahující některé proměnné
- b) term neobsahující žádné proměnné
- c) term obsahující všechny proměnné

Která z možností obsahuje predikát?

- a) x
- b) true
- c) rodina(matka, otec, dítě)

Jaké kvantifikátory existují v predikátové logice?

- a) existenční a časový
- b) časový a univerzální
- c) existenční a univerzální

Predikátová logika má:

- a) větší expresivitu, než logika výroková
- b) menší expresivitu, než logika výroková
- c) stejnou expresivitu, jako logika výroková

Kolik typů symbolů užívá predikátová logika?

- a) 3
- b) 8
- c) 17

Kolik tablových pravidel se používá v predikátové logice?

- a) 2 – alfa a beta
- b) 4 – alfa, beta, gama a delta
- c) 6 – alfa, beta, gama, delta, epsilon a zeta

Jak se nazývá odstranění existenčních kvantifikátorů a zavedení nových konstant?

- a) skolemizace
- b) herbrand
- c) rezoluce

Prenexní operace v predikátové logice:

- a) slouží k převodu formule do klauzulární formy
- b) slouží k převodu formule do výrokové logiky
- c) slouží k převodu formule do prenexní normální formy

PŘEVZATO Z [4].

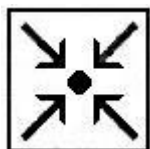
LITERATURA K DOPORUČENÍ

- [1] Lukasová Alena, Logické základy umělé inteligence 1,2. Skriptum. Ostravská univerzita v Ostravě. 2004.
- [2] Lukasová, A. Formální logika v umělé inteligenci. Praha: Computer Press, 2003.
- [3] Lukasová, A. Telnarová, Z. Habiballa, H. Vajgl, M. Formální reprezentace znalostí. první. vyd. Ostrava: Universum, 2010. 345 s. dvanáctý svazek edice Universum. ISBN 978-80-7368-900-1.
- [4] Rombová Z.: Využití moderních multimediálních prvků výuky v logice, Bakalářská práce, 2011. Ostravská univerzita v Ostravě.
- [5] Hromada J.: Hromada J.: Podpora stromové reprezentace úlohvýrokové logiky. Bakalářská práce, 2011. Ostravská univerzita v Ostravě.
- [6] Veselý P.: Stromová reprezentace úloh predikátové logiky. Bakalářská práce, 2012. Ostravská univerzita v ostravě.
- [7] Codd, E.F.: The Relational Model for Database Management. Addison-Wesley, 1990.
- [8] Gallier, J.H.: Logic for Computer Science. Foundations of Automatic Theorem Proving. John Willey & Sons, 1987.
- [9] Genesereth, M.R., Nilsson, N.J. : Logical Foundation of Artificial Intelligence.
- [10] Gray, P.M.D.: Logic, Algebra and Databases. Ellis Horwood Ltd., 1984.
- [11] Kleene, S.C.: Mathematical Logic. John Willey & Sons, 1967.
- [12] Kleene, S.C.: Introduction to Metamathematics. Wolters-Noordhoff Publishing, 1971.
- [13] Kolář, J., Štěpánková, O., Chytil, M.: Logika, algebry a grafy. SNTL/Alfa Praha, 1989.
- [14] Manna, Z.: Matematická teorie programů. SNTL Praha, 1981.
- [15] Mendelson, E. : Introduction to Mathematical Logic. D. Van Nostrand Comp., Inc., 1966.
- [16] Štěpán, J.: Logika a logické systémy. Votobia, Olomouc 1992.
- [17] Rožňák M.: Sémantické tablo jako formální procedura důkazu v logice I. řádu, bakalářská práce, Ostravská univerzita, 2013.

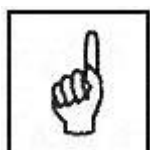
Vysvětlivky k používaným symbolům



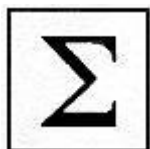
Průvodce studiem – vstup autora do textu, specifický způsob kterým se studentem komunikuje, povzbuzuje jej, doplňuje text o další informace.



Příklad – objasnění nebo konkretizování problematiky na příkladu ze života, z praxe, ze společenské reality apod.



K zapamatování



Shrnutí – shrnutí předcházející látky, shrnutí kapitoly.



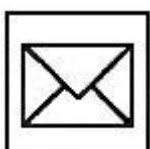
Literatura – použita ve studijním materiálu, pro doplnění a rozšíření poznatků.



Kontrolní otázky a úkoly – prověřují, do jaké míry studující text a problematiku pochopil, zapamatoval si podstatné a důležité informace a zda je dokáže aplikovat při řešení problémů.



Úkoly k textu – je potřeba je splnit neprodleně, neboť pomáhají k dobrému zvládnutí následující látky.



Korespondenční úkoly – při jejich plnění postupuje studující podle pokynů s notnou dávkou vlastní iniciativy. Úkoly se průběžně evidují a hodnotí v průběhu celého kurzu.



Otázky k zamyšlení



Část pro zájemce – přináší látku a úkoly rozšiřující úroveň základního kurzu. Pasáže i úkoly jsou dobrovolné.