

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

BAKALÁŘSKÁ PRÁCE

Program na podporu výuky logiky



2014

Mgr. Iva Navrátilová

Anotace

Cílem této práce bylo prostudovat téma „Výroková logika“ a poté vytvořit program na podporu výuky části předmětu Matematická logika. Vstupem je posloupnost symbolů, u které program pozná, zdali je či není formule výrokové logiky. Posloupnost obsahuje nejvýše čtyři různé výrokové symboly. Hlavní důraz byl kladen na co možná nejjednodušší ovládání a názornost, aby byl program určen komukoliv, kdo by se rád seznámil se základními pojmy výrokové logiky. Program obsahuje i výukový text poskytující uživateli informace o těchto základních pojmech.

Děkuji všem, kteří přispěli připomínkou či podnětem k výsledné podobě této práce. Zvláště pak svému vedoucímu práce RNDr. Miroslavu Kolaříkovi, Ph.D. za ochotu, podnětné rady, vstřícnost a čas, který mi věnoval při konzultacích.

Obsah

1. Úvod	7
2. Teoretická část	8
2.1. Co je to logika	8
2.2. Výroková logika	8
2.3. Jazyk a formule výrokové logiky	10
2.4. Pravdivostní ohodnocení výroku	11
2.5. Tabulková metoda	12
2.6. Sémantické vyplývání	13
2.7. Normální formy výrokových formulí	14
2.7.1. Disjunktivní normální formy formulí	14
2.7.2. Konjunktivní normální formy formulí	15
2.8. Úplný systém logických spojek	15
2.8.1. Úplný systém spojek $\{\neg, \wedge\}$	16
2.8.2. Úplný systém spojek $\{\neg, \vee\}$	16
2.8.3. Úplný systém spojek $\{\neg, \Rightarrow\}$	17
2.8.4. Úplný systém spojek $\{\Downarrow\}$	17
2.8.5. Úplný systém spojek $\{\Uparrow\}$	17
3. Uživatelská dokumentace	18
3.1. Systémové požadavky	18
3.2. Instalace a spuštění aplikace	18
3.3. Popis aplikace	18
3.3.1. Hlavní menu	18
3.3.2. Formule výrokové logiky	20
3.3.3. Záložky	20
3.4. Ovládání aplikace	20
3.4.1. Zadání formule výrokové logiky	20
3.4.2. Vytvoření tabulek, převod	21
3.5. Ukončení aplikace	24
3.6. Odinstalování aplikace	24
4. Programátorská dokumentace	25
4.1. Použité technologie	25
4.2. Popis tříd	25
Závěr	29
Conclusions	30
Reference	31

5. Obsah příloženého CD/DVD

32

Seznam obrázků

1.	Pravdivostní hodnoty formulí.	12
2.	Pravdivostní hodnoty spojek \Downarrow, \Uparrow	16
3.	Hlavní okno aplikace.	19
4.	Okno zadávání formule výrokové logiky.	20
5.	Hlavní okno aplikace - záložka Tabulková metoda.	21
6.	Hlavní okno aplikace - záložka Sémantické vyplývání.	22
7.	Hlavní okno aplikace - záložka ÚDNF.	23
8.	Hlavní okno aplikace - záložka Převod formule.	24

1. Úvod

Práce popisuje tvorbu programu sloužícího pro podporu výuky předmětu Matematická logika. Se základními pojmy výrokové logiky se setkáváme již na střední škole. Z důvodu zopakování či připomenutí jsou v následujícím textu definovány všechny pojmy vyskytující se v programu i výukovém textu, který je povinnou součástí programu.

Zásady pro vypracování programu:

- pozná, zda zadaná posloupnost symbolu je či není formule výrokové logiky
- bude umět demonstrovat pomocí tabulkové metody pojmy: tautologie, kontradikce, . . .
- dokáže posoudit sémantické vyplývání jedné formule z druhé a naopak
- bude umět převádět zadané formule do úplné disjunktivní normální formy
- bude umět převádět zadané formule do úplné konjunktivní normální formy
- převede formuli na sémanticky ekvivalentní formule s danými výrokovými spojkami
- demonstrace bude prováděna na formulích majících nejvýše čtyři různé výrokové symboly
- součástí práce bude výukový text o výše zmíněných pojmech.

2. Teoretická část

2.1. Co je to logika

Třebaže je obtížné najít univerzálně přijatelnou definici, můžeme zjednodušeně říci, že logika je (především) věda o správném usuzování, o umění správné argumentace, o korektním vyvozování důsledků z daných předpokladů. Logika zkoumá pojmy, jako jsou pravdivost, dokazatelnost, vyvratitelnost a zabývá se jejich vzájemnými vztahy.

Přirozené jazyky jsou složité a často nejednoznačné. Ve vědách jako je matematika, fyzika, informatika, potřebujeme naše znalosti vyjadřovat přesným a jednoznačným způsobem.

Logika, jako vědní disciplína, se vyvíjela během staletí a vyvíjely se i její cíle. Za zakladatele logiky je všeobecně pokládán řecký filosof Aristotelés (384 – 322 př.n.l.), který svou naukou o sylogismech položil základy systematickému zkoumání našeho způsobu vyvozování důsledků v pětisvazkovém díle Organon (Nástroj). Zkoumal také pravdivostní módy a položil tak základy modální logiky. Zkoumání logiky se rozvíjelo také v megarsko-stoické škole, za jejíhož prvního představitele se uvádí Eukleidés z Megary (†360 př.n.l.). Rozkvět logiky u Řeků trval asi 150 let. Na antické výsledky navázali středověcí myslitelé, např. formulací dalších principů logiky. Vrchol středověké logiky je kladen do 13. století, po 14. století nastává útlum. [8]

Na přelomu 19. a 20. století došlo k výrazné formalizaci logiky, začíná rozvoj matematické (nazývané též symbolické) logiky, vyvolaný zejména potřebou upřesnit základy matematiky. Používání matematických metod a s ním spojená idealizace a formalizace našeho vyvozování se staly jedním z nejdůležitějších rysů matematické logiky.

2.2. Výroková logika

Výroková logika umožňuje analyzovat věty až do úrovně elementárních výroků (výrokové symboly). Strukturu těchto elementárních výroků již dále nezkoumá a jediným jejich atributem je pravdivostní hodnota.

Definice 1. *Výrok je vyslovené nebo napsané tvrzení, o kterém má smysl rozhodnout, zda je pravdivé nebo nepravdivé, přičemž musí nastat právě jedna z těchto dvou možností (podle toho pak výrok můžeme nazývat pravdivým nebo nepravdivým).*

Tato „definice“ se zdá být až banální, pokud si neuvědomíme, že ne každá věta vyjadřuje výrok. Klasická výroková logika ctí princip dvojhodnotovosti (tercium non datur – Chrisipos ze Solov) [2]. Například „číslo 6 je dělitelné 2“ je

výrok, který je pravdivý. Naproti tomu „Slunce obíhá kolem Země“ je nepravdivý výrok. Některé výroky jsou pravdivé za všech okolností, např. „kočka je černá nebo není černá“, jiné jsou nepravdivé za všech okolností, např. „kočka je černá a zároveň není černá“. Není nutné okamžitě vědět, zda je dané tvrzení pravdivé nebo nepravdivé, abychom o něm řekli, že se jedná o výrok. Musí ale být smysluplné zabývat se otázkou pravdivosti tohoto tvrzení, jinak řečeno musí existovat cesta, jak se k pravdivosti tvrzení dobrat. Může to být oznamovací věta přirozeného jazyka, tvrzení vyjádřené matematickým zápisem (symboly) nebo kombinace obojího.

Příklad 1. *„V roce 1998 získala hokejová reprezentace České republiky zlatou medaili na olympijských hrách v Naganu.“ – uvedená věta je výrokem. Dokonce je pravdivým výrokem, jehož pravdivost si můžeme snadno ověřit nahlédnutím do historických sportovních tabulek.*

Příklad 2. *„ $4 < 5$ “ – opět máme výrok. Ačkoli tak na první pohled nemusí vypadat, jedná se vlastně o zkrácený zápis jednoduché věty, která říká: „Číslo čtyři je menší než číslo pět.“ Taková věta je samozřejmě pravdivá.*

Příklad 3. *„Sedni si!“ – uvedené tvrzení je větou rozkazovací. U rozkazu nemá smysl hovořit o pravdivosti. Můžeme uvažovat pouze o tom, zda bude či nebude splněn, ale to nesouvisí s pravdivostí. Toto tvrzení není výrokem.*

Příklad 4. *„Učitel drží v ruce křídu.“ – při prvním pohledu bychom se mohli nechat unést myšlenkou, že jde o výrok. Vždyť se stačí na učitele podívat a hned víme, zda křídu skutečně drží. Zde narážíme na první rozdíl mezi běžným a matematickým jazykem. V takové situaci ovšem nemůžeme vědět, o kterém z milionů učitelů se tato věta vyjadřuje. Nemá tedy smysl zabývat se pravdivostí tohoto tvrzení, protože nikdy nebudeme vědět, na kterého z učitelů se podívat. Věty obsahující takový „nezjistitelný“ prvek nebudeme za výroky považovat.*

Příklad 5. *„ $x > 10$ “ – co máme dosadit za x ? Tvrzení opět není dostatečně přesné. Abychom mohli hovořit o pravdivosti takového tvrzení, museli bychom vzít v úvahu nějaké další dodatečné informace. Tvrzení tedy není výrokem.*

Výroky dělíme na jednoduché a složené. Výroky, o nichž jsme doposud mluvili, se nazývají jednoduché výroky. Elementární (jednoduchý) výrok je takové tvrzení, jehož žádná vlastní část již není výrokem. Složený výrok pak má vlastní části – výroky.

Příklad 6. *V Praze prší a v Brně je hezky - složený výrok*

Výroková logika zkoumá strukturu těchto složených výroků v tom smyslu, že zkoumá způsob skládání jednoduchých výroků do složených pomocí logických spojek. Je to tedy teorie logických spojek. Přitom ovšem zachovává žádoucí princip skladebnosti (kompozicionality), podle něhož je pravdivostní hodnota složeného výroku jednoznačně určena jen pravdivostními hodnotami jeho složek a povahou spojení těchto složek (tj. logickou povahou spojek).

2.3. Jazyk a formule výrokové logiky

Jazyk výrokové logiky musí obsahovat symboly zastupující jednotlivé elementární výroky, tzv. výrokové symboly (proměnné), které budou nabývat hodnot pravda, nepravda, symboly pro logické spojky a případné pomocné symboly. Každý jazyk je nejlépe definován abecedou a gramatikou. Abeceda určuje typy symbolů, které mohou být v jazyce použity, gramatika pak zadává pravidla tvorby (nekonečně mnoha) dobře vytvořených výrazů, tj. v tomto případě formulí.

Definice 2. *Abeceda jazyka výrokové logiky je množina následujících symbolů:*

- výrokové symboly: p, q, r, s, \dots (případně i s indexy),
- symboly logických spojek (funktorů): $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$,
- pomocné symboly (závorky): $(,)$ (případně $[,], ,$).

Symboly $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$, nazýváme po řadě funktoři negace, konjunkce, disjunkce, implikace, ekvivalence.

Definice 3. *Gramatika jazyka výrokové logiky rekurzivně definuje formule:*

- (1) *Výrokové symboly jsou formule.*
- (2) *Jsou-li výrazy φ, ψ formule, pak jsou formulemi i výrazy $(\neg\varphi)$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \Rightarrow \psi)$, $(\varphi \Leftrightarrow \psi)$.*
- (3) *Každá formule je pouze výsledkem konečného počtu užití pravidel (1), (2). Jiné formule výrokové logiky nejsou.*

Jazyk výrokové logiky je množina všech formulí výrokové logiky. Formule vzniklé podle bodu (1) nazýváme elementárními (atomárními, primitivními) formulemi a formule vzniklé podle bodu (2) složenými formulemi [2]. Část formule, která je sama opět formulí, se nazývá podformule.

Příklad 7. *Podformulemi formule $(p \Rightarrow q) \Rightarrow (\neg q \Rightarrow p)$ jsou například p , $(p \Rightarrow q)$ a $(\neg q \Rightarrow p)$.*

Předchozí definice popisuje způsob jakým se konstruují složitější výrokové formule z formulí již sestrojených. Jde o induktivní definici opírající se o konstruktivní pravidla (1), (2) a uzavírací klauzuli (3), která zaručuje, že každá výroková formule je konečné slovo [3].

Pomocné symboly, které používáme při zápisu formulí slouží především k lepší čitelnosti těchto výrazů. I přesto, že definice stanoví psaní závorek jednoznačně, můžeme si při zápisu dovést určitou volnost pokud to nebude na újmu srozumitelnosti. Je například obvyklé vynechávat krajní párové závorky, které jsou definicí formule předepsány, ale ve skutečnosti nic neoddělují.

Abychom zpřehlednili zápis formulí, přijmeme tedy konvenci o vynechávání vnějších závorek. Formule tvaru $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \Rightarrow \psi)$, $(\varphi \Leftrightarrow \psi)$ budeme psát bez vnějších závorek, to jest $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \Rightarrow \psi$, $\varphi \Leftrightarrow \psi$. Závorky, které mohou být uvnitř formulí φ , ψ , ale nevypouštíme.

Příklad 8. *Například místo formule $((p \wedge q) \vee r)$ budeme psát $(p \wedge q) \vee r$.*

2.4. Pravdivostní ohodnocení výroku

Zatím jsme se formulím věnovali pouze z pohledu jejich tvaru, případně zápisu a čtení, ale doposud se nebavili o jejich pravdivosti či nepravdivosti. Část výrokové logiky, která se zabývá formulemi pouze jako řetězci symbolů jazyka, se nazývá syntaxe. Syntaxe výrokové logiky nedefinuje pravdivost formulí ani význam spojek. Formule samy o sobě nemají žádný význam. Přiřazení významu syntaktickým objektům je záležitostí další části výrokové logiky – sémantiky [1].

Definice 4. *Pravdivostní ohodnocení výroku říká, zda je výrok pravdivý či nepravdivý. Je-li výrok pravdivý, pak můžeme také říct, že výrok platí. Je-li nepravdivý, můžeme říct, že výrok neplatí.*

Uvědomme si, že při zkoumání pravdivostí výrokových formulí není důležité, jak logické spojky čteme, nebo jak je značíme, ale důležité je, jaký jim dáme význam.

Definice 5. *Je samozřejmé, že definice spojek kopíruje jejich jazykový význam. Říkáme tedy, že:*

- formule $\neg\varphi$ je pravdivá, právě tehdy, když je φ nepravdivá,
- formule $\varphi \wedge \psi$ je pravdivá, právě tehdy, když jsou obě formule φ i ψ pravdivé,
- formule $\varphi \vee \psi$ je pravdivá, právě tehdy, je-li alespoň jedna z formulí φ nebo ψ pravdivá,
- formule $\varphi \Rightarrow \psi$ je pravdivá, právě tehdy, je-li pravdivá φ , pak je nutně pravdivá i ψ ,
- formule $\varphi \Leftrightarrow \psi$ je pravdivá, právě tehdy, když jsou obě formule φ a ψ pravdivé nebo jsou obě formule φ a ψ nepravdivé.

Pro zkrácení zápisu se často používá číselné vyjádření pravdivostního ohodnocení. Výrokům se přiřazují tzv. pravdivostní hodnoty, pravdivému výroku pravdivostní hodnota 1 a nepravdivému pravdivostní hodnota 0.

Definice 6. *Pravdivostní ohodnocení (valuace) výrokových symbolů je zobrazení v , které ke každému výrokovému symbolu přiřazuje pravdivostní hodnotu, tj. hodnotu z množiny $\{1, 0\}$, která kóduje množinu pravda, nepravda: $\{p_i\} \rightarrow \{1, 0\}$. Pravdivostní funkce formule výrokové logiky je funkce w , která pro každé pravdivostní ohodnocení výrokových symbolů přiřazuje formuli její pravdivostní hodnotu.*

Tato hodnota je určena takto:

- pravdivostní hodnota elementární formule: $w(p)_v = v(p)$ pro všechny výrokové proměnné p ,
- jsou-li dány pravdivostní funkce formulí φ, ψ , pak pravdivostní funkce formulí, $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \Rightarrow \psi, \varphi \Leftrightarrow \psi$ je možné vyčíslit z následujícího obrázku.

$\neg\varphi$		$\varphi \wedge \psi$			$\varphi \vee \psi$			$\varphi \Rightarrow \psi$			$\varphi \Leftrightarrow \psi$		
0	1	0	0	0	0	0	0	0	0	1	0	0	1
1	0	1	0	0	1	0	1	1	0	0	1	0	0
		0	1	0	0	1	1	0	1	1	0	1	0
		1	1	1	1	1	1	1	1	1	1	1	1

Obrázek 1. Pravdivostní hodnoty formulí.

2.5. Tabulková metoda

Oblíbeným způsobem pro zjišťování pravdivostních hodnot formulí je využívání univerzální tabulkové metody, kterou vytváříme tabulky pravdivostních hodnot (zavedl Charles Sanders Peirce, 1839 – 1914) [6]. Tato metoda představuje jednoduchý způsob, jak přehledně zapsat pravdivostní hodnoty dané formule při všech možných ohodnoceních. Obsahuje-li ověřovaná formule celkem n výrokových proměnných, pak příslušná tabulka má 2^n možností, jak dosadit za tyto proměnné pravdivostní hodnoty $\{1, 0\}$.

Je vhodná pouze pro formule s malým počtem výrokových proměnných, protože jak již bylo uvedeno má 2^n řádků, tj. v případě dvou proměnných $2^2 = 4$ řádků, v případě tří proměnných $2^3 = 8$ řádků, v případě čtyř proměnných $2^4 = 16$ řádků. V každém řádku tabulky musíme vyčerpát jednu možnou distribuci pravdivostních hodnot na výrokové proměnné, přičemž na žádnou možnost nesmíme zapomenout.

Postup sestavení tabulky je takový, že sestavíme tabulku o $2^n + 1$ řádek (pro popis záhlaví). Počet sloupců závisí na konkrétní formuli, minimální počet je $n+1$,

do prvních sloupců záhlaví tabulky vypíšeme n výrokových symbolů, se kterými formule pracuje (p, q, r, s, \dots) a do dalších rozepíšeme dílčí formule.

Pod záhlaví do jednotlivých řádků vyplníme všechny kombinace ohodnocení symbolů, opět na prvních n místech, zapisujeme postupně všechny možnosti dosazení hodnot $\{1, 0\}$ za výrokové proměnné, na jejichž základě postupným používáním spojek dopočteme i zbytek jedniček a nul. Abychom zjistili pravdivostní hodnotu formule (při dané valuaci), musíme tedy nejprve zjistit pravdivostní hodnotu jednotlivých podformulí. V posledním sloupci je odpovídající pravdivostní hodnota uvažované formule.

Poté z tabulky můžeme vyčíst, zda je formule splnitelná, tautologie, kontradikce nebo pomocí ní tvoříme úplnou konjunktivní a disjunktivní normální formu.

Definice 7. Každé ohodnocení v výrokových symbolů obsažených ve formuli φ , pro které je hodnota pravdivostní funkce rovna 1, tedy $w(\varphi)_v = 1$, se nazývá model této formule.

Definice 8. Formule φ výrokové logiky je splnitelná, je-li $w(\varphi)_v = 1$ pro nějaké ohodnocení v , neboli existuje aspoň jeden model formule φ .

Definice 9. Formule φ výrokové logiky je tautologií (logickým zákonem), je-li $w(\varphi)_v = 1$ pro všechna ohodnocení v , neboli každé ohodnocení je modelem formule φ . Skutečnost, že formule φ je tautologií, označujeme zápisem $\models \varphi$.

Definice 10. Formule φ výrokové logiky je kontradikcí, jestliže neexistuje takové ohodnocení výrokových symbolů, pro které by hodnota pravdivostní funkce formule φ byla rovna 1, tj. $w(\varphi)_v = 0$ pro všechna ohodnocení v , formule nemá model.

2.6. Sémantické vyplývání

Definice 11. Jsou-li φ, ψ dvě formule výrokové logiky, říkáme, že formule ψ sémanticky plyne z formule φ , nebo že ψ je sémantickým důsledkem φ , jestliže ψ je pravdivá při každém ohodnocení, při kterém je pravdivá φ (značíme $\varphi \models \psi$).

Definice 12. Pokud ψ sémanticky plyne z φ a naopak, říkáme, že φ a ψ jsou sémanticky ekvivalentní (značíme $\varphi \equiv \psi$).

Je-li φ tautologie, pak dříve zavedená notace $\models \varphi$ odpovídá tomu, že φ platí při každém ohodnocení.

Pro libovolné formule výrokové logiky platí:

- jestliže $\varphi \equiv \psi$, pak $\varphi \models \psi$,
- $\varphi \equiv \psi$, právě když $\varphi \models \psi$ a $\psi \models \varphi$,
- jestliže $\varphi \models \psi$ a $\varphi \equiv \varphi'$ a $\psi \equiv \psi'$, pak $\varphi' \models \psi'$,

- jestliže $\varphi \models \psi$ a $\psi \models \chi$, pak $\varphi \models \chi$,
- $\varphi \models \varphi$.

Další vlastnosti vztahu vyplývání:

- $\varphi \models \psi$ právě tehdy, když $\models \varphi \Rightarrow \psi$,
- $\varphi \models \psi$ právě tehdy, když $\varphi \Leftrightarrow \psi$,
- každé dvě tautologie (i každé dvě kontradikce) jsou sémanticky ekvivalentní,
- formule je tautologie právě tehdy, když je sémanticky ekvivalentní s tautologií,
- je-li ϑ nějaká tautologie, pak pro každou formuli φ je $\varphi \models (\varphi \wedge \vartheta)$,
- Je-li ξ nějaká kontradikce, pak pro každou formuli φ je $\varphi \models (\varphi \wedge \xi)$.

Mezi každými dvěma formulemi můžeme rozhodnout o vztahu vyplývání nebo ekvivalence pomocí pravdivostních tabulek.

2.7. Normální formy výrokových formulí

Mezi normální formy výrokových formulí řadíme úplná disjunktivní normální forma (UDNF) a úplná konjunktivní normální forma (UKNF), které nazýváme harmonickými (standardními) tvary této formule.

- Každou formuli, která není kontradikcí, lze vyjádřit ve tvaru UDNF.
- Každou formuli, která není tautologií, lze vyjádřit ve tvaru UKNF.

Definice 13. *Literál je výrokový symbol nebo jeho negace.*

- *Elementární konjunkce (EK) je konjunkce literálů.*
- *Elementární disjunkce (ED) je disjunkce literálů.*

2.7.1. Disjunktivní normální formy formulí

Definice 14. *Disjunktivní normální forma (DNF) dané formule je formule ekvivalentní s danou formulí, mající tvar disjunkce elementárních konjunkcí.*

Úplná elementární disjunkce (ÚED) dané množiny výrokových symbolů je elementární disjunkce, ve které se každý symbol z dané množiny vyskytuje právě jednou (buďto prostě nebo negovaný).

Úplná disjunktivní normální forma (UDNF) dané formule je formule ekvivalentní s danou formulí a mající tvar disjunkce úplných elementárních konjunkcí.

Výroková formule je v disjunktivní normální formě, je-li disjunkcí několika podformulí (disjunktů) o nichž platí, že každý disjunkt je konjunkcí konečně mnoha literálů (prvotních proměnných nebo jejich negací) a v žádném disjunktě se sobě odpovídající pozitivní a negativní literály nevyskytují současně.

Úplná disjunktivní normální forma je taková, v níž každý disjunkt obsahuje literály všech proměnných formule.

Každá výroková formule, která není kontradikcí, je ekvivalentní jisté formuli v disjunktivní normální formě. Dvě úplné disjunktivní normální formy téže formule se mohou lišit nejvýš pořadím složek v téže disjunkci nebo konjunkci. Disjunktivní normální forma kontradikce je prázdná disjunkce.

2.7.2. Konjunktivní normální formy formulí

Definice 15. *Konjunktivní normální forma (KNF) dané formule je formule ekvivalentní s danou formulí a mající tvar konjunkce elementárních disjunkcí.*

Úplná elementární konjunkce (ÚEK) dané množiny výrokových symbolů je elementární konjunkce, ve které se každý symbol z dané množiny vyskytuje právě jednou (buďto prostě nebo negovaný).

Úplná konjunktivní normální forma (UKNF) dané formule je formule ekvivalentní s danou formulí, mající tvar konjunkce úplných elementárních disjunkcí.

Výroková formule je konjunktivní normální formě, je-li konjunkcí několika podformulí (konjunktů) o nichž platí, že každý konjunkt je disjunkcí konečně mnoha literálů a v žádném konjunktě se sobě odpovídající pozitivní a negativní literály nevyskytují současně.

Úplná konjunktivní normální forma je taková v níž každý konjunkt obsahuje literály všech proměnných formule.

Každá výroková formule, která není tautologie, je ekvivalentní jisté formuli v konjunktivní normální formě. Dvě úplné konjunktivní normální formy téže formule se mohou lišit nejvýš pořadím složek v téže disjunkci nebo konjunkci. Konjunktivní normální forma tautologie je prázdná konjunkce.

2.8. Úplný systém logických spojek

Když v některé formuli nahradíme jen podformuli formulí s ní sémanticky ekvivalentní, tak získaná formule bude sémanticky ekvivalentní s původní formulí.

Říkáme, že množina logických spojek tvoří úplný systém spojek právě tehdy, když pro každou formuli φ existuje formule ψ taková, že $\varphi \models \psi$ a formule ψ obsahuje jen logické spojky s této množiny.

Čím větší je systém základních spojek, které užíváme v jazyku výrokové logiky, tím jednodušší je vyjadřování ostatních spojek. Naopak, pokud je systém spojek malý, vyjadřování ostatních spojek může být komplikované a formule sloužící k vyjádření zbývajících spojek jsou obvykle dlouhé [1].

Definice 16. *Speciální význam mají Pierceova spojka¹ a Shefferova spojka², které samy o sobě tvoří úplný systém spojek. To znamená, že všechny klasické výrokové spojky lze vyjádřit pomocí jediné binární spojky $(\varphi \Downarrow \psi)$ definované jako $(\neg(\varphi \vee \psi))$ a nazývané Pierceova spojka (Pierce arrow) neboli NOR (not or) nebo spojky $(\varphi \Uparrow \psi)$ definované jako $(\neg(\varphi \wedge \psi))$ a nazývané Shefferova spojka (Sheffer stroke) neboli NAND (not and).*

		$\varphi \Downarrow \psi$			$\varphi \Uparrow \psi$
0	0	1	0	0	1
1	0	0	1	0	1
0	1	0	0	1	1
1	1	0	1	1	0

Obrázek 2. Pravdivostní hodnoty spojek \Downarrow, \Uparrow .

2.8.1. Úplný systém spojek $\{\neg, \wedge\}$

Převod: [8]

- $\varphi \vee \psi \Leftrightarrow \neg(\neg\varphi \wedge \neg\psi)$
- $\varphi \Rightarrow \psi \Leftrightarrow \neg(\varphi \wedge \neg\psi)$
- $\varphi \Leftrightarrow \psi \Leftrightarrow \neg(\varphi \wedge \neg\psi) \wedge \neg(\psi \wedge \neg\varphi)$

2.8.2. Úplný systém spojek $\{\neg, \vee\}$

Převod: [8]

- $\varphi \wedge \psi \Leftrightarrow \neg(\neg\varphi \vee \neg\psi)$
- $\varphi \Rightarrow \psi \Leftrightarrow \neg\varphi \vee \psi$
- $\varphi \Leftrightarrow \psi \Leftrightarrow \neg(\neg((\neg\varphi \vee \psi) \vee \neg(\neg\psi \vee \varphi)))$

¹Charles Sanders Peirce – americký filozof a logik

²Henry Maurice Sheffer – americký logik

2.8.3. Úplný systém spojek $\{\neg, \Rightarrow\}$

Převod: [8]

- $\varphi \wedge \psi \Leftrightarrow \neg(\varphi \Rightarrow \neg\psi)$
- $\varphi \vee \psi \Leftrightarrow \neg\varphi \Rightarrow \psi$
- $\varphi \Leftrightarrow \psi \Leftrightarrow \neg((\varphi \Rightarrow \psi) \Rightarrow \neg(\psi \Rightarrow \varphi))$

2.8.4. Úplný systém spojek $\{\Downarrow\}$

Převod: [8]

- $\neg\varphi \Leftrightarrow \varphi \Downarrow \varphi$
- $\varphi \wedge \psi \Leftrightarrow (\varphi \Downarrow \varphi) \Downarrow (\psi \Downarrow \psi)$
- $\varphi \vee \psi \Leftrightarrow (\varphi \Downarrow \psi) \Downarrow (\varphi \Downarrow \psi)$

2.8.5. Úplný systém spojek $\{\Uparrow\}$

Převod: [8]

- $\neg\varphi \Leftrightarrow \varphi \Uparrow \varphi$
- $\varphi \wedge \psi \Leftrightarrow (\varphi \Uparrow \psi) \Uparrow (\varphi \Uparrow \psi)$
- $\varphi \vee \psi \Leftrightarrow (\varphi \Uparrow \varphi) \Uparrow (\psi \Uparrow \psi)$

3. Uživatelská dokumentace

Tato část textu popisuje systémové požadavky, instalaci, spuštění aplikace s následným zadáním formule výrokové logiky, zvolení činnosti s touto zadanou formulí, vlastní vytvoření tabulek nebo převodů, ukončení a možné odinstalování aplikace.

Celou aplikaci jsem navrhovala tak, aby uživatel nebyl nijak rozptylován, aby práce s aplikací byla pohodlná, z uživatelského hlediska snadno použitelná, s ohledem na jednoduché a intuitivní ovládání. Jednou z priorit byla ergonomie. Při tvorbě uživatelského rozhraní jsem dbala obecných zvyklostí a doporučení.

3.1. Systémové požadavky

Primárně byla aplikace vyvíjena i testována pro operační systém Windows 7 s minimálním zobrazovacím rozlišením 1280 x 800 pixelu. Pro běh je nutná instalace rozhraní .NET Framework 4.

3.2. Instalace a spuštění aplikace

Aplikace „Výuka logiky“ se nainstaluje spuštěním souboru „setup.exe“ umístěném v adresáři bin, který se nachází na přiloženém CD-ROM disku. Po zobrazení okna instalátoru je nutné pokračovat kliknutím na tlačítko „Next“. V dalším okně vyberte adresář pro umístění aplikace a opět klikněte na tlačítko „Next“. Poté potvrďte poslední souhrnné okno, čímž se zahájí instalace. Instalátor vytvoří zástupce pro spuštění aplikace se jménem „Vyuka_logiky“ na pracovní ploše i v nabídce „Start“ pod volbou „Všechny programy“ → „Výuka logiky“.

Jak již bylo zmíněno, je pro chod aplikace nutná instalace .NET Framework. V případě absence dojde k automatickému stažení během instalace. Proto je vhodné provádět instalaci na počítači s trvalým připojením k internetu.

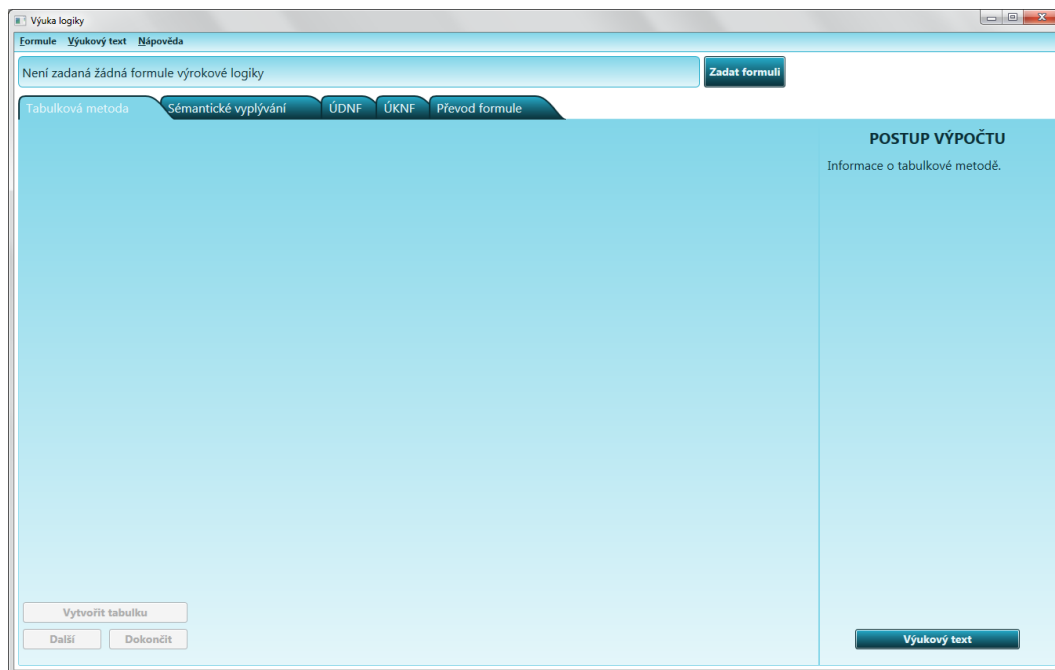
Aplikaci můžeme spustit z cílového adresáře, z plochy nebo z nabídky Start otevřením souboru „Vyuka_logiky.exe“. Po spuštění se otevře hlavní pracovní okno a je možné ihned začít pracovat.

3.3. Popis aplikace

Hlavní okno aplikace se skládá ze tří částí: Hlavní menu, Formule výrokové logiky a Záložky. Všechny jsou znázorněny na následujícím obrázku.

3.3.1. Hlavní menu

Nachází se v nejhornější části programu v celé jeho šíři, hned pod řádkem s názvem programu a tlačítka pro minimalizaci, maximalizaci a zavření aplikace. Obsahuje tři podmenu:



Obrázek 3. Hlavní okno aplikace.

- Soubor

Zadat formuli – slouží k zadání formule výrokové logiky.

Konec - ukončení aplikace.

- Výukový text

Slouží k zobrazení výukového textu k jednotlivým tématům.

Tabulková metoda

Sémantické vyplývání

Úplná disjunktivní normální forma

Úplná konjunktivní normální forma

Převod formule na sémanticky ekvivalentní formule s danými výrokovými spojkami.

- Nápověda

Ovládání aplikace – otevře okno s popisem ovládání aplikace.

O programu - otevře okno s informacemi o aplikaci.

3.3.2. Formule výrokové logiky

Obsahuje textové pole pro zobrazení formule výrokové logiky doplněné o tlačítko „Zadat formuli“, které slouží pro otevření okna s možností zadávat formuli výrokové logiky.

3.3.3. Záložky

Tvoří největší část okna, jsou hlavní a nejdůležitější částí aplikace. Slouží k zobrazení tabulek a převodů. Z důvodu jednoduchosti ve většině případů obsahují totožné prvky (tlačítka, vyhodnocovací textová pole, . . .)

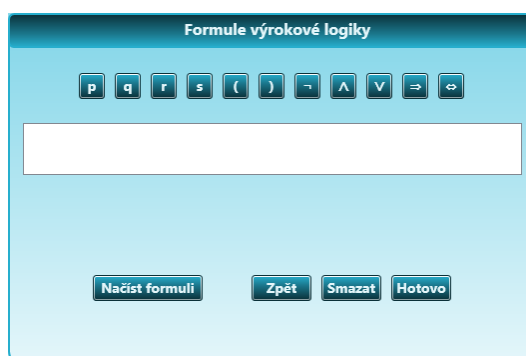
Tato část je rozdělena na pět záložek podle činností, které chceme se zadanou formulí výrokové logiky provádět a to na Tabulková metoda, Sémantické vyplývání, ÚDNF, ÚKNF a Převod formule. Každá záložka má dvě oddělené části. První slouží pro umístění výsledných tabulek a převodů, druhá má funkci informační, zobrazuje informace o kroku, který se právě provádí.

3.4. Ovládání aplikace

3.4.1. Zadání formule výrokové logiky

Před spuštěním samotné demonstrace vytváření tabulek či převodu je třeba nejprve zadat formuli výrokové logiky. V případě nezadání formule, jsou tlačítka pro vytváření tabulek či převody neaktivní.

Formuli výrokové logiky lze zadat pomocí menu „Soubor“ → „Zadat Formuli“ nebo tlačítkem Zadat formuli, které je umístěno v horní části aplikace. Poté dojde k otevření druhého okna „Formule výrokové logiky“ s možností zadávání formule výrokové logiky.



Obrázek 4. Okno zadávání formule výrokové logiky.

Zadávání se provádí dvěma způsoby. Ručně, znak po znaku, pomocí tlačítek pro výrokové symboly, symboly logických spojek a pomocné symboly nebo pomocí načtení formule z textového souboru s koncovkou „*.fvl“.

V případě špatného zadání formule výrokové logiky je možné ji upravit pomocí tlačítka „Zpět“ nebo „Smazat“.

Dokončení zadávání formule se provede pomocí tlačítka „Hotovo“. Formule se zobrazí do textového pole v hlavním okně a okno „Formule výrokové logiky se zavře“.

3.4.2. Vytvoření tabulek, převod

Je-li zadána formule výrokové logiky, můžeme zvolit záložku podle úkonu, který chceme provádět a vytvořit tabulku nebo provést převod. Rozmístění základních ovládacích prvků je většinou totožné, liší se pouze minimálně.

- Záložka Tabulková metoda



Obrázek 5. Hlavní okno aplikace - záložka Tabulková metoda.

Jak již bylo zmíněno, každá záložka obsahuje dvě oddělená pole. Větší je hlavní pracovní pole, které slouží pro zobrazení tabulky. V jeho spodní části jsou umístěna tři tlačítka „Vytvořit tabulku“, „Další“ a „Dokončit“, která jsou zprvu neaktivní. Aktivují se podle dalšího kroku, který se právě provádí.

Vytvořit tabulku – slouží pro vytvoření tabulky s předepsaným počtem řádků a sloupců. Tabulka vždy obsahuje sloupec pro každou proměnnou (tj. p, q, r, s), která se ve formuli vyskytuje, dále sloupce pro podformule dané

formule a sloupec pro výslednou hodnotu zkoumané formule. Pořadí sloupců v tabulce udává pořadí vyhodnocování z čehož je zřejmé, že poslední sloupec je pravdivostním ohodnocením celé zkoumané formule. Pokud formule obsahuje „n“ proměnných, které mohou nabývat pravdivostní ohodnocení 1 i 0, pak tabulka musí mít 2^n řádků. Po zvolení tlačítka dojde k vytvoření tabulky s předepsaným záhlavím a s odpovídajícím počtem sloupců a řádků.

Další – provede postupně další krok vyhodnocení. Každé kliknutí způsobí vyplnění jednoho následujícího řádku. Pod záhlaví se postupně zapíše všechny možnosti dosazení hodnot 1, 0 za výrokové proměnné, na jejichž základě dojde k dopočítání zbytku jedniček a nul. Poslední krok vyhodnotí tabulku a rozhodne, je-li formule splnitelná, tautologie či kontradikce. Toto sdělení se zobrazí ve spodní části záložky.

Dokončit – provede doplnění i vyhodnocení tabulky v jednom kroku.

Druhé menší pole „Postup výpočtu“ zobrazuje informace o kroku, který se právě provádí. Je doplněno o tlačítko „Výukový text“, kterým si může uživatel otevřít podrobnější text s informacemi a definicemi k zvolenému tématu.

- Záložka Sémantické vyplývání



Obrázek 6. Hlavní okno aplikace - záložka Sémantické vyplývání.

Hlavní pracovní pole této záložky obsahuje v porovnání s předchozím navíc pouze jedno textové pole a tlačítko „Zadat další formuli“. Jelikož můžeme mezi každými dvěma formulemi rozhodnout o vztahu vyplývání nebo ekvivalence pomocí pravdivostních tabulek, proto i v této záložce dochází k zobrazení tabulky. Tentokrát ale neobsahuje sloupce pro všechny podformule, ale pouze pro proměnné. Dva poslední sloupce zobrazují výsledné ohodnocení obou posuzovaných formulí. Tlačítko „Vytvořit tabulku“ je tentokrát aktivní až tehdy, když dojde ke správnému zadání obou formulí. Pro posouzení vztahu vyplývání je důležité, aby formule obsahovaly shodné proměnné. Pro demonstraci jsem zvolila sémantické vyplývání pouze z jednoho předpokladu.

- Záložky Úplná disjunktivní a konjunktivní normální forma

Jedná se o dvě naprosto totožné záložky, lišící se pouze závěrečným vyhodnocením a zápisem formy. Opět obsahují tlačítka „Vytvořit tabulku“, „Další“ a „Dokončit“. Jejich funkčnost je stejná jako u předchozích záložek. I pole „Postup výpočtu“ a tlačítko „Výukový text“ slouží ke stejnému účelu.

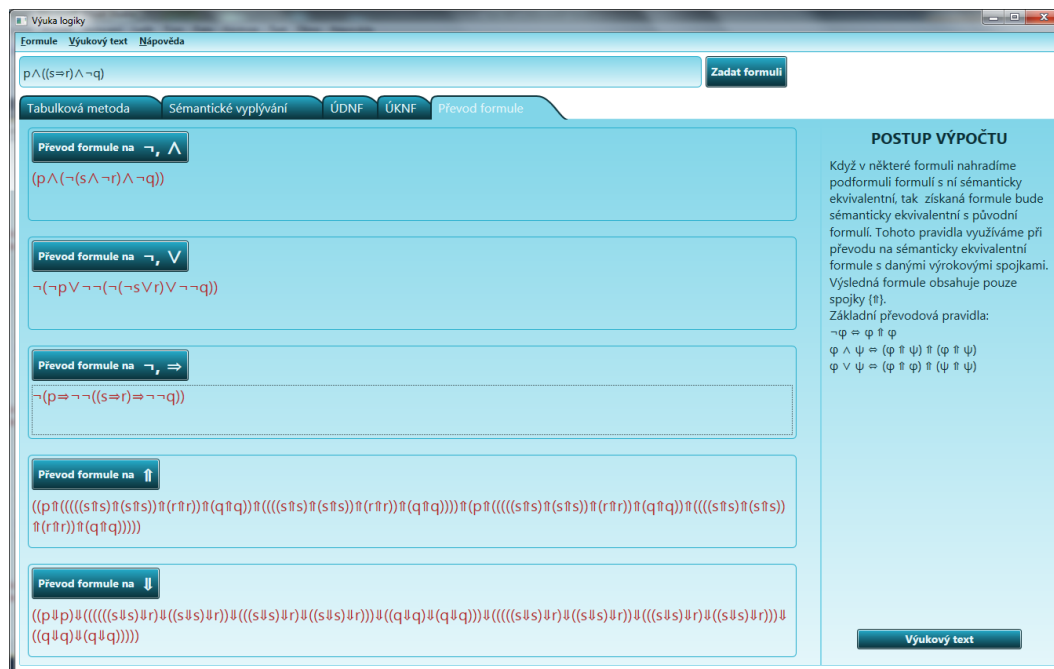


Obrázek 7. Hlavní okno aplikace - záložka ÚDNF.

- Záložka Převod formule

Jedná se o jedinou vzhledově se lišící záložku. Neslouží totiž k zobrazení

tabulky, ale znázorňuje zápisy převodů formulí na sémanticky ekvivalentní formule s danými výrokovými spojkami.



Obrázek 8. Hlavní okno aplikace - záložka Převod formule.

Hlavní pracovní pole je rozděleno na pět podpolí. Každé podpole odpovídá převodu na některý zvolený úplný systém logických spojek. Obsahuje vždy jedno tlačítko, které provede zobrazení zadané formule na sémanticky ekvivalentní formuli, obsahující pouze dané spojky. Pole „Postup výpočtu“ a tlačítko „Výukový text“ již není asi nutné popisovat, jejich funkce je v celé aplikaci naprosto totožná.

3.5. Ukončení aplikace

Aplikaci je možné ukončit pomocí menu „Soubor“ → „Konec“ nebo pomocí křížku v pravém horním rohu okna aplikace.

3.6. Odinstalování aplikace

Odinstalování aplikace se provádí standardním způsobem a to v nabídce „Start“ → „Ovládací panely“ → „Programy“ → „Programy a funkce“ → vybráním položky „Výuka logiky“ a kliknutím na tlačítko „Odinstalovat“.

4. Programátorská dokumentace

4.1. Použité technologie

Aplikace je vytvořena ve vývojovém prostředí Microsoft Visual Studio 2010 na platformě .NET Framework 4., s využitím grafického subsystému Microsoft Windows Presentation Foundation (WPF), který se využívá při praní programů fungujících v systému Microsoft Windows. Programy WPF lze psát kompletně pomocí jazyka C# nebo jiného programovacího jazyka. Kromě toho WPF zahrnuje i značkovací jazyk založený na XML, který se nazývá XAML (Extensible Application Markup Language) [7]. V aplikaci je zdrojový kód programu napsán v programovacím jazyce C# a v jazyce XAML. Jazykem C# je pak napsán kód k obsluze uživatelských vstupů a dalších algoritmů v aplikaci. Jazykem XAML je definován vzhled a grafické prvky aplikace.

4.2. Popis tříd

Třída `MainWindow.xaml` a `MainWindow.xaml.cs` – zavede vzhled a interakční logiku aplikace. Obsahuje především definované služby událostí stisku tlačítek. Obsluha události kliku na tlačítka zavolá příslušnou metodu.

- `void ZadatFormuliClick(object sender, RoutedEventArgs e)` - obsluha událostí zobrazující okno pro zadávání formule výrokové logiky.
- `void VytvoritTabulkuClick_TM(object sender, RoutedEventArgs e)` - obsluha událostí zajišťující vyvolání metody pro vytvoření tabulky.
- `void DalsiClick_TM(object sender, RoutedEventArgs e)` - obsluha událostí zajišťující vyvolání metody zobrazující další krok demonstrace vytvoření tabulky.
- `void DokoncitClick_TM(object sender, RoutedEventArgs e)` - obsluha událostí zajišťující vyvolání metody, která zobrazí kompletní demonstraci vytvoření tabulky.
- `void VyukovyTextClick_TM(object sender, RoutedEventArgs e)` - obsluha událostí zajišťující vyvolání metody zobrazující dokument s výukovým textem.
- `void Prevod_1Click(object sender, RoutedEventArgs e)` - obsluha událostí zajišťující vyvolání metody pro provedení převodu.
- `MainWindow()` - inicializuje hlavní okno.

Třída `ZadatFormuli.xaml` a `ZadatFormuli.xaml.cs` - zavede vzhled a interakční logiku okna pro zadávání formule.

- `string DoplnitVnejsiZavoroky(string vstupniRetezec)` – doplní chybějící vnější závorky.
- `string DoplnitZavorokyNegaci(string vstupniRetezec)` – doplní chybějící závorky u negace.
- `bool KontrolaFormule(string vstupniRetezec)` – zkontroluje správné pořadí symbolů ve formuli.
- `bool KontrolaPlatnychZnaku(string vstupniRetezec)` – zkontroluje zapsané znaky, vrátí true, pokud jsou zapsané pouze platné znaky.
- `bool KontrolaZavorek(string vstupniRetezec)` – zkontroluje správné spárování závorek.
- `bool KontrolaZavorekFunkce(string vstupniRetezec)` – zkontroluje správné uzávorkování všech funkcí.
- `void NacistFormuli()` – načte formuli z textového souboru.
- `void PovolitTlacitka()` – aktivuje všechna tlačítka.
- `ZadatFormuli()` - inicializuje okno.
- `void ZakazatTlacitka()` – deaktivuje všechna tlačítka.
- `void ZapsatFormuli()` - zapíše zadané symboly a znaky formule do seznamů.
- `void SymbolKonjunkceClick(object sender, RoutedEventArgs e)` - obsluha událostí zajišťující vyvolání metody vložení symbolu do formule.
- `void TlacitkoHotovoClick(object sender, RoutedEventArgs e)` - obsluha událostí zajišťující vyvolání metody pro zapsání a kontrolu formule. Zavře okno.
- `void TlacitkoNacistClick(object sender, RoutedEventArgs e)` - obsluha událostí zajišťující vyvolání metody pro načtení formule.
- `void TlacitkoSmazatClick(object sender, RoutedEventArgs e)` - obsluha událostí zajišťující vyvolání metody pro smazání celé formule.
- `void TlacitkoZpetClick(object sender, RoutedEventArgs e)` - obsluha událostí zajišťující vyvolání metody pro smazání posledního zadaného symbolu.

Třída Formule - zajišťuje rozklad formule na podformule, jejich následné vyhodnocování a převody formulí.

- `List<List<string>> ZapsatHodnotyFormule(string form, List<char> sym)` – vyvolá metody pro zápis hodnot záhlaví a řádků, vrátí seznam hodnot.
- `List<string> ZapsatHodnotyZahlavi()` - zapíše hodnoty záhlaví tabulky, vrátí seznam hodnot.
- `List<string> ZapsatHodnotyRadku()` - zapíše hodnoty jednotlivých řádku, vrátí seznam hodnot.
- `List<string> RozdelitRetezec(string text)` - rozdělí formule na podformule.
- `bool JeAtomicka(string text)` – zkontroluje, zda je formule atomická.
- `char VytvoritPodformule(string text)` – vytvoří a zapíše podformule.
- `char VyhodnotitFormuli(string text)` – vyhodnotí formuli, vstupem je ohodnocená správně zadaná formule.
- `char VyhodnotitFunkci(char funkce, char a, char b)` – provede výběr zadaného vyhodnocovacího algoritmu.
- `char Negace(char b), char Konjunkce(char a, char b), char Disjunkce(char a, char b), char Implikace(char a, char b), char Ekvivalence(char a, char b)` – metody vyhodnocující funkce výrokové logiky.
- `string PrevodFormule_1(string text)` – vyvolá metody pro převod formulí na dané spojky.
- `string Prevod_1(char funkce, string a, string b)` – provede výběr zadaného převodového algoritmu.
- `string Negace_1(string b), string Konjunkce_1(string a, string b), string Disjunkce_1(string a, string b), string Implikace_1(string a, string b), string Ekvivalence_1(string a, string b)` - metody provádějící převody formulí.

Třídy `TabulkovalaMetoda`, `UDNF`, `UKNF`, `SemantickeVyplyvani` – obsahují metody pro vytváření tabulek pravdivostních hodnot. Jsou zodpovědné za správné vytvoření tabulky a naplnění hodnotami.

- `void VytvoritTabulku(Grid tabulkaGrid_TM)` - vytvoří prázdnou tabulku, nenaplněnou hodnotami.
- `void VytvoritTabulkaGrid(int rows, int cols)` - vytvoří Grid s požadovaným počtem sloupců a řádků.

- `void Dalsi_TM()` - vyplní do daného počtu řádků hodnoty, počet řádků určuje `krokVypoctu_TM`.
- `void Dokoncit_TM()` – doplní kompletně hodnoty do tabulky a vyvolá zápis vyhodnocení.
- `string KrokVypoctuInfo_TM()` - zapíše informace o kroku, který se právě provádí.
- `string VyhodnoceniInfo_TM()` – zapíše vyhodnocení.

Závěr

Cílem této bakalářské práce bylo vytvořit aplikaci na podporu výuky části předmětu „Matematická logika“. Aplikace provádí demonstraci vyvození a vyhodnocení některých pojmů výrokové logiky na předem zadané a překontrolované formuli výrokové logiky. Tato aplikace je určena především jako pomůcka pro studenty. Měla by uživateli pomoci pochopit základní pojmy výrokové logiky. Je doplněna o podrobný výukový text, který obsahuje definice všech pojmů, se kterými se uživatel v aplikaci setká. Toto doplnění umožňuje užívání aplikace i uživatelům, kteří se s výrokovou logikou setkávají úplně poprvé. Veškeré definice si tedy mohou nastudovat v průběhu užívání aplikace.

Při návrhu uživatelského rozhraní byl kladen důraz na názornost, jednoduchost, přehlednost, snadnou ovladatelnost a uživatelsky přívětivé prostředí.

Jako možné rozšíření aplikace se jeví implementace dalších pojmů výrokové logiky.

Conclusions

The aim of this work was create an application to support teaching part of the subject „Mathematical Logic“. Application does demonstration draw and evaluation of some concepts of propositional logic to the predetermined and checked out formula of propositional logic. This application is intended primarily as a tool for students. It should help the user understand the basic concepts propositional logic. It is supplemented by detailed tutorial text, which contains definitions of all terms with which will the user meet in this application. This addition allows use applications the users, who meet with propositional logic fot the very first time. All the definitions they can study during the using this application.

At the suggestion of the user interface was placed emphasis to the illustration, simplicity, clarity and user-friendly environment.

As a possible extension of this application appears the implementations of other concepts of the propositional logic.

Reference

- [1] Bělohávek Radim, Vychodil Vilém: *Diskrétní matematika pro informatiky I*. Olomouc, 2006.
- [2] Duží Marie: *Logika pro informatiky (a příbuzné obory) učební text*. VŠB-TU Ostrava, Ostrava, 2012.
- [3] Štěpánek Petr: *Predikátová logika*. Praha, 2000.
- [4] Draženská Emílie, Myšková Helena: *Matematická logika*. Equilibria, s.r.o., Košice, 2012.
- [5] Sochor Antonín: *Klasická matematická logika*. Karolinum, Praha, 2001.
- [6] Štěpán Jan: *Logika a logické systémy*. Votobia, 1992.
- [7] Petzold Charles: *Mistrovství ve Windows Presentation Foundation*. Computer Press, a.s., Brno, 2008.
- [8] Wikipedie, otevřená encyklopedie, <http://www.wikipedia.org/>

5. Obsah příloženého CD/DVD

Součástí této práce je i CD. Jeho obsahem jsou tři adresáře a jeden soubor.

bin/

Instalační balíček `SETUP.EXE` vytvořené aplikace.

doc/

Dokumentace práce ve formátu PDF, vytvořená dle závazného stylu KI PŘF pro diplomové práce, včetně všech příloh.

src/

Kompletní zdrojové texty programu `VÝUKA LOGIKY`.

readme.txt

Instrukce pro instalaci a spuštění programu `VÝUKA LOGIKY`.