Matematická výroková logika

13. března 2024

Tento text slouží jako kompilace příprav k přednáškám z výrokové logiky předmětu Logika a grafy. Omluvte prosím mírnou nevyváženost textu: průběžným doplňováním podrobností mohou některé sekce působit celkem plně, zatímco některé sekce jsou zatím stále pouhé *lecture notes* k přednáškám.

- V sekci 1 zavedeme syntax výrokové logiky a předvedeme, jak induktivní definice jazyka výrokové logiky umožňuje definovat funkce na formulích výrokové logiky rekursivně.
- Sekce 2 je kompilací příkladů sloužících k osvětlení důkazového systému *přirozené dedukce* ve výrokové logice. Zavádíme zde *odvozovací pravidla* a ukazujeme, jak jsou využívána ve *formálních důkazech*.
- Klasickou sémantiku výrokové logiky zavádíme v sekci 3: definujeme pojem *významu* formule jako jisté *booleovské funkce*, zavádíme pojem *splnitelnosti* formule, *sémantickou ekvivalenci*, a studujeme pojem *sémantického důsledku*.

  Zodpovíme otázku, zda je každá booleovská funkce representovatelná nějakou formulí výrokové logiky.
- Krátká sekce 4 zmiňuje bez důkazu důležitý výsledek propojující pojmy logického a sémantického důsledku: tzv. *větu o korektnosti a úplnosti*.

1 Syntax výrokové logiky

Definice 1.1 (Jazyk výrokové logiky)**.** Mějme množinu *atomických formulí* At. Množina všech formulí výrokové logiky nad množinou At je definována induktivně:

- Každá atomická formule $a \in \mathsf{At}$ je formule.
- $\top$ a $\bot$ jsou formule.

- Je-li φ formule, pak je i $(\neg\varphi)$ formule.
- Jsou-li φ_1 a φ_2 formule, pak je i $(\varphi_1 \wedge \varphi_2)$ formule.
- Jsou-li φ_1 a φ_2 formule, pak je i $(\varphi_1 \vee \varphi_2)$ formule.
- Jsou-li φ_1 a φ_2 formule, pak je i $(\varphi_1 \Rightarrow \varphi_2)$ formule.
- Jsou-li φ_1 a φ_2 formule, pak je i $(\varphi_1 \Leftrightarrow \varphi_2)$ formule.

Řetězec znaků je formule pouze tehdy, když o něm tuto skutečnost můžeme odvodit pomocí výše zmíněných pravidel v konečně mnoha krocích.

Množinu všech formulí výrokové logiky nad množinou At značíme $\mathsf{Fm}(\mathsf{At})$.

Poznámka 1.2. K jednoduššímu popisu formálních jazyků se často používá Backusova-Naurova forma (BNF). Pro náš jazyk výrokové logiky by příslušná forma vypadala následovně:

$$\mathsf{F} ::= a \mid \top \mid \bot \mid (\neg \mathsf{F}) \mid (\mathsf{F} \wedge \mathsf{F}) \mid (\mathsf{F} \vee \mathsf{F}) \mid (\mathsf{F} \Rightarrow \mathsf{F}) \mid (\mathsf{F} \Leftrightarrow \mathsf{F})$$

kde a označuje atomickou formuli a F formuli výrokové logiky. BNF slouží ke krátkému popisu toho, jakým způsobem lze formule konstruovat. Jednotlivé způsoby konstrukce jsou odděleny svislou čarou $|$. Formuli lze tedy zkonstruovat tak, že vybereme atomickou formuli, symbol $\top$ či $\bot$, nebo vezmeme již zhotovené formule, propojíme je některou z dostupných logických spojek a obalíme závorkami. Spojka $\neg$ (negace) je unární, zbylé spojky $\wedge$ (konjunkce), $\vee$ (disjunkce), $\Rightarrow$ (implikace) a $\Leftrightarrow$ (ekvivalence) jsou binární.

Příklad 1.3. Nechť $\mathsf{At} = \{a, b, c, d\}$. Příkladem formulí z množiny $\mathsf{Fm}(\mathsf{At})$ jsou například formule

$$(a \wedge (b \vee c)), \qquad (\neg b), \qquad c, \qquad ((\neg a) \Rightarrow (\neg b)).$$

Poznámka 1.4. Formule z $\mathsf{Fm}(\mathsf{At})$ splňují takzvanou *unique readability property* (URP), česky *jednoznačnou čitelnost*. Neformálně řečeno, u každé formule $\varphi \in \mathsf{Fm}(\mathsf{At})$ lze jednoznačně určit

1. pravidlo BNF, kterým byla φ vytvořena,
2. stavební části („přímé podformule"), které byly při konstrukci φ podle pravidla BNF použity.

Například u formule $\varphi = (a \wedge (b \vee c))$ je vidět, že

1. její *hlavní spojkou* je konjunkce $\wedge$ (vznikla spojením dvou formulí φ_1 a φ_2 logickou spojkou $\wedge$),
2. formule φ_1 a φ_2 jsou jednoznačně dané; jsou to formule a a $(b \vee c)$.

Pozorování 1.5. Formule jsou jednoznačně čitelné díky svému uzávorkování. Doplňte závorky v nějaké formuli na ovály, a tyto ovály ve vaší formuli vytvoří systém „vrstevnic", které umožňují „detekovat" hlavní spojky a přímé podformule.

Značení 1.6. Dovolíme si při zápisu formulí dvěma způsoby *relaxovat* značení:

1. U formulí nebudeme psát vnější závorky. Například formuli

$$((\neg(p \Rightarrow (\neg q))) \vee (\neg(\neg r)))$$

   můžeme zapisovat jako

$$(\neg(p \Rightarrow (\neg q))) \vee (\neg(\neg r)).$$

2. U formulí, ve kterých se vyskytuje negace, nebudeme kolem negace používat závorky. Výše zmíněnou formuli

$$((\neg(p \Rightarrow (\neg q))) \vee (\neg(\neg r)))$$

   tedy můžeme zapisovat jako

$$\neg(p \Rightarrow \neg q) \vee \neg\neg r$$

Striktně vzato řetězec

$$\neg(p \Rightarrow \neg q) \vee \neg\neg r$$

podle naší definice formule formulí *není*. Jelikož je však tento zápis čitelnější a lze z něj jednoznačně určit, jakou formuli by popisoval, kdybychom značení nerelaxovali, nebude naše dohoda o zápisu dělat žádné problémy.

Pozor! V řetězci

$$\neg(p \Rightarrow \neg q)$$

jsou závorky důležité. Řetězec

$$\neg p \Rightarrow \neg q$$

je v relaxovaném značení *jinou formulí*, a to formulí

$$((\neg p) \Rightarrow (\neg q)).$$

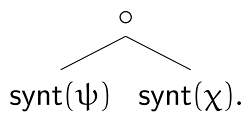
Definice 1.7 (Syntaktický strom formule)**.** Syntaktický strom formule φ je kořenový strom $\mathsf{synt}(\varphi)$ s označenými vrcholy zadaný rekursivním výpočtem následovně:

1. Pokud je φ atomickou formulí či pokud $\varphi = \top$ nebo $\varphi = \bot$, pak $\mathsf{synt}(\varphi)$ je jediný vrchol s označením φ.

2. Pokud $\varphi = \neg\psi$ pro nějakou formuli ψ, pak $\mathsf{synt}(\varphi)$ je strom

$$\begin{array}{c} \neg \\ | \\ \mathsf{synt}(\psi). \end{array}$$

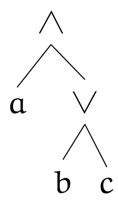
3. Pokud $\varphi = \psi \circ \chi$ (kde $\circ$ je jakákoli binární logická spojka), pak $\mathsf{synt}(\varphi)$ je strom



Příklad 1.8. Sestrojíme syntaktické stromy formulí

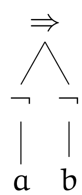
$$\alpha = (a \wedge (b \vee c)), \qquad \beta = \neg b, \qquad \gamma = \neg a \Rightarrow \neg b.$$

Syntaktický strom formule α vypadá následovně:



Syntaktický strom formule β vypadá následovně:

Syntaktický strom formule γ vypadá následovně:



Definice 1.9 (Hloubka formule)**.** Hloubka formule φ je přirozené číslo $\mathsf{hl}(\varphi)$ zadané rekursivním výpočtem následovně:

1. Pokud je φ atomickou formulí či pokud $\varphi = \top$ nebo $\varphi = \bot$, pak $\mathsf{hl}(\varphi) = 0$.
2. Pokud $\varphi = \neg\psi$ pro nějakou formuli ψ, pak $\mathsf{hl}(\varphi) = 1 + \mathsf{hl}(\psi)$.
3. Pokud $\varphi = \psi \circ \chi$ (kde $\circ$ je jakákoli binární logická spojka), pak $\mathsf{hl}(\varphi) = 1 + \max(\mathsf{hl}(\psi), \mathsf{hl}(\chi))$.

Příklad 1.10. U formule $a \wedge (b \vee c)$ spočtěme její hloubku:

$$\begin{aligned}\mathsf{hl}(a \wedge (b \vee c)) &= 1 + \max(\mathsf{hl}(a), \mathsf{hl}(b \vee c)) \\ &= 1 + \max(0, 1 + \max(\mathsf{hl}(b), \mathsf{hl}(c))) \\ &= 1 + \max(0, 1 + \max(0, 0)) \\ &= 1 + \max(0, 1) \\ &= 1 + 1 = 2.\end{aligned}$$

Definice 1.11 (Podformule)**.** Množina podformulí formule $\varphi \in \mathsf{Fm}(\mathsf{At})$ je podmnožina $\mathsf{pf}(\varphi) \subseteq \mathsf{Fm}(\mathsf{At})$ zadaná rekursivním výpočtem následovně:

1. Pokud je φ atomickou formulí či pokud $\varphi = \top$ nebo $\varphi = \bot$, pak $\mathsf{pf}(\varphi) = \{\varphi\}$.
2. Pokud $\varphi = \neg\psi$ pro nějakou formuli ψ, pak $\mathsf{pf}(\varphi) = \{\varphi\} \cup \mathsf{pf}(\psi)$.
3. Pokud $\varphi = \psi \circ \chi$ (kde $\circ$ je jakákoli binární logická spojka), pak $\mathsf{pf}(\varphi) = \{\varphi\} \cup \mathsf{pf}(\psi) \cup \mathsf{pf}(\chi)$.

Příklad 1.12. U formule $a \wedge (b \vee c)$ spočtěme množinu jejích podformulí:

$$\begin{aligned}\mathsf{pf}(a \wedge (b \vee c)) &= \{a \wedge (b \vee c)\} \cup \mathsf{pf}(a) \cup \mathsf{pf}(b \vee c) \\ &= \{a \wedge (b \vee c)\} \cup \{a\} \cup \{b \vee c\} \cup \mathsf{pf}(b) \cup \mathsf{pf}(c) \\ &= \{a \wedge (b \vee c),\ a,\ b \vee c\} \cup \{b\} \cup \{c\} \\ &= \{a \wedge (b \vee c),\ a,\ b \vee c,\ b,\ c\}.\end{aligned}$$

2 Přirozená dedukce ve výrokové logice

Přirozená dedukce je *důkazový systém*. Její hlavní částí jsou *odvozovací pravidla*, jejichž skládáním je možné tvořit (formální) *důkazy*.

Základní úlohou, kterou budeme řešit, je ověřování *logického důsledku*.

Mějme formule $\varphi_1, \varphi_2, \ldots, \varphi_n$ (tzv. *předpoklady*) a formuli ψ (*závěr*). Je ψ logickým důsledkem předpokladů $\varphi_1, \varphi_2, \ldots, \varphi_n$? Pokud ano, budeme tento vztah značit jako

$$\varphi_1, \varphi_2, \ldots, \varphi_n \vdash \psi,$$

kde $\vdash$ je znak pro vztah logického důsledku, a celému zápisu $\varphi_1, \varphi_2, \ldots, \varphi_n \vdash \psi$ říkáme *sekvent*.

Rozhodnout, že je ψ logickým důsledkem předpokladů $\varphi_1, \varphi_2, \ldots, \varphi_n$, můžeme konstrukcí *formálního důkazu* formule ψ z množiny předpokladů $\{\varphi_1, \varphi_2, \ldots, \varphi_n\}$. Tvorbu důkazů předvedeme na vhodně zvolených příkladech, které budou ilustrovat odvozovací pravidla pro práci s jednotlivými logickými spojkami.

Konjunkce a implikace

Spojka	Zavedení spojky	Eliminace spojky
$\wedge$	$\dfrac{\varphi \qquad \psi}{\varphi \wedge \psi}\ \mathrm{i}\wedge$	$\dfrac{\varphi \wedge \psi}{\varphi}\ \mathrm{e}\wedge_1 \qquad \dfrac{\varphi \wedge \psi}{\psi}\ \mathrm{e}\wedge_2$
$\Rightarrow$	$\dfrac{\boxed{\begin{array}{c}\varphi \\ \vdots \\ \psi\end{array}}}{\varphi \Rightarrow \psi}\ \mathrm{i}\Rightarrow$	$\dfrac{\varphi \qquad \varphi \Rightarrow \psi}{\psi}\ \mathrm{e}\Rightarrow$

Příklady ilustrující odvozovací pravidla pro konjunkci a implikaci

Příklad 2.1. Dokažme logický úsudek

$$a \wedge b,\ b \Rightarrow c \vdash c.$$

Důkaz.

1.	$a \wedge b$	P
2.	$b \Rightarrow c$	P
3.	b	$\mathrm{e}\wedge_2$, 1
4.	c	$\mathrm{e}\Rightarrow$, 3, 2

□

Příklad 2.2. Dokažme logický úsudek

$$a \Rightarrow b,\ b \Rightarrow c,\ a \vdash b \wedge c.$$

Důkaz.

1.	$a \Rightarrow b$	P
2.	$b \Rightarrow c$	P
3.	a	P
4.	b	$\mathrm{e}\Rightarrow$, 3, 1
5.	c	$\mathrm{e}\Rightarrow$, 4, 2
6.	$b \wedge c$	$\mathrm{i}\wedge$, 4, 5

□

Příklad 2.3. Dokažme logický úsudek

$$a \Rightarrow b,\ b \Rightarrow c, \vdash a \Rightarrow c.$$

Důkaz.

1.	$a \Rightarrow b$	P
2.	$b \Rightarrow c$	P
3.	a	P
4.	b	$e\Rightarrow$, 3, 1
5.	c	$e\Rightarrow$, 4, 2
6.	$a \Rightarrow c$	$i\Rightarrow$, 3-5

□

Příklad 2.4. Dokažme logický úsudek

$$a \Rightarrow b,\ c \Rightarrow d, \vdash (a \wedge c) \Rightarrow (b \wedge d).$$

Důkaz.

1.	$a \Rightarrow b$	P
2.	$b \Rightarrow c$	P
3.	$a \wedge c$	P
4.	a	$e\wedge_1$, 3
5.	c	$e\wedge_2$, 3
6.	b	$e\Rightarrow$, 4, 1
7.	d	$e\Rightarrow$, 5, 2
8.	$b \wedge d$	$i\wedge$, 6, 7
9.	$(a \wedge c) \Rightarrow (b \wedge d)$	$i\Rightarrow$, 3-8

□

Příklad 2.5. Dokažme logický úsudek

$$a \vdash a.$$

Důkaz.

1.	a	P

□

Příklad 2.6. Dokažme logický úsudek

$$\vdash a \Rightarrow a.$$

Důkaz.

1.	a	P
2.	$a \Rightarrow a$	i$\Rightarrow$, 1-1

□

Rámečky je občas nutné vnořovat.

Příklad 2.7. Dokažme logický úsudek

$$(a \wedge b) \Rightarrow c \vdash a \Rightarrow (b \Rightarrow c).$$

Důkaz.

1.	$(a \wedge b) \Rightarrow c$	P
2.	a	P
3.	b	P
4.	$a \wedge b$	i$\wedge$, 2, 3
5.	c	e$\Rightarrow$, 4, 1
6.	$b \Rightarrow c$	i$\Rightarrow$, 3-5
7.	$a \Rightarrow (b \Rightarrow c)$	i$\Rightarrow$, 2-6

□

Pomocné pravidlo iterace umožňuje na řádku zopakovat tvrzení, které již bylo odvozeno dříve a které je z daného řádku viditelné (tj. neleží mimo *scope*, tedy *obor platnosti* daného řádku).

Příklad 2.8. Dokažme logický úsudek

$$\vdash a \Rightarrow (b \Rightarrow a).$$

Důkaz.

1.	a	P
2.	b	P
3.	a	it, 1
4.	$b \Rightarrow a$	i$\Rightarrow$, 2-3
5.	$a \Rightarrow (b \Rightarrow a)$	i$\Rightarrow$, 1-4

□

Disjunkce

Spojka	Zavedení spojky	Eliminace spojky
$\vee$	$\dfrac{\varphi}{\varphi \vee \psi}\,\mathrm{i}\vee_1 \qquad \dfrac{\psi}{\varphi \vee \psi}\,\mathrm{i}\vee_2$	$\dfrac{\varphi \vee \psi \quad \boxed{\begin{array}{c}\varphi\\ \vdots\\ \chi\end{array}} \quad \boxed{\begin{array}{c}\psi\\ \vdots\\ \chi\end{array}}}{\chi}\,\mathrm{e}\vee$

Příklady ilustrující pravidla pro disjunkci

Příklad 2.9. Dokažme logický úsudek

$$a \vee b,\ a \Rightarrow (c \wedge d),\ b \Rightarrow d \vdash d$$

Důkaz.

1.	$a \vee b$	P
2.	$a \Rightarrow (c \wedge d)$	P
3.	$b \Rightarrow d$	P
4.	a	P
5.	$c \wedge d$	e$\Rightarrow$, 4, 2
6.	d	e$\wedge_2$, 5
7.	b	P
8.	d	e$\Rightarrow$, 7, 3
9.	d	e$\vee$, 1, 4-6, 7-8

□

Příklad 2.10. Dokažme logický úsudek

$$a \vee b \vdash b \vee a.$$

Důkaz.

1.	$a \vee b$	P
2.	a	P
3.	$b \vee a$	i$\vee_2$, 2
4.	b	P
5.	$b \vee a$	i$\vee_1$, 4
6.	$b \vee a$	e$\vee$, 1, 2-3, 4-5

□

Negace a absurdita (spor, „false")

Spojka	Zavedení spojky	Eliminace spojky
$\neg$	$\dfrac{\boxed{\begin{matrix}\varphi\\ \vdots\\ \bot\end{matrix}}}{\neg\varphi}\ \mathrm{i}\neg$	$\dfrac{\varphi \qquad \neg\varphi}{\bot}\ \mathrm{e}\neg$
$\bot$	není	$\dfrac{\bot}{\varphi}\ \mathrm{e}\bot$

Příklady ilustrující pravidla pro negaci a absurditu

Příklad 2.11. Dokažme logický úsudek

$$a \Rightarrow b,\ a \Rightarrow \neg b \vdash \neg a.$$

Důkaz.

1.	$a \Rightarrow b$	P
2.	$a \Rightarrow \neg b$	P
3.	a	P
4.	b	e$\Rightarrow$, 3, 1
5.	$\neg b$	e$\Rightarrow$, 3, 2
6.	$\bot$	e$\neg$, 4, 5
7.	$\neg a$	i$\neg$, 3-6

□

Příklad 2.12. Dokažme logický úsudek

$$a \Rightarrow b \vdash \neg b \Rightarrow \neg a.$$

Důkaz.

1.	$a \Rightarrow b$	P
2.	$\neg b$	P
3.	a	P
4.	b	e$\Rightarrow$, 3, 1
5.	$\bot$	e$\neg$, 4, 2
6.	$\neg a$	i$\neg$, 3-5
7.	$\neg b \Rightarrow \neg a$	i$\Rightarrow$, 2-6

$\square$

Příklad 2.13. Dokažme logický úsudek

$$a \lor b,\ \neg b \vdash a.$$

Důkaz.

1.	$a \lor b$	P
2.	$\neg b$	P
3.	a	P
4.	b	P
5.	$\bot$	e¬, 4, 2
6.	a	e⊥, 5
7.	a	e∨, 1, 3-3, 4-6

$\square$

Ekvivalence a verum („true“)

Spojka	Zavedení spojky	Eliminace spojky
$\Leftrightarrow$	$\dfrac{\boxed{\begin{array}{c}\varphi\\ \vdots\\ \psi\end{array}} \qquad \boxed{\begin{array}{c}\psi\\ \vdots\\ \varphi\end{array}}}{\varphi \Leftrightarrow \psi}\ \mathrm{i}\Leftrightarrow$	$\dfrac{\varphi \qquad \varphi \Leftrightarrow \psi}{\psi}\ \mathrm{e}\Leftrightarrow_1 \qquad \dfrac{\varphi \Leftrightarrow \psi \qquad \psi}{\varphi}\ \mathrm{e}\Leftrightarrow_2$
$\top$	$\dfrac{}{\top}\ \mathrm{i}\top$	není

Nepřímý důkaz

Do této chvíle jsme zavedli pravidla přirozené dedukce pro takzvanou *intuicionistickou výrokovou logiku*. Naše poslední pravidlo, *eliminace dvojité negace*, upraví chování spojky negace:

$$\frac{\neg\neg\varphi}{\varphi}\ \mathrm{e}\neg\neg$$

Příklad 2.14. Dokažme logický úsudek

$$\neg(a \land b) \vdash \neg a \lor \neg b.$$

Důkaz.

1.	$\neg(a \wedge b)$	P
2.	$\neg(\neg a \vee \neg b)$	P
3.	$\neg a$	P
4.	$\neg a \vee \neg b$	$\mathrm{i}\vee_1$, 3
5.	$\perp$	$\mathrm{e}\neg$, 4, 2
6.	$\neg\neg a$	$\mathrm{i}\neg$, 3-5
7.	a	$\mathrm{e}\neg\neg$, 6
8.	$\neg b$	P
9.	$\neg a \vee \neg b$	$\mathrm{i}\vee_2$, 8
10.	$\perp$	$\mathrm{e}\neg$, 10, 2
11.	$\neg\neg b$	$\mathrm{i}\neg$, 8-10
12.	b	$\mathrm{e}\neg\neg$, 11
13.	$a \wedge b$	$\mathrm{i}\wedge$, 7, 12
14.	$\perp$	$\mathrm{e}\neg$, 13, 2
15.	$\neg\neg(\neg a \vee \neg b)$	$\mathrm{i}\neg$, 2-14
16.	$\neg a \vee \neg b$	$\mathrm{e}\neg\neg$, 15

□

Užitím pravidla e¬¬lze dokázat korektnost odvozeného pravidla: *pravidla vyloučeného třetího* (anglicky *law of excluded middle*, LEM):

$$\frac{}{\varphi \vee \neg\varphi}\ \mathrm{LEM}$$

Příklad 2.15. Dokažme logický úsudek

$$\vdash \varphi \vee \neg\varphi.$$

Důkaz.

1.	$\neg(\varphi \vee \neg\varphi)$	P
2.	φ	P
3.	$\varphi \vee \neg\varphi$	$\mathrm{i}\vee_1$, 2
4.	$\perp$	$\mathrm{e}\neg$, 3, 1
5.	$\neg\varphi$	$\mathrm{i}\neg$, 2-4
6.	$\varphi \vee \neg\varphi$	$\mathrm{i}\vee_2$, 5
7.	$\perp$	$\mathrm{e}\neg$, 6, 1
8.	$\neg\neg(\varphi \vee \neg\varphi)$	$\mathrm{i}\neg$, 1-7
9.	$\varphi \vee \neg\varphi$	$\mathrm{e}\neg\neg$, 8

$\square$

Poznámka 2.16. Pravidlem e¬¬lze též odvodit pravidlo „důkazu sporem", takzvané *reductio ad absurdum* (RAA):

$$\dfrac{\boxed{\begin{array}{c}\neg\varphi\\ \vdots\\ \bot\end{array}}}{\varphi}\ \text{RAA}$$

Pozor! Toto pravidlo *není totožné s pravidlem zavedení negace*, i když vypadá podobně. V klasické matematice se však důkazové postupy odpovídající zavedení negace (důkaz negace) a důkazu sporem (reductio ad absurdum) obvykle nerozlišují, neboť klasická logika rozdíly mezi nimi stírá.

Poznámka 2.17. Nabízí se otázka, proč nezavést následující pravidlo zavedení dvojité negace (označené třeba i¬¬)

$$\frac{\varphi}{\neg\neg\varphi}\ \text{i}\neg\neg$$

když jsme zavedli pravidlo eliminace dvojité negace e¬¬. Toto pravidlo, ač korektní, lze odvodit z již zavedených pravidel:

1.	φ	P
2.	$\neg\varphi$	P
3.	$\bot$	e¬, 1, 2
4.	$\neg\neg\varphi$	i¬, 2-3

Na následující straně naleznete soupis všech pravidel přirozené dedukce zavedených na přednášce.

Spojka	Zavedení spojky	Eliminace spojky
$\wedge$	$\dfrac{\varphi \qquad \psi}{\varphi \wedge \psi}\ \mathrm{i}\wedge$	$\dfrac{\varphi \wedge \psi}{\varphi}\ \mathrm{e}\wedge_1 \qquad \dfrac{\varphi \wedge \psi}{\psi}\ \mathrm{e}\wedge_2$
$\vee$	$\dfrac{\varphi}{\varphi \vee \psi}\ \mathrm{i}\vee_1 \qquad \dfrac{\psi}{\varphi \vee \psi}\ \mathrm{i}\vee_2$	$\dfrac{\varphi \vee \psi \qquad \boxed{\begin{array}{c}\varphi\\ \vdots\\ \chi\end{array}} \qquad \boxed{\begin{array}{c}\psi\\ \vdots\\ \chi\end{array}}}{\chi}\ \mathrm{e}\vee$
$\Rightarrow$	$\dfrac{\boxed{\begin{array}{c}\varphi\\ \vdots\\ \psi\end{array}}}{\varphi \Rightarrow \psi}\ \mathrm{i}\Rightarrow$	$\dfrac{\varphi \qquad \varphi \Rightarrow \psi}{\psi}\ \mathrm{e}\Rightarrow$
$\Leftrightarrow$	$\dfrac{\boxed{\begin{array}{c}\varphi\\ \vdots\\ \psi\end{array}} \qquad \boxed{\begin{array}{c}\psi\\ \vdots\\ \varphi\end{array}}}{\varphi \Leftrightarrow \psi}\ \mathrm{i}\Leftrightarrow$	$\dfrac{\varphi \qquad \varphi \Leftrightarrow \psi}{\psi}\ \mathrm{e}\Leftrightarrow_1 \qquad \dfrac{\varphi \Leftrightarrow \psi \qquad \psi}{\varphi}\ \mathrm{e}\Leftrightarrow_2$
$\neg$	$\dfrac{\boxed{\begin{array}{c}\varphi\\ \vdots\\ \bot\end{array}}}{\neg\varphi}\ \mathrm{i}\neg$	$\dfrac{\varphi \qquad \neg\varphi}{\bot}\ \mathrm{e}\neg$
$\top$	$\dfrac{}{\top}\ \mathrm{i}\top$	není
$\bot$	není	$\dfrac{\bot}{\varphi}\ \mathrm{e}\bot$
$\neg\neg$	(netřeba) $\dfrac{\varphi}{\neg\neg\varphi}\ \mathrm{i}\neg\neg$	$\dfrac{\neg\neg\varphi}{\varphi}\ \mathrm{e}\neg\neg$

Pomocné pravidlo: pravidlo iterace (it).
Odvozené pravidlo LEM (zákon vyloučeného třetího, tertium non datur):

$$\dfrac{}{\varphi \vee \neg\varphi}\ \mathrm{LEM}$$

Ekvivalence a negace jako „syntactic sugar“

Spojka ekvivalence ($\Leftrightarrow$) je v našem jazyce v jistém smyslu „nepotřebná“: můžeme ji chápat jako konjunkci dvou implikací (tedy tak, jak ji v matematice obvykle používáme). K formalisaci našeho pozorování zavedeme pojem logické ekvivalence:

Definice 2.18. Řekneme, že formule φ a ψ jsou *logicky ekvivalentní*, pokud platí $\varphi \vdash \psi$ a současně $\psi \vdash \varphi$.

Příklad 2.19. Nechť φ a ψ jsou dvě formule výrokové logiky. Dokažme, že formule $\varphi \Leftrightarrow \psi$ a $(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$ jsou logicky ekvivalentní.

Důkaz. Sestrojme nejprve důkaz

$$\varphi \Leftrightarrow \psi \vdash (\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi).$$

1.	$\varphi \Leftrightarrow \psi$	P
2.	φ	P
3.	ψ	$e{\Leftrightarrow}_1$, 2, 1
4.	$\varphi \Rightarrow \psi$	$i{\Rightarrow}$, 2-3
5.	ψ	P
6.	φ	$e{\Leftrightarrow}_2$, 1, 5
7.	$\psi \Rightarrow \varphi$	$i{\Rightarrow}$, 5-6
8.	$(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$	$i{\wedge}$, 4, 7

Nyní sestrojme důkaz

$$(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi) \vdash \varphi \Leftrightarrow \psi.$$

1.	$(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$	P
2.	$\varphi \Rightarrow \psi$	$e{\wedge}_1$, 1
3.	$\psi \Rightarrow \varphi$	$e{\wedge}_2$, 1
4.	φ	P
5.	ψ	$e{\Rightarrow}$, 4, 1
6.	ψ	P
7.	φ	$e{\Rightarrow}$, 6, 3
8.	$\varphi \Leftrightarrow \psi$	$i{\Leftrightarrow}$, 4-5, 6-7

□

Podobně je také možné „vypustit“ logickou spojku negace:

Příklad 2.20. Nechť je φ formule výrokové logiky. Dokažme, že formule $\neg\varphi$ a $\varphi \Rightarrow \bot$ jsou logicky ekvivalentní.

Důkaz. Sestrojme nejprve důkaz

$$\neg\varphi \vdash \varphi \Rightarrow \bot.$$

1.	$\neg\varphi$	P
2.	φ	P
3.	$\bot$	e$\neg$, 2, 1
4.	$\varphi \Rightarrow \bot$	i$\Rightarrow$, 2-3

Nyní sestrojme důkaz

$$\varphi \Rightarrow \bot \vdash \neg\varphi.$$

1.	$\varphi \Rightarrow \bot$	P
2.	φ	P
3.	$\bot$	e$\Rightarrow$, 2, 1
4.	$\neg\varphi$	i$\neg$, 2-3

$\square$

Proč byly předchozí dva důkazy triviální? Pravidla pro zavedení a eliminaci negace přesně odpovídají pravidlům pro práci s implikací ve speciálním případě, kdy za formuli ψ dosadíme formuli $\bot$.

3 Sémantika výrokové logiky

Sémantika je, obecně řečeno, nauka o *významu*. V této sekci se budeme zabývat sémantikou výrokové logiky. To znamená, že vezmeme náš *formální jazyk* (jazyk výrokové logiky), a formulím z tohoto jazyka se pokusíme přiřadit jejich význam. Nejprve je nutno mít představu, co by význam formule měl být. Obecně se formule používají k representaci *výroků*, u nichž nás zajímá jejich *pravdivost*. Význam nějaké formule φ by tedy měl být buď pravda, nebo nepravda. Pravdivost většiny formulí však není pojem *absolutní*, nýbrž relativní: to, zda je formule pravdivá, závisí na tom, zda jsou pravdivé její jednotlivé části (podformule). Tento problém vyřešíme elegantně: definice významu a pravdivosti formule bude mít *rekursivní charakter*. Způsob, jakým význam (sémantiku) formulí výrokové logiky zavedeme, je velmi podobný způsobu, jakým se v teoretické computer science popisuje sémantika programovacích jazyků.

Definice 3.1. Číslem 0 budeme označovat nepravdu a číslem 1 budeme označovat pravdu. Tyto dvě hodnoty budeme označovat jako *pravdivostní hodnoty*, a množinu pravdivostních hodnot označíme jako

$$\mathsf{Bool} = \{0, 1\}.$$

Booleovské funkce

Definice 3.2. Nechť n je přirozené číslo. Funkci f typu

$$f : \mathsf{Bool}^n \to \mathsf{Bool}$$

nazveme n-*ární booleovskou funkcí*.

Poznámka 3.3. Mírně obecněji můžeme definovat booleovské funkce jako funkce tvaru

$$f : \mathsf{Bool}^X \to \mathsf{Bool},$$

kde X je jakákoli konečná množina. Pro „booleovskost" je podstatné, aby výstupy i vstupy dané funkce byly booleovské hodnoty (resp. jejich n-tice).

Příklad 3.4. Příkladem booleovské funkce pro $n = 3$ je funkce f zadaná následovně:

$$\begin{aligned} f : \mathsf{Bool}^3 &\to \mathsf{Bool} \\ (0,0,0) &\mapsto 0 \\ (0,0,1) &\mapsto 0 \\ (0,1,0) &\mapsto 0 \\ (0,1,1) &\mapsto 1 \\ (1,0,0) &\mapsto 0 \\ (1,0,1) &\mapsto 1 \\ (1,1,0) &\mapsto 1 \\ (1,1,1) &\mapsto 1 \end{aligned}$$

Poznámka 3.5. Abychom definovali ternární ($n = 3$) booleovskou funkci, musíme definovat její výstup pro všech $2^n = 2^3 = 8$ možných vstupů. To jsme učinili v předchozím příkladu. Náš zápis je však (pro booleovské funkce) mírně typograficky nestandardní – zpravidla se k úplnému popisu booleovské funkce používá tabulka, přesněji řečeno *pravdivostní tabulka* dané booleovské funkce. Pro funkci f z předchozího příkladu by tabulka vypadala následovně:

vstup 1	vstup 2	vstup 3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Často jsou vstupy nějak pojmenovány. Kdyby v našem případě byl první vstup pojmenován x_1, druhý vstup x_2 a třetí vstup x_3, mohli bychom označit množinu „vstupů" jako $X = \{x_1, x_2, x_3\}$, a danou booleovskou funkci representovat jako funkci

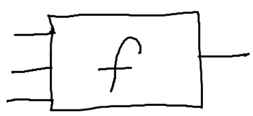
$$\mathsf{f} : \mathsf{Bool}^X \to \mathsf{Bool}$$

zadanou pravdivostní tabulkou

x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Rozdíl je spíše kosmetický, v dalším textu se nám ale druhý způsob popisu booleovských funkcí bude hodit více.

Poznámka 3.6. Často je vhodné si booleovské funkce představovat jako krabičky (*black boxes*), do kterých vede n drátů (jeden pro každý vstup) a ze kterých vede jeden drát (výstupní).



Podle toho, jaký vstup (tj. kombinaci nul a jedniček) „posíláme" do vstupních drátů, booleovská funkce vyšle nulu či jedničku do výstupního drátu. Přesné chování booleovské funkce je dáno její tabulkou. Tento přístup je velmi častý v praktické computer science při tvorbě *logických obvodů* pomocí *logických hradel* (více viz např. předmět Architektura počítačů).

Sémantika formulí výrokové logiky

Představme si naivní přístup ke snaze o definici významu formulí. Každá formule by měla být buď pravdivá, nebo nepravdivá. Jak u formule rozhodnout, zda je pravdivá či nepravdivá? Pokud by naším cílem bylo pouze přiřadit každé formuli pravdivostní hodnotu, stačilo nám zvolit si zcela libovolnou funkci

$$\mathsf{s} : \mathsf{Fm}(\mathsf{At}) \to \mathsf{Bool}.$$

Funkce tohoto typu by přesně splnila náš požadavek: množinou vstupů je množina všech formulí našeho jazyka, a výstupem pro každou formuli φ je jedna z hodnot 0, 1. Tento přístup má však několik zjevných vad:

1. Jelikož na s neklademe další požadavky, může nějaké dvě formule $\varphi_1, \varphi_2 \in \mathsf{Fm}(\mathsf{At})$ obě ohodnotit jako nepravdivé:

   $$s(\varphi_1) = s(\varphi_2) = 0.$$

   Současně může s ohodnotit formuli $\varphi_1 \wedge \varphi_2$ jako pravdivou:

   $$s(\varphi_1 \wedge \varphi_2) = 1.$$

   To je však v ostrém kontrastu s naším očekáváním toho, jak by konjunkce dvou pravdivých formulí měla být ohodnocena. Konjunkcí se snažíme modelovat spojku „a“, a jsou-li oba dva konjunkty φ_1, φ_2 pravdivé, měla by být pravdivá i celá konjunkce $\varphi_1 \wedge \varphi_2$.

2. Formule $\perp$ formalisuje absurditu. Náš přístup však funkci s nezakazuje přiřadit formuli $\perp$ pravdivostní hodnotu 1.

3. Předchozí dva problémy naráží na to, že formule obsahující logické spojky by neměly mít význam daný „náhodně“, ale jejich význam by měl být dán významem jejich podformulí a významem jednotlivých logických spojek. Nabízí se tedy následující postup:

   (a) Formuli $\perp$ je nutné přiřadit pravdivostní hodnotu 0.

   (b) Formuli $\top$ je nutné přiřadit pravdivostní hodnotu 1.

   (c) Každé logické spojce $\circ$ je nutné nějaký „návod“, který určuje význam formulí typu $\varphi_1 \circ \varphi_2$, pokud známe význam formulí φ_1 a φ_2 (a pro unární spojku $\neg$ popsat, jak význam formule typu $\neg\varphi$ závisí na významu formule φ).

   Tím se však nabízí další otázka: jak zjistit pravdivost formulí, které žádné logické spojky neobsahují? Uvažujme například nějakou atomickou formuli $a \in \mathsf{At}$. Má být pravdivá? Má být nepravdivá? Kdo to rozhodne? Ukážeme, že tuto nepříjemnost lze vyřešit elegantně: v případě, že pravdivost atomických formulí známe předem (*a priori*), o pravdivosti všech neatomických formulí už bude rozhodnuto *automaticky*. Pokud pravdivost atomických formulí předem neznáme, pak *významem* formule φ bude zobrazení

   $$[\![\varphi]\!] : \mathsf{Bool}^{\mathsf{At}} \to \mathsf{Bool},$$

   které jako vstup očekává ohodnocení atomických formulí, tedy zobrazení typu

   $$u_{\mathsf{At}} : \mathsf{At} \to \mathsf{Bool},$$

   a jako výstup vrátí pravdivostní hodnotu formule φ za předpokladu, že pravdivost atomických formulí je dána zobrazením u_{At}.

Poznámka 3.7 (odvození booleovských funkcí pro logické spojky)**.** Než přejdeme k úplné definici sémantiky výrokových formulí, pokusíme se popsat sémantiku samotných logických spojek. Logické spojky, velmi zhruba řečeno, používáme jako formální protějšek spojek z přirozeného jazyka. Tímto způsobem tedy jejich sémantiku (význam) budeme motivovat.

Každé logické spojce přiřadíme booleovskou funkci, a tuto funkci budeme chápat jako význam dané spojky.

Konjunkce Spojka $\wedge$, tedy konjunkce, přibližně odpovídá spojce „a“ přirozeného jazyka v poměru slučovacím. Výrok „A a B“ chápeme jako pravdivý přesně v případech, kdy jsou pravdivé oba výroky, „A“ i „B“.

Disjunkce Spojka $\vee$, tedy disjunkce, přibližně odpovídá spojce „nebo“ přirozeného jazyka (a *ne*chápeme ji v poměru vylučovacím). Výrok „A nebo B“ chápeme jako pravdivý v případech, kdy je pravdivý alespoň jeden z výroků „A“ a „B“.

Negace Spojka $\neg$, tedy negace, odpovídá přibližně slovnímu obratu „není pravda, že…“. Výrok „není pravda, že A“ chápeme jako pravdivý přesně v případě, kdy „A“ pravdivý *není*.

Implikace Spojka $\Rightarrow$, tedy implikace, *velmi volně* odpovídá obratu „pokud …, pak …“. Výrok „pokud A, pak B“ lze chápat jako *slib*: ten je pravdivý, pokud slib neporušíme. Jediný způbob, jakým lze slib „pokud A, pak B“ porušit je v případě, že „A“ je pravda a „B“ je nepravda.

Ekvivalence Spojka $\Leftrightarrow$, tedy exvivalence, odpovídá obratu „právě tehdy, když“. Výrok „A právě tehdy, když B“ chápeme jako pravdivý v případě, že se pravdivost A a B shoduje: jsou oba pravdivé či oba nepravdivé.

Definice 3.8 (booleovské funkce pro logické spojky)**.** *Význam negace* je booleovská funkce $[\![\neg]\!]$ zadaná následující pravdivostní tabulkou:

vstup 1	$[\![\neg]\!]$
0	1
1	0

Významy binárních spojek *konjunkce, disjunkce, implikace a ekvivalence* jsou booleovské funkce $[\![\wedge]\!]$, $[\![\vee]\!]$, $[\![\Rightarrow]\!]$ a $[\![\Leftrightarrow]\!]$ zadané následujícími pravdivostními tabulkami:

vstup 1	vstup 2	$[\![\wedge]\!]$	$[\![\vee]\!]$	$[\![\Rightarrow]\!]$	$[\![\Leftrightarrow]\!]$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

Významy logických konstant $\bot$ a $\top$ jsou pravdivostní hodnoty $[\![\bot]\!] = 0$ a $[\![\top]\!] = 1$:

	$[\![\bot]\!]$	$[\![\top]\!]$
	0	1

Definice 3.9. Zobrazení $u : \mathsf{Fm}(\mathsf{At}) \to \mathsf{Bool}$ nazveme *pravdivostním ohodnocením*, pokud splňuje následující podmínky (pro každou dvojici formulí $\varphi, \psi \in \mathsf{Fm}(\mathsf{At})$):

- $u(\top) = [\![\top]\!] = 1$.
- $u(\bot) = [\![\bot]\!] = 0$.
- $u(\varphi \land \psi) = [\![\land]\!](u(\varphi), u(\psi))$.
- $u(\varphi \lor \psi) = [\![\lor]\!](u(\varphi), u(\psi))$.
- $u(\varphi \Rightarrow \psi) = [\![\Rightarrow]\!](u(\varphi), u(\psi))$.
- $u(\varphi \Leftrightarrow \psi) = [\![\Leftrightarrow]\!](u(\varphi), u(\psi))$.
- $u(\neg\varphi) = [\![\neg]\!](u(\varphi))$.

Pozorování 3.10. Z definice vidíme, že každé pravdivostní ohodnocení je „svázáno“ našimi požadavky na to, jak interpretovat jednotlivé logické spojky. Ukážeme, že jediná „volnost“, kterou pravdivostní ohodnocení má, je volnost v přiřazení pravdivostních hodnot *atomickým formulím*.

Definice 3.11. Zobrazení tvaru $u_{\mathsf{At}} : \mathsf{At} \to \mathsf{Bool}$ říkáme *ohodnocení atomů*. Pokud pro pravdivostní ohodnocení $u : \mathsf{Fm}(\mathsf{At}) \to \mathsf{Bool}$ platí

$$u_{\mathsf{At}}(a) = u(a)$$

pro všechna $a \in \mathsf{At}$, říkáme, že je u *rozšířením* ohodnocení atomů u_{At}.

Poznámka 3.12. Každé ohodnocení atomů u_{At} lze jednoznačně rozšířit na pravdivostní ohodnocení u. Máme-li zadáno ohodnocení atomů

$$u_{\mathsf{At}} : \mathsf{At} \to \mathsf{Bool},$$

způsob konstrukce pravdivostního ohodnocení

$$u : \mathsf{Fm}(\mathsf{At}) \to \mathsf{Bool}$$

rozšiřujícího u_{At} je rekursivní: víme, že pro atomické formule $a \in \mathsf{At}$ musí platit $u(a) = u_{\mathsf{At}}(a)$ (to je základní krok rekurse), a máme-li ohodnotit složitější formuli, např. formuli ψ tvaru $\varphi_1 \Rightarrow \varphi_2$, víme z definice pravdivostního ohodnocení, že $u(\varphi_1 \Rightarrow \varphi_2)$ musí být rovno $[\![\Rightarrow]\!](u(\varphi_1), u(\varphi_2))$. Pokud již formule φ_1 a φ_2 umíme pomocí pravdivostního ohodnocení u vyhodnotit (což je náš indukční předpoklad), umíme zjevně vyhodnotit i formuli ψ.

Principu, kdy význam složitého objektu (u nás formule) lze vyvodit z významu jeho jednodušších částí, se říká *komposicionalita*, česky možná *skladatelnost*.

Pozorování 3.13. Každé pravdivostní lze *omezit* na ohodnocení atomů: máme-li zadáno pravdivostní ohodnocení $u : \mathsf{Fm}(\mathsf{At}) \to \mathsf{Bool}$, můžeme definovat zobrazení

$$\begin{aligned} u_{\mathsf{At}} : \mathsf{At} &\to \mathsf{Bool} \\ a &\mapsto u(a). \end{aligned}$$

Pravdivostní ohodnocení u rozšiřuje u_{At}.

Důsledek 3.14. Každé pravdivostní ohodnocení $u : \mathsf{Fm}(\mathsf{At}) \to \mathsf{Bool}$ je jednoznačně zadáno svým ohodnocením atomů. Skutečně: jakmile zjistíme, že dvě pravdivostní ohodnocení u a v rozšiřují to samé ohodnocení atomů, můžeme využít faktu, že ohodnocení atomů lze rozšířit na pravdivostní ohodnocení *jednoznačně*, a tedy platí $u = v$.

Poznámka 3.15. Kolik existuje různých pravdivostních ohodnocení

$$u : \mathsf{Fm}(\mathsf{At}) \to \mathsf{Bool},$$

když At obsahuje přesně n prvků?

Jelikož je každé u jednoznačně zadáno zobrazením

$$u_{\mathsf{At}} : \mathsf{At} \to \mathsf{Bool},$$

a zadat zobrazení

$$\begin{aligned} u_{\mathsf{At}} : \mathsf{At} &\to \mathsf{Bool} \\ a_1 &\mapsto b_1 \\ a_2 &\mapsto b_2 \\ &\dots \\ a_n &\mapsto b_n \end{aligned}$$

znamená zadat n-tici booleovských hodnot $(b_1, b_2, \dots, b_n)$, je takových různých ohodnocení přesně 2^n (počet různých n-tic booleovských hodnot).

Důsledek 3.16. Pravdivostní tabulka formule $\varphi \in \mathsf{Fm}(\mathsf{At})$ pro n-prvkovou množinu At obsahuje 2^n řádků.

Příklad 3.17. Nalezněme z pravdivostní tabulky nějaké pravdivostní ohodnocení u, ve kterém je formule $\varphi = (a \wedge b) \Rightarrow (b \wedge c)$ pravdivá, a nějaké pravdivostní ohodnocení v, ve kterém je φ nepravdivá.

a	b	c	$($	a	$\wedge$	b	$)$	$\Rightarrow$	$($	b	$\wedge$	c	$)$
0	0	0		0	0	0		1		0	0	0	
0	0	1		0	0	0		1		0	0	1	
0	1	0		0	0	1		1		1	0	0	
0	1	1		0	0	1		1		1	1	1	
1	0	0		1	0	0		1		0	0	0	
1	0	1		1	0	0		1		0	0	1	
1	1	0		1	1	1		0		1	0	0	
1	1	1		1	1	1		1		1	1	1	

Formule φ je pravdivá například v pravdivostním ohodnocení $u : \mathsf{Fm}(\mathsf{At}) \to \mathsf{Bool}$, kde

$$\begin{aligned} u(a) &= 0, \\ u(b) &= 1, \\ u(c) &= 0. \end{aligned}$$

Formule φ je nepravdivá v pravdivostním ohodnocení $v : \mathsf{Fm}(\mathsf{At}) \to \mathsf{Bool}$, kde

$$\begin{aligned} v(a) &= 1, \\ v(b) &= 1, \\ v(c) &= 0. \end{aligned}$$

Definice 3.18. Mějme formuli $\varphi \in \mathsf{Fm}(\mathsf{At})$. Její *význam* je její pravdivostní tabulka, tedy booleovská funkce

$$\begin{aligned} [\![\varphi]\!] : \mathsf{Bool}^{\mathsf{At}} &\to \mathsf{Bool} \\ u_{\mathsf{At}} &\mapsto u(\varphi). \end{aligned}$$

Definice 3.19. Formule φ a ψ jsou *sémanticky ekvivalentní*, jestliže pro každé pravdivostní ohodnocení u platí, že

$$u(\varphi) = u(\psi).$$

Tento fakt značíme $\varphi \mathrel{\text{⫤⊨}} \psi$.

Poznámka 3.20. Formule φ a ψ jsou sémanticky ekvivalentní, pokud mají stejný význam: to jest, pokud $[\![\varphi]\!] = [\![\psi]\!]$ (pravdivostní tabulky φ a ψ se shodují).

Příklad 3.21. Formule $a \Rightarrow b$ a $\neg a \vee b$ jsou sémanticky ekvivalentní. To lze zjistit například porovnáním jejich pravdivostních tabulek.

a	b	$a \Rightarrow b$	$\neg a \vee b$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	1

Tvrzení 3.22. Vztah (relace) sémantické ekvivalence je relací ekvivalence. To znamená, že pro každou trojici formulí φ, ψ, χ platí:

Reflexivita $\varphi \mathrel{\text{⫤⊨}} \varphi$.

Symetrie Pokud $\varphi \mathrel{\text{⫤⊨}} \psi$, pak $\psi \mathrel{\text{⫤⊨}} \varphi$.

Transitivita Pokud $\varphi \mathrel{\text{⫤⊨}} \psi$ a $\psi \mathrel{\text{⫤⊨}} \chi$, pak $\varphi \mathrel{\text{⫤⊨}} \chi$.

Důkaz. Tvrzení plyne přímo z pozorování, že sémanticky ekvivalentní formule jsou ty se stejným významem:

Reflexivita Pro φ platí $[\![\varphi]\!] = [\![\varphi]\!]$, tedy $\varphi \mathrel{\text{H}} \varphi$.

Symetrie Pokud $\varphi \mathrel{\text{H}} \psi$, pak $[\![\varphi]\!] = [\![\psi]\!]$, tedy $[\![\psi]\!] = [\![\varphi]\!]$, což znamená, že $\psi \mathrel{\text{H}} \varphi$.

Transitivita Pokud $\varphi \mathrel{\text{H}} \psi$ a $\psi \mathrel{\text{H}} \chi$, znamená to, že $[\![\varphi]\!] = [\![\psi]\!]$ a $[\![\psi]\!] = [\![\chi]\!]$, z čehož plyne $[\![\varphi]\!] = [\![\chi]\!]$. To znamená, že $\varphi \mathrel{\text{H}} \chi$.

$\square$

Pozorování 3.23. Relace sémantické ekvivalence rozkládá množinu všech formulí $\mathsf{Fm}(\mathsf{At})$ na třídy ekvivalence: pro každou formuli $\varphi \in \mathsf{Fm}(\mathsf{At})$ máme množinu $E_\varphi = \{\psi \in \mathsf{Fm}(\mathsf{At}) \mid \varphi \mathrel{\text{H}} \psi\}$ všech formulí sémanticky ekvivalentních s φ (*synonyma* formule φ).

> **K zamyšlení.** Kolik formulí obsahuje každá třída ekvivalence z předchozího pozorování? Umíme už „detekovat synonyma“, to jest, máme algoritmus, který o dvojici formulí rozhodne, zda jsou sémanticky ekvivalentní?

Tvrzení 3.24. Pro každé dvě formule φ a ψ platí:

$$\varphi \mathrel{\text{H}} \psi \text{ právě tehdy, když je formule } \varphi \Leftrightarrow \psi \text{ tautologie.}$$

Důkaz. Formule φ a ψ jsou sémanticky ekvivalentní právě tehdy, když pro každé pravdivostní ohodnocení u platí $u(\varphi) = u(\psi)$. To je pravda právě tehdy, když pro každé pravivostní ohodnocení platí $u(\varphi \Leftrightarrow \psi) = 1$, což přesně znamená, že $\varphi \Leftrightarrow \psi$ je tautologie. $\square$

Definice 3.25. Formuli φ nazveme

1. *splnitelnou*, pokud je pravdivá v alespoň jednom pravdivostním ohodnocení,
2. *tautologií*, pokud je pravdivá ve všech pravdivostních ohodnoceních,
3. *kontradikcí*, pokud není pravdivá v žádném pravdivostním ohodnocení.

Příklad 3.26. Formule $\varphi = (a \wedge b) \Rightarrow (b \wedge c)$ z předchozího příkladu je splnitelná, ale není to tautologie.

Pojem splnitelnosti lze rozšířit na *množiny formulí*.

Definice 3.27. Množina formulí S je *splnitelná*, pokud existuje alespoň jedno pravdivostní ohodnocení, ve kterém jsou pravdivé všechny formule $\varphi \in S$ (zároveň). Jinak množinu S nazveme *nesplnitelnou*.

Pokud jsou v pravdivostním ohodnocení u pravdivé všechny formule z množiny S, značíme tuto situaci jako $u(S) = 1$.

Příklad 3.28. Množina $S = \{a \vee b, \neg a, \neg b\}$ je nesplnitelná.

a	b	$a \vee b$	$\neg a$	$\neg b$
0	0	0	1	1
0	1	1	1	0
1	0	1	0	1
1	1	1	0	0

Přitom každá jednotlivá formule z množiny S je sama o sobě splnitelná.

3.1 Sémantický důsledek

Definice 3.29. Formule φ je *sémantickým důsledkem* množiny formulí S, pokud je v každém pravdivostním ohodnocení, v němž je S pravdivá, pravdivá i formule φ. Tento fakt značíme $S \models \varphi$.

Příklad 3.30. Formule b je sémantickým důsledkem množiny formulí $\{a, a \Rightarrow b\}$, to jest, platí $\{a, a \Rightarrow b\} \models b$. (Je častým zvykem nepsat nalevo od znaku $\models$ množinové závorky, a psát místo toho například $a, a \Rightarrow b \models b$.) To lze ověřit například studiem pravdivostní tabulky:

a	b	$a \Rightarrow b$
0	0	1
0	1	1
1	0	0
1	1	1

V každém řádku (pravdivostním ohodnocení), ve kterém jsou pravdivé obě formule a a $a \Rightarrow b$, musí být pravdivá i formule b.

Pozorování 3.31. Následující vlastnosti sémantického důsledku je snadné ověřit:

1. $\varphi \mathrel{\reflectbox{$\models$}}\!\models \psi$ platí právě tehdy, když platí současně $\varphi \models \psi$ a $\psi \models \varphi$.
2. Pro $S = \emptyset$ platí, že $S \models \varphi$ právě tehdy, když je φ tautologie.
3. Množina formulí N je nesplnitelná právě tehdy, když pro všechny formule φ platí $N \models \varphi$.
4. $\varphi \models \psi$ platí právě tehdy, když je $\varphi \Rightarrow \psi$ tautologie.

Tvrzení 3.32 (Věta o dedukci)**.** Pro každou množinu formulí $S \subseteq \mathsf{Fm}(\mathsf{At})$ a všechny formule $\varphi, \psi \in \mathsf{Fm}(\mathsf{At})$ platí:

$$S \cup \{\varphi\} \models \psi \text{ právě tehdy, když } S \models \varphi \Rightarrow \psi.$$

Důkaz. Dokážeme každý směr ekvivalence zvlášť.

1. Nechť platí $S \cup \{\varphi\} \vDash \psi$. Dokážeme, že $S \vDash \varphi \Rightarrow \psi$. Mějme pravdivostní ohodnocení u takové, že $u(S) = 1$. Naším úkolem je ukázat, že $u(\varphi \Rightarrow \psi) = 1$. Pokud $u(\varphi) = 0$, pak $u(\varphi \Rightarrow \psi) = 1$. Pokud naopak $u(\varphi) = 1$, znamená to, že $u(S \cup \{\varphi\}) = 1$, a tedy z předpokladu $u(\psi) = 1$. To znamená, že $u(\varphi \Rightarrow \psi) = 1$. Důsledek $S \vDash \varphi \Rightarrow \psi$ tedy platí.

2. Nechť platí $S \vDash \varphi \Rightarrow \psi$. Dokážeme, že $S \cup \{\varphi\} \vDash \psi$. Mějme pravdivostní ohodnocení u takové, že $u(S \cup \{\varphi\}) = 1$. Naším úkolem je ukázat, že $u(\psi) = 1$. Jelikož $u(S \cup \{\varphi\}) = 1$, pak i $u(S) = 1$, a z předpokladu tedy plyne, že $u(\varphi \Rightarrow \psi) = 1$. Jelikož $u(S \cup \{\varphi\}) = 1$, pak i $u(\varphi) = 1$. Jediný způsob, jak může platit $u(\varphi) = u(\varphi \Rightarrow \psi) = 1$, je ten, že $u(\psi) = 1$, což jsme měli dokázat.

□

Definice 3.33. Ať $S \subseteq \mathsf{Fm}(\mathsf{At})$. Množinu

$$\{\varphi \in \mathsf{Fm}(\mathsf{At}) \mid S \vDash \varphi\}$$

všech sémantických důsledků množiny formulí S označujeme jako $\mathsf{Con}(S)$, tzv. *důsledkový obal* množiny S.

Důsledkový obal množiny S skutečně má vlastnosti, na které jsme u pojmu „obalu“ zvyklí:

Tvrzení 3.34. Ať $S, T \subseteq \mathsf{Fm}(\mathsf{At})$. Pak platí:

1. $S \subseteq \mathsf{Con}(S)$.
2. Pokud $S \subseteq T$, pak $\mathsf{Con}(S) \subseteq \mathsf{Con}(T)$.
3. $\mathsf{Con}(\mathsf{Con}(S)) \subseteq \mathsf{Con}(S)$.

Důkaz. Cvičení. □

> **K zamyšlení.** Vzpomeňte si na své studium lineární algebry. Navrhněte definici pojmu „sémanticky důsledkově nezávislá množina formulí“. Uveďte příklad závislé a nezávislé množiny formulí.

3.2 Normální formy

Každá formule $\varphi \in \mathsf{Fm}(\mathsf{At})$ určuje pravdivostní tabulku, tedy booleovskou funkci

$$[\![\varphi]\!] : \mathsf{Bool}^{\mathsf{At}} \to \mathsf{Bool}.$$

Lze každou booleovskou funkci „representovat“ nějakou formulí výrokové logiky? Uvidíme, že ano, a předvedeme dokonce dva (duální) způsoby.

Definice 3.35. Každou formuli tvaru a či $\neg a$ (kde $a \in \mathsf{At}$) nazveme *literál*.

Poznámka 3.36. Literály můžeme definovat též pomocí Backusovy-Naurovy formy, a to následovně:

$$L ::= a \mid \neg a.$$

Poznámka 3.37. Připomeňte si značení „velké disjunkce“ a „velké konjunkce“ z materiálů ke cvičení.

Disjunktivní normální forma

Definice 3.38. Řekneme, že formule φ je v *disjunktivní normální formě*, pokud

$$\varphi = \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} \varphi_{i,j}$$

pro nějaké $n \in \mathbb{N}$, nějaká $m_i \in \mathbb{N}$ $(i = 1, \ldots, n)$, a všechny formule $\varphi_{i,j}$ jsou literály.

Tvrzení 3.39. Každá formule $\varphi \in \mathsf{Fm}(\mathsf{At})$ (kde At je konečná) je sémanticky ekvivalentní nějaké formuli ψ v disjunktivní normální formě.

Důkaz. Ať

$$\begin{aligned} l : \mathsf{At} \times \mathsf{Bool}^{\mathsf{At}} &\to \mathsf{Fm}(\mathsf{At}) \\ (a, u_{\mathsf{At}}) &\mapsto \begin{cases} a & \text{pokud } u_{\mathsf{At}}(a) = 1, \\ \neg a & \text{pokud } u_{\mathsf{At}}(a) = 0. \end{cases} \end{aligned}$$

Označme

$$P = \{u_{\mathsf{At}} \mid u(\varphi) = 1\}.$$

Pak $\varphi \mathrel{\text{⫤}\!\models} \psi$, kde

$$\psi = \bigvee_{u_{\mathsf{At}} \in P} \bigwedge_{a \in \mathsf{At}} l(a, u_{\mathsf{At}}).$$

$\square$

Konjunktivní normální forma

Definice 3.40. Řekneme, že formule φ je v *konjunktivní normální formě*, pokud

$$\varphi = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{m_i} \varphi_{i,j}$$

pro nějaké $n \in \mathbb{N}$, nějaká $m_i \in \mathbb{N}$ $(i = 1, \ldots, n)$, a všechny formule $\varphi_{i,j}$ jsou literály.

Tvrzení 3.41. Každá formule $\varphi \in \mathsf{Fm}(\mathsf{At})$ (kde At je konečná) je sémanticky ekvivalentní nějaké formuli ψ v konjunktivní normální formě.

Důkaz. Ať

$$nl : \mathsf{At} \times \mathsf{Bool}^{\mathsf{At}} \to \mathsf{Fm}(\mathsf{At})$$
$$(a, u_{\mathsf{At}}) \mapsto \begin{cases} a & \text{pokud } u_{\mathsf{At}}(a) = 0, \\ \neg a & \text{pokud } u_{\mathsf{At}}(a) = 1. \end{cases}$$

Označme

$$N = \{u_{\mathsf{At}} \mid u(\varphi) = 0\}.$$

Pak $\varphi \models\!\!\!\!\dashv \psi$, kde

$$\psi = \bigwedge_{u_{\mathsf{At}} \in N} \bigvee_{a \in \mathsf{At}} nl(a, u_{\mathsf{At}}).$$

□

Negační normální forma

Definice 3.42. Formule v *negační normální formě* jsou formule vymezené následující BNF (Backusovou-Naurovou formou):

$$\mathsf{N} ::= \mathsf{L} \mid \top \mid \bot \mid (\mathsf{N} \wedge \mathsf{N}) \mid (\mathsf{N} \vee \mathsf{N}) \mid (\mathsf{N} \Rightarrow \mathsf{N}) \mid (\mathsf{N} \Leftrightarrow \mathsf{N}).$$

(Symbol L označuje literál, viz Definici 3.35.)

Tvrzení 3.43. Každá formule $\varphi \in \mathsf{Fm}(\mathsf{At})$ je sémanticky ekvivalentní nějaké formuli v negační normální formě.

Důkaz. Ukážeme algoritmus, který pro zadanou formuli φ sestrojí sémanticky ekvivalentní formuli v negační normální formě (viz Algoritmus 1). Daný algoritmus na výstupu zjevně vrací formuli v negační normální formě, což lze dokázat indukcí. Formule NNF(φ) je sémanticky ekvivalentní formuli φ, což lze dokázat indukcí podle struktury formule φ. □

Algoritmus 1: Rekursivní algoritmus pro převod do negační normální formy

Vstup: Formule $\varphi \in \mathsf{Fm}(\mathsf{At})$.
Výstup: Formule NNF(φ) v NNF splňující $\varphi \mathrel{\text{⫤}\!\models} \mathrm{NNF}(\varphi)$.
switch φ **do**
- **case** $\top\ \bot$, a, $\neg a$ **do return** φ;
- **case** $\neg\top$ **do return** $\bot$;
- **case** $\neg\bot$ **do return** $\top$;
- **case** $\psi \circ \chi$ **do return** $\mathsf{NNF}(\psi) \circ \mathsf{NNF}(\chi)$;
- **case** $\neg\neg\psi$ **do return** $\mathsf{NNF}(\psi)$;
- **case** $\neg(\psi \wedge \chi)$ **do return** $\mathsf{NNF}(\neg\psi) \vee \mathsf{NNF}(\neg\chi)$;
- **case** $\neg(\psi \vee \chi)$ **do return** $\mathsf{NNF}(\neg\psi) \wedge \mathsf{NNF}(\neg\chi)$;
- **case** $\neg(\psi \Rightarrow \chi)$ **do return** $\mathsf{NNF}(\psi) \wedge \mathsf{NNF}(\neg\chi)$;
- **case** $\neg(\psi \Leftrightarrow \chi)$ **do return** $(\mathsf{NNF}(\psi) \wedge \mathsf{NNF}(\neg\chi)) \vee (\mathsf{NNF}(\neg\psi) \wedge \mathsf{NNF}(\chi))$;

K zamyšlení. Promyslete, zda (a případně jak) by k hledání negační normální formy šlo i využít konstrukce disjunktivní či konjunktivní normální formy z předchozích odstavců.

3.3 Úplné systémy logických spojek

Z podsekce 3.2 plyne, že každou booleovskou funkci

$$b : \mathsf{Bool}^{\mathsf{At}} \to \mathsf{Bool}$$

lze *representovat* nějakou formulí $\varphi \in \mathsf{Fm}(\mathsf{At})$, tj. existuje $\varphi \in \mathsf{Fm}(\mathsf{At})$ taková, že

$$[\![\varphi]\!] = b.$$

Odvodili jsme dokonce silnější výsledek: formuli φ lze zvolit tak, aby byla v konjunktivní (či disjunktivní) normální formě. To znamená, že každou booleovskou funkci lze representovat formulí, která ve své konstrukci využívá pouze logických spojek $\wedge$, $\vee$ a $\neg$, případně logických konstant $\top$ a $\bot$.

V případě, že se omezíme na booleovské funkce

$$b : \mathsf{Bool}^{\mathsf{At}} \to \mathsf{Bool}$$

kde At je *neprázdná* množina, lze snadno ukázat, že takové funkce lze dokonce representovat formulemi, které využívají pouze logických spojek $\wedge$, $\vee$ a $\neg$, a nevyužívají logické konstanty $\top$ a $\bot$.

K zamyšlení. Čím je způsobeno, že pro neprázdné množiny At nepotřebujeme logické konstanty $\top$ a $\bot$? Nebo duálně: v čem je problém, pokud uvažujeme prázdnou množinu atomických formulí At, a jak tento problém logické konstanty $\top$ a $\bot$ řeší?

4 Vztah logického a sémantického důsledku

Tvrzení 4.1 (Věta o korektnosti a úplnosti)**.** Ať $S \subseteq \mathsf{Fm}(\mathsf{At})$ a $\varphi \in \mathsf{Fm}(\mathsf{At})$.

$$S \vdash \varphi \text{ platí právě tehdy, když } S \vDash \varphi.$$

Poznámka 4.2. Předchozí tvrzení má dvě části, *korektnost* a *úplnost* přirozené dedukce ve výrokové logice:

Korektnost Pokud $S \vdash \varphi$, pak $S \vDash \varphi$.

Úplnost Pokud $S \vDash \varphi$, pak $S \vdash \varphi$.

Důkaz. Důkaz úplnosti je relativně náročný, bude proveden v případě dostatku času na přednášce. □

Poznámka 4.3.

- Jak zjistit, zda $S \vdash \varphi$? Stačí nalézt důkaz φ z množiny předpokladů S.
- Jak zjistit, zda $S \vDash \varphi$? Pokud $|\mathsf{At}| = n$, je třeba ověřit 2^n pravdivostních ohodnocení (tabulku o 2^n řádcích).
- Jak zjistit, že *neplatí* $S \vdash \varphi$? (To jest, $S \nvdash \varphi$.) Museli bychom dokázat, že *neexistuje žádný důkaz* φ z množiny předpokladů S.
- Jak zjistit, že *neplatí* $S \vDash \varphi$? Stačí nalézt *jediné* pravdivodstní ohodnocení $u : \mathsf{Fm}(\mathsf{At}) \to \mathsf{Bool}$, kde $u(S) = 1$ a $u(\varphi) = 0$.

Reference

[1] M. Huth a M. Ryan, *Logic in Computer Science: Modelling and Reasoning about Systems*, Cambridge University Press, 2004.

[2] J. Velebil, *Velmi jemný úvod do matematické logiky*, 2007. Dostupné online na `https://math.fel.cvut.cz/en/people/velebil/files/y01mlo/logika.pdf`.