

První přednáška

NAIL062 Výroková a predikátová logika

Jakub Bulín (KTIML MFF UK)

Zimní semestr 2023

Cesta k jistému úspěchu u zkoušky²

- Studujte **průběžně (každý týden)**, a průběžně také **testujte své znalosti**: umíte sami **napsat(!)** definici, větu, důkaz?
- Před každou přednáškou alespoň zběžně projděte příslušné sekce v **Zápiscích z přednášky**. Ty obsahují vše, co po vás bude vyžadováno.¹ Snažte se pochopit smysl definic a tvrzení.
- Po každé přednášce si skripta **podrobně přečtěte**. Pokud něčemu nebudete rozumět, využijte konzultačních hodin.
- Snažte se zúčastnit všech přednášek, pokud nemůžete, včas se materiál doučte, případně využijte konzultačních hodin.
- Ujistěte se, že rozumíte nejen myšlenkám, ale umíte pracovat i s **formalizmem**: ten je neoddělitelnou součástí logiky.
- Stejnou pozornost věnujte i přípravě na **cvičení**.

¹Nedoporučujeme učit se primárně podle slidů! Mnoho detailů v nich chybí.

²Podrobnosti o zkoušce včas upřesníme. Základní formát i většina otázek ale pravděpodobně zůstanou stejné, viz loňské **informace o zkouškách**.

Program

- úvod do logiky
- neformální představení výrokové a predikátové logiky
- syntaxe výrokové logiky

Materiály

Zápisky z přednášky, Kapitola 1 a Sekce 2.1 z Kapitoly 2

KAPITOLA 1: ÚVOD DO LOGIKY

Dvě definice:

1. soubor principů, které jsou základem uspořádání prvků nějakého systému (např. počítačového programu, elektronického zařízení, komunikačního protokolu)
2. věda o uvažování prováděném podle striktních pravidel zachovávajících platnost

V informatice obojí: daný systém nejprve *formálně popíšeme*, a poté o něm *formálně uvažujeme* (automaticky!), tj. odvozujeme *platné inference* za použití nějakého *dokazovacího systému*

Filozofie → Matematika → Teoretická informatika →

Aplikovaná informatika

- logic programming
- discrete optimization (SAT solving, scheduling, planning)
- database theory
- verification (software, hardware, protocol)
- automated reasoning and proving
- knowledge-based representation
- artificial intelligence

1.1 Výroková logika

Příklad ze života: Hledání pokladu

Při hledání pokladu jsme narazili na rozcestí dvou chodeb. Víme, že na konci každé chodby je buď poklad, nebo drak, ale ne obojí.

Trpaslík nám řekl, že:

- *“Alespoň jedna z těch dvou chodeb vede k pokladu”,* a že
- *“První chodba vede k drakovi.”*

Je známo, že trpaslíci buď vždy mluví pravdu, nebo vždy lžou. Kterou cestou se máme vydat?

Výroky neformálně

Výrok je tvrzení, kterému lze přiřadit pravdivostní hodnotu:

pravdivý (*True*, 1), nebo **lživý** (*False*, 0)

Prvovýroky (**atomické výroky**, **výrokové proměnné**) zkombinované pomocí logických spojek a závorek do **složených výroků**:

“(Trpaslík lže,) *právě když* (druhá chodba vede k drakovi.)”

¬ “neplatí X”, *negace*

∧ “X a Y”, *konjunkce*

∨ “X nebo Y”, *disjunkce* (není exkluzivní)

→ “pokud X, potom Y”, *implikace* (čistě logická)

↔ “X, právě když Y”, *ekvivalence*

Formalizace ve výrokové logice

Volba množiny prvovýroků: *bity informace popisující daný systém*

$p_1 = \text{"Poklad je v první chodbě."}$

$p_2 = \text{"Poklad je ve druhé chodbě."}$

(Co nejmenší, např. hodnota $t = \text{"Trpaslík mluví pravdu."}$ je jednoznačně určená hodnotami $\mathbb{P} = \{p_1, p_2\}$.)

- *Poklad nebo drak, ale ne obojí:* zakódované do volby \mathbb{P} (přítomnost draka je absence pokladu)
- *"První chodba vede k drakovi."* $\Leftrightarrow \neg p_1$
- *"Alespoň jedna z chodeb vede k pokladu."* $\Leftrightarrow p_1 \vee p_2$
- *Trpaslík buď mluví pravdu, nebo lže:*

$$\varphi = (\neg p_1 \wedge (p_1 \vee p_2)) \vee (\neg(\neg p_1) \wedge \neg(p_1 \vee p_2))$$

Teorie $T = \{\varphi\}$ v **jazyce** $\mathbb{P} = \{p_1, p_2\}$, φ je **axiom** T .

Modely a důsledky

Lze určit, kde je poklad? Je p_1 nebo p_2 **důsledkem** φ resp. T ?

“**Svět**”, ve kterém je např. v první chodbě poklad a ve druhé drak, popíšeme pomocí **pravdivostního ohodnocení** $p_1 = 1, p_2 = 0$, neboli **modelu** $v = (1, 0)$ jazyka \mathbb{P} . Celkem máme 4 “světy” a modely:

$$M_{\mathbb{P}} = \{(0, 0), (0, 1), (1, 0), (1, 1)\}.$$

Je “svět” popsán modelem $v = (1, 0)$ *konzistentní* s tím, co víme, tj. **platí** v modelu v výrok φ resp. teorie T ? Vyhodnotíme podle stromové struktury φ :

$$v(p_1) = 1, v(p_2) = 0, v(\neg p_1) = 0, v(p_1 \vee p_2) = 1, \dots, v(\varphi) = 0$$

Množina **modelů výroku** φ (resp. *modelů teorie* T):

$$M_{\mathbb{P}}(\varphi) = M_{\mathbb{P}}(T) = \{(0, 1)\}.$$

V **každém modelu** teorie T platí výrok p_2 , neboli p_2 je **důsledek** T .

Ověřovat všechny modely je nepraktické, pro $|\mathbb{P}| = n$ máme 2^n modelů, a \mathbb{P} může být i nekonečná.

Dokazovací systém

- **důkaz** výroku ψ z teorie T je formálně definovaný syntaktický objekt, snadno (mechanicky) ověřitelný
- lze hledat algoritmicky čistě na základě struktury ψ a axiomů T (“*syntaxe*”), nemusíme se zabývat modely (“*sémantikou*”).

Klíčové vlastnosti:

- **korektnost**: pokud existuje důkaz ψ z T , potom ψ platí v T
- **úplnost**, pokud ψ platí v T , potom existuje důkaz ψ z T

Ukážeme si **metodu analytického tabla** a **rezoluční metodu**. Obě dokazují *sporem*: předpokládají platnost T a $\neg\psi$, hledají spor.

Metoda analytického tabla

- důkaz je strom olabelovaný předpoklady o platnosti výroků
- v kořeni: **neplatí** dokazovaný výrok ψ (důkaz sporem)
- připojíme platnost axiomů z T
- dále budujeme tablo zjednodušováním výroků ve vrcholech, podle pravidel zaručujících následující **invariant**:

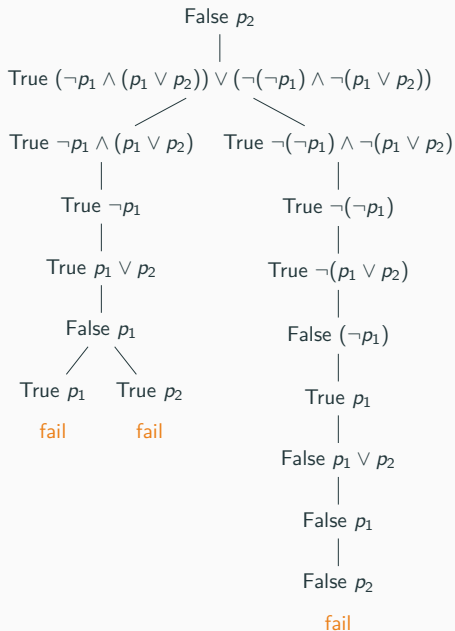
Každý model teorie T , ve kterém neplatí ψ , se musí shodovat s některou z větví tabla.

- například:

True $(\varphi_1 \rightarrow \varphi_2)$ zredukujeme rozvětvením na **False** φ_1 a **True** φ_2 ,
False $(\varphi_1 \rightarrow \varphi_2)$ zredukujeme připojením **True** φ_1 a **False** φ_2 .

- **sporná** větev = předpokládá True i False stejného výroku
- **důkaz** = všechny větve sporné (tj. nemůže existovat model T , ve kterém neplatí ψ)

Příklad tablo důkazu



Konjunktivní normální forma (CNF)

literál $p, \neg p$ **klauzule** disjunkce literálů **CNF** konjunkce klauzulí

každý výrok má **ekvivalentní** CNF ($\psi \sim \psi'$, stejné modely)

např. pro výrok $(\neg p_1 \wedge (p_1 \vee p_2)) \vee (\neg(\neg p_1) \wedge \neg(p_1 \vee p_2))$

nahradíme $\neg(\neg p_1) \sim p_1$ a $\neg(p_1 \vee p_2) \sim (\neg p_1 \wedge \neg p_2)$ (*De Morgan*)

$$(\neg p_1 \wedge (p_1 \vee p_2)) \vee (p_1 \wedge \neg p_1 \wedge \neg p_2)$$

a dále opakovaně použijeme **distributivitu** \vee vůči \wedge :

$$\begin{aligned} &(\neg p_1 \vee p_1) \wedge (\neg p_1 \vee \neg p_1) \wedge (\neg p_1 \vee \neg p_2) \wedge (p_1 \vee p_2 \vee p_1) \wedge \\ &\quad (p_1 \vee p_2 \vee \neg p_1) \wedge (p_1 \vee p_2 \vee \neg p_2) \end{aligned}$$

už je CNF, ještě zjednodušíme: odstraníme duplicitní literály, a klauzule obsahující p_i a zároveň $\neg p_i$ (to jsou **tautologie**)

$$\neg p_1 \wedge (\neg p_1 \vee \neg p_2) \wedge (p_1 \vee p_2)$$

Rezoluční důkaz

Důkaz sporem: převedeme **negaci** dokazovaného do CNF a přidáme **p_2 platí v T** , právě když je následující CNF výrok **nesplnitelný**:

$$\neg p_1 \wedge (\neg p_1 \vee \neg p_2) \wedge (p_1 \vee p_2) \wedge \neg p_2$$

množinový zápis: $S = \{\{\neg p_1\}, \{\neg p_1, \neg p_2\}, \{p_1, p_2\}, \{\neg p_2\}\}$

rezoluční pravidlo: je-li $p \in C_1$ a $\neg p \in C_2$, potom *rezolventa*

$$C = (C_1 \setminus \{p\}) \cup (C_2 \setminus \{\neg p\})$$

platí v každém modelu, ve kterém platí C_1 i C_2

rezoluční zamítnutí S : posloupnost klauzulí, kde každá je buď z S nebo rezolventa předchozích, poslední je prázdná klauzule \square

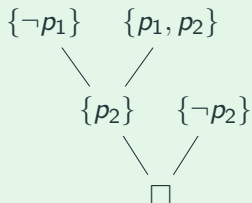
myšlenka: protože \square nemá žádný model, je i S nesplnitelná

Příklad rezolučního důkazu

rezoluční zamítnutí (třetí klauzule je rezolventou 1. a 2., pátá je rezolventou 3. a 4.)

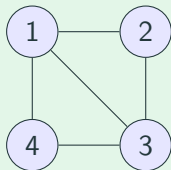
$$\{\neg p_1\}, \{p_1, p_2\}, \{p_2\}, \{\neg p_2\}, \square$$

rezoluční strom (listy klauzule z S , vnitřní vrcholy rezolventy synů)



Příklad: Barvení grafů

Najděte vrcholové obarvení následujícího grafu třemi barvami.



graf: množina vrcholů a množina (libovolně) **orientovaných** hran

$$\mathcal{G} = \langle V; E \rangle = \langle \{1, 2, 3, 4\}; \{(1, 2), (1, 3), (1, 4), (2, 3), (3, 4)\} \rangle$$

jak formalizovat? pro $v \in V$ a $c \in C = \{R, G, B\}$:

$$p_v^c = \text{"vrchol } v \text{ má barvu } c"$$

$$\mathbb{P} = \{p_v^c \mid c \in C, v \in V\} = \{p_1^R, p_1^G, p_1^B, p_2^R, p_2^G, p_2^B, p_3^R, p_3^G, p_3^B, p_4^R, p_4^G, p_4^B\}$$

máme celkem $|\mathbb{M}_{\mathbb{P}}| = 2^{12} = 4096$ **modelů jazyka** (12-dim. vektorů)

Formalizace hranového obarvení

- každý vrchol má nejvýše jednu barvu: $4^4 = 2^8 = 256$ modelů

$$T_1 = \{(\neg p_v^R \vee \neg p_v^G) \wedge (\neg p_v^R \vee \neg p_v^B) \wedge (\neg p_v^G \vee \neg p_v^B) \mid v \in V\}$$

- a každý vrchol má alespoň jednu barvu: $3^4 = 81$ modelů

$$T_2 = T_1 \cup \{p_v^R \vee p_v^G \vee p_v^B \mid v \in V\} = T_1 \cup \left\{ \bigvee_{c \in C} p_v^c \mid v \in V \right\}$$

T_2 je **extenze** teorie T_1 neboť **každý důsledek** T_1 **platí i v** T_2 ,
zde dokonce $M_{\mathbb{P}}(T_2) \subseteq M_{\mathbb{P}}(T_1)$

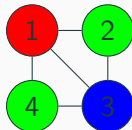
- nakonec přidáme **hranovou podmínku**:

$$T_3 = T_2 \cup \left\{ \bigwedge_{c \in C} (\neg p_u^c \vee \neg p_v^c) \mid (u, v) \in E \right\}$$

Výsledná teorie T_3 je **splnitelná** (má model), právě když je
graf \mathcal{G} 3-obarvitelný.

Všechna obarvení?

T_3 má 6 modelů: $v = (1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0)$ a další získané permutací barev



Obarvení, ve kterých je vrchol 1 modrý a vrchol 2 zelený?

Odpovídají modelům teorie $T_3 \cup \{p_1^B, p_2^G\}$

Důkaz, že vrcholy 2 a 4 musí mít stejnou barvu?

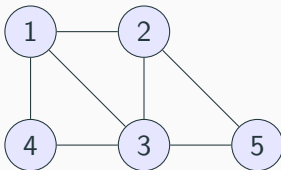
Tablo s kořenem False $(p_2^R \wedge p_4^R) \vee (p_2^G \wedge p_4^G) \vee (p_2^B \wedge p_4^B)$

Nebo **rezolucí**: přidáme **negaci** $(p_2^R \wedge p_4^R) \vee (p_2^G \wedge p_4^G) \vee (p_2^B \wedge p_4^B)$,
vše převedeme do CNF a zamítneme

1.2 Predikátová logika

Nevýhody formalizace ve výrokové logice

Teorie T_3 je poměrně velká, a 'natvrdo' kóduje graf \mathcal{G} .



Obohatit jazyk $\mathbb{P}' = \mathbb{P} \cup \{p_5^R, p_5^G, p_5^B\}$ a vytvořit ještě větší teorii T'_3 přidáním axiomů o vrcholu 5 a hranách $(2, 5), (3, 5)$?

A co vlastnosti obecně platné o všech nebo mnoha grafech?

V **predikátové logice** můžeme mluvit o **vrcholech** grafu pomocí **proměnných** a přirozeně vyjádřit vlastnosti jako:

- “z vrcholu u vede hrana do vrcholu v ”
- “vrchol u je zelený”

Modely už nejsou 0–1 vektory, ale **struktury**, např. naše (orientované) grafy:

$$\mathcal{G} = \langle V^{\mathcal{G}}; E^{\mathcal{G}} \rangle = \langle \{1, 2, 3, 4\}; \{(1, 2), (1, 3), (1, 4), (2, 3), (3, 4)\} \rangle$$

$$\mathcal{G}' = \langle V^{\mathcal{G}'}; E^{\mathcal{G}'} \rangle = \langle \{1, 2, 3, 4, 5\}; \{(1, 2), (1, 3), (1, 4), (2, 3), (3, 4), (2, 5), (3, 5)\} \rangle$$

- množina vrcholů, a binární relace na této množině
- **jazyk** specifikuje kolik **relací** jakých arit má struktura mít, a symboly pro ně
- např. **jazyk grafů** $\mathcal{L} = \langle E \rangle$ (kde E je binární relační symbol)
- \mathcal{G} a \mathcal{G}' jsou **struktury v jazyce** \mathcal{L} (**\mathcal{L} -struktury**)
- můžeme mít také **funkce** a **konstanty**, a symbol $=$ pro **rovnost**

Syntaxe: místo prvovýroků **atomické formule**, např. $E(x, y)$, kde x, y jsou **proměnné** reprezentující vrcholy; stejné logické spojky, ale navíc **kvantifikátory**:

$(\forall x)$ “pro všechny vrcholy x ”

$(\exists y)$ “existuje vrchol y ”

(hrají roli “konjunkce” a “disjunkce” přes všechny prvky)

- “V grafu nejsou smyčky”: $(\forall x)(\neg E(x, x))$

- “Existuje vrchol výstupního stupně 1”:

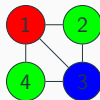
$(\exists x)(\exists y)(E(x, y) \wedge (\forall z)(E(x, z) \rightarrow y = z))$

Sémantika: V daném grafu \mathcal{G} a při **dosazení** vrcholu u za proměnnou x a vrcholu v za proměnnou y **vyhodnotíme** $E(x, y)$ jako **True**, právě když $(u, v) \in E^{\mathcal{G}}$.

Barvení grafů v predikátové logice

Jazyk $\mathcal{L}' = \langle E, R, G, B \rangle$, kde E je binární a R, G, B jsou unární relační symboly ($R(x)$ znamená “vrchol x je červený”)

\mathcal{L}' -struktura: graf s trojicí množin vrcholů



$$\begin{aligned}\mathcal{G}_C &= \langle V^{\mathcal{G}_C}; E^{\mathcal{G}_C}, R^{\mathcal{G}_C}, G^{\mathcal{G}_C}, B^{\mathcal{G}_C} \rangle \\ &= \langle \{1, 2, 3, 4\}; \{(1, 2), (1, 3), (1, 4), (2, 3), (3, 4)\}, \{1\}, \{2, 4\}, \{3\} \rangle\end{aligned}$$

\mathcal{G}_C je **expanze** \mathcal{L} -struktury \mathcal{G} **do jazyka** \mathcal{L}'

Nejvýše jedna barva, alespoň jedna barva, hranová podmínka:

- $(\forall x)((\neg R(x) \vee \neg G(x)) \wedge (\neg R(x) \vee \neg B(x)) \wedge (\neg G(x) \vee \neg B(x)))$
- $(\forall x)(R(x) \vee G(x) \vee B(x))$
- $(\forall x)(\forall y)(E(x, y) \rightarrow ((\neg R(x) \vee \neg R(y)) \wedge (\neg G(x) \vee \neg G(y)) \wedge (\neg B(x) \vee \neg B(y))))$

1.3 Další druhy logických systémů

- Predikátová logika, kde proměnné reprezentují jednotlivé vrcholy, je logika **prvního řádu** (**first-order**, **FO**)
- Logika **druhého řádu** (**second-order**, **SO**): proměnné i pro množiny vrcholů a n -tic vrcholů (tj. relace, funkce)

$$(\exists S)(\forall x)(\forall y)(E(x, y) \rightarrow (S(x) \leftrightarrow \neg S(y)))$$

“Graf je bipartitní.”

- A v logice *třetího řádu* máme i množiny množin (např. v topologii).

Kromě toho lze zobecnit pojem platnosti (pravdy):

- **temporální logiky** (platnost 'vždy', 'někdy v budoucnosti', 'dokud' apod.) – např. v paralelním programování
- **modální logiky** ('je možné', 'je nutné') – v umělé inteligenci, uvažování autonomních agentů o svém okolí
- **fuzzy logiky** ('je 0.35 pravdivé') – v automatických pračkách
- **intuicionistická logika** (povoluje jen konstruktivní důkazy, nemá *zákon vyloučeného třetího*)

1.4 O přednášce

I. Výroková logika (5 přednášek)

- Syntaxe a sémantika
- Problém SAT
- Tablo metoda
- Rezoluční metoda

II. Predikátová logika (6 přednášek)

- Syntaxe a sémantika
- Tablo metoda v predikátové logice
- Rezoluční metoda v predikátové logice
- Aplikace: databáze, Prolog, verifikace

III. Pokročilé partie (2 přednášky)

- Teorie modelů
- Nerozhodnutelnost a neúplnost

ČÁST I – VÝROKOVÁ LOGIKA

KAPITOLA 2: SYNTAXE A SÉMANTIKA VÝROKOVÉ LOGIKY

syntaxe dává pravidla pro tvoření korektních formálních výrazů sestávajících ze symbolů, a pro operace s nimi (*výrok*, *důkaz*, ...)

sémantika popisuje význam syntaktických objektů “v reálném světě” (*model*, ...)

Klíčem k logice je **vztah mezi syntaxí a sémantikou**:

- sémantické objekty studujeme pomocí syntaxe (‘jaké výroky platí v modelu?’)
- syntaktické pomocí sémantiky, např. ekvivalence výroků:
 $\psi \sim \psi'$ právě když $M_{\mathbb{P}}(\psi) = M_{\mathbb{P}}(\psi')$

2.1 Syntaxe výrokové logiky

- určený množinou **prvovýroků** (**výrokových proměnných**, **atomických výroků**) – neprázdná, konečná nebo i *nekonečná*

$$\mathbb{P}_1 = \{p, q, r\}$$

$$\mathbb{P}_2 = \{p_0, p_1, p_2, p_3, \dots\} = \{p_i \mid i \in \mathbb{N}\}$$

(obvykle *spočetná*, *uspořádaná*)

- dále do jazyka patří **logické symboly**:
 - logické spojky $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
 - závorky $(,)$

Výrok

Výrok (**výroková formule**) v jazyce \mathbb{P} je prvek množiny $VF_{\mathbb{P}}$ definované *induktivně*: $VF_{\mathbb{P}}$ je nejmenší množina splňující

- pro každý prvovýrok $p \in \mathbb{P}$ platí $p \in VF_{\mathbb{P}}$,
- pro každý výrok $\varphi \in VF_{\mathbb{P}}$ je $(\neg\varphi)$ také prvek $VF_{\mathbb{P}}$
- pro každé $\varphi, \psi \in VF_{\mathbb{P}}$ jsou $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$, a $(\varphi \leftrightarrow \psi)$ také prvky $VF_{\mathbb{P}}$.

Výroky jsou nutně *konečné* řetězce!

Var(φ): množina všech prvovýroků ve φ (vždy konečná)

podvýrok: podřetězec, který je sám výrok

$\varphi = ((p \vee (\neg q)) \leftrightarrow (r \rightarrow (p \wedge q)))$, $\text{Var}(\varphi) = \{p, q\}$

podvýroky: $p, q, (\neg q), (p \vee (\neg q)), r, (p \wedge q), (r \rightarrow (p \wedge q)), \varphi$

pravda: $\top = (p \vee (\neg p))$, **spor**: $\perp = (p \wedge (\neg p))$

(kde $p \in \mathbb{P}$ je pevně daný)

Konvence zápisu

při *zápisu* výroků můžeme vynechat některé závorky:

$\varphi = ((p \vee (\neg q)) \leftrightarrow (r \rightarrow (p \wedge q)))$ lze zapsat jako $p \vee \neg q \leftrightarrow (r \rightarrow p \wedge q)$

- priorita operátorů: \neg nejvyšší, dále \wedge a \vee , nakonec \rightarrow a \leftrightarrow
- asociativita \wedge a \vee : nápis $p \wedge q \wedge r$ znamená výrok $(p \wedge (q \wedge r))$
- vnější závorky nemusíme psát

Poznámka: v definici jsme mohli místo *infixového* zápisu zvolit *prefixový* (“polskou notaci”): “každý prvovýrok je výrok, jsou-li φ, ψ výroky, jsou výroky také $\neg\varphi$, $\wedge\varphi\psi$, $\vee\varphi\psi$, $\rightarrow\varphi\psi$, a $\leftrightarrow\varphi\psi$ ” nebo i *postfixový*

$\varphi = \leftrightarrow \vee p \neg q \rightarrow r \wedge p q$

$\varphi = p q \neg \vee r p q \wedge \rightarrow \leftrightarrow$

Důležitá je jen **stromová struktura** výroků!

Strom výroku

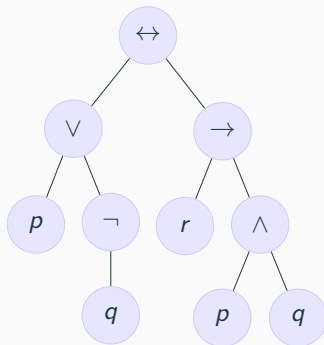
$\text{Tree}(\varphi)$ je zakořeněný uspořádaný strom, definovaný induktivně:

- $\varphi = p \in \mathbb{P}$: jediný vrchol, s labelem p
- $\varphi = (\neg\varphi')$: kořen s labelem \neg , jediný syn je kořen $\text{Tree}(\varphi')$.
- $\varphi = (\varphi' \square \varphi'')$ pro $\square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$: kořen s labelem \square a dvěma syny: levý syn je kořen $\text{Tree}(\varphi')$, pravý $\text{Tree}(\varphi'')$.

$\varphi = ((p \vee (\neg q)) \leftrightarrow (r \rightarrow (p \wedge q)))$

rekonstrukce φ průchodem stromu,
podvýroky odpovídají podstromům

$\text{Tree}(\varphi)$ je jednoznačně určený!



Teorie v jazyce \mathbb{P} je libovolná množina výroků $T \subseteq VF_{\mathbb{P}}$. Výrokům $\varphi \in T$ říkáme také **axiomy**.

$T = \emptyset$ a $T = VF_{\mathbb{P}}$ nad libovolným jazykem,

$T = \{p \wedge q, q \rightarrow (p \vee r)\}$ v jazyce $\mathbb{P} = \{p, q, r\}$

$T = \{p_0\} \cup \{p_i \rightarrow p_{i+1} \mid i \in \mathbb{N}\}$ nad *nekonečným* $\mathbb{P} = \{p_i \mid i \in \mathbb{N}\}$

Poznámka: *Konečnou* teorii by bylo možné (byť ne praktické!) nahradit jediným výrokem: konjunkcí všech axiomů.

Připouštíme ale i *nekonečné teorie*; hodí se např. pro popis systému v (diskrétním) čase $t = 0, 1, 2, \dots$