

# Logika pre informatikov a Úvod do matematickej logiky

Poznámky z prednášok

Ján Kl'uka, Ján Mazák, Jozef Šiška

Letný semester 2024/2025

Posledná aktualizácia: 7. apríla 2025

## Obsah

<b>P1</b>	<b>Úvod. Atomické formuly a štruktúry</b>	<b>5</b>
<b>0</b>	<b>Úvod</b>	<b>5</b>
0.1	O logike . . . . .	5
0.2	O kurzoch LPI a UdML . . . . .	14
<b>1</b>	<b>Atomické formuly a štruktúry</b>	<b>15</b>
1.1	Syntax atomických formúl . . . . .	21
1.2	Štruktúry . . . . .	25
1.3	Sémantika atomických formúl . . . . .	28
1.4	Zhrnutie . . . . .	29

<b>P2</b>	<b>Výrokovologické spojky a ohodnotenia</b>	<b>31</b>
<b>2</b>	<b>Výrokovologické spojky a ohodnotenia</b>	<b>31</b>
2.1	Boolovské spojky . . . . .	32
2.2	Implikácia . . . . .	38
2.3	Ekvivalencia . . . . .	40
2.4	Správnosť a vernosť formalizácie . . . . .	41
2.5	Syntax výrokovologických formúl . . . . .	44
2.6	Sémantika výrokovologických formúl . . . . .	53
2.7	Teórie a ich modely . . . . .	55
2.8	Výrokovologické ohodnotenia . . . . .	56
<b>P3</b>	<b>Výrokovologické vyplývanie, sémantické vlastnosti formúl a ekvivalencia</b>	<b>63</b>
<b>3</b>	<b>Výrokovologické vyplývanie</b>	<b>63</b>
3.1	Výrokovologické teórie a modely . . . . .	64
3.2	Vyplývanie, nezávislosť a nesplniteľnosť . . . . .	65
<b>4</b>	<b>Sémantické vlastnosti a vzťahy formúl</b>	<b>72</b>
4.1	Tautológie, splniteľné, falzifikovateľné a nesplniteľné formuly	72
4.2	Hľadanie ohodnotení . . . . .	77
4.3	Ekvivalencia . . . . .	79
4.4	Vzťah tautológií, vyplývania a ekvivalencie . . . . .	83
4.5	Ekvivalentné úpravy a CNF . . . . .	85
4.6	CNF vs. XOR . . . . .	91
<b>P4</b>	<b>Dôkazy a výrokovologické tablá</b>	<b>93</b>
<b>5</b>	<b>Dôkazy a výrokovologické tablá</b>	<b>93</b>
5.1	Druhy dôkazov . . . . .	96
5.2	Výrokovologické tablá . . . . .	98

<b>P5</b>	<b>Korektnosť a úplnosť výrokovologických tabiel</b>	<b>108</b>
5.3	Korektnosť tabiel . . . . .	108
5.4	Testovanie nesplniteľnosti, splniteľnosti a falzifikovateľnosti	112
5.5	Úplnosť . . . . .	114
5.6	Nové korektné pravidlá . . . . .	115
5.7	Iné dokazovacie systémy . . . . .	123
<b>P6</b>	<b>SAT solvery</b>	<b>129</b>
<b>6</b>	<b>SAT solvery</b>	<b>129</b>
6.1	Problém výrokovologickej splniteľnosti (SAT) . . . . .	129
6.2	Výpočtová zložitosť: teória a prax ( <i>informatívne</i> ) . . . . .	131
6.3	Algoritmy na riešenie problému splniteľnosti . . . . .	133
6.4	Backtracking . . . . .	134
6.5	DPLL a sledované literály . . . . .	138
6.6	CDCL . . . . .	140
6.7	Ďalšie aspekty ( <i>informatívne</i> ) . . . . .	148
6.8	Verifikácia hardvéru ( <i>informatívne</i> ) . . . . .	150
6.9	Kombinatorické problémy ( <i>informatívne</i> ) . . . . .	152
6.10	SMT solvery ( <i>informatívne</i> ) . . . . .	156
<b>P7</b>	<b>Kvantifikátory</b>	<b>159</b>
<b>7</b>	<b>Kvantifikátory</b>	<b>159</b>
7.1	Kvantifikácia . . . . .	159
7.2	Kvantifikátory a premenné . . . . .	160
7.3	Syntax relačnej logiky prvého rádu . . . . .	162
7.4	Sémantika relačnej logiky prvého rádu . . . . .	167
7.5	Neexistencia . . . . .	174
7.6	Aristotelovské formy . . . . .	174
7.7	Zamĺčané a zdanlivo opačné kvantifikátory . . . . .	177
7.8	Nutné a postačujúce podmienky . . . . .	179
7.9	Zložené kvantifikované vlastnosti . . . . .	181
7.10	Konverzačné implikatúry . . . . .	183

7.11	Vyhýbanie sa chybám . . . . .	184
<b>P8</b>	<b>Tablá pre kvantifikátory. Viackvantifikátorové tvrdenia</b>	<b>186</b>
<b>8</b>	<b>Tablá s kvantifikátormi</b>	<b>186</b>
8.1	Logické vlastnosti a vzťahy v logike prvého rádu . . . . .	186
8.2	Dokazovanie s kvantifikátormi . . . . .	190
8.3	Substitúcia a substituovateľnosť . . . . .	198
<b>9</b>	<b>Formalizácia s viacerými kvantifikátormi</b>	<b>200</b>
9.1	Rovnaký kvantifikátor . . . . .	200
9.2	Alternácia kvantifikátorov . . . . .	202
9.3	Postupná formalizácia a parafrázovanie . . . . .	203
9.4	Závislosť od kontextu . . . . .	207
9.5	Dodatky k formalizácii s jedným kvantifikátorom . . . . .	207

## 1. prednáška

# Úvod

## Atomické formuly a štruktúry

---

### 0 Úvod

#### 0.1 O logike

##### Čo je logika

Logika je vedná disciplína, ktorá študuje usudzovanie.

Správne, racionálne usudzovanie je základom vedy a inžinierstva.

Vyžaduje rozoznať

- správne úsudky z predpokladaných princípov a pozorovania
- od chybných úvah a špekulácií.

Správnosť úsudkov, zdá sa, nie je iba vec konvencie a dohody.

Logika skúma, *aké* sú zákonitosti správneho usudzovania a *prečo* sú zákonitosťami.

Historicky sa logika venovala najmä filozofickým hľadiskám, dnes kladieme väčší dôraz na výpočtové aspekty.

##### Ako logika študuje usudzovanie

Logika má dva hlavné predmety záujmu:

**Jazyk** zápis pozorovaní, definície pojmov, formulovanie teórií

*Syntax* pravidlá zápisu tvrdení

*Sémantika* význam tvrdení

**Usudzovanie (inferencia)** odvodzovanie nových *logických dôsledkov* z doterajších poznatkov. (Úzko súvisí s jazykom: čím viac možno v jazyku vyjadriť, tým ťažšie je definovať či algoritmicke rozhodovať logické vyplývanie.)

## Jazyk, poznatky a teórie

*Jazyk* slúži na formulovanie tvrdení, ktoré vyjadrujú poznatky o svete (princípy jeho fungovania aj pozorované fakty).

Súboru poznatkov, ktoré považujeme za pravdivé, hovoríme *teória*.

*Príklad 0.1* (Party time!). Máme troch nových známych — Kim, Jima a Sarah. Organizujeme párty a P0: chceme na ňu pozvať niekoho z nich. Od spoločných kamarátov sme sa ale dozvedeli o ich požiadavkách:

P1: Sarah nepôjde na párty, ak pôjde Kim.

P2: Jim pôjde na párty, len ak pôjde Kim.

P3: Sarah nepôjde bez Jima.

## Možné stavy sveta a modely

Jedna z otázok, ktoré si o teórii o party môžeme položiť, je: „*Môžu noví známi prísť na párty tak, aby boli všetky podmienky splnené? Ak áno, v akých zostavách?*“

Priamočiaro (aj keď práce) to zistíme tak, že:

1. vymenujeme *všetky možné stavy sveta* (účasti nových známych),
2. zistíme, v ktorých sú všetky podmienky splnené.

K	J	S	P0	P1	P2	P3
n	n	n	n			
n	n	p	p	p	p	n
n	p	n	p	p	n	
n	p	p	p	p	n	
p	n	n	p	p	p	p
p	n	p	p	n		
p	p	n	p	p	p	p
p	p	p	p	n		

## Možné stavy sveta a modely

Teória rozdeľuje *možné stavy sveta* (interpretácie) na:

⊢ stavy, v ktorých je pravdivá — *modely* teórie,

⊣ stavy, v ktorých je nepravdivá.

Tvrdenie aj teória môžu mať viacero modelov, ale aj žiaden.

*Príklad 0.2.* Modelmi teórie P0, P1, P2, P3 sú dve situácie: keď Kim príde na párty a ostatní noví známi nie, a keď Kim a Jim prídu na párty a Sarah nie.

K	J	S	P0	P1	P2	P3	
n	n	n	n				≠ P0, P1, P2, P3
n	n	p	p	p	p	n	≠ P0, P1, P2, P3
n	p	n	p	p	n		≠ P0, P1, P2, P3
n	p	p	p	p	n		≠ P0, P1, P2, P3
p	n	n	p	p	p	p	≠ P0, P1, P2, P3
p	n	p	p	n			≠ P0, P1, P2, P3
p	p	n	p	p	p	p	≠ P0, P1, P2, P3
p	p	p	p	n			≠ P0, P1, P2, P3

## Logické dôsledky

Často je zaujímavá iná otázka o teórii — musí byť nejaké tvrdenie pravdivé *vždy*, keď je pravdivá teória?

V našom prípade: Kto *musí* a kto *nesmie* prísť na párty, aby boli podmienky P0, ..., P3 splnené?

K	J	S	P0	P1	P2	P3	
n	n	n	n				≠ P0, P1, P2, P3
n	n	p	p	p	p	n	≠ P0, P1, P2, P3
n	p	n	p	p	n		≠ P0, P1, P2, P3
n	p	p	p	p	n		≠ P0, P1, P2, P3
p	n	n	p	p	p	p	≠ P0, P1, P2, P3
p	n	p	p	n			≠ P0, P1, P2, P3
p	p	n	p	p	p	p	≠ P0, P1, P2, P3
p	p	p	p	n			≠ P0, P1, P2, P3

## Logické dôsledky

Logickými dôsledkami teórie sú tvrdenia, ktoré sú pravdivé vo *všetkých* modeloch teórie.

*Príklad 0.3.* Logickými dôsledkami teórie P0, P1, P2, P3 sú napríklad:

- Kim príde na párty.
- Sarah nepríde na párty.

Logických dôsledkov je nekonečne veľa, môžu nimi byť ľubovoľne zložité tvrdenia:

- Na party príde Kim alebo Jim.
- Ak príde Sarah, tak príde aj Jim.
- Ak príde Jim, tak nepríde Sarah.

⋮

## Logické usudzovanie

Preskúmať všetky stavy sveta je často nepraktické až nemožné.

Logické dôsledky ale môžeme *odvodzovať usudzovaním (inferovať)*.

Pri odvodení vychádzame z *premís* (predpokladov) a postupnosťou *správnych úsudkov* dospievame k *záverom*.

*Príklad 0.4.* Vieme, že ak na párty pôjde Kim, tak nepôjde Sarah (P1), a že ak pôjde Jim, tak pôjde Kim (P2).

1. Predpokladajme, že na párty pôjde Jim.
2. Podľa 1. a P2 pôjde aj Kim.
3. Podľa 2. a P1 nepôjde Sarah.

Teda podľa uvedenej úvahy: Ak na párty pôjde Jim, tak nepôjde Sarah.

## Dedukcia

Úsudok je správny (*korektný*) vtedy, keď *vždy*, keď sú pravdivé jeho premisy, je pravdivý aj jeho záver.

Ak sú všetky úsudky v odvodení správne, záver je *logickým dôsledkom* premís a odvodenie je jeho *dôkazom* z premís.

*Dedukcia* je usudzovanie, pri ktorom sa používajú iba správne úsudky.

Logika študuje dedukciu, ale aj niektoré nededuktívne úsudky, ktoré sú *vo všeobecnosti* nesprávne, ale sú správne v *špeciálnych* prípadoch alebo sú *užitočné*:

- indukcia — zovšeobecnenie;
- abdukcia — odvodzovanie možných príčin z následkov;
- usudzovanie na základe analógie (podobnosti).



## Kontrapríklady

Ak úsudok nie je správny, existuje *kontrapríklad* — stav sveta, v ktorom sú *predpoklady pravdivé*, ale *záver je nepravdivý*.

*Príklad 0.5.* Nesprávny úsudok: Ak platia tvrdenia teórie o party, na party príde Jim.

Kontrapríklad: Stav, kedy príde Kim, nepríde Jim, nepríde Sarah.

Teória je pravdivá, výrok „na party príde Jim“ nie je pravdivý.

K	J	S	
n	n	n	⊭ P0, P1, P2, P3
n	n	p	⊭ P0, P1, P2, P3
n	p	n	⊭ P0, P1, P2, P3
n	p	p	⊭ P0, P1, P2, P3
p	n	n	⊭ P0, P1, P2, P3
p	n	p	⊭ P0, P1, P2, P3
p	p	n	⊭ P0, P1, P2, P3
p	p	p	⊭ P0, P1, P2, P3

## Matematická logika

### Matematická logika

- modeluje jazyk, jeho sémantiku a usudzovanie ako matematické objekty (množiny, postupnosti, zobrazenia, stromy);
- rieši logické problémy matematickými metódami.

Rozvinula sa koncom 19. a v prvej polovici 20. storočia hlavne vďaka *Hilbertovmu programu* — snahe vybudovať základy matematiky bez sporov a paradoxov, mechanizovať overovanie dôkazov alebo priamo hľadanie matematických viet.

## Matematická logika a informatika

Informatika sa vyvinula z matematickej logiky (J. von Neumann, A. Turing, A. Church, ...)

Väčšina *programovacích jazykov* obsahuje logické prvky:

- `all(x > m for x in arr)`,

fragmenty niektorých sú priamo preložiteľné na logické formuly:

- `SELECT t1.x FROM t1 JOIN t2 ON t1.y = t2.y WHERE t1.y > 25,`

niektoré (Prolog, Datalog) sú podmnožinou logických jazykov.

Metódami logiky sa dá *presne špecifikovať*, čo má program robiť, *popísať*, čo robí, a *dokázať*, že robí to, čo bolo špecifikované.

## Matematická logika a informatika

Veľa otázok v logike je *algoritmických*:

- Možno usudzovanie pre danú triedu jazykov automatizovať?
- Dá sa nájsť dôkaz pre tvrdenia s takouto štruktúrou dostatočne rýchlym algoritmom?

*Výpočtová logika* hľadá algoritmické riešenia problémov pre rôzne triedy logických jazykov. Aplikovateľné na iné ťažké problémy (grafové, plánovacie, vysvetľovanie, ...) vyjadriteľné v príslušnej triede.

Logika umožňuje hľadať všeobecné odpovede.

- Ak možno vlastnosť grafu popísať *prvorádovou formulou s najviac dvomi kvantifikátormi* a zároveň ..., existuje pomerne rýchly algoritmus, ktorý rozhodne, či daný graf túto vlastnosť má.

## Matematická logika a informatika

*Automatizované dokazovače*: napr. v r. 1996 počítač dokázal Robbins Conjecture, ktorá odolávala ľudskej snahe 60 rokov.

Donedávna malo automatizované dokazovanie nepresvedčivé výsledky a niektoré oblasti výskumu boli relatívne mŕtve, napr. expertné systémy.

S novými modelmi umelej inteligencie však oživa, napr. AlphaProof rieši 84% úloh z IMO.

## Formálne jazyky a formalizácia

Matematická logika nepracuje s prirodzeným jazykom, ale s jeho zjednodušenými modelmi — *formálnymi jazykmi*.

- Presne definovaná, zjednodušená syntax a sémantika.
- Obchádzajú problémy prirodzeného jazyka:

viacznačnosť slov, nejednoznačné syntaktické vzťahy, zložitá syntaktická analýza, výnimky, obraty s ustáleným významom, ...

- Niekoľko formálnych jazykov už poznáte: aritmetika, jazyky fyzikálnych a chemických vzorcov, programovacie jazyky, ...

Problémy z iných oblastí opísané v prirodzenom jazyku musíme najprv *sformalizovať*, a potom naň môžeme použiť aparát mat. logiky.

Formalizácia vyžaduje cvik — trocha veda, trocha umenie.

## Ťažkosti s prirodzeným jazykom

*Prirodzený jazyk* je problematický:

- Viacznačné slová: Milo *je* v posluchárni A.
- Viacznačné tvrdenia: Chlieb sa predáva v potravinách. Videl som dievča v sále *s ďalekohľadom*.
- Ťažko syntakticky analyzovateľné tvrdenia:

Vlastníci bytov a nebytových priestorov v dome prijímajú rozhodnutia na schôdzi vlastníkov dvojtreťinovou väčšinou hlasov všetkých vlastníkov bytov a nebytových priestorov v dome, ak hlasujú o zmluve o úvere a o každom dodatku k nej, o zmluve o zabezpečení úveru a o každom dodatku k nej, o zmluve o nájme a kúpe veci, ktorú vlastníci bytov a nebytových priestorov v dome užívajú s právom jej kúpy po uplynutí dojednaného času užívania a o každom dodatku k nej, o zmluve o vstavbe alebo nadstavbe a o každom dodatku k nim, o zmene účelu užívania spoločných častí domu a spoločných zariadení domu a o zmene formy výkonu správy; ...

— Zákon č. 182/1993 Z. z. SR v znení neskorších predpisov

- Výnimky a obraty so špeciálnym ustáleným významom: *Nikto nie je dokonalý*.

## Formalizácia poznatkov

S formalizáciou ste sa už stretli — napríklad pri riešení slovných úloh:

Karol je trikrát starší ako Mária.

Súčet Karolovho a Máriinho veku je 12 rokov.

Koľko rokov majú Karol a Mária?

$$k = 3 \cdot m$$

$$k + m = 12$$

Stretli ste sa už aj s formálnym jazykom výrokovej logiky.

*Príklad 0.6.* Sformalizujme náš pártý príklad:

P0: Nieкто z trojice Kim, Jim, Sarah pôjde na párty.  $p(K) \vee p(J) \vee p(S)$

P1: Sarah nepôjde na párty, ak pôjde Kim.  $p(K) \rightarrow \neg p(S)$

P2: Jim pôjde na párty, len ak pôjde Kim.  $p(J) \rightarrow p(K)$

P3: Sarah nepôjde bez Jima.  $\neg p(J) \rightarrow \neg p(S)$

Všimnite si, koľko vetných konštrukcií v slovenčine zodpovedá jednej formálnej spojke  $\rightarrow$ .

### Logika prvého rádu

*Jazyk logiky prvého rádu* (FOL) je jeden zo základných formálnych jazykov, ktorými sa logika zaoberá.

Do dnešnej podoby sa vyvinul koncom 19. a v prvej polovici 20. storočia — G. Frege, G. Peano, C. S. Peirce.

Výrokové spojky + kvantifikátory  $\forall$  a  $\exists$ .

Dá sa v ňom vyjadriť veľa zaujímavých tvrdení, bežne sa používa v matematike.

$$\forall \varepsilon > 0 \exists \delta > 0 \dots$$

### Kalkuly — formalizácia usudzovania

Pre mnohé logické jazyky sú známe *kalkuly* — množiny usudzovacích pravidiel, ktoré sú

**korektné** — odvodzujú iba logické dôsledky,

**úplné** — umožňujú odvodiť všetky logické dôsledky.

Kalkuly sú bežné v matematike

- kalkul elementárnej aritmetiky: na počítanie s číslami, zlomkami,
- kalkul lineárnej algebry: riešenie lineárnych rovníc,
- kalkul matematickej analýzy: derivovanie, integrovanie, riešenie diferenciálnych rovníc

⋮

Sú korektné, ale nie vždy úplné.

Poznáte už aj jeden logický kalkul – ekvivalentné úpravy.

### Symbolické vs. aproximačné výpočty

Symbolický výpočet:

$$\begin{aligned}x^2 &= 2 \\(x + \sqrt{2})(x - \sqrt{2}) &= 0 \\x &= \pm\sqrt{2}\end{aligned}$$

Symbols majú jasný význam, výpočet pozostáva z overiteľných krokov, ktoré samé osebe „dávajú zmysel“.

Aproximačný výpočet:

$$\begin{aligned}x^2 &= 2 \\x &\in (1, 2) \\x &\in (1.4, 1.5) \\&\dots \\x &\approx 1.4142\end{aligned}$$

Kroky výpočtu nenesú samé osebe zmysel, sú to len aritmetické operácie, výsledok je nespoľahlivý.

### Symbolické vs. aproximačné výpočty

Symbolické:

- úprava výrazov
- derivovanie elem. f.
- matematické dôkazy
- expertné systémy (kľúč na určovanie druhu húb)

Aproximačné / data-driven:

- numerická optimalizácia
- strojové učenie
- neurónové siete
- LLM (ChatGPT)

### Symbolické vs. aproximačné výpočty

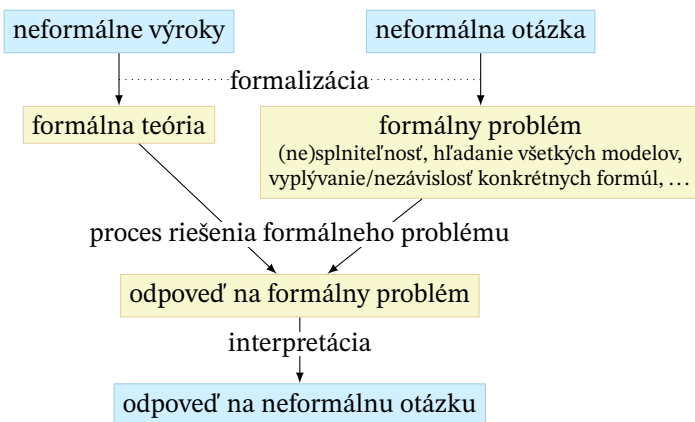
Nevýhodou výpočtov založených na dátach je chýbajúca kontrola nad smerovaním výpočtu a nemožnosť pochopenia/overenia.

Napr. ChatGPT generuje text, ktorý je „pravdepodobný“ (vzhľadom na texty v tréningových vstupoch). Nevie merať ani overovať správnosť. Na začiatku nezvládal ani sčítanie jednociferných čísel; užitočnosť a spoľahlivosť

LLM výrazne stúpne, ak majú prístup ku kalkulačke, ktorá vie robiť symbolickú aritmetiku.

V kontraste s tým symbolické kalkuly garantujú správnosť. Ukážeme si dva (tablá a rezolvenciu), existuje mnoho ďalších (napr. AlphaProof používa Lean).

### Schéma riešenia problémov pomocou logiky



## 0.2 O kurzoch LPI a UdML

### Prístup k logike na tomto predmete

Stredoškolský prístup príliš *neoddeľuje jazyk výrokov od jeho významu* a vlastne ani jednu stránku *redefinuje jasne*.

Prevedieme vás základmi matematickej a výpočtovej logiky pre (postupne čoraz zložitejšie) fragmenty jazykov logiky prvého rádu.

Teoretická časť:

- Matematické definície logických pojmov (výrok, model, logický dôsledok, dôkaz, ...)
- Dôkazy ich vlastností

Praktická časť

- *Dátové štruktúry* na reprezentáciu logických objektov

- *Algoritmické riešenie logických problémov*
- *Formalizácia rôznych problémov v logických jazykoch a ich riešenie nástrojmi na riešenie logických problémov*

### Organizácia kurzu — rozvrh, kontakty, pravidlá

Organizácia — rozvrh, kontakty a pravidlá absolvovania — je popísaná na oficiálnych webových stránkach predmetov:

1-AIN-412 [https://dai.fmph.uniba.sk/w/Course:Logic\\_for\\_CS](https://dai.fmph.uniba.sk/w/Course:Logic_for_CS)



1-INF-210 <http://www.dcs.fmph.uniba.sk/~mazak/vyucba/udml/>



## 1 Atomické formuly a štruktúry

### Problémy s výrokmi

Ukážeme si, prečo je formalizácia pomocou výrokov nedostatočná.

Čo je *výrok*? Oznamovacia veta,

- ktorej sa dá priradiť pravdivostná hodnota.
- pre ktorú má zmysel otázka na jej platnosť, správnosť, pravdivosť.

### Problémy s výrokmi

Nech  $x$  je kladné reálne číslo. Potom  $x^2 > 0$ . [5mm]

Počet hviezd je nepárny. [5mm]

Zajtra vznikne jadro hélia. [5mm]

Odtiaľ až navždy každú sekundu vznikne jadro hélia.

## Problémy s výrokmi

Postupnosť 0, 1, 2, 3, 4, 5, 6, ... je rastúca.[4mm]

- Je to výrok?
- Ako vieme, ako bude postupnosť pokračovať?
- Aký je význam troch bodiek ako symbolu? Môže byť súčasťou výroku popis algoritmu?
- Môže byť výrok nekonečne dlhý?

## Problémy s výrokmi

Bu bayonot emas.[4mm]

- Znamená „toto nie je výrok“ v uzbečtine. Aký jazyk je prípustný?
- Čo ak má v rôznych jazykoch ten istý reťazec rôzny význam?
- Môže výrok hovoriť o sebe?

Môže viesť k paradoxom: Zoberme množinu  $X$  všetkých množín, ktoré neobsahujú samé seba. Je veta „ $X$  patrí do  $X$ “ výrok? Nemôže to byť ani pravda, ani nepravda, pritom to vyzerá ako neškodné matematické tvrdenie.

## Problémy s výrokmi

Ako navrhnúť jazyk logiky?

- Bez akýchkoľvek odkazov na pravdivosť (ale zároveň aby pravdivosť bolo možné bez paradoxov neskôr definovať).
- Presný a jednoznačný: vieme pomocou jednoduchých pravidiel rozhodnúť, či reťazec je tvrdením v jazyku alebo nie.
- Logická štruktúra (spojky, kvantifikátory) musí byť oddelená od popisovaného sveta.



## Jazyky logiky prvého rádu

Logika prvého rádu je trieda (rodina) formálnych jazykov.

Zdieľajú:

- časti abecedy — *logické symboly* (spojky, kvantifikátory)
- pravidlá tvorby *formúl* (slov)

Líšia sa v *mimologických symboloch* — časť abecedy, pomocou ktorej sa tvoria najjednoduchšie — *atomické formuly* (*atómy*).

## Jazyk logiky: príklad

Každý človek umrie.	$\forall x(C(x) \rightarrow U(x))$
Sokrates je človek.	$C(s)$
Sokrates umrie.	$U(s)$

Konštanty: s Predikáty: C, U

## Jazyk logiky: príklad

$\forall x \in \mathbb{R} : x > 0 \rightarrow x \text{ je celé číslo}$

$\forall x(\in(x, \mathbb{R}) \rightarrow (>(x, 0) \rightarrow \in(x, \mathbb{Z})))$

Konštanty: 0,  $\mathbb{R}$ ,  $\mathbb{Z}$  Predikáty:  $>$ ,  $\in$

Konštanty aj predikáty sú úplne iné, ale logické spojky, kvantifikátory a zátvorkovanie majú uvedené dva jazyky spoločné.

## Atómy

Atómy zodpovedajú jednoduchým výrokom — nemajú žiadnu vnútornú logickú štruktúru.

Sokrates je človek.	$C(s)$
$7 > 0$	$>(7, 0)$
Juraj má psa Rexa.	$ma\_psa(Juraj, Rexo)$

## Atomické formuly a výroky v prirodzenom jazyku

Atomické formuly logiky prvého rádu zodpovedajú *pozitívnym jednoduchým vetám* o vlastnostiach, stavoch, vzťahoch a rovnosti *jednotlivých pomenovaných* objektov.

Príklady 1.1.

- |                                      |                                       |
|--------------------------------------|---------------------------------------|
| ✓ Milo beží.                         | ✗ Jarka nie je doma.                  |
| ✓ Jarka vidí Mila.                   | ✗ Nieкто je doma.                     |
| ✗ Milo beží, ale Jarka ho nevidí.    | ✓ Súčet 2 a 2 je 3.                   |
| ✗ Jarka vidí všetkých.               | ✓ Prezidentkou SR je Zuzana Čáputová. |
| ✓ Jarka dala Milovi Bobyho v piatok. |                                       |

Atomické formuly sa skladajú z *individuových konštánt* a *predikátových symbolov*.

## Individuové konštanty

*Individuové konštanty* sú symboly jazyka logiky prvého rádu, ktoré pomenúvajú jednotlivé, pevne zvolené objekty.

Zodpovedajú *približne* vlastným menám, jednoznačným pomenovaniám, niekedy zámenám; konštantám v matematike a programovacích jazykoch.

Príklady 1.2. Jarka, 2, Zuzana\_Čáputová, sobota,  $\pi$ , ...

Individuová konštantá:

- vždy pomenúva skutočný, existujúci objekt (na rozdiel od vlastného mena *Yeti*);

- nikdy nepomenúva viac objektov (na rozdiel od vlastného mena *Jarka*).
- Objekt z domény, ktorú chceme prvorádovým jazykom opísať,
- *môže byť* pomenovaný aj *viacerými* individuovými konštantami (napr. Prezidentka\_SR a Zuzana\_Čaputová);
  - *nemusi* mať žiadne meno.

## Predikátové symboly a arita

*Predikátové symboly* sú symboly jazyka logiky prvého rádu, ktoré označujú vlastnosti alebo vzťahy.

Zodpovedajú

- prísudkom v slovenských vetách,
- množinám alebo reláciám v matematike,
- identifikátorom funkcií s boolovskou návratovou hodnotou.

Predikátový symbol má pevne určený počet argumentov — *aritu*.

*Vždy* musí mať práve toľko argumentov, aká je jeho arita.

Úloha argumentu v predikáte je daná jeho poradím (podobne ako pozíčné argumenty funkcií/metód v prog. jazykoch).

*Dohoda 1.3.* Aritu budeme *niekedy* písať ako horný index symbolu. Napríklad beží<sup>1</sup>, vidí<sup>2</sup>, dal<sup>4</sup>, <<sup>2</sup>.

## Zamýšľaný význam predikátových symbolov

*Unárny* predikátový symbol (teda s aritou 1) zvyčajne označuje *vlastnosť*, druh, rolu, stav.

*Príklady 1.4.*

$\text{pes}(x)$	$x$ je pes
$\text{čierne}(x)$	$x$ je čierne
$\text{beží}(x)$	$x$ beží

*Binárny, ternárny, ...* predikátový symbol (s aritou 2, 3, ...) zvyčajne označuje *vzťah* svojich argumentov.

*Príklady 1.5.*

$\text{vidí}(x, y)$	$x$ vidí $y$
$\text{dal}(x, y, z, t)$	$x$ dal(a/o) objektu $y$ objekt $z$ v čase $t$

## Kategorickosť významu predikátových symbolov

V bežnom jazyku často nie je celkom jasné, či objekt má alebo nemá nejakú vlastnosť — kedy je niekto *mladý*?

Predikátové symboly predstavujú *kategorické* vlastnosti/vzťahy — pre každý objekt sa dá *jednoznačne rozhodnúť*, či má alebo nemá túto vlastnosť/vzťah s iným objektom či inými objektmi.

Význam predikátového symbolu preto často zodpovedá rovnakému slovenskému predikátu iba približne.

*Príklad 1.6.* Predikát mladší<sup>2</sup> môže označovať vzťah „*x* je mladší ako *y*“ presne.

Predikát mladý<sup>1</sup> zodpovedá vlastnosti „*x* je mladý“ iba približne.

Nekategorickými vlastnosťami sa zaoberajú *fuzzy* logiky. Predikáty v nich zachytávajú význam týchto vlastností presnejšie.

## Atomické formuly

*Atomické formuly* majú tvar

$$\text{predikát}(\text{argument}_1, \text{argument}_2, \dots, \text{argument}_k),$$

alebo

$$\text{argument}_1 \doteq \text{argument}_2,$$

pričom *k* je arita predikátu, a  $\text{argument}_1, \dots, \text{argument}_k$  sú (nateraz) individuové konštanty.

Atomická formula zodpovedá (jednoduchému) *výroku* v slovenčine, t.j. tvrdeniu, ktorého *pravdivostná hodnota* (pravda alebo nepravda) sa dá jednoznačne určiť, lebo predikát označuje kategorickú vlastnosť/vzťah a individuové konštanty jednoznačne označujú objekty.

## Formalizácia jednoduchých výrokov

*Formalizácia* je preklad výrokov z prirodzeného jazyka do formálneho logického jazyka.

*Nie je to jednoznačný proces.*

V spojení s *návrhom vlastného jazyka* (konštant a predikátov) je typicky *iteratívna*.

- Postupne zisťujeme, aké predikáty a konštanty potrebujeme, upravujeme predchádzajúce formalizácie.
- Zanedbávame nepodstatné detaily.
- Doterajší jazyk sa snažíme využiť čo najlepšie.

### Návrh jazyka popri formalizácii

Príklad 1.7.  $A_1$ : Jarka dala Milovi Bobbyho.

↪ ~~d(Jarka)~~ ~~dalBobyho(Jarka, Milo)~~ dal(Jarka, Milo, Boby)

$A_2$ : Evka dostala Bobbyho od Mila.

↪ ~~dalBobyho(Milo, Evka)~~ dal(Milo, Evka, Boby)

$A_3$ : Evka dala Jarke Čilku.

↪ ~~dalČilku(Evka, Jarka)~~ dal(Evka, Jarka, Čilka)

$A_4$ : Boby je pes.

↪ pes(Boby)

### Návrh jazyka pri formalizácii

Minimalizujeme počet predikátov, uprednostňujeme flexibilnejšie, viacúčelovejšie ( $\text{dal}^3$  pred  $\text{dalBobyho}^2$  a  $\text{dalČilku}^2$ ).

Dosiahneme

- expresívnejší jazyk (vyjadrí viac menším počtom prostriedkov),
- zrejmejšie logické vzťahy výrokov.

## 1.1 Syntax atomických formúl

### Presné definície

Cieľom logiky je uvažovať o jazyku, výrokoch, vyplývaní, dôkazoch.

Výpočtová logika sa snaží automaticky riešiť konkrétne problémy vyjadrené v logických jazykoch.

Spolahlivé a overiteľné úvahy a výpočty vyžadujú *presnú* dohodu na tom, o čom hovoríme — *definíciu* logických pojmov (jazyk, výrok, pravdivosť, ...).

Pojmy (napr. *atomická formula*) môžeme zadať napríklad

- *matematicky* ako množiny,  $n$ -tice, relácie, funkcie, postupnosti, ...;
- *informaticky* tým, že ich *naprogramujeme*, napr. zadefinujeme triedu `AtomickaFormula` v Pythone.

Matematický jazyk je univerzálnejší ako programovací — abstraktnejší, menej nie až tak podstatných detailov.

### Syntax atomických formúl logiky prvého rádu

Najprv sa musíme dohodnúť na tom, aká je *syntax* atomických formúl logiky prvého rádu:

- z čoho sa skladajú,
- čím vlastne sú,
- akú majú štruktúru.

### Symbols jazyka atomických formúl logiky prvého rádu

Z čoho sa skladajú atomické formuly?

**Definícia 1.8.** *Symbolmi jazyka  $\mathcal{L}$  atomických formúl logiky prvého rádu sú mimologické, logické a pomocné symboly, pričom:*

*Mimologickými symbolmi sú*

- *individuové konštanty* z nejakej neprázdnej spočítateľnej množiny  $\mathcal{C}_{\mathcal{L}}$
- a *predikátové symboly* z nejakej spočítateľnej množiny  $\mathcal{P}_{\mathcal{L}}$ .

Jediným *logickým symbolom* je  $\doteq$  (symbol rovnosti).

*Pomocnými symbolmi* sú  $(, )$  a  $,$  (ľavá, pravá zátvorka a čiarka).

Množiny  $\mathcal{C}_{\mathcal{L}}$  a  $\mathcal{P}_{\mathcal{L}}$  sú disjunktné. Pomocné symboly sa nevyskytujú v symboloch z  $\mathcal{C}_{\mathcal{L}}$  ani  $\mathcal{P}_{\mathcal{L}}$ . Každému symbolu  $P \in \mathcal{P}_{\mathcal{L}}$  je priradená *arita*  $\text{ar}_{\mathcal{L}}(P) \in \mathbb{N}^+$ .

## Abeceda jazyka atomických formúl logiky prvého rádu

Na Úvode do teoretickej informatiky/Formálnych jazykoch a automatoch by ste povedali, že *abecedou* jazyka  $\mathcal{L}$  atomických formúl logiky prvého rádu je  $\Sigma_{\mathcal{L}} = \mathcal{C}_{\mathcal{L}} \cup \mathcal{P}_{\mathcal{L}} \cup \{=, (, ), ,\}$ .

V logike sa väčšinou pojem *abeceda* nepoužíva, pretože potrebujeme rozlišovať rôzne druhy symbolov.

Namiesto *abeceda jazyka*  $\mathcal{L}$  hovoríme *množina všetkých symbolov jazyka*  $\mathcal{L}$  alebo len *symboly jazyka*  $\mathcal{L}$ .

Na zápise množiny  $\Sigma_{\mathcal{L}}$  však ľahko vidíme, čím sa rôzne jazyky atomických formúl logiky prvého rádu od seba líšia a čo majú spoločné.

## Príklady symbolov jazykov atomických formúl logiky prvého rádu

*Príklad 1.9.* Príklad o deťoch a zvieratkách sme sformalizovali v jazyku  $\mathcal{L}_{dz}$ , v ktorom

$$\begin{aligned}\mathcal{C}_{\mathcal{L}_{dz}} &= \{\text{Boby, Cilka, Evka, Jarka, Milo}\}, \\ \mathcal{P}_{\mathcal{L}_{dz}} &= \{\text{dal, pes}\}, \quad \text{ar}_{\mathcal{L}_{dz}}(\text{dal}) = 3, \quad \text{ar}_{\mathcal{L}_{dz}}(\text{pes}) = 1.\end{aligned}$$

*Príklad 1.10.* Príklad o návštevníkoch party by sme mohli sformalizovať v jazyku  $\mathcal{L}_{party}$ , kde

$$\begin{aligned}\mathcal{C}_{\mathcal{L}_{party}} &= \{\text{Kim, Jim, Sarah}\}, \\ \mathcal{P}_{\mathcal{L}_{party}} &= \{\text{príde}\}, \quad \text{ar}_{\mathcal{L}_{party}}(\text{príde}) = 1.\end{aligned}$$

## Označenia symbolov

Keď budeme hovoriť o ľubovoľnom jazyku  $\mathcal{L}$ , často budeme potrebovať nejak označiť niektoré jeho konštanty alebo predikáty, aj keď nebudeme vedieť, aké konkrétne symboly to sú.

Na označenie symbolov použijeme *meta premenné*: premenné v (matematickej) slovenčine, pomocou ktorých budeme hovoriť o (po grécky *meta*) týchto symboloch.

*Dohoda 1.11.* Individuové konštanty budeme spravidla označovať meta premennými  $a, b, c, d$  s prípadnými dolnými indexmi.

Predikátové symboly budeme spravidla označovať meta premennými  $P, Q, R$  s prípadnými dolnými indexmi.

## Atomické formuly jazyka

Čo sú atomické formuly?

**Definícia 1.12.** Nech  $\mathcal{L}$  je jazyk atomických formúl logiky prvého rádu.

*Rovnostný atóm* jazyka  $\mathcal{L}$  je každá postupnosť symbolov  $c_1 \doteq c_2$ , kde  $c_1$  a  $c_2$  sú individuové konštanty z  $\mathcal{C}_{\mathcal{L}}$ .

*Predikátový atóm* jazyka  $\mathcal{L}$  je každá postupnosť symbolov  $P(c_1, \dots, c_n)$ , kde  $P$  je predikátový symbol z  $\mathcal{P}_{\mathcal{L}}$  s aritou  $n$  a  $c_1, \dots, c_n$  sú individuové konštanty z  $\mathcal{C}_{\mathcal{L}}$ .

*Atomickými formulami* (skrátene *atómami*) jazyka  $\mathcal{L}$  súhrnne nazývame všetky rovnostné a predikátové atómy jazyka  $\mathcal{L}$ .

Množinu všetkých atómov jazyka  $\mathcal{L}$  označujeme  $\mathcal{A}_{\mathcal{L}}$ .

## Slová jazyka atomických formúl logiky prvého rádu

Na UTI/FoJa by ste povedali, že jazyk  $\mathcal{L}$  atomických formúl logiky prvého rádu nad abecedou  $\Sigma_{\mathcal{L}} = \mathcal{C}_{\mathcal{L}} \cup \mathcal{P}_{\mathcal{L}} \cup \{\doteq, (, ), ,\}$  je množina slov

$$\{c_1 \doteq c_2 \mid c_1 \in \mathcal{C}_{\mathcal{L}}, c_2 \in \mathcal{C}_{\mathcal{L}}\} \\ \cup \{P(c_1, \dots, c_n) \mid P \in \mathcal{P}_{\mathcal{L}}, \text{ar}_{\mathcal{L}}(P) = n, c_1 \in \mathcal{C}_{\mathcal{L}}, \dots, c_n \in \mathcal{C}_{\mathcal{L}}\}.$$

V logike sa jazyk takto nedefinuje, pretože potrebujeme rozlišovať *rôzne druhy slov*.

## Príklady atómov jazyka

*Príklad 1.13.* V jazyku  $\mathcal{L}_{\text{dz}}$ , kde  $\mathcal{C}_{\mathcal{L}_{\text{dz}}} = \{\text{Boby, Cilka, Evka, Jarka, Milo}\}$ ,  $\mathcal{P}_{\mathcal{L}_{\text{dz}}} = \{\text{dal, pes}\}$ ,  $\text{ar}_{\mathcal{L}_{\text{dz}}}(\text{dal}) = 3$ ,  $\text{ar}_{\mathcal{L}_{\text{dz}}}(\text{pes}) = 1$ , sú *okrem iných* rovnostné atómy:

Boby  $\doteq$  Boby

Cilka  $\doteq$  Boby

Evka  $\doteq$  Jarka

Boby  $\doteq$  Cilka

a predikátové atómy:

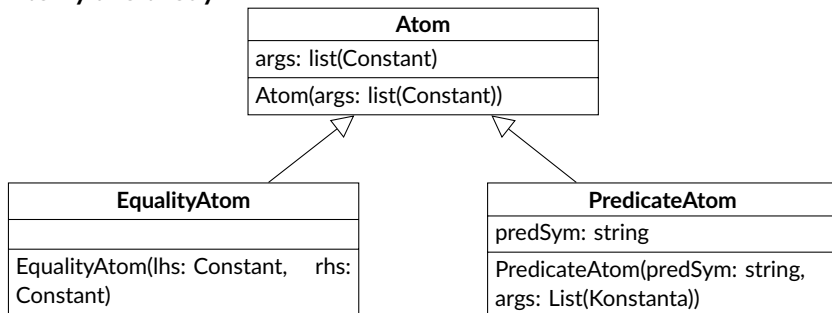
pes(Cilka)

dal(Cilka, Milo, Boby)

dal(Jarka, Evka, Milo).



## Atómy ako triedy



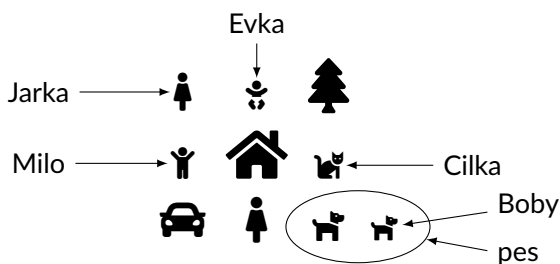
## 1.2 Štruktúry

### Vyhodnotenie atomickej formuly

Ako zistíme, či je atomická formula `pes(Boby)` *pravdivá* v nejakej situácii (napríklad u babky Evky, Jarky a Mila na dedine)?

Pozrieme sa na túto situáciu a zistíme:

1. aký objekt *b* pomenúva konštanta *Boby*;
2. akú vlastnosť *p* označuje predikát *pes*;
3. či objekt *b* má vlastnosť *p*.



### Vyhodnotenie atomickej formuly

Ako môžeme tento postup matematicky alebo informaticky modelovať? Potrebujeme:

- matematický/informatický model situácie (stavu vybranej časti sveta),

- postup na jeho použitie pri vyhodnocovaní pravdivosti formúl.

## Matematický model stavu sveta

Potrebujeme vedieť:

- ktoré objekty sú v popisovanej situácii prítomné,
- množina všetkých týchto objektov — *doména*;
- jednoznačné priradenie významu všetkým individuovým konštantám a predikátom z jazyka  $\mathcal{L}$
- *interpretačná funkcia*;
- pre každú individuovú konštantu  $c$  z jazyka  $\mathcal{L}$ , ktorý *objekt* z domény konštanty  $c$  pomenúva,
- pre každý unárny predikát  $P$  z jazyka  $\mathcal{L}$ , ktoré objekty z domény majú vlastnosť označenú predikátom  $P$ ,
- tvoria *podmnožinu* domény;
- pre každý  $n$ -árny predikát  $R$  z jazyka  $\mathcal{L}$ ,  $n > 1$ , ktoré  $n$ -tice objektov z domény sú vo vzťahu ozn. pred.  $R$ ,
- tvoria  *$n$ -árnu reláciu* na doméne.

## Štruktúra pre jazyk

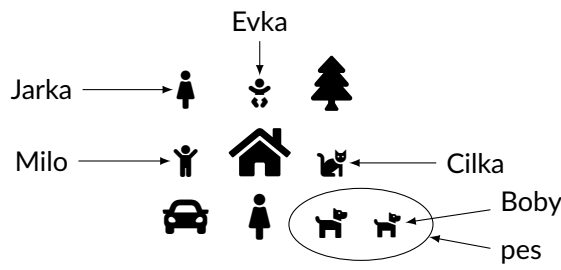
**Definícia 1.14.** Nech  $\mathcal{L}$  je jazyk atomických formúl logiky prvého rádu. *Štruktúrou* pre jazyk  $\mathcal{L}$  (niekedy *interpretáciou* jazyka  $\mathcal{L}$ ) nazývame dvojicu  $\mathcal{M} = (D, i)$ , kde  $D$  je ľubovoľná neprázdna množina nazývaná *doména* štruktúry  $\mathcal{M}$ ;  $i$  je zobrazenie, nazývané *interpretačná funkcia* štruktúry  $\mathcal{M}$ , ktoré

- každej individuovej konštanty  $c$  jazyka  $\mathcal{L}$  priraduje prvok  $i(c) \in D$ ;
- každému predikátovému symbolu  $P$  jazyka  $\mathcal{L}$  s aritou  $n$  priraduje množinu  $i(P) \subseteq D^n$ .

*Dohoda 1.15.* Štruktúry označujeme veľkými písanými písmenami  $\mathcal{M}, \mathcal{N}$ ,

....

# Príklad štruktúry



Príklad 1.16.

$$\begin{aligned} \mathcal{M} &= (D, i), \quad D = \left\{ \text{person}, \text{person}, \text{tree}, \text{person}, \text{house}, \text{cat}, \text{car}, \text{person}, \text{dog}, \text{dog} \right\} \\ i(\text{Boby}) &= \text{dog} & i(\text{Cilka}) &= \text{cat} \\ i(\text{Evka}) &= \text{person} & i(\text{Jarka}) &= \text{person} & i(\text{Milo}) &= \text{person} \\ i(\text{pes}) &= \{ \text{dog}, \text{dog} \} \\ i(\text{dal}) &= \left\{ (\text{person}, \text{person}, \text{dog}), (\text{person}, \text{person}, \text{dog}), (\text{person}, \text{person}, \text{cat}) \right\} \end{aligned}$$

## Štruktúra ako informatický objekt

Štruktúru sme definovali pomocou *matematických* objektov.

Aký *informatický* objekt sa podobá na štruktúru? *Databáza*.

Predikátové symboly jazyka  $\sim$  zjednodušená databázová schéma (arita  $\sim$  počet stĺpcov)

Interpretácia predikátových symbolov  $\sim$  konkrétne tabuľky s dátami (doména  $\sim$  dátový typ)

$i(\text{pes}^1)$	$i(\text{dal}^3)$			$i(\text{clovek}^2)$	
1	1	2	3	Meno	Rodné číslo
dog	person	person	dog	Petra	1234
dog	person	person	dog	Michal	5678
	person	person	cat		

## Štruktúra ako informatický objekt

Dopytom zodpovedajú logické formuly:

„rodné čísla ľudí, ktorí sa volajú Michal“  $= \{rc \mid \text{clovek}(\text{Michal}, rc)\}$   
„rodné čísla všetkých ľudí“  $= \{rc \mid \exists m \text{ clovek}(m, rc)\}$

## Štruktúry — upozornenia

Štruktúr pre daný jazyk je *nekonečne veľa*.

Doména štruktúry

- *nesúvisí so zamýšľaným významom* interpretovaného jazyka;
- môže mať ľubovoľné prvky;
- môže byť *nekonečná*.

Interpretácia symbolov konštánt:

- každej konštante je priradený objekt domény;
- nie každý objekt domény musí byť priradený nejakej konštante;
- rôznym konštantám môže byť priradený rovnaký objekt.

Interpretácie predikátových symbolov môžu byť *nekonečné*.

*Príklad 1.17* (Štruktúra s nekonečnou doménou).  $\mathcal{M} = (\mathbb{N}, i)$   $i(\text{pes}) = \{2n \mid n \in \mathbb{N}\}$   $i(\text{dal}) = \{(n, m, n + m) \mid n, m \in \mathbb{N}\}$   
 $i(\text{Boby}) = 0$   $i(\text{Cilka}) = 1$   $i(\text{Evka}) = 3$   $i(\text{Jarka}) = 5$   $i(\text{Milo}) = 0$

## 1.3 Sémantika atomických formúl

### Pravdivosť atomickej formuly v štruktúre

Ako zistíme, či je atomická formula pravdivá v štruktúre?

**Definícia 1.18.** Nech  $\mathcal{M} = (D, i)$  je štruktúra pre jazyk  $\mathcal{L}$  atomických formúl jazyka logiky prvého rádu.

Rovnostný atóm  $c_1 \doteq c_2$  jazyka  $\mathcal{L}$  je *pravdivý* v štruktúre  $\mathcal{M}$  vtedy a len vtedy, keď  $i(c_1) = i(c_2)$ .

Predikátový atóm  $P(c_1, \dots, c_n)$  jazyka  $\mathcal{L}$  je *pravdivý* v štruktúre  $\mathcal{M}$  vtedy a len vtedy, keď  $(i(c_1), \dots, i(c_n)) \in i(P)$ .

Vzťah *atóm*  $A$  je *pravdivý* v štruktúre  $\mathcal{M}$  skráteno zapisujeme  $\mathcal{M} \models A$ . Hovoríme aj, že  $\mathcal{M}$  je *modelom*  $A$ .

Vzťah *atóm*  $A$  *nie je pravdivý* v štruktúre  $\mathcal{M}$  zapisujeme  $\mathcal{M} \not\models A$ . Hovoríme aj, že  $A$  je *nepravdivý* v  $\mathcal{M}$  a  $\mathcal{M}$  *nie je modelom*  $A$ .

*Príklad 1.19* (Určenie pravdivosti atómov v štruktúre).

$$\begin{aligned} \mathcal{M} &= (D, i), \quad D = \left\{ \text{ľudia, strom, dieťa, dom, mačka, auto, žena, pes, pes} \right\} \\ i(\text{Boby}) &= \text{ľudia} & i(\text{Cilka}) &= \text{mačka} \\ i(\text{Evka}) &= \text{ľudia} & i(\text{Jarka}) &= \text{ľudia} & i(\text{Milo}) &= \text{dieťa} \\ i(\text{pes}) &= \{ \text{ľudia}, \text{pes} \} \\ i(\text{dal}) &= \left\{ (\text{ľudia}, \text{ľudia}, \text{pes}), (\text{ľudia}, \text{ľudia}, \text{pes}), (\text{ľudia}, \text{ľudia}, \text{mačka}) \right\} \end{aligned}$$

Atóm  $\text{pes}(\text{Boby})$  je *pravdivý* v štruktúre  $\mathcal{M}$ , t.j.,  $\mathcal{M} \models \text{pes}(\text{Boby})$ , lebo objekt  $i(\text{Boby}) = \text{ľudia}$  je prvkom množiny  $\{ \text{ľudia}, \text{pes} \} = i(\text{pes})$ .

Atóm  $\text{dal}(\text{Evka}, \text{Jarka}, \text{Cilka})$  je *pravdivý* v  $\mathcal{M}$ , t.j.,  $\mathcal{M} \models \text{dal}(\text{Evka}, \text{Jarka}, \text{Cilka})$ , lebo  $(i(\text{Evka}), i(\text{Jarka}), i(\text{Cilka})) = (\text{ľudia}, \text{ľudia}, \text{mačka}) \in i(\text{dal})$ .

Atóm  $\text{Cilka} \doteq \text{Boby}$  *nie je pravdivý* v  $\mathcal{M}$ , t.j.,  $\mathcal{M} \not\models \text{Cilka} \doteq \text{Boby}$ , lebo  $i(\text{Cilka}) = \text{mačka} \neq \text{ľudia} = i(\text{Boby})$ .

## 1.4 Zhrnutie

### Zhrnutie

- Logika prvého rádu je rodina formálnych jazykov.
- Každý jazyk logiky prvého rádu je daný neprázdnu množinou individuových konštánt a množinou predikátových symbolov.
- Atomické formuly sú základnými výrazmi prvorádového jazyka.
  - Postupnosti symbolov  $P(c_1, \dots, c_n)$  (predikátové) a  $c_1 \doteq c_2$  (rovnostné).
  - Zodpovedajú pozitívnym jednoduchým výrokom o vlastnostiach, stavoch, vzťahoch, rovnosti jednotlivých pomenovaných objektov.

- Význam jazyku dáva štruktúra — matematický opis stavu sveta
  - Skladá sa z neprázdnej domény a z interpretačnej funkcie.
  - Konštanty interpretuje ako prvky domény.
  - Predikáty interpretuje ako podmnožiny domény/relácie na doméne.
- Pravdivosť atómu určíme interpretovaním argumentov a zistením, či je výsledná  $n$ -tica objektov prvkom interpretácie predikátu, resp. pri rovnostnom atóme, či sa objekty rovnajú.

## 2. prednáška

# Výrokovologické spojky a ohodnotenia

---

### Rekapitulácia

Minulý týždeň sme si povedali:

- čo sú symboly jazyka *atomických formúl* logiky prvého rádu;
- čo sú atomické formuly;
- čo sú štruktúry:
  - modely stavu sveta,
  - neprázdna doména + interpretačná funkcia,
  - konštanty označujú objekty,
  - predikáty označujú vzťahy a vlastnosti;
- kedy sú atomické formuly pravdivé v danej štruktúre.
- Jazyk atomických formúl je oproti slovenčine veľmi slabý.
- Môžu byť pravdivé vo veľmi čudných štruktúrach.

## 2 Výrokovologické spojky a ohodnotenia

### Výrokovologické spojky

Atomické formuly logiky prvého rádu môžeme spájať do zložitejších tvrdení *výrokovologickými spojkami*.

- Zodpovedajú spojkám v slovenčine, ktorými vytvárame súvetia.
- Významom spojky je vždy *boolovská funkcia*, teda funkcia na pravdivostných hodnotách spájaných výrokov. Pravdivostná hodnota zloženého výroku závisí *iba* od pravdivostných hodnôt podvýrokov.

*Príklad 2.1.* Negácia, konjunkcia, disjunkcia, implikácia, ekvivalencia, ...

## Nevýrokovologické spojky

### Negatívny príklad

Spojka *pretože* nie je výrokovologická.

*Dôkaz.* Uvažujme o výroku „*Karol je doma, pretože Jarka je v škole*“.

*Je pravdivý v situácii:* Je 18:00 a Karol je doma, aby šiel na prechádzku s ich psom. Ten by inak musel čakať na Jarku, ktorá sa zo školy vráti až o 19:30.

*Nie je pravdivý v situácii:* Jarka išla ráno do školy, ale Karol ostal doma, lebo je chorý. S Jarkinou prítomnosťou v škole to nesúvisí.

V oboch situáciách sú výroky „*Karol je doma*“ aj „*Jarka je v škole*“ pravdivé, ale pravdivostná hodnota zloženého výroku je rôzna. *Nezávisí* iba od pravdivostných hodnôt podvýrokov (ale od existencie vzťahu *príčina-následok* medzi nimi).

Spojka *pretože* teda nie je *funkciou* na pravdivostných hodnotách.  $\square$

## 2.1 Boolovské spojky

### Negácia

Negácia  $\neg$  je *unárna* spojka — má jeden argument, formulu.

Zodpovedá výrazom *nie*, „*nie je pravda, že ...*“, predpone *ne*.

Lubovoľne vnárateľná.

Formula vytvorená negáciou sa *nezátvorkuje*.

Okolo argumentu negácie *nepridávame* zátvorky, ale môže ich mať on sám, ak to jeho štruktúra vyžaduje.

*Príklad 2.2.*

$\neg \text{doma}(\text{Karol})$	Karol <i>nie</i> je doma.
$\neg \text{Jarka} \doteq \text{Karol}$	Jarka <i>nie</i> je Karol.
$\neg \neg \neg \text{poslúcha}(\text{Cilka})$	<i>Nie</i> je pravda, že <i>nie</i> je pravda, že Cilka <i>neposlúcha</i> .
<del><math>(\neg \text{doma}(\text{Karol}))</math></del>	nesprávna
<del><math>\neg(\text{doma}(\text{Karol}))</math></del>	syntax

### Negácia rovnostného atómu

Rovnosť nie je spojka, preto:



✓  $\neg \text{Jarka} \doteq \text{Karol}$  — Jarka *nie* je Karol.

✗  $\neg (\text{Jarka} \doteq \text{Karol})$

Zátvorky sú zbytočné, lebo čítanie „*«Nie je pravda, že Jarka» sa rovná Karol*“ je nezmyselné:

1. Syntakticky: Negácia sa vzťahuje na formulu. Konštanta nie je formula, rovnosť s oboma argumentmi je.
2. Sémanticky: Negácia je funkcia na pravdivostných hodnotách. Konštanty označujú objekty domény. Objekty nie sú pravdivé ani nepravdivé.

*Dohoda 2.3.* Formulu  $\neg \tau \doteq \sigma$  budeme skrátene zapisovať  $\tau \neq \sigma$ .

## Konjunkcia

Konjunkcia  $\wedge$  je *binárna* spojka.

Zodpovedá spojkám *a, aj, i, tiež, ale, avšak, no, hoci, ani, ba (aj/ani), ...*

Formalizujeme ňou zlučovacie, stupňovacie a odporovacie súvetia:

- Jarka je doma *aj* Karol je doma.  
 $(\text{doma}(\text{Jarka}) \wedge \text{doma}(\text{Karol}))$
- Jarka je v škole, *no* Karol je doma.  
 $(\text{v\_škole}(\text{Jarka}) \wedge \text{doma}(\text{Karol}))$
- *Ani* Jarka nie je doma, *ani* Karol tam nie je.  
 $(\neg \text{doma}(\text{Jarka}) \wedge \neg \text{doma}(\text{Karol}))$
- *Nielen* Jarka je chorý, *ale aj* Karol je chorý.  
 $(\text{chorý}(\text{Jarka}) \wedge \text{chorý}(\text{Karol}))$

Zloženú formulu vždy *zátvorkujeme*.

## Formalizácia viacnásobných vetných členov konjunkciou

Zlučovacie viacnásobné vetné členy tiež formalizujeme ako konjunkcie:

- *Jarka aj Karol sú doma.*  
(doma(Jarka)  $\wedge$  doma(Karol))
- *Karol sa potkol a spadol.*  
(potkol\_sa(Karol)  $\wedge$  spadol(Karol))
- *Jarka dostala Bobyho od mamy a otca.*  
(dostal(Jarka, Boby, mama)  $\wedge$  dostal(Jarka, Boby, otec))

Podobne (jednoduché a viacnásobné zlučovacie) prívlastky vlastností:

- *Eismann je ruský špión.*  
(Rus(Eismann)  $\wedge$  špión(Eismann))
- *Boby je malý čierny pes.*  
((malý(Boby)  $\wedge$  čierny(Boby))  $\wedge$  pes(Boby))

## Stratené v preklade

Zlučovacie súvetia niekedy vyjadrujú časovú následnosť, ktorá sa pri priamočiarom preklade do logiky prvého rádu *stráca*:

- *Jarka a Karol sa stretli a išli do kina.* (stretli\_sa(Jarka, Karol)  $\wedge$  (do\_kina(Jarka)  $\wedge$  do\_kina(Karol)))
- *Jarka a Karol išli do kina a stretli sa.* ((do\_kina(Jarka)  $\wedge$  do\_kina(Karol))  $\wedge$  stretli\_sa(Jarka, Karol))

## Disjunkcia

Disjunkcia  $\vee$  je binárna spojka, ktorá zodpovedá spojkám *alebo*, či v *inkluzívnom* význame (môžu nastať aj obe možnosti). Inkluzívnu disjunkciu vyjadruje tiež „*alebo aj/i*“ a častice *respektíve*, *eventuálne*, *popripade*, *pripadne*.

Disjunkciou formalizujeme vylučovacie súvetia s inkluzívnym významom:

- *Jarka je doma alebo Karol je doma.*  
(doma(Jarka)  $\vee$  doma(Karol))

- Bobyho kúpe Jarka, prípadne ho kúpe Karol. ( $\text{kúpe}(\text{Jarka}, \text{Boby}) \vee \text{kúpe}(\text{Karol}, \text{Boby})$ )

Zloženú formulu vždy *zátvorkujeme*.

### Formalizácia viacnásobných vetných členov disjunkciou

Viacnásobné vetné členy s vylučovacou spojkou (v inkluzívnom význame) tiež prekladáme ako disjunkcie:

- Doma je Jarka alebo Karol. ( $\text{doma}(\text{Jarka}) \vee \text{doma}(\text{Karol})$ )
- Jarka je doma alebo v škole. ( $\text{doma}(\text{Jarka}) \vee \text{v\_škole}(\text{Jarka})$ )
- Jarka dostala Bobyho od mamy alebo otca. ( $\text{dostal}(\text{Jarka}, \text{Boby}, \text{mama}) \vee \text{dostal}(\text{Jarka}, \text{Boby}, \text{otec})$ )
- Boby je čierny či tmavohnedý psík. ( $((\text{čierny}(\text{Boby}) \vee \text{tmavohnedý}(\text{Boby})) \wedge \text{pes}(\text{Boby}))$ )

### Exkluzívna disjunkcia

Konštrukcie „*bud' ... , alebo ...*“, „*bud' ... , bud' ...*“, „*alebo ... , alebo ...*“ *spravidla* (v matematike vždy) vyjadrujú *exkluzívnu* disjunkciu.

- Bud' je batéria vybitá alebo svieti kontrolka.

Exkluzívnu disjunkciu môžeme vyjadriť zložitejšou formulou:

$$((\text{vybitá}(\text{batéria}) \vee \text{svieti}(\text{kontrolka})) \wedge \neg(\text{vybitá}(\text{batéria}) \wedge \text{svieti}(\text{kontrolka})))$$

Niekedy aj samotné *alebo* spája možnosti, o ktorých vieme, že sú vzájomne vylučné (na základe znalostí o fungovaní domény alebo z kontextu):

- Jarka sa nachádza doma alebo v škole. (Nemôže byť súčasne na dvoch miestach.)

Vid' *Znalosti na pozadí* ďalej.

## Jednoznačnosť rozkladu

Formuly s binárnymi spojkami sú vždy uzátvorkované. Dajú sa jednoznačne rozložiť na podformuly a interpretovať.

Slovenské tvrdenia so spojками nie sú vždy jednoznačné:

- Karol je doma a Jarka je doma alebo je Boby šťastný.

❓  $((\text{doma}(\text{Karol}) \wedge \text{doma}(\text{Jarka})) \vee \text{šťastný}(\text{Boby}))$

❓  $(\text{doma}(\text{Karol}) \wedge (\text{doma}(\text{Jarka}) \vee \text{šťastný}(\text{Boby})))$

- Karol je doma alebo Jarka je doma a Boby je šťastný.

❓  $((\text{doma}(\text{Karol}) \vee \text{doma}(\text{Jarka})) \wedge \text{šťastný}(\text{Boby}))$

❓  $(\text{doma}(\text{Karol}) \vee (\text{doma}(\text{Jarka}) \wedge \text{šťastný}(\text{Boby})))$

## Jednoznačnosť rozkladu v slovenčine

Slovenčina má prostriedky podobné zátvorkám:

- Viacnásobný vetný člen (+*obaja*, *niekto* z):

- Karol aj Jarka sú (obaja) doma alebo je Boby šťastný.

$((\text{doma}(\text{Karol}) \wedge \text{doma}(\text{Jarka})) \vee \text{šťastný}(\text{Boby}))$

- Doma je Karol alebo Jarka a Boby je šťastný.

Niekoľko z dvojice Karol a Jarka je doma a Boby je šťastný.

$((\text{doma}(\text{Karol}) \vee \text{doma}(\text{Jarka})) \wedge \text{šťastný}(\text{Boby}))$

- Kombinácie spojok *bud'* ..., *alebo* ...; *alebo* ..., *alebo* ...; *aj* ..., *aj* ...; *ani* ..., *ani* ...; a pod.

- Karol je doma a *bud'* je doma Jarka, alebo je Boby šťastný, alebo jedno aj druhé.

$(\text{doma}(\text{Karol}) \wedge (\text{doma}(\text{Jarka}) \vee \text{šťastný}(\text{Boby})))$

- Alebo je doma Karol, alebo je doma Jarka a Boby je šťastný, alebo aj aj.  $(\text{doma}(\text{Karol}) \vee (\text{doma}(\text{Jarka}) \wedge \text{šťastný}(\text{Boby})))$

## Jednoznačnosť rozkladu

Aj Karol je doma, aj Jarka je doma alebo je Bobby šťastný. Aj Karol je doma, aj Jarka je doma, alebo je Bobby šťastný.

- Má čiarka vplyv na význam tvrdenia?
- Chybovosť pri čiarkach je vysoká, pravidlá nie sú jednoznačné a v čase sa menia.
- Pri formalizácii pomáhajú dodatočné znalosti (napr. o spoločnom fungovaní Karola, Jarky, Bobbyho).

## Oblasť platnosti negácie

Výskyt negácie sa vzťahuje na *najkratšiu nasledujúcu formulu* – *oblasť platnosti* tohto výskytu.

- $((\neg \text{doma}(\text{Karol}) \wedge \text{doma}(\text{Jarka})) \vee \text{šťastný}(\text{Boby}))$
- $(\neg (\text{doma}(\text{Karol}) \wedge \text{doma}(\text{Jarka}))) \vee \text{šťastný}(\text{Boby})$

Argument negácie je *uzátvorkovaný práve vtedy*, keď je *priamo* vytvorený binárnou spojkou:

✓  $\neg \neg (\text{doma}(\text{Karol}) \wedge \text{doma}(\text{Jarka}))$

✗  $\neg (\neg (\text{doma}(\text{Karol}) \wedge \text{doma}(\text{Jarka})))$

## Interakcia negácie s alebo v slovenčine

### Zamyslite sa 2.1

Ako by ste sformalizovali: „Doma nie je Jarka alebo Karol“?

A.  $(\neg \text{doma}(\text{Jarka}) \vee \neg \text{doma}(\text{Karol}))$

B.  $\neg (\text{doma}(\text{Jarka}) \vee \text{doma}(\text{Karol}))$

Zvyčajné chápanie v slovenčine je A.

Formalizácii B zodpovedá „Nie je pravda, že je doma Jarka alebo Karol.“

## 2.2 Implikácia

### Implikácia

Implikácia  $\rightarrow$  je binárna spojka približne zodpovedajúca podmienkovému podrad'ovaciemu súvetiu *ak ... , tak ...*.

Vo formule  $(A \rightarrow B)$  hovoríme podformule  $A$  *antecedent* a podformule  $B$  *konzekvent*.

Formula vytvorená implikáciou je *nepravdivá* v *jedinom* prípade: antecedent je pravdivý a konzekvent nepravdivý.

⚠ Tomuto významu nezodpovedajú všetky súvetia *ak ... , tak ...*:

Napr. veta „Ak by Sarah prišla, Jim by prišiel tiež“ je nepravdivá, keď ňou chceme povedať, že si myslíme, že išli rovnakým autobusom, ale v skutočnosti Jim išiel iným a zmeškal ho.

Implikácia plne nevystihuje prípady, keď *ak ... , tak ...* vyjadruje (neboolovský) vzťah príčina-následok (ako *pretože*).

*Keď ... , potom ...* má často význam časovej následnosti, ktorý implikácia tiež nepostihuje.

### Nutná a postačujúca podmienka

Implikáciu vyjadrujú aj súvetia:

Jim príde, *ak* príde Kim.

Jim príde, *iba ak* príde Kim.

Vedľajšie vety (*príde Kim*) sú *podmienkami* hlavnej vety (*Jim príde*).

Ale je medzi nimi *podstatný rozdiel*:

Jim príde, *ak* príde Kim.  
*postačujúca*  
*podmienka*

Jim príde, *iba ak* príde Kim.  
*nutná*  
*podmienka*

### Postačujúca podmienka

Jim príde, *ak* príde Kim.

- Na to, aby prišiel Jim, *stačí*, aby prišla Kim.
- Teda, *ak* príde Kim, *tak* príde aj Jim.
- Nepravdivé, keď Kim príde, ale Jim *nepríde*.

- Zodpovedá teda ( $\text{príde}(\text{Kim}) \rightarrow \text{príde}(\text{Jim})$ ).

Vo všeobecnosti:

$$A, \text{ ak } B. \quad \rightsquigarrow \quad (B \rightarrow A)$$

Iné vyjadrenia:

- Jim príde, *pokiaľ* príde Kim.

### Nutná podmienka

Jim príde, *iba ak* príde Kim.

- Na to, aby prišiel Jim, *je nevyhnutné*, aby prišla Kim, ale nemusí to stačiť.
- Teda, ak Jim príde, tak príde aj Kim.
- Nepravdivé, keď Jim príde, ale Kim *nepríde*.
- Zodpovedá teda ( $\text{príde}(\text{Jim}) \rightarrow \text{príde}(\text{Kim})$ ).

Vo všeobecnosti:

$$A, \text{ iba ak } B. \quad \rightsquigarrow \quad (A \rightarrow B)$$

Iné vyjadrenia:

- Jim príde, *iba pokiaľ* príde Kim.
- Jim príde *iba* spolu s Kim.
- Jim *nepríde bez* Kim.

### Nutná a postačujúca podmienka rukolapne

Určite by sa vám páčilo, keby z pravidiel predmetu vyplývalo:

Z logiky prejdete, *ak* pridete na písomnú aj ústnu skúšku.

*Stačilo* by prísť na obe časti skúšky a *nebolo by nutné* urobiť nič iné.

Žiaľ, z našich pravidiel vyplýva:

Z logiky prejdete, *iba ak* pridete na písomnú aj ústnu skúšku.

Prísť na obe časti skúšky *je nutné*, ale na prejdienie to *nestačí*.

## Súvetia formalizované implikáciou

$(A \rightarrow B)$  formalizuje (okrem iných) zložené výroky:

- Ak  $A$ , tak  $B$ .
- Ak  $A$ , tak aj  $B$ .
- Ak  $A$ ,  $B$ .
- Pokiaľ  $A$ , [tak (aj)]  $B$ .
- $A$ , iba/len/jedine ak/pokiaľ(/keď)  $B$ .
- $A$  nastane iba spolu s  $B$ .
- $A$  nenastane bez  $B$ .
- $B$ , ak/pokiaľ(/keď)  $A$ .

## 2.3 Ekvivalencia

### Ekvivalencia

Ekvivalencia  $\leftrightarrow$  vyjadruje, že ňou spojené výroky majú rovnakú pravdivostnú hodnotu.

Zodpovedá slovenským výrazom *ak a iba ak; vtedy a len vtedy, keď; práve vtedy, keď; rovnaký ... ako ...; taký ... ako ...*.

- Jim príde, ak a iba ak príde Kim. ( $\text{príde}(\text{Jim}) \leftrightarrow \text{príde}(\text{Kim})$ )
- Číslo  $n$  je párne práve vtedy, keď  $n^2$  je párne. ( $\text{párne}(n) \leftrightarrow \text{párne}(n^2)$ )
- Müller je taký Nemec, ako je Stirlitz Rus. ( $\text{Nemec}(\text{Müller}) \leftrightarrow \text{Rus}(\text{Stirlitz})$ )

### Ekvivalencia

Ekvivalencia  $(A \leftrightarrow B)$  zodpovedá tvrdeniu, že  $A$  je nutnou aj postačujúcou podmienkou  $B$ .

Budeme ju preto považovať za skratku za formulu

$$((A \rightarrow B) \wedge (B \rightarrow A)).$$



## Ďalšie spojky a vetné konštrukcie

V slovenčine a iných prirodzených aj umelých jazykoch sa dajú tvoriť aj oveľa komplikovanejšie podmienené tvrdenia:

- Karol je doma, *ak* je Jarka v škole, *inak* má Jarka obavy.
- Karol je doma, *ak* je Jarka v škole, *inak* má Jarka obavy, *okrem* prípadov, keď je s ním Boby.

Výrokovologické spojky sa dajú vytvoriť aj pre takéto konštrukcie, ale väčšinou sa to nerobí.

Na ich vyjadrenie stačia aj základné spojky. Mohli by sme pre ne vymyslieť označenia a považovať ich za skratky, podobne ako ekvivalenciu.

## 2.4 Správnosť a vernosť formalizácie

### Skúška správnosti formalizácie

*Správnou formalizáciou* výroku je taká formula, ktorá je pravdivá *za tých istých okolností* ako formalizovaný výrok.

Formuly dokážeme vyhodnocovať iba v štruktúrach.

Preto *za tých istých okolností* znamená *v tých istých štruktúrach*.

### Vernosť formalizácie

Výrok „*Nie je pravda, že Jarka a Karol sú doma*“ sa dá *správne* formalizovať ako

$$\neg(\text{doma}(\text{Jarka}) \wedge \text{doma}(\text{Karol})),$$

ale rovnako *správna* je aj formalizácia

$$(\neg \text{doma}(\text{Jarka}) \vee \neg \text{doma}(\text{Karol})),$$

lebo je pravdivá v rovnakých štruktúrach.

Pri formalizácii sa snažíme o správnosť, ale zároveň *uprednostňujeme* formalizácie, ktoré *vernejšie* zachytávajú štruktúru výroku.

Zvyšuje to pravdepodobnosť, že sme neurobili chybu, a uľahčuje hľadanie chýb.

Prvá formalizácia je vernejšia ako druhá, a preto ju uprednostníme.

## Znalosti na pozadí

Na praktických cvičeniach ste sa stretli so *znalosťami na pozadí* (background knowledge): vzájomná výlučnosť vlastností *je Nemec a je Rus*, ktorá v úlohe nebola explicitne uvedená.

Uprednostňujeme ich vyjadrovanie *samostatnými formulami*.

Rovnaké dôvody ako pre vernosť.

## Skutočné súčasti významu a konverzačné implikatury

Niektoré tvrdenia *vyznievajú* silnejšie, ako naozaj sú:

- „*Prílohou sú zemiaky alebo šalát*“ môže niekomu znieť ako exkluzívna disjunkcia.
- „*Prejdete, ak všetky úlohy vyriešite na 100 %*“ znie mnohým ako ekvivalencia.

*Skutočnú časť významu* tvrdenia *nemôžeme poprieť* v dodatku k pôvodnému tvrdeniu bez sporu s ním.

- Keď k tvrdeniu „*Karol a Jarka sú doma*“ dodáme „*Ale Karol nie je doma*,“ dostaneme sa do sporu.

Takže „*Karol je doma*“ je skutočne časťou významu pôvodného výroku.

## Skutočné súčasti významu a konverzačné implikatury

Časť významu tvrdenia, ktorú *môžeme poprieť* dodatkami bez sporu s pôvodným tvrdením, sa nazýva *konverzačná implikatura* (H. P. Grice). *Nie je skutočnou časťou významu pôvodného tvrdenia.*

- Prílohou sú zemiaky alebo šalát. *Ale môžete si (pol na pol alebo za príplatok) dať aj oboje.*

Dodatok popiera exkluzívnosť, ale nie je v spore s tvrdením. Takže exkluzívnosť nie je súčasťou významu základného tvrdenia, je to iba konverzačná implikatura.

- Prejdete, ak všetky úlohy vyriešite na 100 %. *Ale nemusíte mať všetko na 100 %, aby ste prešli.*

Dodatok popiera implikáciu „*Prejdete*, iba ak *všetky úlohy vyriešite na 100 %*“, ale nie je v spore s pôvodným tvrdením. Táto implikácia teda nie je skutočne časťou významu základného tvrdenia, je to len konverzačná implikatúra.

### Formalizácia je ťažká

- Jedna formula môže byť rozdelená do viac viet. Nech  $x$  je kladné reálne. Potom  $x^2 > 0$ .
- Niektoré tvrdenia sú vnútorne veľmi zložité. Postupnosť prvočísel usporiadaných podľa veľkosti je rastúca.
- Existujú jazyky, kde pre niektoré logické spojky neexistuje slovo, využívajú sa iné prvky gramatiky.
- Logickú spojku občas treba hádať z kontextu. It's raining. The game is cancelled. It's raining *and therefore* the game is cancelled.
- Aj intonácia môže ovplyvniť formalizáciu. I said I *might* go. I did not say I am surely going, I only suggested the possibility.

### Formalizácia je ťažká

- Mnohé tvrdenia z praxe majú veľmi ďaleko od ideálnych jazykových schém (či už z hľadiska aplikácie gramatických pravidiel alebo formalizácie do presného jazyka).

„Na druhej strane si myslím, že Slovensko, niekedy žiaľ-bohu a niekedy je to aj chvalabohu, že žiaľ-bohu, na Slovensku predsa len tá pracovná sila je ešte stále lacnejšia a teraz, chvalabohu, že je lacnejšia.“

- Jednotlivci majú svoje vnímanie jazykových konštruktov založené na ich osobnej histórii a niekedy sa nezhodnú. Trénovacie množiny pre AI modely tak sú nekonzistentné a výsledok tréningu nemôže byť dokonalý.

## Formalizácia je ťažká

Aj najlepšie súčasné AI systémy potrebujú s formalizáciou ľudskú pomoc. Z článku o [AlphaGeometry](#) (2025, parafrázované):

A major weakness is the need to manually transform input problems from natural language into a domain-specific language. Automating this process is an active area of research. It is significantly more complicated than translation between human languages.

Formalization frequently requires re-formulating the original problem into an alternative *equivalent* form, and disambiguating the nuances in the original problem statement. *Automated formalization thus demands significant background knowledge and problem-solving skills* on its own. Using Gemini prompts containing examples obtained manually, we are able to formalize 30 out of 39 formalizable IMO geometry problems.

## 2.5 Syntax výrokovologických formúl

### Syntax a sémantika formúl s výrokovologickými spojkami

Podobne ako pri atomických formulách, aj pri formulách s výrokovologickými spojkami potrebujeme *zadefinovať* — presne a záväzne — ich *syntax* (skladbu) a *sémantiku* (význam).

Niektoré definície preberieme, iné rozšírime alebo modifikujeme, ďalšie pridáme.

*Syntax* výrokovologických formúl logiky prvého rádu špecifikuje:

- z čoho sa skladajú,
- čím sú a akú majú štruktúru.

### Symbole výrokovologickej časti logiky prvého rádu

**Definícia 2.4.** *Symbolmi jazyka  $\mathcal{L}$  výrokovologickej časti logiky prvého rádu sú:*

*mimologické symboly*, ktorými sú

- *individuové konštanty* z nejakej neprázdnej spočítateľnej množiny  $\mathcal{C}_{\mathcal{L}}$
- a *predikátové symboly* z nejakej spočítateľnej množiny  $\mathcal{P}_{\mathcal{L}}$ ;

*logické symboly*, ktorými sú

- *výrokovologické spojky*  $\neg, \wedge, \vee, \rightarrow$  (nazývané, v uvedenom poradí, *symbol negácie, symbol konjunkcie, symbol disjunkcie, symbol implikácie*);
- a *symbol rovnosti*  $\doteq$ ;

*pomocné symboly*  $(, )$  a  $,$  (ľavá zátvorka, pravá zátvorka a čiarka).

Množiny  $\mathcal{C}_{\mathcal{L}}$  a  $\mathcal{P}_{\mathcal{L}}$  sú disjunktné. Pomocné ani logické symboly sa nevyskytujú v symboloch z  $\mathcal{C}_{\mathcal{L}}$  ani  $\mathcal{P}_{\mathcal{L}}$ . Každému symbolu  $P \in \mathcal{P}_{\mathcal{L}}$  je priradená *arita*  $\text{ar}_{\mathcal{L}}(P) \in \mathbb{N}^+$ .

## Atomické formuly

Definícia atomických formúl je takmer rovnaká ako doteraz:

**Definícia 2.5.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu.

*Rovnostný atóm* jazyka  $\mathcal{L}$  je každá postupnosť symbolov  $c_1 \doteq c_2$ , kde  $c_1$  a  $c_2$  sú individuové konštanty z  $\mathcal{C}_{\mathcal{L}}$ .

*Predikátový atóm* jazyka  $\mathcal{L}$  je každá postupnosť symbolov  $P(c_1, \dots, c_n)$ , kde  $P$  je predikátový symbol z  $\mathcal{P}_{\mathcal{L}}$  s aritou  $n$  a  $c_1, \dots, c_n$  sú individuové konštanty z  $\mathcal{C}_{\mathcal{L}}$ .

*Atomickými formulami* (skrátene *atómami*) jazyka  $\mathcal{L}$  súhrnne nazývame všetky rovnostné a predikátové atómy jazyka  $\mathcal{L}$ .

Množinu všetkých atómov jazyka  $\mathcal{L}$  označujeme  $\mathcal{A}_{\mathcal{L}}$ .

## Čo sú výrokovologické formuly?

Majme jazyk  $\mathcal{L}$ , kde  $\mathcal{C}_{\mathcal{L}} = \{\text{Kim}, \text{Jim}, \text{Sarah}\}$  a  $\mathcal{P}_{\mathcal{L}} = \{\text{príde}^1\}$ .

Čo sú formuly tohto jazyka?

- Samotné atómy, napr.  $\text{príde}(\text{Sarah})$ .
- Negácie atómov, napr.  $\neg \text{príde}(\text{Sarah})$ .

- Atómy alebo aj ich negácie spojené spojkou, napr.  $(\neg \text{príde}(\text{Kim}) \vee \text{príde}(\text{Sarah}))$ .
- Ale negovať a spájať spojkami môžeme aj zložitejšie formuly, napr.  $(\neg(\text{príde}(\text{Kim}) \wedge \text{príde}(\text{Sarah})) \rightarrow (\neg \text{príde}(\text{Kim}) \vee \neg \text{príde}(\text{Sarah})))$ .

Ako to presne a úplne popíšeme?

### Čo sú výrokovologické formuly?

Ako presne a úplne popíšeme, čo je formula?

*Induktívnou* definíciou:

1. Povieme, čo sú základné formuly, ktoré sa nedajú rozdeliť na menšie formuly.
  - Podobne ako báza pri matematickej indukcii.
2. Opíšeme, ako sa z jednoduchších formúl skladajú zložitejšie.
  - Podobne ako indukčný krok pri matematickej indukcii.
3. Zabezpečíme, že nič iné nie je formulou.

### Formuly jazyka výrokovologickej časti logiky prvého rádu

**Definícia 2.6.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Množina  $\mathcal{E}_{\mathcal{L}}$  formúl jazyka  $\mathcal{L}$  je (3.) *najmenšia* množina postupností symbolov, ktorá spĺňa všetky nasledujúce podmienky:

1. Každý atóm z  $\mathcal{A}_{\mathcal{L}}$  je formulou z  $\mathcal{E}_{\mathcal{L}}$ .
- 2.1. Ak  $A$  patrí do  $\mathcal{E}_{\mathcal{L}}$ , tak aj postupnosť symbolov  $\neg A$  patrí do  $\mathcal{E}_{\mathcal{L}}$  a nazývame ju *negácia* formuly  $A$ .
- 2.2. Ak  $A$  a  $B$  sú v  $\mathcal{E}_{\mathcal{L}}$ , tak aj postupnosti symbolov  $(A \wedge B)$ ,  $(A \vee B)$  a  $(A \rightarrow B)$  patria do  $\mathcal{E}_{\mathcal{L}}$  a nazývame ich postupne *konjunkcia*, *disjunkcia* a *implikácia* formúl  $A$  a  $B$ .

Každý prvok  $A$  množiny  $\mathcal{E}_{\mathcal{L}}$  nazývame *formulou* jazyka  $\mathcal{L}$ .

## Dohody • Vytvorenie formuly

*Dohoda 2.7.* Formuly označujeme meta premennými  $A, B, C, X, Y, Z$ , podľa potreby aj s dolnými indexmi.

*Dohoda 2.8.* Pre každú dvojicu formúl  $A, B \in \mathcal{E}_{\mathcal{L}}$  je zápis  $(A \leftrightarrow B)$  skratka za formulu  $((A \rightarrow B) \wedge (B \rightarrow A))$ .

Technicky  $(\cdot \leftrightarrow \cdot) : \mathcal{E}_{\mathcal{L}} \times \mathcal{E}_{\mathcal{L}} \rightarrow \mathcal{E}_{\mathcal{L}}$  je funkcia na formulách definovaná ako  $(A \leftrightarrow B) = ((A \rightarrow B) \wedge (B \rightarrow A))$  pre každé dve formuly  $A$  a  $B$ .

*Príklad 2.9.* Ako by sme podľa definície 2.6 mohli dokázať, že  $(\neg \text{príde}(\text{Kim}) \rightarrow (\text{príde}(\text{Jim}) \vee \text{príde}(\text{Sarah})))$  je formula? Teda, ako by sme ju podľa definície 2.6 mohli vytvoriť?

## Vytvárajúca postupnosť

**Definícia 2.10.** *Vytvárajúcou postupnosťou* nad jazykom  $\mathcal{L}$  výrokovologickej časti logiky prvého rádu je ľubovoľná konečná postupnosť  $A_0, \dots, A_n$  postupností symbolov, ktorej každý člen

- je atóm z  $\mathcal{A}_{\mathcal{L}}$ , alebo
- má tvar  $\neg A$ , pričom  $A$  je niektorý predchádzajúci člen postupnosti, alebo
- má jeden z tvarov  $(A \wedge B)$ ,  $(A \vee B)$ ,  $(A \rightarrow B)$ , kde  $A$  a  $B$  sú niektoré predchádzajúce členy postupnosti.

*Vytvárajúcou postupnosťou pre  $X$*  je ľubovoľná vytvárajúca postupnosť, ktorej posledným prvkom je  $X$ .

## Indukcia (vzhľadom) na konštrukciu formuly

**Veta 2.11** (Princíp indukcie na konštrukciu formuly). *Nech  $P$  je ľubovoľná vlastnosť formúl ( $P \subseteq \mathcal{E}_{\mathcal{L}}$ ). Ak platí súčasne*

1. *každý atóm z  $\mathcal{A}_{\mathcal{L}}$  má vlastnosť  $P$ ,*
- 2.1. *ak formula  $A$  má vlastnosť  $P$ , tak aj  $\neg A$  má vlastnosť  $P$ ,*
- 2.2. *ak formuly  $A$  a  $B$  majú vlastnosť  $P$ , tak aj každá z formúl  $(A \wedge B)$ ,  $(A \vee B)$  a  $(A \rightarrow B)$  má vlastnosť  $P$ ,*

*tak všetky formuly majú vlastnosť  $P$  ( $P = \mathcal{E}_{\mathcal{L}}$ ).*

## Formula a existencia vytvárajúcej postupnosti

**Tvrdenie 2.12.** *Postupnosť symbolov  $A$  je výrokovologickou formulou vtt existuje vytvárajúca postupnosť pre  $A$ .*

Osnova dôkazu. ( $\Rightarrow$ ) Indukciou na konštrukciu formuly

( $\Leftarrow$ ) Indukciou na dĺžku vytvárajúcej postupnosti

□

vtt skracuje „vtedy a len vtedy, keď“.

Výrokovologické formuly by sa dali alternatívne zadefinovať ako postupnosti symbolov jazyka  $\mathcal{L}$ , pre ktoré existuje vytvárajúca postupnosť nad  $\mathcal{L}$ .

Výhoda: Dĺžka vytvárajúcej postupnosti je číslo, tvrdenia o všetkých formulách sa potom dajú dokazovať matematickou alebo úplnou indukciou.

## (Ne)jednoznačnosť rozkladu formúl výrokovej logiky

Čo keby sme zdefinovali „formuly“ takto?

### Definícia „formúl“



Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Množina  $\mathcal{E}_{\mathcal{L}}$  „formúl“ jazyka  $\mathcal{L}$  je (3.) *najmenšia* množina postupností symbolov, ktorá spĺňa všetky nasledujúce podmienky:

1. Každý atóm z  $\mathcal{A}_{\mathcal{L}}$  je „formulou“ z  $\mathcal{E}_{\mathcal{L}}$ .
- 2.1. Ak  $A$  patrí do  $\mathcal{E}_{\mathcal{L}}$ , tak aj postupnosť symbolov  $\neg A$  patrí do  $\mathcal{E}_{\mathcal{L}}$ .
- 2.2. Ak  $A$  a  $B$  sú v  $\mathcal{E}_{\mathcal{L}}$ , tak aj postupnosti symbolov  $A \wedge B$ ,  $A \vee B$  a  $A \rightarrow B$  patria do  $\mathcal{E}_{\mathcal{L}}$ .
- 2.3. ak  $A$  patrí do  $\mathcal{E}_{\mathcal{L}}$ , tak aj postupnosť symbolov  $(A)$  je v  $\mathcal{E}_{\mathcal{L}}$ .

Každý prvok  $A$  množiny  $\mathcal{E}_{\mathcal{L}}$  nazývame „formulou“ jazyka  $\mathcal{L}$ .

Čo znamená „formula“ (príde(Jim)  $\rightarrow$  príde(Kim)  $\rightarrow$   $\neg$ príde(Sarah))?

Formulu by sme mohli čítať ako  $A = (\text{príde}(\text{Jim}) \rightarrow (\text{príde}(\text{Kim}) \rightarrow \neg \text{príde}(\text{Sarah})))$  alebo ako  $B = ((\text{príde}(\text{Jim}) \rightarrow \text{príde}(\text{Kim})) \rightarrow \neg \text{príde}(\text{Sarah}))$ .

Čítanie  $A$  hovorí, že Sarah nepríde, ak prídu Jim a Kim súčasne. To neplatí v *práve jednej* situácii: keď všetci prídu.

Čítanie  $B$  hovorí, že Sarah nepríde, ak alebo nepríde Jim alebo príde Kim. To však neplatí v *aspoň dvoch* rôznych situáciách: keď prídu všetci a keď príde Sarah a Kim, ale nie Jim.



## Jednoznačnosť rozkladu formúl výrokovej logiky

Pre našu definíciu formúl platí:

**Tvrdenie 2.13** (o jednoznačnosti rozkladu). *Pre každú formulu  $X \in \mathcal{E}_{\mathcal{L}}$  v jazyku  $\mathcal{L}$  platí práve jedna z nasledujúcich možností:*

- $X$  je atóm z  $\mathcal{A}_{\mathcal{L}}$ .
- Existuje práve jedna formula  $A \in \mathcal{E}_{\mathcal{L}}$  taká, že  $X = \neg A$ .
- Existujú práve jedna dvojica formúl  $A, B \in \mathcal{E}_{\mathcal{L}}$  a jedna binárna spojka  $b \in \{\wedge, \vee, \rightarrow\}$  také, že  $X = (A \ b \ B)$ .

## Problémy s vytvárajúcou postupnosťou

Vytvárajúca postupnosť popisuje konštrukciu formuly podľa definície formúl:

príde(Jim), príde(Sarah),  $\neg$ príde(Jim), príde(Kim),  
 $\neg$ príde(Sarah),  $(\neg$ príde(Jim)  $\wedge$  príde(Kim)),  
 $((\neg$ príde(Jim)  $\wedge$  príde(Kim))  $\rightarrow \neg$ príde(Sarah))

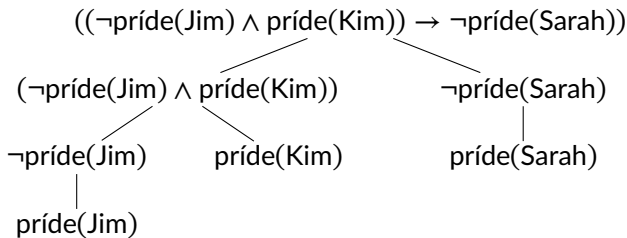
ale

- môže obsahovať „zbytočné“ prvky;
- nie je jasné, ktoré z predchádzajúcich formúl sa *bezprostredne* použijú na vytvorenie nasledujúcej formuly.

Akou „dátovou štruktúrou“ vieme vyjadriť konštrukciu formuly bez týchto problémov?

## Vytvárajúci strom

Konštrukciu si vieme predstaviť ako *strom*:



Takéto stromy voláme *vytvárajúce*.

Ako ich *presne* a *všeobecne* popíšeme — zdefinujeme?

Podobne ako sa definuje napr. binárny vyhľadávací strom.

### Vytvárajúci strom formuly

**Definícia 2.14.** *Vytvárajúci strom*  $T$  pre formulu  $X$  je binárny strom obsahujúci v každom vrchole formulu, pričom platí:

- v koreni  $T$  je formula  $X$ ,
- ak vrchol obsahuje formulu  $\neg A$ , tak má práve jedno dieťa, ktoré obsahuje formulu  $A$ ,
- ak vrchol obsahuje formulu  $(A \ b \ B)$ , kde  $b$  je jedna z binárnych spojok, tak má dve deti, pričom ľavé dieťa obsahuje formulu  $A$  a pravé formulu  $B$ ,
- vrcholy obsahujúce atómy sú listami.

### Syntaktické vzťahy formúl

Uvažujme formulu:

$$((\neg \text{príde}(\text{Jim}) \wedge \text{príde}(\text{Kim})) \rightarrow \neg \text{príde}(\text{Sarah}))$$

Ako nazveme formuly, z ktorých vznikla?

$$\text{príde}(\text{Sarah}), \neg \text{príde}(\text{Jim}), (\neg \text{príde}(\text{Jim}) \wedge \text{príde}(\text{Kim})), \dots$$

Ako nazveme formuly, z ktorých *bezprostredne/priamo* vznikla?

$$(\neg \text{príde}(\text{Jim}) \wedge \text{príde}(\text{Kim})) \quad \text{a} \quad \neg \text{príde}(\text{Sarah})$$

Ako tieto pojmy presne zdefinujeme?

## Podformuly

**Definícia 2.15** (Priama podformula). Pre všetky formuly  $A$  a  $B$ :

- Priamou podformulou  $\neg A$  je formula  $A$ .
- Priamymi podformulami  $(A \wedge B)$ ,  $(A \vee B)$  a  $(A \rightarrow B)$  sú formuly  $A$  (ľavá priama podformula) a  $B$  (pravá priama podformula).

**Definícia 2.16** (Podformula). Vzťah *byť podformulou* je najmenšia relácia na formulách spĺňajúca pre všetky formuly  $X$ ,  $Y$  a  $Z$ :

- $X$  je podformulou  $X$ .
- Ak  $X$  je priamou podformulou  $Y$ , tak  $X$  je podformulou  $Y$ .
- Ak  $X$  je podformulou  $Y$  a  $Y$  je podformulou  $Z$ , tak  $X$  je podformulou  $Z$ .

Formula  $X$  je *vlastnou podformulou* formuly  $Y$  práve vtedy, keď  $X$  je podformulou  $Y$  a  $X \neq Y$ .

## Meranie syntaktickej zložitosti formúl

Miera zložitosti/veľkosti formuly:

- Jednoduchá: dĺžka, teda počet symbolov
  - Počíta aj pomocné symboly.
  - Nič nemá mieru 0, ani atómy.
- Lepšia: počet netriviálnych krokov pri konštrukcii formuly
  - pridanie negácie,
  - spojenie formúl spojkou.

Túto lepšiu mieru nazývame *stupeň formuly*.

*Príklad 2.17.* Aký je stupeň formuly  $((\text{príde}(\text{Jim}) \vee \neg \text{príde}(\text{Kim})) \wedge \neg (\text{príde}(\text{Sarah}) \rightarrow \text{príde}(\text{Kim})))$ ?

## Meranie syntaktickej zložitosti formúl

Ako stupeň zadefinujeme?

Podobne ako sme zadefinovali formuly – induktívne:

1. určíme hodnotu stupňa pre atomické formuly,
2. určíme, ako zo stupňa priamych podformúl vypočítame stupeň z nich zloženej formuly.

## Stupeň formuly

**Definícia 2.18** (Stupeň formuly). Pre všetky formuly  $A$  a  $B$  a všetky  $n, n_1, n_2 \in \mathbb{N}$ :

- Atomická formula je stupňa 0.
- Ak  $A$  je formula stupňa  $n$ , tak  $\neg A$  je stupňa  $n + 1$ .
- Ak  $A$  je formula stupňa  $n_1$  a  $B$  je formula stupňa  $n_2$ , tak  $(A \wedge B)$ ,  $(A \vee B)$  a  $(A \rightarrow B)$  sú stupňa  $n_1 + n_2 + 1$ .

**Definícia 2.18** (Stupeň formuly presnejšie a symbolicky). *Stupeň*  $\deg(X)$  formuly  $X \in \mathcal{E}_{\mathcal{L}}$  definujeme pre všetky formuly  $A, B \in \mathcal{E}_{\mathcal{L}}$  nasledovne:

- $\deg(A) = 0$ , ak  $A \in \mathcal{A}_{\mathcal{L}}$ ,
- $\deg(\neg A) = \deg(A) + 1$ ,
- $\deg((A \wedge B)) = \deg((A \vee B)) = \deg((A \rightarrow B)) = \deg(A) + \deg(B) + 1$ .

## Indukcia na stupeň formuly

Pomocou stupňa vieme indukciu na konštrukciu formuly zredukovať na špeciálny prípad matematickej indukcie:

**Veta 2.19** (Princíp indukcie na stupeň formuly). *Nech  $P$  je ľubovoľná vlastnosť formúl ( $P \subseteq \mathcal{E}_{\mathcal{L}}$ ). Ak platí súčasne*

1. *báza indukcie: každá formula stupňa 0 má vlastnosť  $P$ ,*
2. *indukčný krok: pre každú formulu  $X$  z predpokladu, že všetky formuly menšieho stupňa ako  $\deg(X)$  majú vlastnosť  $P$ , vyplýva, že aj  $X$  má vlastnosť  $P$ ,*

*tak všetky formuly majú vlastnosť  $P$  ( $P = \mathcal{E}_{\mathcal{L}}$ ).*

## 2.6 Sémantika výrokovologických formúl

### Sémantika výrokovej logiky

Význam formúl výrokovologickej časti logiky prvého rádu popíšeme podobne ako význam atomických formúl pomocou *štruktúr*.

#### Štruktúra pre jazyk

Definícia štruktúry sa takmer nemení:

**Definícia 2.20.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. *Štruktúrou* pre jazyk  $\mathcal{L}$  nazývame dvojicu  $\mathcal{M} = (D, i)$ , kde  $D$  je ľubovoľná neprázdna množina nazývaná *doména* štruktúry  $\mathcal{M}$ ;  $i$  je zobrazenie, nazývané *interpretačná funkcia* štruktúry  $\mathcal{M}$ , ktoré

- každému symbolu konštanty  $c$  jazyka  $\mathcal{L}$  priraduje prvok  $i(c) \in D$ ;
- každému predikátovému symbolu  $P$  jazyka  $\mathcal{L}$  s aritou  $n$  priraduje množinu  $i(P) \subseteq D^n$ .

#### Pravdivosť formuly v štruktúre

**Definícia 2.21.** Nech  $\mathcal{M} = (D, i)$  je štruktúra pre jazyk  $\mathcal{L}$  výrokovologickej časti logiky prvého rádu. Reláciu *formula  $A$  je pravdivá v štruktúre  $\mathcal{M}$*  ( $\mathcal{M} \models A$ ) definujeme *induktívne* pre všetky arity  $n > 0$ , všetky predikátové symboly  $P$  s aritou  $n$  všetky konštanty  $c_1, c_2, \dots, c_n$ , a všetky formuly  $A, B$  jazyka  $\mathcal{L}$  nasledovne:

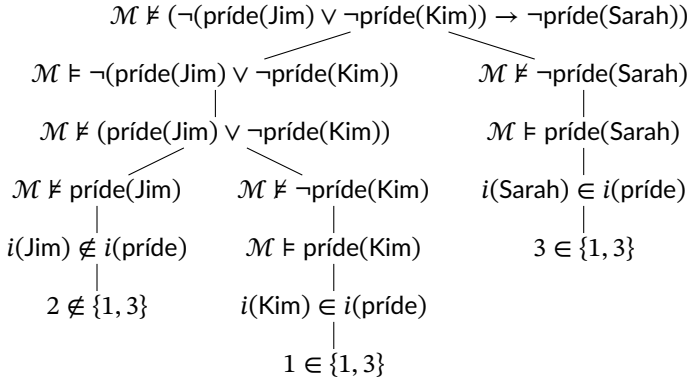
- $\mathcal{M} \models c_1 \doteq c_2$  vtt  $i(c_1) = i(c_2)$ ,
- $\mathcal{M} \models P(c_1, \dots, c_n)$  vtt  $(i(c_1), \dots, i(c_n)) \in i(P)$ ,
- $\mathcal{M} \models \neg A$  vtt  $\mathcal{M} \not\models A$ ,
- $\mathcal{M} \models (A \wedge B)$  vtt  $\mathcal{M} \models A$  a zároveň  $\mathcal{M} \models B$ ,
- $\mathcal{M} \models (A \vee B)$  vtt  $\mathcal{M} \models A$  alebo  $\mathcal{M} \models B$ ,
- $\mathcal{M} \models (A \rightarrow B)$  vtt  $\mathcal{M} \not\models A$  alebo  $\mathcal{M} \models B$ ,

kde  $\mathcal{M} \not\models A$  skrakuje  *$A$  nie je pravdivá v  $\mathcal{M}$* .

## Vyhodnotenie pravdivosti formuly

*Príklad 2.22* (Vyhodnotenie pravdivosti formuly v štruktúre). Majme štruktúru  $\mathcal{M} = (D, i)$  pre jazyk o party, kde  $D = \{0, 1, 2, 3\}$ ,  $i(\text{Kim}) = 1$ ,  $i(\text{Jim}) = 2$ ,  $i(\text{Sarah}) = 3$ ,  $i(\text{príde}) = \{1, 3\}$ .

Formuly vyhodnocujeme podľa definície postupom zdola nahor (od atómov cez zložitejšie podformuly k cieľovej formule):



## Vyhodnotenie pravdivosti formuly

*Príklad 2.23* (Vyhodnotenie pravdivosti formuly v štruktúre). Majme štruktúru  $\mathcal{M} = (D, i)$  pre jazyk o party, kde  $D = \{0, 1, 2, 3\}$ ,  $i(\text{Kim}) = 1$ ,  $i(\text{Jim}) = 2$ ,  $i(\text{Sarah}) = 3$ ,  $i(\text{príde}) = \{1, 3\}$ .

Vyhodnotenie pravdivosti môžeme zapísať aj tabuľkou:

	$p(J)$	$p(K)$	$\neg p(K)$	$(p(J) \vee \neg p(K))$	$\neg(p(J) \vee \neg p(K))$	...
$\mathcal{M}$	$\not\models$	$\models$	$\not\models$	$\not\models$	$\models$	

	$p(S)$	$\neg p(S)$	$(\neg(p(J) \vee \neg p(K)) \rightarrow \neg p(S))$
...	$\models$	$\not\models$	$\not\models$

kde  $p = \text{príde}$ ,  $K = \text{Kim}$ ,  $J = \text{Jim}$  a  $S = \text{Sarah}$ .

Všimnite si, že v záhlaví tabuľky je vytvárajúca postupnosť vyhodnocovanej formuly.

## Hľadanie štruktúry

*Príklad 2.24* (Nájdenie štruktúry, v ktorej je formula pravdivá). V akej štruktúre  $\mathcal{M} = (D, i)$  je pravdivá formula  $\mathcal{M} \models (\neg(\text{príde}(\text{Jim}) \vee \neg \text{príde}(\text{Kim})) \rightarrow \neg \text{príde}(\text{Sarah}))$ ?

Na zodpovedanie je dobré postupovať podľa definície pravdivosti zhora nadol (od cieľovej formuly cez podformuly k atómom):

$\mathcal{M} \models (\neg(\text{príde}(\text{Jim}) \vee \neg \text{príde}(\text{Kim})) \rightarrow \neg \text{príde}(\text{Sarah}))$  vtt  $\mathcal{M} \not\models \neg(\text{príde}(\text{Jim}) \vee \neg \text{príde}(\text{Kim}))$  alebo  $\mathcal{M} \models \neg \text{príde}(\text{Sarah})$  vtt  $\mathcal{M} \models (\text{príde}(\text{Jim}) \vee \neg \text{príde}(\text{Kim}))$  alebo  $\mathcal{M} \not\models \text{príde}(\text{Sarah})$  vtt  $\mathcal{M} \models \text{príde}(\text{Jim})$  alebo  $\mathcal{M} \models \neg \text{príde}(\text{Kim})$  alebo  $\mathcal{M} \not\models \text{príde}(\text{Sarah})$  vtt  $i(\text{Jim}) \in i(\text{príde})$  alebo  $i(\text{Kim}) \notin i(\text{príde})$  alebo  $i(\text{Sarah}) \notin i(\text{príde})$ .

## 2.7 Teórie a ich modely

### Teórie v neformálnej logike

Medzi základnými logickými pojmami z úvodnej prednášky boli teória a model.

Neformálne je *teória* súbor tvrdení, ktoré pokladáme za pravdivé.

Zvyčajne popisujú našu predstavu o zákonitostiach platných v nejakej časti sveta a pozorovania o jej stave.

*Príklad 2.25.* Máme troch nových známych — Kim, Jima a Sarah. Organizujeme párty a P0: chceme, aby na ňu prišiel niekto z nich. Od spoločných kamarátov sme sa ale dozvedeli o ich požiadavkách:

P1: Sarah nepríde na párty, ak príde Kim.

P2: Jim príde na párty, len ak príde Kim.

P3: Sarah nepríde bez Jima.

### Výrokovologické teórie

V logike prvého rádu tvrdenia zapisujeme formulami. Teóriu preto budeme chápať ako súbor (čiže množinu) formúl.

**Definícia 2.26.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu.

Každú množinu formúl jazyka  $\mathcal{L}$  budeme nazývať *teóriou* v jazyku  $\mathcal{L}$ .

Príklad 2.27.

$$T_{\text{party}} = \{((\text{príde}(\text{Kim}) \vee \text{príde}(\text{Jim})) \vee \text{príde}(\text{Sarah})), \\ (\text{príde}(\text{Kim}) \rightarrow \neg \text{príde}(\text{Sarah})), \\ (\text{príde}(\text{Jim}) \rightarrow \text{príde}(\text{Kim})), \\ (\text{príde}(\text{Sarah}) \rightarrow \text{príde}(\text{Jim}))\}$$

## Modely teórií

Neformálne je *modelom* teórie stav vybranej časti sveta, v ktorom sú všetky tvrdenia v teórii pravdivé.

Pre logiku prvého rádu stavy sveta vyjadrujú štruktúry.

Príklad 2.28 (Model teórie o party).

$$\begin{aligned} \mathcal{M} &= (\{k, j, s, e, h\}, i), \\ i(\text{Kim}) &= k, \quad i(\text{Jim}) = j, \quad i(\text{Sarah}) = s, \\ i(\text{príde}) &= \{k, j, e\}; \\ \left. \begin{aligned} \mathcal{M} &\models ((\text{príde}(\text{Kim}) \vee \text{príde}(\text{Jim})) \vee \text{príde}(\text{Sarah})) \\ \mathcal{M} &\models (\text{príde}(\text{Kim}) \rightarrow \neg \text{príde}(\text{Sarah})) \\ \mathcal{M} &\models (\text{príde}(\text{Jim}) \rightarrow \text{príde}(\text{Kim})) \\ \mathcal{M} &\models (\text{príde}(\text{Sarah}) \rightarrow \text{príde}(\text{Jim})) \end{aligned} \right\} \mathcal{M} \models T_{\text{party}} \end{aligned}$$

## Model teórie

**Definícia 2.29** (Model). Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu a nech  $T$  je teória v jazyku  $\mathcal{L}$  a  $\mathcal{M}$  je štruktúra pre jazyk  $\mathcal{L}$ .

Teória  $T$  je *pravdivá* v  $\mathcal{M}$ , skrátene  $\mathcal{M} \models T$ , vtt každá formula  $X$  z  $T$  je pravdivá v  $\mathcal{M}$  (teda  $\mathcal{M} \models X$ ).

Hovoríme tiež, že  $\mathcal{M}$  je *modelom*  $T$ .

Teória  $T$  je *nepravdivá* v  $\mathcal{M}$ , skrátene  $\mathcal{M} \not\models T$ , vtt  $T$  nie je pravdivá v  $\mathcal{M}$ .

## 2.8 Výrokovologické ohodnotenia

### Nekonečne veľa štruktúr



Logickými dôsledkami teórie sú tvrdenia, ktoré sú pravdivé vo všetkých modeloch teórie.

$$T_{\text{party}} = \{((\text{príde}(\text{Kim}) \vee \text{príde}(\text{Jim})) \vee \text{príde}(\text{Sarah})), \\ (\text{príde}(\text{Kim}) \rightarrow \neg \text{príde}(\text{Sarah})), \\ (\text{príde}(\text{Jim}) \rightarrow \text{príde}(\text{Kim})), \\ (\text{príde}(\text{Sarah}) \rightarrow \text{príde}(\text{Jim}))\}$$

Ale štruktúra je nekonečne veľá a ak má teória jeden model, má aj nekonečne veľá ďalších:

$\mathcal{M}_1 = (\{k, j, s\}, i_1)$	$\mathcal{M}'_1 = (\{k, j, s, 0, 1\}, i'_1)$	$\mathcal{M}''_1 = (\{2, 4, 6\}, i''_1) \quad \dots$
$i_1(\text{Kim}) = k$	$i'_1(\text{Kim}) = k$	$i''_1(\text{Kim}) = 2$
$i_1(\text{Jim}) = j$	$i'_1(\text{Jim}) = j$	$i''_1(\text{Jim}) = 4$
$i_1(\text{Sarah}) = s$	$i'_1(\text{Sarah}) = s$	$i''_1(\text{Sarah}) = 6$
$i_1(\text{príde}) = \{k, j\}$	$i'_1(\text{príde}) = \{k, j, 1\}$	$i''_1(\text{príde}) = \{2, 4\}$

## Rozdiely modelov

V čom sa líšia a čo majú spoločné nasledujúce modely  $T_{\text{party}}$ ?

$\mathcal{M}_1 = (\{k, j, s, e, h\}, i_1)$	$\mathcal{M}_2 = (\{1, 2, 3\}, i_2)$	$\mathcal{M}_3 = (\{kj, s\}, i_3)$
$i_1(\text{Kim}) = k$	$i_2(\text{Kim}) = 1$	$i_3(\text{Kim}) = kj$
$i_1(\text{Jim}) = j$	$i_2(\text{Jim}) = 2$	$i_3(\text{Jim}) = kj$
$i_1(\text{Sarah}) = s$	$i_2(\text{Sarah}) = 3$	$i_3(\text{Sarah}) = s$
$i_1(\text{príde}) = \{k, j, e\}$	$i_2(\text{príde}) = \{1, 2\}$	$i_3(\text{príde}) = \{kj\}$

Líšia sa doménami aj v interpretáciách.

Líšia sa v pravdivosti rovnostných atómov, napr.  $\text{Kim} \doteq \text{Jim}$ .

Zhodujú sa na pravdivosti všetkých predikátových atómov  $\text{príde}(\text{Kim})$ ,  $\text{príde}(\text{Jim})$ ,  $\text{príde}(\text{Sarah})$ .

💡 V  $T_{\text{party}}$  na ničom inom nezáleží.

## Ohodnotenie atómov

Z každej zo štruktúr

$\mathcal{M}_1 = (\{k, j, s, e, h\}, i_1)$	$\mathcal{M}_2 = (\{1, 2, 3\}, i_2)$	$\mathcal{M}_3 = (\{kj, s\}, i_3)$
$i_1(\text{Kim}) = k$	$i_2(\text{Kim}) = 1$	$i_3(\text{Kim}) = kj$
$i_1(\text{Jim}) = j$	$i_2(\text{Jim}) = 2$	$i_3(\text{Jim}) = kj$
$i_1(\text{Sarah}) = s$	$i_2(\text{Sarah}) = 3$	$i_3(\text{Sarah}) = s$
$i_1(\text{príde}) = \{k, j, e\}$	$i_2(\text{príde}) = \{1, 2\}$	$i_3(\text{príde}) = \{kj\}$

môžeme skonštruovať to isté *ohodnotenie predikátových atómov*:

$v(\text{príde}(\text{Kim})) = t$	lebo $\mathcal{M}_j \models \text{príde}(\text{Kim})$ ,
$v(\text{príde}(\text{Jim})) = t$	lebo $\mathcal{M}_j \models \text{príde}(\text{Jim})$ ,
$v(\text{príde}(\text{Sarah})) = f$	lebo $\mathcal{M}_j \not\models \text{príde}(\text{Sarah})$ .

Všetky tieto štruktúry (a nekonečne veľa ďalších) vieme pri vyhodnocovaní formúl jazyka  $\mathcal{L}_{\text{party}}$  nahradiť týmto ohodnotením.

## Výrokovologické formuly, teórie a ohodnotenia

**Definícia 2.30.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu.

Množinu všetkých predikátových atómov jazyka  $\mathcal{L}$  označujeme  $\mathcal{PA}_{\mathcal{L}}$ .

*Výrokovologickými formulami* jazyka  $\mathcal{L}$  nazveme všetky formuly jazyka  $\mathcal{L}$ , ktoré *neobsahujú symbol rovnosti*. Množinu všetkých výrokovologických formúl jazyka  $\mathcal{L}$  označujeme  $\mathcal{PE}_{\mathcal{L}}$ .

**Definícia 2.31.** Nech  $(f, t)$  je usporiadaná dvojica *pravdivostných hodnôt*,  $f \neq t$ , kde  $f$  predstavuje *nepravdu* a  $t$  predstavuje *pravdu*. Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu.

*Výrokovologickým ohodnotením* pre  $\mathcal{L}$ , skrátene *ohodnotením*, nazveme každé zobrazenie  $v : \mathcal{PA}_{\mathcal{L}} \rightarrow \{f, t\}$ .

## Pravdivé formuly v ohodnutí

Ako vyhodnotíme, či je formula pravdivá v nejakom ohodnutí?

**Definícia 2.32.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu, nech  $(f, t)$  sú pravdivostné hodnoty a nech  $v : \mathcal{PA}_{\mathcal{L}} \rightarrow \{f, t\}$  je výrokovologické ohodnotenie pre  $\mathcal{L}$ . Reláciu *výrokovologická formula  $A$  je pravdivá v ohodnotení  $v$*  ( $v \models_p A$ ) definujeme *induktívne* pre všetky predikátové atómy  $a$  a všetky výrokovologické formuly  $A, B$  jazyka  $\mathcal{L}$  nasledovne:

- $v \models_p a$  vtt  $v(a) = t$ ,
- $v \models_p \neg A$  vtt  $v \not\models_p A$ ,
- $v \models_p (A \wedge B)$  vtt  $v \models_p A$  a zároveň  $v \models_p B$ ,
- $v \models_p (A \vee B)$  vtt  $v \models_p A$  alebo  $v \models_p B$ ,
- $v \models_p (A \rightarrow B)$  vtt  $v \not\models_p A$  alebo  $v \models_p B$ ,

kde vtt skrakuje *vtedy a len vtedy* a  $v \not\models_p A$  skrakuje  *$A$  nie je pravdivá vo  $v$* .

### Vyhodnotenie formuly v ohodnotení

*Príklad 2.33.* Vyhodnoťme formulu

$$X = ((\text{príde}(\text{Jim}) \vee \neg \text{príde}(\text{Kim})) \rightarrow \text{príde}(\text{Sarah}))$$

vo výrokovologickom ohodnotení

$$v = \{\text{príde}(\text{Kim}) \mapsto t, \text{príde}(\text{Jim}) \mapsto t, \text{príde}(\text{Sarah}) \mapsto f\}$$

zdola nahor:

	p(Kim)	p(Jim)	p(Sarah)	$\neg p(\text{Kim})$	$(p(\text{Jim}) \vee \neg p(\text{Kim}))$	$X$
$v$	$\models_p$	$\models_p$	$\not\models_p$	$\not\models_p$	$\models_p$	$\not\models_p$

príde sme skrátili na p.

### Ohodnotenie zhodné so štruktúrou

**Definícia 2.34.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu, nech  $\mathcal{M}$  je štruktúra pre  $\mathcal{L}$ , nech  $(f, t)$  sú pravdivostné hodnoty,  $v : \mathcal{PA}_{\mathcal{L}} \rightarrow \{f, t\}$  je výrokovologické ohodnotenie pre  $\mathcal{L}$  a  $S \subseteq \mathcal{PA}_{\mathcal{L}}$  je množina predikátových atómov.

Ohodnotenie  $v$  a štruktúra  $\mathcal{M}$  sú navzájom *zhodné na  $S$*  vtt pre každý predikátový atóm  $A \in S$  platí

$$v(A) = t \text{ vtt } \mathcal{M} \models A.$$

Ohodnotenie  $v$  a štruktúra  $\mathcal{M}$  sú navzájom *zhodné* vtt sú zhodné na  $\mathcal{PA}_{\mathcal{L}}$ .

### Konštrukcia ohodnotenia zhodného so štruktúrou

Ohodnotenie zhodné so štruktúrou zostrojíme ľahko:

**Tvrdenie 2.35.** *Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu, nech  $\mathcal{M}$  je štruktúra pre  $\mathcal{L}$  a  $(f, t)$  sú pravdivostné hodnoty. Zobrazenie  $v : \mathcal{PA}_{\mathcal{L}} \rightarrow \{f, t\}$  definované pre každý atóm  $A \in \mathcal{PA}_{\mathcal{L}}$  nasledovne:*

$$v(A) = \begin{cases} t, & \text{ak } \mathcal{M} \models A, \\ f, & \text{ak } \mathcal{M} \not\models A \end{cases}$$

*je výrokovologické ohodnotenie zhodné s  $\mathcal{M}$ .*

*Dôkaz.* Pre každý atóm  $A \in \mathcal{PA}_{\mathcal{L}}$  musíme dokázať, že  $v(A) = t$  vtt  $\mathcal{M} \models A$ :

( $\Leftarrow$ ) Priamo: Ak  $\mathcal{M} \models A$ , tak  $v(A) = t$  podľa jeho definície v leme.

( $\Rightarrow$ ) Nepriamo: Ak  $\mathcal{M} \not\models A$ , tak  $v(A) = f$  podľa jeho definície v leme, a pretože  $t \neq f$ , tak  $v(A) \neq t$ . □

Dokážeme zostrojiť aj štruktúru z ohodnotenia, aby boli zhodné?

**Príklad 2.36** (Konštrukcia štruktúry zhodnej s ohodnotením). Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu, kde  $\mathcal{C}_{\mathcal{L}} = \{\text{Kim, Jim, Sarah}\}$  a  $\mathcal{P}_{\mathcal{L}} = \{\text{príde}\}$ .

Nech  $v$  je výrokovologické ohodnotenie pre  $\mathcal{L}$ , kde

$$v(\text{príde}(\text{Kim})) = t \quad v(\text{príde}(\text{Jim})) = t \quad v(\text{príde}(\text{Sarah})) = f$$

Zostrojme štruktúru pre  $\mathcal{L}$  zhodnú s  $v$ .

Možnosťou, ktorú ľahko zovšeobecníme na všetky jazyky, je použiť ako doménu množinu konštánt:

$$\mathcal{M} = (\underbrace{\{\text{Kim, Jim, Sarah}\}}_{\mathcal{C}_{\mathcal{L}}}, i)$$

Každú konštantu interpretujeme ňou samou:

$$i(\text{Kim}) = \text{Kim} \qquad i(\text{Jim}) = \text{Jim} \qquad i(\text{Sarah}) = \text{Sarah}$$

predikát príde ako množinu tých  $c$ , pre ktoré  $v(\text{príde}(c)) = t$ :

$$i(\text{príde}) = \{\text{Kim}, \text{Jim}\}$$

### Konštrukcia štruktúry zhodnej s ohodnotením

Ako zostrojíme štruktúru zhodnú s ohodnotením pre hocikajký jazyk?

**Tvrdenie 2.37.** *Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu, nech  $(f, t)$  sú pravdivostné hodnoty a  $v : \mathcal{PA}_{\mathcal{L}} \rightarrow \{f, t\}$  je výrokovologické ohodnotenie pre  $\mathcal{L}$ .*

*Nech  $\mathcal{M} = (D, i)$  je štruktúra pre  $\mathcal{L}$  s doménou  $D = \mathcal{C}_{\mathcal{L}}$  a interpretačnou funkciou definovanou pre všetky  $n > 0$ , všetky konštanty  $c$  a všetky predikátové symboly  $P \in \mathcal{P}_{\mathcal{L}}$  s aritou  $n$  takto:*

$$i(c) = c$$

$$i(P) = \{(c_1, \dots, c_n) \in \mathcal{C}_{\mathcal{L}}^n \mid v(P(c_1, \dots, c_n)) = t\}$$

*Potom  $\mathcal{M}$  je zhodná s  $v$ .*

Štruktúram zo syntaktického materiálu sa hovorí *herbrandovské*.

Zhoda ohodnotenia a štruktúry je definované iba na *atómoch*.

Ako sa správajú na *zložitejších* formulách?

### Zhoda na všetkých výrokovologických formulách

**Tvrdenie 2.38.** *Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu,  $\mathcal{M}$  je štruktúra pre  $\mathcal{L}$  a  $v$  je výrokovologické ohodnotenie pre  $\mathcal{L}$  zhodné s  $\mathcal{M}$ . Potom pre každú výrokovologickú formulu  $X \in \mathcal{PE}_{\mathcal{L}}$  platí, že  $v \models_p X$  vtt  $\mathcal{M} \models X$ .*

*Dôkaz (indukciou na konštrukciu formuly).* 1.1: Nech  $X$  je rovnostný atóm. Potom nie je výrokovologickou formulou a tvrdenie preň triviálne platí.

1.2: Nech  $X$  je predikátový atóm. Potom  $v \models_p X$  vtt  $v(X) = t$  vtt  $\mathcal{M} \models X$  podľa def. zhodnosti  $v$  a  $\mathcal{M}$ .

2.1: Indukčný predpoklad: Nech tvrdenie platí pre formulu  $X$ . Dokážme tvrdenie pre  $\neg X$ . Ak  $X$  neobsahuje symbol rovnosti  $\doteq$ , potom  $v \models_p \neg X$  vtt (podľa def.  $\models_p$ )  $v \not\models_p X$  vtt (podľa IP)  $\mathcal{M} \not\models X$  vtt (podľa def.  $\models$ )  $\mathcal{M} \models \neg X$ . Ak  $X$  obsahuje  $\doteq$ ,  $\neg X$  ho obsahuje tiež, teda nie je výrokovologická a tvrdenie pre ňu platí triviálne.

2.2: IP: Nech tvrdenie platí pre formuly  $X$  a  $Y$ . Dokážme ho pre  $(X \wedge Y)$ ,  $(X \vee Y)$ ,  $(X \rightarrow Y)$ . Ak  $X$  alebo  $Y$  obsahuje  $\doteq$ , tvrdenie platí pre  $(X \wedge Y)$ ,  $(X \vee Y)$ ,  $(X \rightarrow Y)$  triviálne, lebo nie sú výrokovologické.

Nech teda  $X$  ani  $Y$  neobsahuje  $\doteq$ . Potom platí  $v \models_p (X \rightarrow Y)$  vtt  $v \not\models_p X$  alebo  $v \models_p Y$  vtt (podľa IP) vtt  $\mathcal{M} \not\models X$  alebo  $\mathcal{M} \models Y$  vtt  $\mathcal{M} \models (X \rightarrow Y)$ .

Ďalej  $v \models_p (X \wedge Y)$  vtt  $v \models_p X$  a  $v \models_p Y$  vtt (podľa IP) vtt  $\mathcal{M} \models X$  a  $\mathcal{M} \models Y$  vtt  $\mathcal{M} \models (X \wedge Y)$ .

Nakoniec  $v \models_p (X \vee Y)$  vtt  $v \models_p X$  alebo  $v \models_p Y$  vtt (podľa IP) vtt  $\mathcal{M} \models X$  alebo  $\mathcal{M} \models Y$  vtt  $\mathcal{M} \models (X \vee Y)$ . □

### 3. prednáška

## Výrokovologické vyplývanie, sémantické vlastnosti formúl a ekvivalencia

---

### Rekapitulácia

Minulý týždeň sme hovorili o tom,

- čo sú výrokovologické spojky,
- ako zodpovedajú slovenským spojkám,
- čo sú symboly jazyka výrokovologickej časti logiky prvého rádu,
- čo sú formuly tohto jazyka,
- kedy sú formuly pravdivé v danej štruktúre.
- čo je výrokovologická teória a jej model,
- ako zjednodušíme štruktúry na výrokovologické ohodnotenia.

## 3 Výrokovologické vyplývanie

### Logické dôsledky

Na 1. prednáške:

- Hovorili sme o tom, že logiku zaujíma, čo a prečo sú zákonitosti správneho usudzovania.
- Správne úsudky odvodzujú z predpokladov (teórií) závery, ktoré sú ich logickými dôsledkami.
- *Logickými dôsledkami* teórie sú tvrdenia, ktoré sú pravdivé vo *všetkých modeloch* teórie.

Minulý týždeň sme začali pracovať s *výrokovologickou* časťou logiky prvého rádu.

Už vieme, čo sú v nej teórie a modely.

Čo sú logické dôsledky?

### 3.1 Výrokovologické teórie a modely

#### Výrokovologické teórie

Vráťme sa naspäť k teóriám, modelom a vyplývaniu.

**Definícia 3.1.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Každú množinu výrokovologických formúl jazyka  $\mathcal{L}$  budeme nazývať *výrokovologickou teóriou* v jazyku  $\mathcal{L}$ .

*Príklad 3.2.* Výrokovologickou teóriou je

$$\begin{aligned} T_{\text{party}} = \{ & ((\text{príde}(\text{Kim}) \vee \text{príde}(\text{Jim})) \vee \text{príde}(\text{Sarah})), \\ & (\text{príde}(\text{Kim}) \rightarrow \neg \text{príde}(\text{Sarah})), \\ & (\text{príde}(\text{Jim}) \rightarrow \text{príde}(\text{Kim})), \\ & (\text{príde}(\text{Sarah}) \rightarrow \text{príde}(\text{Jim})) \}, \end{aligned}$$

ale nie

$$T_{\text{party}} \cup \{\text{Kim} \doteq \text{Sarah}\}.$$

#### Príklad výrokovologického modelu

*Príklad 3.3* (Výrokovologický model teórie o party).

$$\left. \begin{aligned} v &= \{\text{príde}(\text{Kim}) \mapsto t, \text{príde}(\text{Jim}) \mapsto t, \text{príde}(\text{Sarah}) \mapsto f\} \\ v &\models_p ((\text{príde}(\text{Kim}) \vee \text{príde}(\text{Jim})) \vee \text{príde}(\text{Sarah})) \\ v &\models_p (\text{príde}(\text{Kim}) \rightarrow \neg \text{príde}(\text{Sarah})) \\ v &\models_p (\text{príde}(\text{Jim}) \rightarrow \text{príde}(\text{Kim})) \\ v &\models_p (\text{príde}(\text{Sarah}) \rightarrow \text{príde}(\text{Jim})) \end{aligned} \right\} v \models_p T_{\text{party}}$$



## Výrokovologický model

**Definícia 3.4** (Výrokovologický model). Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu a nech  $T$  je teória v jazyku  $\mathcal{L}$  a  $v$  je výrokovologické ohodnotenie pre jazyk  $\mathcal{L}$ .

Teória  $T$  je *pravdivá* v ohodnotení  $v$ , skráteno  $v \models_p T$ , vtt každá formula  $X$  z  $T$  je pravdivá vo  $v$  (teda  $v \models_p X$  pre každú  $X \in T$ ).

Hovoríme tiež, že  $v$  je *výrokovologickým modelom*  $T$ .

Teória  $T$  je *nepravdivá* vo  $v$ , skráteno  $v \not\models_p T$ , vtt  $T$  nie je pravdivá vo  $v$ .

Zrejme  $v \models_p T$  vtt  $v \not\models_p X$  pre *nejakú*  $X \in T$ .

## Model teórie, splniteľnosť a nespľniteľnosť

**Definícia 3.5** (Splniteľnosť a nespľniteľnosť). Teória je *výrokovologicky splniteľná* vtt má aspoň jeden výrokovologický model.

Teória je *výrokovologicky nespľniteľná* vtt nemá žiaden výrokovologický model.

Zrejme teória nie je splniteľná vtt keď je nespľniteľná.

*Príklad 3.6.*  $T_{\text{party}}$  je evidentne splniteľná.

## 3.2 Vyplývanie, nezávislosť a nespľniteľnosť

### Výrokovologické vyplývanie

Ak sú množiny konštánt a predikátových symbolov jazyka konečné, jazyk má konečne veľa predikátových atómov a teda aj *konečne veľa* ohodnotení.

Uvažovať o všetkých ohodnoteniach a modeloch teórie nie je také odstrašujúce. Napríklad si ľahšie predstavíme logický dôsledok:

**Definícia 3.7.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu a nech  $T$  je výrokovologická teória a  $X$  je výrokovologická formula, obe v jazyku  $\mathcal{L}$ .

Formula  $X$  je *výrokovologickým dôsledkom* teórie  $T$  vtt pre každé ohodnotenie  $v$  pre jazyk  $\mathcal{L}$  platí, že ak  $v \models_p T$ , tak  $v \models_p X$ .

Hovoríme tiež, že  $X$  *vyplýva* z  $T$  a píšeme  $T \models_p X$ .

Ak  $X$  *nevyplýva* z  $T$ , píšeme  $T \not\models_p X$ .

### Príklad výrokovologickeho vyplývania

*Príklad 3.8.* Vyplýva príde(Kim) výrokovologicky z  $T_{\text{party}}$ ? Pretože vieme vymenovať všetky ohodnotenia pre  $\mathcal{L}_{\text{party}}$ , zistíme to ľahko:

	$v_i$			$((p(K) \vee p(J)) \vee p(S))$	$(p(K) \rightarrow \neg p(S))$	$(p(J) \rightarrow p(K))$	$(p(S) \rightarrow p(J))$	$T_{\text{party}}$	$p(K)$
	$p(K)$	$p(J)$	$p(S)$						
$v_0$	<i>f</i>	<i>f</i>	<i>f</i>	$\not\vdash_p$				$\not\vdash_p$	
$v_1$	<i>f</i>	<i>f</i>	<i>t</i>	$\vdash_p$	$\vdash_p$	$\vdash_p$	$\not\vdash_p$	$\not\vdash_p$	
$v_2$	<i>f</i>	<i>t</i>	<i>f</i>	$\vdash_p$	$\vdash_p$	$\not\vdash_p$		$\not\vdash_p$	
$v_3$	<i>f</i>	<i>t</i>	<i>t</i>	$\vdash_p$	$\vdash_p$	$\not\vdash_p$		$\not\vdash_p$	
$v_4$	<i>t</i>	<i>f</i>	<i>f</i>	$\vdash_p$	$\vdash_p$	$\vdash_p$	$\vdash_p$	$\vdash_p$	$\vdash_p$
$v_5$	<i>t</i>	<i>f</i>	<i>t</i>	$\vdash_p$	$\not\vdash_p$			$\not\vdash_p$	
$v_6$	<i>t</i>	<i>t</i>	<i>f</i>	$\vdash_p$	$\vdash_p$	$\vdash_p$	$\vdash_p$	$\vdash_p$	$\vdash_p$
$v_7$	<i>t</i>	<i>t</i>	<i>t</i>	$\vdash_p$	$\not\vdash_p$			$\not\vdash_p$	

Skrátili sme príde na *p*, Kim na *K*, Jim na *J*, Sarah na *S*.

*Logický záver:* Formula príde(Kim) výrokovologicky vyplýva z  $T_{\text{party}}$ .

*Praktický záver:* Aby boli všetky požiadavky splnené, Kim *musí* prísť na párty.

### Príklad nezávislosti

*Príklad 3.9.* Vyplýva príde(Jim) výrokovologicky z  $T_{\text{party}}$ ?

	$v_i$			$((p(K) \vee p(J)) \vee p(S))$	$(p(K) \rightarrow \neg p(S))$	$(p(J) \rightarrow p(K))$	$(p(S) \rightarrow p(J))$	$T_{\text{party}}$	$p(J)$
	$p(K)$	$p(J)$	$p(S)$						
$v_0$	<i>f</i>	<i>f</i>	<i>f</i>	$\not\vdash_p$				$\not\vdash_p$	
$v_1$	<i>f</i>	<i>f</i>	<i>t</i>	$\vdash_p$	$\vdash_p$	$\vdash_p$	$\not\vdash_p$	$\not\vdash_p$	
$v_2$	<i>f</i>	<i>t</i>	<i>f</i>	$\vdash_p$	$\vdash_p$	$\not\vdash_p$		$\not\vdash_p$	
$v_3$	<i>f</i>	<i>t</i>	<i>t</i>	$\vdash_p$	$\vdash_p$	$\not\vdash_p$		$\not\vdash_p$	
$v_4$	<i>t</i>	<i>f</i>	<i>f</i>	$\vdash_p$	$\vdash_p$	$\vdash_p$	$\vdash_p$	$\vdash_p$	$\not\vdash_p$
$v_5$	<i>t</i>	<i>f</i>	<i>t</i>	$\vdash_p$	$\not\vdash_p$			$\not\vdash_p$	
$v_6$	<i>t</i>	<i>t</i>	<i>f</i>	$\vdash_p$	$\vdash_p$	$\vdash_p$	$\vdash_p$	$\vdash_p$	$\vdash_p$
$v_7$	<i>t</i>	<i>t</i>	<i>t</i>	$\vdash_p$	$\not\vdash_p$			$\not\vdash_p$	

*Logický záver:* Formula príde(Jim) *nevyplýva* z  $T_{\text{party}}$ .

## Výrokovologická nezávislosť

Vzťahu medzi  $\text{príde}(\text{Jim})$  a  $T_{\text{party}}$  hovoríme *nezávislosť*.

**Definícia 3.10.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu a nech  $T$  je výrokovologická teória a  $X$  je výrokovologická formula, obe v jazyku  $\mathcal{L}$ .

Formula  $X$  je *výrokovologicky nezávislá* od teórie  $T$  vtt existujú také ohodnotenia  $v_0$  a  $v_1$  pre jazyk  $\mathcal{L}$ , že  $v_0 \models_p T$  aj  $v_1 \models_p T$ , ale  $v_0 \not\models_p X$  a  $v_1 \models_p X$ .

*Príklad 3.11* (pokračovanie príkladu 3.9). *Logický záver*: Formula  $\text{príde}(\text{Jim})$  je *nezávislá* od  $T_{\text{party}}$ .

*Praktický záver*: Všetky požiadavky budú naplnené *bez ohľadu na to*, či Jim príde alebo nepríde na párty. *Nie je nutné*, aby bol prítomný ani aby bol neprítomný. *Môže, ale nemusí* prísť. Jeho prítomnosť od požiadaviek *nezávisí*.

## Príklad vyplývania negácie

*Príklad 3.12.* Je  $\text{príde}(\text{Sarah})$  výrokovologickým dôsledkom  $T_{\text{party}}$  alebo nezávislá od  $T_{\text{party}}$ ?

	$v_i$			$((p(K) \vee p(J)) \vee p(S))$	$(p(K) \rightarrow \neg p(S))$	$(p(J) \rightarrow p(K))$	$(p(S) \rightarrow p(J))$	$T_{\text{party}}$	$p(S)$
	$p(K)$	$p(J)$	$p(S)$						
$v_0$	<i>f</i>	<i>f</i>	<i>f</i>	$\not\models_p$				$\not\models_p$	
$v_1$	<i>f</i>	<i>f</i>	<i>t</i>	$\models_p$	$\models_p$	$\models_p$	$\not\models_p$	$\not\models_p$	
$v_2$	<i>f</i>	<i>t</i>	<i>f</i>	$\models_p$	$\models_p$	$\not\models_p$		$\not\models_p$	
$v_3$	<i>f</i>	<i>t</i>	<i>t</i>	$\models_p$	$\models_p$	$\not\models_p$		$\not\models_p$	
$v_4$	<i>t</i>	<i>f</i>	<i>f</i>	$\models_p$	$\models_p$	$\models_p$	$\models_p$	$\models_p$	$\not\models_p$
$v_5$	<i>t</i>	<i>f</i>	<i>t</i>	$\models_p$	$\not\models_p$			$\not\models_p$	
$v_6$	<i>t</i>	<i>t</i>	<i>f</i>	$\models_p$	$\models_p$	$\models_p$	$\models_p$	$\models_p$	$\not\models_p$
$v_7$	<i>t</i>	<i>t</i>	<i>t</i>	$\models_p$	$\not\models_p$			$\not\models_p$	

*Logický záver*: Formula  $\text{príde}(\text{Sarah})$  *nevyplýva* z  $T_{\text{party}}$ , ale ani *nie je nezávislá* od  $T_{\text{party}}$ .

## Vyplývanie negácie

**Tvrdenie 3.13.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu a nech  $T$  je *splniteľná* výrokovologická teória a  $X$  je výrokovologická formula, obe v jazyku  $\mathcal{L}$ .

Formula  $X$  nevyplýva z teórie  $T$  a nie je výrokologicky nezávislá od  $T$  vtt  $\neg X$  vyplýva z  $T$ .

**Príklad 3.14** (pokračovanie príkladu 3.12). *Logický záver:* Z  $T_{\text{party}}$  vyplýva  $\neg \text{príde}(\text{Sarah})$ .

*Praktický záver:* Aby boli všetky požiadavky naplnené, Sarah *nesmie* prísť na party.

### Vzťahy teórií a formúl

Medzi *ohodnotením* a *formulou* sú iba dva vzájomne výlučné vzťahy:

Buď  $v \models_p X$ , alebo  $v \not\models_p X$ .

Medzi *teóriou* a *formulou* je viac možných vzťahov:

	existuje $v$ také, že $v \models_p T$ a $v \models_p X$	pre všetky $v$ , ak $v \models_p T$ , tak $v \models_p X$
existuje $v$ také, že $v \models_p T$ a $v \not\models_p X$	$X$ je nezávislá od $T$ $T \not\models_p X$ a $T \not\models_p \neg X$	$T \models_p \neg X$ a $T \not\models_p X$
pre všetky $v$ , ak $v \models_p T$ , tak $v \models_p X$	$T \models_p X$ a $T \not\models_p \neg X$	$T$ je <i>nesplniteľná</i> $T \models_p X$ aj $T \models_p \neg X$

### Nesplniteľná teória

**Príklad 3.15.** Je teória  $T'_{\text{party}} = T_{\text{party}} \cup \{(\neg \text{príde}(\text{Sarah}) \rightarrow \neg \text{príde}(\text{Kim}))\}$  splniteľná?

	$v_i$			$((p(K) \vee p(J)) \vee p(S))$	$(p(K) \rightarrow \neg p(S))$	$(p(J) \rightarrow p(K))$	$(p(S) \rightarrow p(J))$	$(\neg p(S) \rightarrow \neg p(K))$	$T'_{\text{party}}$
	$p(K)$	$p(J)$	$p(S)$						
$v_0$	$f$	$f$	$f$	$\not\models_p$					$\not\models_p$
$v_1$	$f$	$f$	$t$	$\models_p$	$\models_p$	$\models_p$	$\not\models_p$		$\not\models_p$
$v_2$	$f$	$t$	$f$	$\models_p$	$\models_p$	$\not\models_p$			$\not\models_p$
$v_3$	$f$	$t$	$t$	$\models_p$	$\models_p$	$\not\models_p$			$\not\models_p$
$v_4$	$t$	$f$	$f$	$\models_p$	$\models_p$	$\models_p$	$\models_p$	$\not\models_p$	$\not\models_p$
$v_5$	$t$	$f$	$t$	$\models_p$	$\not\models_p$				$\not\models_p$
$v_6$	$t$	$t$	$f$	$\models_p$	$\models_p$	$\models_p$	$\models_p$	$\not\models_p$	$\not\models_p$
$v_7$	$t$	$t$	$t$	$\models_p$	$\not\models_p$				$\not\models_p$

*Logický záver:*  $T'_{\text{party}}$  je nesplniteľná, vyplýva z nej každá formula.

*Praktický záver:*  $T'_{\text{party}}$  nemá praktické dôsledky, lebo *nevypovedá o žiadnom stave sveta*. Na jej základe *nevieme rozhodnúť*, kto musí alebo nesmie ísť na párty.

### Vyplývanie a nesplniteľnosť

Nesplniteľnosť ale nie je neužitočná vlastnosť.

**Tvrdenie 3.16.** *Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu a nech  $T$  je splniteľná výrokovologická teória a  $X$  je výrokovologická formula, obe v jazyku  $\mathcal{L}$ .*

*Formula  $X$  výrokovologicky vyplýva z teórie  $T$  vtt  $T \cup \{X\}$  je výrokovologicky nesplniteľná.*

Podľa tohto tvrdenia sa rozhodnutie vyplývania dá *zredukovať* na rozhodnutie splniteľnosti.

Výrokovologickú splniteľnosť rozhoduje SAT solver.

### Množina atómov formuly a teórie

**Definícia 3.17.** *Množinu atómov  $\text{atoms}(X)$  formuly  $X \in \mathcal{E}_{\mathcal{L}}$  definujeme pre všetky formuly  $A, B \in \mathcal{E}_{\mathcal{L}}$  nasledovne:*

- $\text{atoms}(A) = \{A\}$ , ak  $A$  je atóm,
- $\text{atoms}(\neg A) = \text{atoms}(A)$ ,
- $\text{atoms}((A \wedge B)) = \text{atoms}((A \vee B)) = \text{atoms}((A \rightarrow B)) = \text{atoms}(A) \cup \text{atoms}(B)$ .

*Množinou atómov teórie  $T$  je*

$$\text{atoms}(T) = \bigcup_{X \in T} \text{atoms}(X).$$

## Ohodnotenia zhodné na atónoch teórie

**Definícia 3.18.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu, nech  $M \subseteq \mathcal{PA}_{\mathcal{L}}$ . Ohodnotenia  $v_1$  a  $v_2$  sa *zhodujú* na množine  $M$  vtt  $v_1(A) = v_2(A)$  pre každý atóm  $A \in M$ .

**Tvrdenie 3.19.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Pre každú výrokovologickú teóriu  $T$  a formulu  $X$  jazyka  $\mathcal{L}$  a všetky ohodnotenia  $v_1$  a  $v_2$ , ktoré zhodujú na množine  $\text{atoms}(T) \cup \text{atoms}(X)$  platí

- $v_1 \models_p T$  vtt  $v_2 \models_p T$ ,
- $v_1 \models_p X$  vtt  $v_2 \models_p X$ .

## Ohodnotenia postačujúce na skúmanie teórií

Inak povedané: Pravdivosť formuly/teórie v ohodnotení závisí *iba* od pravdivostných hodnôt tých atómov, ktoré sa v nej vyskytujú.

Takže na zistenie vyplývania, nezávislosti, splniteľnosti stačí preskúmať všetky ohodnotenia, ktoré sa *lišia* na atónoch *vyskytujúcich* sa vo formule a teórii.

Pokiaľ je teória je konečná, stačí skúmať konečne veľa ohodnotení, aj keby bol jazyk nekonečný. (A podľa vety o kompaktnosti ak máme nejaký spor v nekonečnej teórii, nájde sa sporná konečná podmnožina.)

## Dôkaz indukciou na konštrukciu formuly

- (1)  $X$  je výrokovologická formula jazyka  $\mathcal{L}$   
(2)  $v_1$  a  $v_2$  sú ohodnotenia zhodné na  $\text{atoms}(X)$   $\Downarrow v_1 \models_p X$  vtt  $v_2 \models_p X$

---

Báza:  $X$  je atóm.

- (3)  $X$  predikátový atóm      podľa 1  
(4)  $v_1 \models_p X$  vtt  $v_1(X) = t$       def. pravdivosti  
(5)  $v_2 \models_p X$  vtt  $v_2(X) = t$       def. pravdivosti  
(6)  $v_1(X) = v_2(X)$       podľa 2  
     $v_1 \models_p X$  vtt  $v_2 \models_p X$       podľa 4, 5, 6

## Dôkaz indukciou na konštrukciu formuly

- (1)  $Z$  je výrokovologická formula jazyka  $\mathcal{L}$
  - (2)  $v_1$  a  $v_2$  sú ohodnotenia zhodné na  $\text{atoms}(Z)$   $\Downarrow v_1 \models_p Z \text{ vtt } v_2 \models_p Z$
- 

Ind. krok pre  $\neg$ : Formula v tvare  $Z = \neg X$ .

- |      |  |   |
|------|--|---|
| (IP) | Tvrdenie platí pre $X$                                   |   |
| (3)  | $v_1, v_2$ sa zhodujú na $\text{atoms}(X)$               | 2, $\text{atoms}(\neg X) = \text{atoms}(X)$ |
| (4)  | $v_1 \models_p X \text{ vtt } v_2 \models_p X$           | 3, IP pre $Z = X$                           |
| (5)  | $v_1 \models_p \neg X \text{ vtt } v_1 \not\models_p X$  | def. $\models_p$                            |
| (6)  | $v_2 \models_p \neg X \text{ vtt } v_2 \not\models_p X$  | def. $\models_p$                            |
| (7)  | $v_1 \not\models_p X \text{ vtt } v_2 \not\models_p X$   | 4, def. $\not\models_p$                     |
|      | $v_1 \models_p \neg X \text{ vtt } v_2 \models_p \neg X$ | 5, 6, 7                                     |

## Dôkaz indukciou na konštrukciu formuly

- (1)  $Z$  je výrokovologická formula jazyka  $\mathcal{L}$
  - (2)  $v_1$  a  $v_2$  sú ohodnotenia zhodné na  $\text{atoms}(Z)$   $\Downarrow v_1 \models_p Z \text{ vtt } v_2 \models_p Z$
- 

Ind. krok pre  $\wedge$ : Formula v tvare  $Z = (X \wedge Y)$ .

- |      |  |                     |
|------|--|---------------------|
| (IP) | Tvrdenie platí pre $X$ aj pre $Y$  |                     |
| (3)  | $\text{atoms}((X \wedge Y)) = \text{atoms}(X) \cup \text{atoms}(Y)$                  | def. $\text{atoms}$ |
| (4)  | $v_1, v_2$ sa zhodujú na $\text{atoms}(X)$   | 2, 3                |
| (5)  | $v_1 \models_p X \text{ vtt } v_2 \models_p X$                                       | 4, IP pre $Z = X$   |
| (6)  | $v_1, v_2$ sa zhodujú na $\text{atoms}(Y)$   | 2, 3                |
| (7)  | $v_1 \models_p Y \text{ vtt } v_2 \models_p Y$                                       | 6, IP pre $Z = Y$   |
| (8)  | $v_1 \models_p (X \wedge Y) \text{ vtt } v_1 \models_p X \text{ a } v_1 \models_p Y$ | def. $\models_p$    |
| (9)  | $v_2 \models_p (X \wedge Y) \text{ vtt } v_2 \models_p X \text{ a } v_2 \models_p Y$ | def. $\models_p$    |
|      | $v_1 \models_p (X \wedge Y) \text{ vtt } v_2 \models_p (X \wedge Y)$                 | 5, 7, 8, 9          |

*Dôkaz tvrdenia 3.19 (ešte raz, vo vetách).* Tvrdenie dokážeme indukciou na konštrukciu formuly:

1.1. Ak  $X$  je rovnostný atóm, nie je výrokovologickou formulou a tvrdenie preň platí triviálne.

1.2. Nech  $X$  je predikátový atóm. Zoberme ľubovoľné ohodnotenia  $v_1$  a  $v_2$ , ktoré sa zhodujú na  $\text{atoms}(X)$ , teda na samotnom  $X$ . Podľa definície pravdivosti platí  $v_1 \models_p X \text{ vtt } v_1(X) = t \text{ vtt } v_2(X) = t \text{ vtt } v_2 \models_p X$ .

2.1 Indukčný predpoklad (IP): Predpokladajme, že tvrdenie platí pre formulu  $X$ . Dokážme ho pre  $\neg X$ . Zoberme ľubovoľné ohodnotenia  $v_1$  a  $v_2$ , ktoré sa zhodujú na  $\text{atoms}(\neg X)$ . Pretože  $\text{atoms}(\neg X) = \text{atoms}(X)$ ,  $v_1$  a  $v_2$  sa zhodujú na  $\text{atoms}(X)$ , a teda podľa IP  $v_1 \models_p X$  vtt  $v_2 \models_p X$ . Preto  $v_1 \models_p \neg X$  vtt (def.  $\models_p$ )  $v_1 \not\models_p X$  vtt (IP)  $v_2 \not\models_p X$  vtt (def.  $\models_p$ )  $v_2 \models_p \neg X$ .

2.2 Indukčný predpoklad (IP): Predpokladajme, že tvrdenie platí pre formulu  $X$  a  $Y$ . Dokážme ho pre  $(X \wedge Y)$ . Zoberme ľubovoľné ohodnotenia  $v_1$  a  $v_2$ , ktoré sa zhodujú na  $\text{atoms}((X \wedge Y))$ . Pretože  $\text{atoms}((X \wedge Y)) = \text{atoms}(X) \cup \text{atoms}(Y)$ ,  $v_1$  a  $v_2$  sa zhodujú na  $\text{atoms}(X)$ , a teda podľa IP  $v_1 \models_p X$  vtt  $v_2 \models_p X$ ; tiež sa zhodujú na  $\text{atoms}(Y)$ , a teda podľa IP  $v_1 \models_p Y$  vtt  $v_2 \models_p Y$ . Preto  $v_1 \models_p (X \wedge Y)$  vtt (def.  $\models_p$ )  $v_1 \models_p X$  a  $v_1 \models_p Y$  vtt (IP)  $v_2 \models_p X$  a  $v_2 \models_p Y$  vtt (def.  $\models_p$ )  $v_2 \models_p (X \wedge Y)$ .

Podobne postupujeme pre ďalšie binárne spojky. □

## 4 Sémantické vlastnosti a vzťahy formúl

### 4.1 Tautológie, splniteľné, falzifikovateľné a nesplniteľné formuly

#### Logické dôsledky prázdnej teórie

Tvrdenie vyplýva z nejakej teórie (je jej logickým dôsledkom), keď je pravdivé v každom modeli teórie, teda v každom stave sveta, v ktorom sú pravdivé všetky tvrdenia teórie.

Čo keď je teória *prázdna*?

- Je pravdivá v *každom* stave sveta.
- Jej logické dôsledky sú teda *tiež* pravdivé v každom stave sveta.

Navyše:

- Každý model hocijakej neprázdnej teórie  $T$  je aj modelom prázdnej teórie.
- Logické dôsledky prázdnej teórie sú v ňom pravdivé.
- Preto sú aj logickými dôsledkami  $T$ .

Logické dôsledky prázdnej teórie sú teda dôsledkami *všetkých* teórií.



## Príklady logických dôsledkov prázdnej teórie

*Existujú vôbec logické dôsledky prázdnej teórie?*

*Áno*, napríklad:

- pre každú konštantu  $c$  je pravdivé tvrdenie  $c \doteq c$ ;
- pre každý atóm  $A$  je pravdivé  $(A \vee \neg A)$ .

Pretože sú pravdivé bez ohľadu na teóriu a sú pravdivé v každom stave sveta, sú *logickými pravdami* a sú *nutne* pravdivé.

## Rozpoznateľné logické pravdy

Jazyk a spôsob pohľadu na stavy sveta ovplyvňuje, ktoré logické pravdy dokážeme rozpoznať:

- $c \doteq c$  aj  $(A \vee \neg A)$  sú pravdivé v každej štruktúre.
- Výrokovologické ohodnotenia sa nezaoberajú rovnostnými atómami. Pomocou nich nezistíme, že  $c \doteq c$  je nutne pravda. Ale zistíme, že  $(A \vee \neg A)$  pre každý *predikátový* atóm  $A$  je pravdivé v každom ohodnotení, a teda je nutne pravdou.

Logickým pravdám, ktorých nutnú pravdivosť dokážeme určiť rozborom všetkých výrokovologických ohodnotení, hovoríme *tautológie*.

## Príklad tautológie

*Príklad 4.1* (Peirceov zákon). Majme jazyk  $\mathcal{L}$  s  $\mathcal{C}_{\mathcal{L}} = \{a, b\}$ ,  $\mathcal{P}_{\mathcal{L}} = \{p^1\}$ . Je formula  $X = (((p(a) \rightarrow p(b)) \rightarrow p(a)) \rightarrow p(a))$  tautológiou?

Označme  $A = p(a)$  a  $B = p(b)$ , teda  $X = (((A \rightarrow B) \rightarrow A) \rightarrow A)$  a preskúmame všetky výrokovologické ohodnotenia týchto atómov:

$v_i$		$X$			
$A$	$B$	$(A \rightarrow B)$	$((A \rightarrow B) \rightarrow A)$	$((((A \rightarrow B) \rightarrow A) \rightarrow A)$	
$v_0$	$f$	$f$	$\models_p$	$\not\models_p$	$\models_p$
$v_1$	$f$	$t$	$\models_p$	$\not\models_p$	$\models_p$
$v_2$	$t$	$f$	$\not\models_p$	$\models_p$	$\models_p$
$v_3$	$t$	$t$	$\models_p$	$\models_p$	$\models_p$

Pretože  $X$  je pravdivá vo všetkých ohodnoteniach pre  $\mathcal{L}$ ,  $X$  je tautológiou.

## Tautológia

**Definícia 4.2.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Nech  $X$  je výrokovologická formula. Formulu  $X$  nazveme *tautológiou* (skrátene  $\models_{\mathcal{L}} X$ ) vtt  $X$  je *pravdivá v každom* výrokovologickom ohodnotení  $v$  pre  $\mathcal{L}$  (teda *pre každé* výrokovologické ohodnotenie  $v$  pre  $\mathcal{L}$  platí  $v \models_{\mathcal{L}} X$ ).

$v_i$				
	$A_1$	$A_2$	$\dots$	$X$
$v_0$	$f$	$f$	$\dots$	$\models_{\mathcal{L}} X$
$v_1$	$f$	$f$	$\dots$	$\models_{\mathcal{L}} X$
		$\dots$		
$v_k$	$t$	$f$	$\dots$	$\models_{\mathcal{L}} X$
		$\dots$		

Definícia vyžaduje preveriť *všetky možné* ohodno-

tenia, ale podľa 3.19 stačí preverovať *ohodnotenia atómov vyskytujúcich sa vo formule*:

**Dôsledok 4.3.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu a nech  $X$  je výrokovologická formula jazyka  $\mathcal{L}$ . Formula  $X$  je tautológiou vtt  $X$  je pravdivá v každom výrokovologickom ohodnotení  $v : \text{atoms}(X) \rightarrow \{f, t\}$ .

## Tautológie a vyplývanie

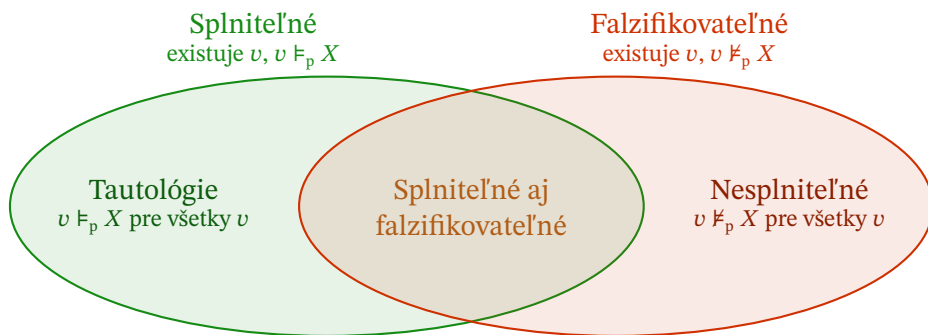
**Tvrdenie 4.4** (Tautológie, vyplývanie a jeho monotónnosť). Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Nech  $A$  je výrokovologická formula v  $\mathcal{L}$ . Nech  $T_1$  a  $T_2$  sú výrokovologické teórie v  $\mathcal{L}$ . Potom:

- a)  $\models_{\mathcal{L}} A$  ( $A$  je tautológia) vtt  $\emptyset \models_{\mathcal{L}} A$  ( $A$  vyplýva z prázdnej teórie).
- b)  $T_1 \models_{\mathcal{L}} A$  a  $T_1 \subseteq T_2$ , tak  $T_2 \models_{\mathcal{L}} A$ .
- c)  $\models_{\mathcal{L}} A$  vtt pre každú teóriu  $T$  v  $\mathcal{L}$ ,  $T \models_{\mathcal{L}} A$ .

Vlastnosti b) hovoríme *monotónnosť*: pridávaním ďalších formúl do teórie nemožno zrušiť platnosť jej dôsledkov.

## „Geografia“ formúl podľa pravdivosti vo všetkých ohodnoteniach

Tautológie (vždy pravdivé) a kontradikcie (vždy nepravdivé) sú „výnimočné“ formuly. Väčšina formúl je kdesi medzi týmito extrémami.



Obrázok podľa [Papadimitriou \[1994\]](#)

### Splniteľnosť

Kým tautológie sú *nutne* pravdivé, teda pravdivé vo *všetkých* ohodnoteniach, mnohé formuly iba *môžu* byť pravdivé, teda sú pravdivé v *niektorých* ohodnoteniach.

Nazývame ich *splniteľné*.

**Definícia 4.5.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Nech  $X$  je výrokovologická formula. Formulu  $X$  nazveme *splniteľnou* vtt  $X$  je *pravdivá* v *nejakom* výrokovologickom ohodnotení pre  $\mathcal{L}$  (teda *existuje* také výrokovologické ohodnotenie  $v$  pre  $\mathcal{L}$ , že  $v \models_p X$ ).

	$v_i$			
	$A_1$	$A_2$	$\dots$	$X$
$v_0$	$f$	$f$	$\dots$	$\not\models_p$
$v_1$	$f$	$f$	$\dots$	$\not\models_p$
		$\dots$		
$v_k$	$t$	$f$	$\dots$	$\models_p$
		$\dots$		

## Falzifikovateľnosť

Na rozdiel od tautológií, ktoré sú *nutne* pravdivé, a teda *nemôžu* byť *nepravdivé*, mnohé formuly *môžu* byť *nepravdivé*, teda sú *nepravdivé* v *niektorých* ohodnoteniach.

Nazývame ich *falzifikovateľné*.

**Definícia 4.6.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Nech  $X$  je výrokovologická formula. Formulu  $X$  nazveme *falzifikovateľnou* vtt  $X$  je *nepravdivá* v *nejakom* výrokovologickom ohodnotení pre  $\mathcal{L}$  (teda *existuje* také výrokovologické ohodnotenie  $v$  pre  $\mathcal{L}$ , že  $v \not\models_p X$ ).

	$v_i$			
	$A_1$	$A_2$	$\dots$	$X$
$v_0$	$f$	$f$	$\dots$	$\models_p$
$v_1$	$f$	$f$	$\dots$	$\models_p$
		$\dots$		
$v_k$	$t$	$f$	$\dots$	$\not\models_p$
		$\dots$		

## Nesplniteľnosť

Nakoniec, mnohé formuly sú *nutne* *nepravdivé*, teda sú *nepravdivé* vo *všetkých* ohodnoteniach.

Nazývame ich *nesplniteľné*.

**Definícia 4.7.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Nech  $X$  je výrokovologická formula. Formulu  $X$  nazveme *nesplniteľnou* vtt  $X$  je *nepravdivá* v *každom* výrokovologickom ohodnotení pre  $\mathcal{L}$  (teda pre *každé* výrokovologické ohodnotenie  $v$  pre  $\mathcal{L}$ , platí  $v \not\models_p X$ ).

	$v_i$			
	$A_1$	$A_2$	$\dots$	$X$
$v_0$	$f$	$f$	$\dots$	$\not\models_p$
$v_1$	$f$	$f$	$\dots$	$\not\models_p$
		$\dots$		
$v_k$	$t$	$f$	$\dots$	$\not\models_p$
		$\dots$		

## 4.2 Hľadanie ohodnotení

### Ohodnotenia pre danú formulu

Ak máme dané ohodnotenie atómov, pravdivosť formuly zistíme indukciou podľa jej štruktúry. Čo však spraviť, ak naopak máme danú formulu a hľadáme ohodnotenia, v ktorých je pravdivá či nepravdivá?

- Ak by sme hľadali všetky štruktúry, nemáme šancu, je ich nekonečne veľa.
- Všetkých vyhovujúcich ohodnotení môže byť exponenciálne veľa (nemôže teda existovať rýchly všeobecný algoritmus).
- Ohodnotenia možno hľadať rozborom všetkých možností pomocou tabuľky, ktorá má v záhlaví vytvárajúcu postupnosť danej formuly a v riadkoch jednotlivé ohodnotenia.
- Mohli by sme lepšie využiť znalosť štruktúry formuly na urýchlenie postupu? Azda áno:  $A \wedge \neg A \wedge B_1 \wedge B_2 \wedge \dots \wedge B_{100}$ .

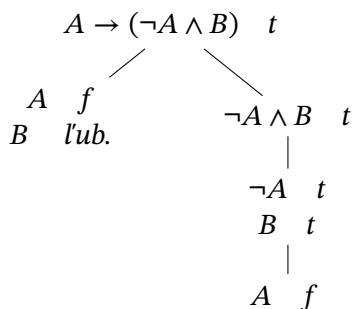
### Ohodnotenia pre danú formulu

Pre formulu so známou pravdivostnou hodnotou vieme určiť pravdivosť priamych podformúl. Napr. ak  $A \vee B$  je nepravda, tak  $A$  aj  $B$  musia byť nepravdivé.

Možností môže byť viac, čo zodpovedá vetveniu. Napr. pre pravdivú implikáciu  $A \rightarrow B$  osobitne skúmame situácie, kde  $A$  je nepravdivá, a kde  $B$  je pravdivá.

Pri takejto postupnej analýze dostávame čoraz jednoduchšie formuly. Končíme, keď sa dostaneme k atómom: ich ohodnotenie presne popisuje, ako má model vyzeráť.

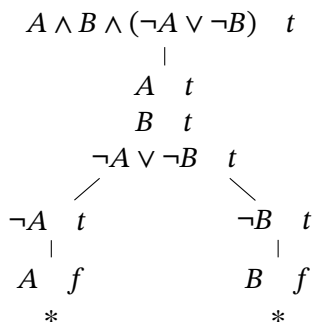
### Hľadanie ohodnotení



Ľavá vetva zodpovedá dvom ohodnoteniam, pravá jednému (avšak už obsiahnutému v ľavej vetve).

### Hľadanie ohodnotení

Môže sa stať, že v niektorej vetve vznikne *spor* (atóm  $A$  má byť zároveň pravdivý aj nepravdivý). Taká vetva neobsahuje žiadne vyhovujúce ohodnotenie a netreba sa ňou ďalej zapodievať.



Spor v každej vetve: formula nie je pravdivá v žiadnom ohodnotení.

### Hľadanie ohodnotení

$$\begin{array}{c}
 A \wedge \neg A \wedge B_1 \wedge B_2 \wedge \cdots \wedge B_n \quad t \\
 | \\
 \begin{array}{cc}
 A & t \\
 \neg A & t \\
 B_1 & t \\
 \cdots & \\
 B_n & t
 \end{array} \\
 | \\
 \begin{array}{cc}
 A & f
 \end{array} \\
 *
 \end{array}$$

Tabuľka všetkých možných ohodnotení by mala  $2^{n+1}$  riadkov, pritom táto úvaha je  $O(n)$ .

## 4.3 Ekvivalencia

### Logická ekvivalencia

Dve tvrdenia sú *ekvivalentné*, ak sú v každom stave sveta buď obe pravdivé alebo obe nepravdivé.

Ekvivalentné tvrdenia sú navzájom nahraditeľné. To je výhodné vtedy, keď potrebujeme, aby tvrdenie malo nejaký požadovaný tvar, alebo používalo iba niektoré spojky. Napríklad vstupom pre SAT solver je teória zložená iba z disjunkcií literálov.

Podobne ako pri tautológiách môžeme pomocou skúmania všetkých ohodnotení rozpoznať *niektoré* ekvivalentné tvrdenia zapísané formulami (ale nie všetky, pretože ohodnotenia napríklad nedávajú význam rovnostným atómom).

### Príklad výrokovologicke ekvivalentných formúl

*Príklad 4.8.* V jazyku  $\mathcal{L}$  z príkladu 4.1 označme  $A = p(a)$  a  $B = p(b)$ . Sú formuly  $X = \neg(A \rightarrow \neg B)$  a  $Y = (A \wedge B)$  výrokovologicke ekvivalentné?

Preskúmajme všetky výrokovologické ohodnotenia atómov  $A$  a  $B$ :


$v_i$				$X$	$Y$
$A$	$B$	$\neg B$	$(A \rightarrow \neg B)$	$\neg(A \rightarrow \neg B)$	$(A \wedge B)$
$v_0$	$f$	$f$	$\models_p$	$\models_p$	$\models_p$
$v_1$	$f$	$t$	$\models_p$	$\models_p$	$\models_p$
$v_2$	$t$	$f$	$\models_p$	$\models_p$	$\models_p$
$v_3$	$t$	$t$	$\models_p$	$\models_p$	$\models_p$

$X$  je pravdivá v *práve tých* ohodnoteniach pre  $\mathcal{L}$ , v ktorých je pravdivá  $Y$ , preto  $X$  a  $Y$  sú výrokovologicky ekvivalentné.

## Výrokovogická ekvivalencia

**Definícia 4.9.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Nech  $X$  a  $Y$  sú výrokovologické formuly jazyka  $\mathcal{L}$ . Formuly  $X$  a  $Y$  sú *výrokovologicky ekvivalentné*, skrátené  $X \Leftrightarrow_p Y$  vtt pre *každé* výrokovologické ohodnotenie  $v$  pre jazyk  $\mathcal{L}$  platí, že  $X$  je pravdivá vo  $v$  vtt  $Y$  je pravdivá vo  $v$ .

$\Leftrightarrow_p$  **verzus**  $\leftrightarrow$

 **Pozor!** Nemýľte si zápis  $X \Leftrightarrow_p Y$  s formulou  $(X \leftrightarrow Y)$ .

- $X \Leftrightarrow_p Y$  je skrátené vyjadrenie vzťahu dvoch formúl podľa definície 4.9. Keď napíšeme  $X \Leftrightarrow_p Y$ , tvrdíme tým, že  $X$  a  $Y$  sú výrokovologicky ekvivalentné formuly (alebo sa pýtame, či to tak je).
- $(X \leftrightarrow Y)$  je formula, postupnosť symbolov, ktorá môže byť pravdivá v nejakom ohodnotení a nepravdivá v inom, môže byť splniteľná, tautológia, falzifikovateľná, nespĺniteľná, môže vyplývať, či byť nezávislá od nejakej teórie, alebo môže byť výrokovologicky ekvivalentná s inou formulou.

Medzi  $X \Leftrightarrow_p Y$  a  $(X \leftrightarrow Y)$  je vzťah, ktorý si ozrejníme neskôr.

## Známe ekvivalencie

O mnohých dvojiciach formúl už viete, že sú vzájomne ekvivalentné. Zhrnuli sme ich do nasledujúcej vety.



**Veta 4.10.** *Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Nech  $A$ ,  $B$  a  $C$  sú ľubovoľné výrokovologické formuly jazyka  $\mathcal{L}$ . Potom:*

$(A \rightarrow B) \Leftrightarrow_p (\neg A \vee B)$	nahradenie $\rightarrow$
$(A \wedge (B \wedge C)) \Leftrightarrow_p ((A \wedge B) \wedge C)$	asociatívnosť $\wedge$
$(A \vee (B \vee C)) \Leftrightarrow_p ((A \vee B) \vee C)$	asociatívnosť $\vee$
$(A \wedge B) \Leftrightarrow_p (B \wedge A)$	komutatívnosť $\wedge$
$(A \vee B) \Leftrightarrow_p (B \vee A)$	komutatívnosť $\vee$
$(A \wedge (B \vee C)) \Leftrightarrow_p ((A \wedge B) \vee (A \wedge C))$	distributívnosť $\wedge$ cez $\vee$
$(A \vee (B \wedge C)) \Leftrightarrow_p ((A \vee B) \wedge (A \vee C))$	distributívnosť $\vee$ cez $\wedge$

**Veta 4.10** (pokračovanie).

$\neg(A \wedge B) \Leftrightarrow_p (\neg A \vee \neg B)$	de Morganove
$\neg(A \vee B) \Leftrightarrow_p (\neg A \wedge \neg B)$	zákony
$\neg\neg A \Leftrightarrow_p A$	zákon dvojitej negácie
$(A \wedge A) \Leftrightarrow_p A$	idempotencia pre $\wedge$
$(A \vee A) \Leftrightarrow_p A$	idempotencia pre $\vee$
$(A \wedge \top) \Leftrightarrow_p A$	identita pre $\wedge$
$(A \vee \perp) \Leftrightarrow_p A$	identita pre $\vee$
$(A \vee (A \wedge B)) \Leftrightarrow_p A$	absorpcia
$(A \wedge (A \vee B)) \Leftrightarrow_p A$	
$(A \vee \neg A) \Leftrightarrow_p \top$	vylúčenie tretieho ( <i>tertium non datur</i> )
$(A \wedge \neg A) \Leftrightarrow_p \perp$	spor,

kde  $\top$  je ľubovoľná tautológia a  $\perp$  je ľubovoľná nesplniteľná formula.

## Všeobecné dôkazy známych ekvivalencií

Pre *konkrétne* dvojice formúl v konkrétnom jazyku sa ekvivalencia dá dokázať rozborom všetkých ohodnotení ako v príklade 4.8.

Dôkaz ekvivalencie  $(A \rightarrow B)$  a  $(\neg A \vee B)$  pre ľubovoľné formuly  $A$  a  $B$  vyžaduje *opatrnější* postup.

Nemôžeme predpokladať, že  $A$  a  $B$  sú atomické a ohodnotenia im *priamo* priradujú pravdivostné hodnoty  $f$  a  $t$  (ak napr.  $A = (p(a) \wedge \neg p(a))$ , tak  $v(A)$  nie je definované, definované sú iba  $v(p(a))$  a  $v(p(b))$ ).

Môžeme však:

1. zobrať ľubovoľné ohodnotenie  $v$ ,
2. rozobrať všetky prípady, akými môžu byť  $A$  a  $B$  pravdivé alebo nepravdivé v tomto ohodnotení (teda  $v \models_p A$  a  $v \models_p B$ ,  $v \models_p A$  a  $v \not\models_p B$ ,  $v \not\models_p A$  a  $v \models_p B$ ,  $v \not\models_p A$  a  $v \not\models_p B$ )
3. a ukázať, že v každom prípade je  $(A \rightarrow B)$  pravdivá vo  $v$  vtt je  $(\neg A \vee B)$  pravdivá vo  $v$ .

*Príklad 4.11* (Dôkaz prvej ekvivalentnej dvojice z vety 4.10). Nech  $A$  a  $B$  sú ľubovoľné výrokovologické formuly v ľubovoľnom jazyku  $\mathcal{L}$ .

Nech  $v$  je ľubovoľné ohodnotenie pre  $\mathcal{L}$ . V tomto ohodnotení môže byť každá z formúl  $A$  a  $B$  buď pravdivá alebo nepravdivá, a teda môžu nastať nasledovné prípady:

- $v \not\models_p A$  a  $v \not\models_p B$ , vtedy  $v \models_p (A \rightarrow B)$  a  $v \models_p (\neg A \vee B)$ ;
- $v \not\models_p A$  a  $v \models_p B$ , vtedy  $v \models_p (A \rightarrow B)$  a  $v \models_p (\neg A \vee B)$ ;
- $v \models_p A$  a  $v \not\models_p B$ , vtedy  $v \not\models_p (A \rightarrow B)$  a  $v \not\models_p (\neg A \vee B)$ ;
- $v \models_p A$  a  $v \models_p B$ , vtedy  $v \models_p (A \rightarrow B)$  a  $v \models_p (\neg A \vee B)$ .

Rozobrali sme *všetky prípady* pravdivosti  $A$  a  $B$  v ohodnotení  $v$  a aj keď sa prípady od seba líšia pravdivosťou  $(A \rightarrow B)$  a  $(\neg A \vee B)$ , v *každom prípade* platí, že  $v \models_p (A \rightarrow B)$  vtt  $v \models_p (\neg A \vee B)$ . Preto môžeme konštatovať, že bez ohľadu na to, ktorý prípad nastáva, v ohodnotení  $v$  platí, že  $v \models_p (A \rightarrow B)$  vtt  $v \models_p (\neg A \vee B)$ .

Pretože ohodnotenie  $v$  bolo ľubovoľné, môžeme toto konštatovanie *zo-všeobecniť* na všetky ohodnotenia pre  $\mathcal{L}$  a podľa definície 4.9 sú  $(A \rightarrow B)$  a  $(\neg A \vee B)$  výrokovologicky ekvivalentné.

## Dôkazy rozborom prípadov

Rozbor prípadov z odrážkového zoznamu v predchádzajúcom dôkaze môžeme zapísať do *podobnej* tabuľky ako v príklade 4.8:

	$A$	$B$	$(A \rightarrow B)$	$(\neg A \vee B)$
$v$	$\not\models_p$	$\not\models_p$	$\not\models_p$	$\not\models_p$
$v$	$\not\models_p$	$\models_p$	$\models_p$	$\models_p$
$v$	$\models_p$	$\not\models_p$	$\not\models_p$	$\not\models_p$
$v$	$\models_p$	$\models_p$	$\models_p$	$\models_p$

Vždy ju však treba doplniť

1. úvodom o ľubovoľnom ohodnotení,
2. úvodom k rozboru prípadov,
3. záverom o všetkých prípadoch,
4. záverom o všetkých ohodnoteniach.

Podobne môžeme uvažovať o tautológiách, nesplniteľnosti, aj vyplývaní.

## 4.4 Vzťah tautológií, vyplývania a ekvivalencie

### Tautológie a vyplývanie

Tautológie nie sú zaujímavé iba preto, že sú logickými pravdami.

Kedy je formula  $((A_1 \wedge A_2) \rightarrow B)$  tautológia?

Vtedy, keď je pravdivá v každom ohodnotení, teda keď v každom ohodnotení  $v$  máme  $v \not\models_p (A_1 \wedge A_2)$  alebo  $v \models_p B$ , čiže keď v každom ohodnotení  $v$ , v ktorom  $v \models_p (A_1 \wedge A_2)$ , máme aj  $v \models_p B$  teda keď v každom ohodnotení  $v$ , v ktorom  $v \models_p A_1$  a  $v \models_p A_2$ , máme aj  $v \models_p B$ , teda keď z  $\{A_1, A_2\}$  výrokovologicky vyplýva  $B$ .

### Vzťahy výrokovologického vyplývania a tautológií

Pripomeňme, že podľa tvrdenia 4.4:  $\emptyset \models_p A$  vtt  $\models_p A$ .

**Tvrdenie 4.12** (Sémantická verzia vety o dedukcii). *Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Nech  $T$  je výrokovologická teória, nech  $A, B, C$  sú výrokovologické formuly v  $\mathcal{L}$ . Potom:*

a)  $T \cup \{A\} \models_p C$  vtt  $T \models_p (A \rightarrow C)$ .

b)  $T \cup \{A, B\} \models_p C$  vtt  $T \cup \{(A \wedge B)\} \models_p C$ .

**Dôsledok 4.13** (Redukcia vyplývania na tautológiu). *Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Nech  $A_1, A_2, \dots, A_n$  a  $C$  sú výrokovologické formuly v jazyku  $\mathcal{L}$ . Potom  $\{A_1, \dots, A_n\} \models_p C$  vtt  $\models_p (((\dots (A_1 \wedge A_2) \wedge \dots) \wedge A_n) \rightarrow C)$ .*

*Dôkaz tvrdenia 4.12.* a) Nech  $T$  je teória a  $A$  a  $C$  sú výrokovologické formuly v ľubovoľnom jazyku  $\mathcal{L}$ .

( $\Leftarrow$ ) Predpokladajme, že  $T \models_p (A \rightarrow C)$  a dokážme *priamo*, že z  $T \cup \{A\}$  vyplýva  $C$ .

Zoberme ľubovoľné výrokovologické ohodnotenie  $v$  pre  $\mathcal{L}$ , ktoré je modelom  $T \cup \{A\}$ . Vo  $v$  sú teda pravdivé všetky formuly z  $T \cup \{A\}$ . Preto  $v \models_p T$  a tiež  $v \models_p A$ .

Z  $v \models_p T$  na základe predpokladu  $T \models_p (A \rightarrow C)$  dostávame, že vo  $v$  je pravdivá implikácia  $(A \rightarrow C)$ , teda podľa definície pravdivosti  $v \models_p A$  alebo  $v \models_p C$ . Pretože ale vieme, že  $v \models_p A$ , musí  $v \models_p C$ .

Keďže  $v$  bol ľubovoľný model  $T \cup \{A\}$ , môžeme toto zistenie zovšeobecniť na všetky ohodnotenia a podľa definície vyplývania potom  $T \cup \{A\} \models_p C$ .

( $\Rightarrow$ ) Predpokladajme, že z  $T \cup \{A\}$  vyplýva  $C$  a dokážme *sporom*, že z  $T$  vyplýva  $(A \rightarrow C)$ .

Nech by existovalo ohodnotenie  $v$ , ktoré je modelom  $T$ , ale nie formuly  $(A \rightarrow C)$ , teda podľa definície pravdivosti  $v \models_p A$  a  $v \not\models_p C$ . Z  $v \models_p T$  a  $v \models_p A$  máme  $v \models_p T \cup \{A\}$  a z predpokladu  $T \cup \{A\} \models_p C$  dostávame  $v \models_p C$ , čo je spor.

b) Dôkaz je podobný ako v časti a). □

*Dôkaz dôsledku 4.13.* Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Nech  $A_1, A_2, \dots, A_n$  a  $C$  sú výrokovologické formuly v jazyku  $\mathcal{L}$ .

Opakovaným použitím tvrdenia 4.12 a pomocou 4.4 dostávame:

$$\begin{aligned}
 \{A_1, A_2, \dots, A_n\} \models_p C & \quad \text{vtt} \quad \{(A_1 \wedge A_2), \dots, A_n\} \models_p C \\
 & \quad \text{vtt} \quad \dots \\
 & \quad \text{vtt} \quad \emptyset \cup \{((\dots (A_1 \wedge A_2) \wedge \dots) \wedge A_n)\} \models_p C \\
 & \quad \text{vtt} \quad \emptyset \models_p (((\dots (A_1 \wedge A_2) \wedge \dots) \wedge A_n) \rightarrow C) \\
 & \quad \text{vtt} \quad \models_p (((\dots (A_1 \wedge A_2) \wedge \dots) \wedge A_n) \rightarrow C) \quad \square
 \end{aligned}$$

### Tautológie a ekvivalencia

Kedy je formula  $(X \leftrightarrow Y)$ , teda  $((X \rightarrow Y) \wedge (Y \rightarrow X))$  tautológia?

Vtedy a len vtedy, keď je pravdivá v každom ohodnotení, teda vtt v každom ohodnotení  $v$  máme  $v \models_p (X \rightarrow Y)$  a  $v \models_p (Y \rightarrow X)$ , vtt v každom ohodnotení  $v$  máme buď  $v \models_p X$  alebo  $v \models_p Y$  a zároveň buď  $v \models_p Y$  alebo  $v \models_p X$ , vtt v každom ohodnotení  $v$  platí, že ak  $v \models_p X$ , tak  $v \models_p Y$ , a ak  $v \models_p Y$ , tak  $v \models_p X$ , vtt v každom ohodnotení  $v$  máme  $v \models_p X$  vtt  $v \models_p Y$ , vtt  $X$  je výrokologicky ekvivalentná s  $Y$ .

**Tvrdenie 4.14.** *Nech  $\mathcal{L}$  je jazyk výrokologickej časti logiky prvého rádu. Nech  $X$  a  $Y$  sú výrokologické formuly v  $\mathcal{L}$ . Potom  $(X \leftrightarrow Y)$  je tautológia vtt  $X$  a  $Y$  sú výrokologicky ekvivalentné. (Skrátene:  $\models_p (X \leftrightarrow Y)$  vtt  $X \Leftrightarrow_p Y$ .)*

## 4.5 Ekvivalentné úpravy a CNF

### Reťazenie ekvivalentných úprav

Určite ste už robili ekvivalentné úpravy formúl, pri ktorých ste *reťazili dvojice* vzájomne ekvivalentných formúl:

$$\neg(A \rightarrow \neg B) \Leftrightarrow_p \neg(\neg A \vee \neg B) \Leftrightarrow_p (\neg\neg A \wedge \neg\neg B) \Leftrightarrow_p (A \wedge B)$$

a nakoniec ste prehlásili, že prvá  $\neg(A \rightarrow \neg B)$  a posledná formula  $(A \wedge B)$  sú ekvivalentné.

Mohli ste to urobiť, lebo  $\Leftrightarrow_p$  je *tranzitívna* relácia na formulách, dokonca viac než iba tranzitívna.

## Výrokovologická ekvivalencia ako relácia ekvivalencie

**Tvrdenie 4.15.** *Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu.*

*Vzťah výrokovologickej ekvivalencie  $\Leftrightarrow_p$  je reláciou ekvivalencie na výrokovologických formulách jazyka  $\mathcal{L}$ , teda pre všetky výrokovologické formuly  $X, Y, Z$  jazyka  $\mathcal{L}$  platí:*

- *Reflexivita:  $X \Leftrightarrow_p X$ .*
- *Symetria: Ak  $X \Leftrightarrow_p Y$ , tak  $Y \Leftrightarrow_p X$ .*
- *Tranzitivita: Ak  $X \Leftrightarrow_p Y$  a  $Y \Leftrightarrow_p Z$ , tak  $X \Leftrightarrow_p Z$ .*

*Dôkaz.* Priamym dôkazom dokážeme tranzitivitu. Ostatné vlastnosti sa dajú dokázať podobne.

Nech  $X, Y$  a  $Z$  sú výrokovologické formuly jazyka  $\mathcal{L}$ . Nech (1)  $X$  je výrokovologicky ekvivalentná s  $Y$  a (2)  $Y$  je ekvivalentná so  $Z$ .

Aby sme dokázali, že  $X$  je výrokovologicky ekvivalentná so  $Z$ , musíme ukázať, že pre každé ohodnotenie pre jazyk  $\mathcal{L}$  platí, že  $v \models_p X$  vtt  $v \models_p Z$ .

Nech teda  $v$  je ľubovoľné ohodnotenie pre  $\mathcal{L}$ .

- Ak  $v \models_p X$ , tak podľa predpokladu (1) a definície výrokovologickej ekvivalencie 4.9 musí platiť  $v \models_p Y$ , a teda podľa predpokladu (2) a definície ekvivalencie máme  $v \models_p Z$ .
- Nezávisle od toho, ak  $v \models_p Z$ , tak  $v \models_p Y$  podľa (2) a def. 4.9, a teda  $v \models_p X$  podľa (1) a def. 4.9.

Preto  $v \models_p X$  vtt  $v \models_p Z$ .

Pretože  $v$  bolo ľubovoľné, môžeme náš záver zovšeobecniť na všetky ohodnotenia, a teda podľa definície ekvivalencie 4.9 sú  $X$  a  $Z$  výrokovologicky ekvivalentné.  $\square$

## Substitúcia pri ekvivalentných úpravách

V reťazci ekvivalentných úprav

$$\begin{aligned}\neg(A \rightarrow \neg B) &\Leftrightarrow_p \neg(\neg A \vee \neg B) \Leftrightarrow_p (\neg\neg A \wedge \neg\neg B) \\ &\Leftrightarrow_p (A \wedge \neg\neg B) \Leftrightarrow_p (A \wedge B)\end{aligned}$$

v prvom, treťom a štvrtom kroku *nezodpovedá celá* formula niektorej zo známych ekvivalencií z vety 4.10.

Podľa známej ekvivalencie sme *nahrádzali podformuly* – *substituovali* sme ich.

**Definícia 4.16** (Substitúcia). Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu a nech  $X, A, B$  sú formuly jazyka  $\mathcal{L}$ . *Substitúciou*  $B$  za  $A$  v  $X$  (skrátene  $X[A|B]$ ) nazývame formulu, ktorá vznikne nahradením každého výskytu  $A$  v  $X$  formulou  $B$ .

### Substitúcia rekurzívne

Substitúciu si vieme predstaviť aj ako indukzívne definovanú (rekurzívnu) operáciu:

### Substitúcia rekurzívne

Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Pre všetky formuly  $A, B, X, Y$  jazyka  $\mathcal{L}$  a všetky binárne spojky  $b \in \{\wedge, \vee, \rightarrow\}$ :

$$\begin{array}{ll} X[A|B] = B, & \text{ak } A = X \\ X[A|B] = X, & \text{ak } X \text{ je atóm a } A \neq X \\ (\neg X)[A|B] = \neg(X[A|B]), & \text{ak } A \neq \neg X \\ (X \ b \ Y)[A|B] = ((X[A|B]) \ b \ (Y[A|B])), & \text{ak } A \neq (X \ b \ Y). \end{array}$$

### Korektnosť substitúcie ekvivalentnej formuly

Substitúciou ekvivalentnej podformuly, napríklad

$$(\neg\neg O \wedge \neg\neg C)[\neg\neg O|O] = (O \wedge \neg\neg C),$$

skutočne dostávame formulu ekvivalentnú s pôvodnou:

**Veta 4.17** (Ekvivalentné úpravy substitúciou). *Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu a nech  $X$  je formula,  $A$  a  $B$  sú výrokovologicky ekvivalentné formuly jazyka  $\mathcal{L}$ . Potom formuly  $X$  a  $X[A|B]$  sú tiež výrokovologicky ekvivalentné.*

Toto tvrdenie môžeme dokázať indukciou na konštrukciu formuly.

## Ekvivalentné úpravy a vstup pre SAT solver

Mnoho teoretických i praktických problémov možno formulovať ako otázku splniteľnosti výrokovologickej formuly – SAT (napr. riešiteľnosť sudoku, existenciu cesty či farbenia v grafe, ekvivalenciu logických hardvérových obvodov).

Na riešenie SAT existujú dlhoročne vyvíjané solvery. Ukázalo sa, že najvýhodnejšie je mať vstup v špecifickom tvare: v konjunktívnej normálnej forme (CNF). Transformácia formuly do tejto formy je tak jedným z najčastejších využití ekvivalentných úprav.

Aby sme CNF mohli popísať, potrebujeme pomenovať viacnásobne vnorené konjunkcie a viacnásobne vnorené disjunkcie a dohodneme sa na skracovaní ich zápisu vynechaním vnútorných zátvoriek.

## Konjunkcia a disjunkcia postupnosti formúl

**Definícia 4.18.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Nech  $A_1, A_2, \dots, A_n$  je konečná postupnosť formúl jazyka  $\mathcal{L}$ .

- *Konjunkciou postupnosti*  $A_1, \dots, A_n$  je formula  $((A_1 \wedge A_2) \wedge A_3) \wedge \dots \wedge A_n$ , skrátene  $(A_1 \wedge A_2 \wedge A_3 \wedge \dots \wedge A_n)$ .
  - Konjunkciu *prázdnej* postupnosti formúl ( $n = 0$ ) označujeme  $\top$ . Chápeme ju ako ľubovoľnú *tautológiu*, napríklad  $(P(c) \vee \neg P(c))$  pre nejaký unárny predikát  $P$  a nejakú konštantu  $c$  jazyka  $\mathcal{L}$ .
- *Disjunkciou postupnosti*  $A_1, \dots, A_n$  je formula  $((A_1 \vee A_2) \vee A_3) \vee \dots \vee A_n$ , skrátene  $(A_1 \vee A_2 \vee A_3 \vee \dots \vee A_n)$ .
  - Disjunkciu *prázdnej* postupnosti formúl označujeme  $\perp$  alebo  $\square$ . Chápeme ju ako ľubovoľnú *nesplniteľnú* formulu, napríklad  $(P(c) \wedge \neg P(c))$ .
- Pre  $n = 1$  chápeme samotnú formulu  $A_1$  ako konjunkciu aj ako disjunkciu jednoprvkovej postupnosti formúl  $A_1$ .

## Literál, klauzula, konjunktívny normálny tvar

Vstup do SAT solvera je formula v konjunktívnom normálnom tvare.

## Definícia 4.19.



*Literál* je atóm alebo negácia atómu.

*Klauzula* (tiež „klauza“, angl. *clause*) je *disjunkcia* postupnosti literálov.

*Formula v konjunktívnom normálnom tvare* (angl. conjunctive normal form, *CNF*) je *konjunkcia* postupnosti klauzúl.

**Príklad 4.20. Literály:**  $P, C, \neg C, \neg O$

**Klauzuly:**  $(\neg P \vee O \vee \neg C)$ , ale aj  $P, \neg O, \square$ ,

**CNF:**  $((P \vee O) \wedge \square), ((\neg P \vee O) \wedge (O \vee C))$ , ale aj  $P, \neg O, \top, (P \vee \neg O) (P \wedge \neg O \wedge C)$ ,  
 $\square$ ,

kde  $P, O, C$  sú ľubovoľné atómy.

### Existencia ekvivalentnej formuly v CNF

**Veta 4.21.** *Ku každej výrokovologickej formule  $X$  existuje ekvivalentná formula  $C$  v konjunktívnom normálnom tvare.*

*Dôkaz.* Zoberme všetky ohodnotenia  $v_1, \dots, v_n$  také, že  $v_i \models_p \neg X$  a  $v_i(A) = f$  pre všetky atómy  $A \notin \text{atoms}(\neg X)$ . Pre každé  $v_i$  zostrojme formulu  $C_i$  ako konjunkciu obsahujúcu  $A$ , ak  $v_i(A) = t$ , alebo  $\neg A$ , ak  $v_i(A) = f$ , pre každý atóm  $A \in \text{atoms}(\neg X)$ . Očividne formula  $D = (C_1 \vee \dots \vee C_n)$  je ekvivalentná s  $\neg X$  (vymenúva všetky možnosti, kedy je  $\neg X$  pravdivá).

Znegovaním  $D$  a aplikáciou de Morganových pravidiel dostaneme formulu  $C$  v CNF, ktorá je ekvivalentná s  $X$ .  $\square$

### Konverzia formuly do ekvivalentnej v CNF

Skúmanie všetkých ohodnotení podľa dôkazu vety 4.21 nie je ideálny spôsob ako upraviť formulu do CNF – najmä keď má veľa premenných a jej splniteľnosť chceme rozhodnúť SAT solverom.

Jednoduchý algoritmus na konverziu formuly do ekvivalentnej formuly v CNF založený na ekvivalentných úpravách si naprogramujete ako **4. praktické cvičenie**.

## Konverzia formuly do ekvivalentnej v CNF

Základný algoritmus konverzie do CNF má dve fázy:

1. Upravíme formulu na *negačný normálny tvar* (NNF) – nevyskytuje sa v ňom implikácia a negované sú iba atómy:
  - Nahradíme implikácie disjunkciami:  $(A \rightarrow B) \Leftrightarrow_p (\neg A \vee B)$
  - Presunieme  $\neg$  k atómom opakovaným použitím de Morganových zákonov a zákona dvojitej negácie.
2. Odstránime konjunkcie vnorené v disjunkciách „roznásobením“ podľa distributívnosti a komutatívnosti:

$$\begin{aligned}(A \vee (B \wedge C)) &\Leftrightarrow_p ((A \vee B) \wedge (A \vee C)) \\ ((B \wedge C) \vee A) &\Leftrightarrow_p (A \vee (B \wedge C)) \Leftrightarrow_p ((A \vee B) \wedge (A \vee C)) \\ &\Leftrightarrow_p ((B \vee A) \wedge (A \vee C)) \\ &\Leftrightarrow_p ((B \vee A) \wedge (C \vee A))\end{aligned}$$

## Konverzia formuly do ekvivalentnej v CNF

*Príklad 4.22.* Úprava formuly do NNF:

$$\begin{aligned}((\neg S \wedge P) \rightarrow \neg(Z \vee \neg O)) &\Leftrightarrow_p (\neg(\neg S \wedge P) \vee \neg(Z \vee \neg O)) \quad (\text{nahr. } \rightarrow) \\ &\Leftrightarrow_p ((\neg\neg S \vee \neg P) \vee (\neg Z \wedge \neg\neg O)) \quad (2 \times \text{de Morgan}) \\ &\Leftrightarrow_p ((S \vee \neg P) \vee (\neg Z \wedge O)) \quad (2 \times \text{dvoj. neg.})\end{aligned}$$

Úprava formuly v NNF do CNF:

$$\begin{aligned}((S \vee \neg P) \vee (\neg Z \wedge O)) \\ &\Leftrightarrow_p (((S \vee \neg P) \vee \neg Z) \wedge ((S \vee \neg P) \vee O)) \quad (\text{distr. } \wedge \text{ cez } \vee)\end{aligned}$$

Podľa dohody v def. 4.18 výslednú formulu v CNF skráteno zapíšeme:

$$((S \vee \neg P \vee \neg Z) \wedge (S \vee \neg P \vee O))$$

## 4.6 CNF vs. XOR

### XOR

Logická spojka exclusive or (XOR):

$a$	$b$	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

- zodpovedá sčítaniu v poli  $\mathbb{Z}_2$
- komutatívna a asociatívna
- rýchlo vypočítateľná, aj na úrovni hardvéru
- dôležitá v kryptológii

### XOR

Ideálna šifra: všetky zašifrované texty sú rovnako pravdepodobné. Napr. vezmeme náhodný reťazec (kľúč) rovnako dlhý ako správa a spravíme XOR bit po bite. Použitý kľúč zahodíme a nikdy viac nepoužijeme.

Reálne šifry: kľúč je krátky (napr. 1024 B). Ak by sme ho nakopírovali veľakrát za sebou, bity správy šifrované tým istým bitom kľúča vytvoria slabinu (možno dešifrovať aj bez znalosti kľúča, stačí uhádnuť jeho dĺžku). Preto napr. použijeme kľúč ako seed do pseudonáhodného generátora a vygenerujeme reťazec potrebnej dĺžky.

Útoky na šifry: o.i. pomocou SAT solvera, ktorý vie pracovať s XOR (aktívna oblasť výskumu).

### XOR

Ku XOR existuje prepis do CNF, napr. z  $a \oplus b \oplus c$  sa stane  $(a \vee b \vee c) \wedge (a \vee \neg b \vee \neg c) \wedge (b \vee \neg a \vee \neg c) \wedge (c \vee \neg a \vee \neg b)$

Ale s počtom premenných rastie dĺžka ekvivalentnej CNF formuly exponenciálne. Preto sa oplatí predspracovanie: XOR formuly vnímame ako súčty nad  $\mathbb{Z}_2$  a použijeme Gaussovu elimináciu.

$$a_1 \oplus a_2 \oplus a_3 = 0$$

$$a_1 \oplus a_3 \oplus a_4 = 0$$

$$\begin{pmatrix} 1 & 1 & 1 & 0 & | & 0 \\ 1 & 0 & 1 & 1 & | & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 1 & 1 & | & 0 \\ 0 & 1 & 0 & 1 & | & 0 \end{pmatrix}$$

## Rekapitulácia

### Rekapitulácia

Dnes sme prebrali:

- Logické vyplývanie z teórie a logický dôsledok teórie
- Nezávislosť formuly od teórie
- Štyri situácie vo vzťahoch teórií a formúl a ich praktické dôsledky
- Splniteľné a nespĺniteľné teórie
- Vzťah nespĺniteľnosti a vyplývania
- Význačné sémantické vlastnosti formúl: tautologickosť, splniteľnosť, nespĺniteľnosť, falzifikovateľnosť
- Ekvivalencia — sémantický vzťah formúl
- Syntaktické odvodenie ekvivalencie pomocou substitúcií podľa známych ekvivalencií
- NNF a CNF
- Vzťah tautológií s vyplývaním a ekvivalenciou

## 4. prednáška

# Dôkazy a výrokovologické tablá

---

### Rekapitulácia

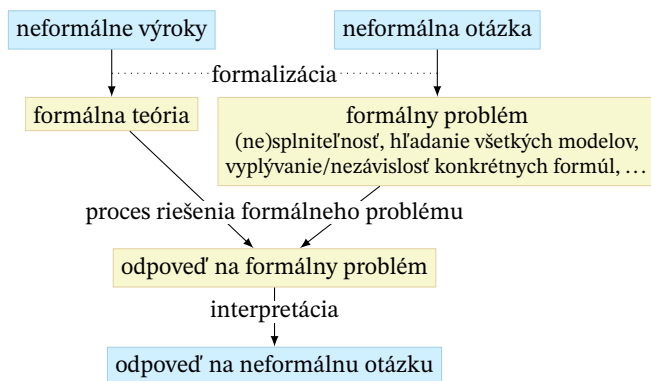
Minulý týždeň sme sa zaoberali:

- výrokovologickým *vyplývaním* formuly z teórie a nezávislosťou formuly od teórie
- vlastnosťami formúl vzhľadom na všetky ohodnotenia:
  - *tautológia*,
  - splniteľnosť,
  - falzifikovateľnosť,
  - nesplniteľnosť;
- vzťahmi formúl:
  - ekvivalencia;
- vzťahom vyplývania a ekvivalencie s tautológiami;
- transformáciou formúl medzi jazykmi so zachovaním splniteľnosti.

## 5 Dôkazy a výrokovologické tablá

### Riešenie slovných úloh pomocou formálnej logiky

Na 1. praktickom cvičení a v 1. domácej úlohe (AIN) sme riešili neformálne zadané problémy pomocou ich formálnej verzie:



Formálny problém sme riešili hrubou silou a sémanticky — rozborom všetkých ohodnotení. Žiadne naozajstné usudzovanie. Výsledok zodpovedal výsledku neformálneho úsudku o probléme.

### Dôkazy neformálnych meta tvrdení

V 1. domácej úlohe sme dokazovali tvrdenia o vyplývaní, splniteľnosti a tautológiách:

- tvrdenia v slovenčine;
- dôkazy tiež v slovenčine.

Usudzovanie, ale neformálne.

### Formalizácia dôkazov

Logiku zaujíma *jazyk* a *usudzovanie*.

Výroky v slovenčine (jazyk) sme *sformalizovali* ako *formuly* v jazyku logiky prvého rádu

- matematická „dátová štruktúra“: postupnosti symbolov s indukčnými pravidlami konštrukcie;
- javovská dátová štruktúra: stromy objektov podtried triedy Formula.

*Dôkazy* (usudzovanie) začneme *formalizovať* tento týždeň.

## Čo sú dôkazy a prečo sa dokazuje

*Dôkaz* je úvaha, ktorá zdôvodňuje, prečo je nejaký záver logickým dôsledkom predpokladov.

*Načo* sú vlastne dobré *dôkazy*?

- Môžeme nimi *presvedčiť* iných o pravdivosti svojich záverov.
- Zvyčajne sú menej prácne a *pochopiteľnejšie* ako rozbor všetkých možností.

Už rozobrať 16 možností je prácne.

Ak je možností nekonečne veľa, rozbor všetkých možností ani nie je možný.

- Odvodzovaním podľa pravidiel dôkazov môžeme skúmať, aké dôsledky má naša teória aj bez konkrétneho cieľa.

## Prečo formalizovať dôkazy

*Načo* je dobré *formalizovať* dôkazy?

- Aby sme si ujasnili, čo sú dôkazy a kedy sú *správne*. Správna argumentácia nie je dôležitá iba v matematike:
  - uvažovanie o správnosti našich programov či dopytov,
  - základ kritického/vedeckého myslenia v bežnom živote.
- Aby sme vedeli naprogramovať *dátové štruktúry* na ich reprezentáciu v počítači.
- Aby sme mohli dokazovanie *automatizovať*.
  - jeden z cieľov klasickej umelej inteligencie
- Aby sme zistili, čo sa dá a čo sa *nedá* dokázať.
  - Prakticky: Čo sa nedá dokázať, toho dôkaz sa nedá automatizovať.
  - Filozoficky: Hranice poznania a chápania.

## 5.1 Druhy dôkazov

### Druhy dôkazov

V matematike sa na to používa viac typov dôkazov:

- priamy,
- sporom,
- nepriamy,
- analýzou prípadov,

ktoré sa často kombinujú.

### Priamy dôkaz a analýza prípadov

*Priamy dôkaz* Z predpokladov postupným odvodzovaním jednoduchých logických dôsledkov dospejeme k požadovanému záveru.

*Dôkaz analýzou (rozborom) prípadov* Keď predpoklady obsahujú *disjunkciu*, dokážeme požadovaný záver z *každého disjunkt*u a ostatných predpokladov *nezávisle* od ostatných disjunktov.

Ak aj predpoklady disjunkciu neobsahujú, môžeme rozoberať prípady, že je nejaké pomocné tvrdenie pravdivé alebo nepravdivé.

### Príklad priameho dôkazu s analýzou prípadov

*Príklad 5.1* (Párty 2 · priamy dôkaz s analýzou prípadov). ( $A_1$ ) Anka príde, iba ak príde Betka a Cyril. ( $A_2$ ) Ak príde Betka alebo Dávid, príde aj Evka. ( $A_3$ ) Evka nepríde, ak príde Fero.

Teda: ( $X$ ) Ak príde Anka, tak nepríde Fero.

*Dôkaz (priamo)*. Predpokladajme, že tvrdenia  $A_1$  až  $A_3$  sú pravdivé. Dokážme  $X$ .

Ak nepríde Anka,  $X$  je pravdivé ( $X$  je implikácia a jej antecedent je nepravdivý).

Preto predpokladajme, že Anka príde. Podľa  $A_1$  potom musia prísť aj Betka a Cyril. Preto príde Betka, a teda príde Betka alebo Dávid. Podľa  $A_2$  potom príde aj Evka. Pretože podľa  $A_3$  by Evka neprišla, ak by prišiel Fero, ale Evka príde, musí byť pravda, že Fero nepríde. Preto je tvrdenie  $X$  opäť pravdivé ( $X$  je implikácia a jej konzekvent je pravdivý).



## Dôkaz sporom a nepriamy dôkaz

*Dôkaz sporom* Prijmeme predpoklady, ale *spochybíme záver* — predpokladáme, že je nepravdivý. Postupným odvodzovaním jednoduchých logických dôsledkov dospejeme k *sporu* s predpokladom alebo iným dôsledkom.

Záver teda nemôže byť nepravdivý, preto ak sú pravdivé predpoklady, je nutne pravdivý, vyplýva z nich.

*Nepriamy dôkaz* — variácia dôkazu sporom Predpokladáme, že záver je nepravdivý. Postupným odvodzovaním jednoduchých logických dôsledkov dospejeme k nepravdivosti niektorého z predpokladov.

Tým dokážeme: Ak je nepravdivý záver, tak sú nepravdivé predpoklady. Obmena: Ak sú pravdivé predpoklady, je pravdivý záver.

## Príklad dôkazu sporom

*Príklad 5.2 (Párty 2 · dôkaz sporom).*

( $A_1$ ) Anka príde, iba ak príde Betka a Cyril. ( $A_2$ ) Ak príde Betka alebo Dávid, príde aj Evka. ( $A_3$ ) Evka nepríde, ak príde Fero.

Teda: ( $X$ ) Ak príde Anka, tak nepríde Fero.

*Dôkaz (sporom).* Predpokladajme, že tvrdenia  $A_1$  až  $A_3$  sú pravdivé, ale  $X$  je nepravdivé.

Predpokladáme teda, že príde Anka a príde aj Fero. Preto príde Fero, a teda podľa predpokladu  $A_3$  Evka nepríde. Zároveň vieme, že príde Anka, a podľa  $A_1$  teda prídu aj Betka a Cyril. Preto príde Betka, a teda príde Betka alebo Dávid. Podľa  $A_2$  potom príde aj Evka. To je však spor z predchádzajúcim dôsledkom  $A_3$ , že Evka nepríde.

Predpoklad, že  $X$  je nepravdivé viedol k sporu, preto  $X$  je pravdivé.

## Výhody dôkazu sporom

Dôkaz sporom je veľmi konkrétna ukážka kritického, vedeckého myslenia:

1. Pochybujeme o pravdivosti tvrdenia.
2. Vyvrátením tejto pochybnosti sa presvedčíme o pravdivosti.

Má ale aj „technickú“ výhodu: Nemusíme pri ňom až tak tápať, ako dospejeme k cieľu, pretože

- dostaneme viac predpokladov;
- máme jednoduchý cieľ: nájsť spor;
- väčšinou stačí tvrdenia iba zjednodušovať.

### Odvodzovanie jednoduchých dôsledkov

*Kroky dôkazu by mali odvodzovať jednoduché dôsledky.*

Tie potom používame na odvodenie ďalších dôsledkov.

*Aký dôsledok je jednoduchý?*

Závisí od čitateľa dôkazu — musí byť schopný ho overiť.

Matematici (a učitelia) radi robia väčšie skoky a nechajú čitateľa (študenta) domýšľať si, prečo ich mohli urobiť.

Vyučujúci chcú od študentov malé kroky — aby si overili, že študent skutočne uvažuje správne.

## 5.2 Výrokovologické tablá

### Jednoduché dôsledky podľa definície pravdivosti formúl

Pozrime sa znova na príklad dôkazu sporom:

1. Sformalizujme ho.
2. Uvedomme si, čo vlastne dokazujeme.
3. Všímajme si, aké kroky robíme.

### Príklad dôkazu sporom s formulami

*Príklad 5.3 (Párty 2 · formalizovaný dôkaz sporom). Dokážme, že z teórie  $T = \{A_1, A_2, A_3\}$ , kde*

$A_1 = (p(A) \rightarrow (p(B) \wedge p(C)))$	Anka príde, iba ak príde Betka a Cyril.
$A_2 = ((p(B) \vee p(D)) \rightarrow p(E))$	Ak príde Betka alebo Dávid, príde aj Evka.
$A_3 = (p(F) \rightarrow \neg p(E)),$	Evka nepríde, ak príde Fero.
vyplyva formula $X$ , pričom	
$X = (p(A) \rightarrow \neg p(F))$	Ak príde Anka, tak nepríde Fero.

*Príklad 5.3* (Párty 2 · formal. dôkaz sporom, pokrač.).

*Dôkaz (sporom).* Predpokladajme, pre nejaké ohodnotenie  $v$  platí, že

(1)  $v \models_p (p(A) \rightarrow (p(B) \wedge p(C)))$ ,

(2)  $v \models_p ((p(B) \vee p(D)) \rightarrow p(E))$ ,

(3)  $v \models_p (p(F) \rightarrow \neg p(E))$ , ale

(4)  $v \not\models_p (p(A) \rightarrow \neg p(F))$ .

Podľa definície pravdivosti v ohodnotení, potom máme:

(5)  $v \models_p p(A)$  zo (4) a súčasne

(6)  $v \not\models_p \neg p(F)$  zo (4), teda

(7)  $v \models_p p(F)$  z (6). Ďalej

(8)  $v \not\models_p p(F)$ , alebo (9)  $v \models_p \neg p(E)$  podľa (3).

čo je (10)  $v \not\models_p p(E)$  z (9). Zároveň

v spore (11)  $v \not\models_p p(A)$ , alebo (12)  $v \models_p (p(B) \wedge p(C))$  podľa (1).

so (7), čo je (13)  $v \models_p p(B)$  z (12). Potom podľa (2):

v spore (14)  $v \not\models_p (p(B) \vee p(D))$ , alebo (15)  $v \models_p p(E)$ ,

s (5), (16)  $v \not\models_p p(B)$  zo (14), spor s (10).

spor s (13);

## Tablový kalkul

Z takýchto dôkazov sporom vychádza *tablový kalkul* — jeden z *formálnych deduktívnych systémov* pre výrokovologickú časť logiky prvého rádu

*Formálny deduktívny systém* je systém odvodzovacích pravidiel na konštrukciu dôkazov vyplývania formúl z teórií.

Nami používaná verzia tablového kalkulu pochádza od Raymonda M. Smullyana [Smullyan, 1979].

Postupne si ukážeme, ako predchádzajúci dôkaz premeníme na *tablo* — formálny dôkaz v tablovom kalkule.

## Označené formuly a ich sémantika

Zbavme sa najprv opakovania  $v \models_p \dots$  a  $v \not\models_p \dots$ .

**Definícia 5.4.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu. Nech  $X$  je výrokovologická formula jazyka  $\mathcal{L}$ . Postupnosti symbolov **T**  $X$  a **F**  $X$  nazývame *označené formuly*.

**Definícia 5.5.** Nech  $\mathcal{L}$  je jazyk výrokovologickej časti logiky prvého rádu,  $v$  je ohodnotenie pre  $\mathcal{L}$  a  $X$  je výrokovologická formula v  $\mathcal{L}$ . Potom

- vo  $v$  je pravdivá **T**  $X$  (skrátene  $v \models_p \mathbf{T} X$ ) vtt vo  $v$  je pravdivá  $X$ ;

- vo  $v$  je pravdivá  $\mathbf{F}X$  (skr.  $v \models_p \mathbf{F}X$ ) vtt vo  $v$  nie je pravdivá  $X$ .

Znamienko  $\mathbf{F}$  sa teda správa ako negácia a  $\mathbf{T}$  nemení význam formuly. Znamienka  $\mathbf{F}$  a  $\mathbf{T}$  sa *nesmú* objaviť v podformulách. Vďaka znamienkam stačí hovoriť iba o pravdivých ozn. formulách.

*Příklad 5.5* (Párty 2 · dôkaz s označenými formulami). Predpokladajme, pre nejakom ohodnotení  $v$  sú pravdivé označené formuly

(1)  $\mathbf{T}(p(A) \rightarrow (p(B) \wedge p(C)))$ ,

(2)  $\mathbf{T}((p(B) \vee p(D)) \rightarrow p(E))$ ,

(3)  $\mathbf{T}(p(F) \rightarrow \neg p(E))$ , ale

(4)  $\mathbf{F}(p(A) \rightarrow \neg p(F))$ .

Podľa definície pravdivosti, sú vo  $v$  pravdivé:

(5)  $\mathbf{T} p(A)$  zo (4) a súčasne

(6)  $\mathbf{F} \neg p(F)$  zo (4), teda

(7)  $\mathbf{T} p(F)$  z (6). Ďalej

(8)  $\mathbf{F} p(F)$ , alebo (9)  $\mathbf{T} \neg p(E)$  podľa (3).

čo je (10)  $\mathbf{F} p(E)$  z (9). Zároveň

v spore (11)  $\mathbf{F} p(A)$ , alebo (12)  $\mathbf{T}(p(B) \wedge p(C))$  z (1).

so (7), čo je (13)  $\mathbf{T} p(B)$  z (12). Potom podľa (2)

v spore (14)  $\mathbf{F}(p(B) \vee p(D))$ , alebo (15)  $\mathbf{T} p(E)$ ,

s (5), (16)  $\mathbf{F} p(B)$  zo (14), spor s (10).  
spor s (13);

## Kroky odvodenia

Všimnime si teraz kroky, ktoré sme v dôkaze robili:

- Niektoré z pravdivosti formuly *priamo odvodili* pravdivosť niektorej priamej podformuly, napr.:
  - z (4)  $\mathbf{F}(p(A) \rightarrow \neg p(F))$  sme odvodili (5)  $\mathbf{T} p(A)$ ;
  - z (4)  $\mathbf{F}(p(A) \rightarrow \neg p(F))$  sme odvodili (6)  $\mathbf{F} \neg p(F)$ ;
  - z (9)  $\mathbf{T} \neg p(E)$  sme odvodili (10)  $\mathbf{F} p(E)$ .
- Iné viedli k *analýze prípadov* pravdivosti *oboch* priamych podformúl:
  - (2)  $\mathbf{T}((p(B) \vee p(D)) \rightarrow p(E))$  viedla k analýze prípadov: (14)  $\mathbf{F}(p(B) \vee p(D))$  alebo (15)  $\mathbf{T} p(E)$ .

## Priame odvodenie pravdivosti priamych podformúl

Z definície pravdivosti formúl ľahko dostaneme:

**Pozorovanie 5.6.** *Nech  $v$  je ľubovoľné ohodnotenie pre jazyk  $\mathcal{L}$  výrokovologickej časti logiky prvého rádu. Nech  $X$  a  $Y$  sú ľubovoľné formuly  $\mathcal{L}$ :*

$Ak\ v \models_p \neg X, \text{ tak } v \not\models_p X.$	$Ak\ v \models_p \mathbf{T} \neg X, \text{ tak } v \models_p \mathbf{F} X.$
$Ak\ v \not\models_p \neg X, \text{ tak } v \models_p X.$	$Ak\ v \models_p \mathbf{F} \neg X, \text{ tak } v \models_p \mathbf{T} X.$
$Ak\ v \models_p (X \wedge Y), \text{ tak } v \models_p X.$	$Ak\ v \models_p \mathbf{T}(X \wedge Y), \text{ tak } v \models_p \mathbf{T} X.$
$Ak\ v \models_p (X \wedge Y), \text{ tak } v \models_p Y.$	$Ak\ v \models_p \mathbf{T}(X \wedge Y), \text{ tak } v \models_p \mathbf{T} Y.$
$Ak\ v \not\models_p (X \vee Y), \text{ tak } v \not\models_p X.$	$Ak\ v \models_p \mathbf{F}(X \vee Y), \text{ tak } v \models_p \mathbf{F} X.$
$Ak\ v \not\models_p (X \vee Y), \text{ tak } v \not\models_p Y.$	$Ak\ v \models_p \mathbf{F}(X \vee Y), \text{ tak } v \models_p \mathbf{F} Y.$
$Ak\ v \not\models_p (X \rightarrow Y), \text{ tak } v \models_p X.$	$Ak\ v \models_p \mathbf{F}(X \rightarrow Y), \text{ tak } v \models_p \mathbf{T} X.$
$Ak\ v \not\models_p (X \rightarrow Y), \text{ tak } v \not\models_p Y.$	$Ak\ v \models_p \mathbf{F}(X \rightarrow Y), \text{ tak } v \models_p \mathbf{F} Y.$

## Zjednodušujúce tablové pravidlá

Z pozorovania 5.6 môžeme sformulovať pravidlá, ktoré priamo odvodzujú z označených formúl ich označené podformuly:

$\frac{\mathbf{T} \neg X}{\mathbf{F} X}$	$\frac{\mathbf{F} \neg X}{\mathbf{T} X}$	$\frac{\mathbf{T}(X \wedge Y)}{\mathbf{T} X}$	$\frac{\mathbf{F}(X \vee Y)}{\mathbf{F} X}$	$\frac{\mathbf{F}(X \rightarrow Y)}{\mathbf{T} X}$
		$\frac{\mathbf{T}(X \wedge Y)}{\mathbf{T} Y}$	$\frac{\mathbf{F}(X \vee Y)}{\mathbf{F} Y}$	$\frac{\mathbf{F}(X \rightarrow Y)}{\mathbf{F} Y}$

Na tieto pravidlá sa dá pozeráť ako na *špeciálne prípady jedného pravidla*, ktorému sa hovorí  $\alpha$ , *zjednodušenie* alebo *sploštenie* (angl. *flattening*), pre rôzne spojky.

## Jednotný zápis označených formúl typu $\alpha$

**Definícia 5.7** (Jednotný zápis označených formúl typu  $\alpha$ ).

Označená formula  $A^+$  je typu  $\alpha$  vtt má jeden z tvarov v ľavom stĺpci tabuľky pre nejaké formuly  $X$  a  $Y$ . Takéto formuly budeme označovať písmenom  $\alpha$ ;  $\alpha_1$  bude označovať príslušnú označenú formulu zo stredného stĺpca,  $\alpha_2$  príslušnú formulu z pravého stĺpca.

$\alpha$	$\alpha_1$	$\alpha_2$
$\mathbf{T}(X \wedge Y)$	$\mathbf{T} X$	$\mathbf{T} Y$
$\mathbf{F}(X \vee Y)$	$\mathbf{F} X$	$\mathbf{F} Y$
$\mathbf{F}(X \rightarrow Y)$	$\mathbf{T} X$	$\mathbf{F} Y$
$\mathbf{T} \neg X$	$\mathbf{F} X$	$\mathbf{F} X$
$\mathbf{F} \neg X$	$\mathbf{T} X$	$\mathbf{T} X$

**Pozorovanie 5.8** (Stručne vďaka jednotnému zápisu). *Nech  $v$  je ľubovoľné ohodnotenie pre jazyk  $\mathcal{L}$  výrokovologickej časti logiky prvého rádu. Potom  $v \models_p \alpha$  vtt  $v \models_p \alpha_1$  a  $v \models_p \alpha_2$ .*

### Analýza prípadov pravdivosti priamych podformúl

Z definície pravdivosti formúl ľahko dostaneme:

**Pozorovanie 5.9.** *Nech  $v$  je ľubovoľné ohodnotenie pre jazyk  $\mathcal{L}$  výrokovologickej časti logiky prvého rádu. Nech  $X$  a  $Y$  sú ľubovoľné formuly  $\mathcal{L}$ :*

- *Ak  $v \models_p (X \wedge Y)$ , tak  $v \models_p X$  alebo  $v \models_p Y$ . Ak  $v \models_p \mathbf{F}(X \wedge Y)$ , tak  $v \models_p \mathbf{F}X$  alebo  $v \models_p \mathbf{F}Y$ .*
- *Ak  $v \models_p (X \vee Y)$ , tak  $v \models_p X$  alebo  $v \models_p Y$ . Ak  $v \models_p \mathbf{T}(X \vee Y)$ , tak  $v \models_p \mathbf{T}X$  alebo  $v \models_p \mathbf{T}Y$ .*
- *Ak  $v \models_p (X \rightarrow Y)$ , tak  $v \models_p X$  alebo  $v \models_p Y$ . Ak  $v \models_p \mathbf{T}(X \rightarrow Y)$ , tak  $v \models_p \mathbf{F}X$  alebo  $v \models_p \mathbf{T}Y$ .*

### Rozvetvujúce tablové pravidlá

Z pozorovania 5.9 môžeme sformulovať pravidlá, ktoré vedú k analýze prípadov pravdivosti priamych podformúl:

$$\frac{\mathbf{F}(X \wedge Y)}{\mathbf{F}X \mid \mathbf{F}Y} \qquad \frac{\mathbf{T}(X \vee Y)}{\mathbf{T}X \mid \mathbf{T}Y} \qquad \frac{\mathbf{T}(X \rightarrow Y)}{\mathbf{F}X \mid \mathbf{T}Y}$$

Aj na tieto pravidlá sa dá pozerat' ako na špeciálne prípady jedného pravidla, ktorému sa hovorí  $\beta$ , *vetvenie* alebo *rozdelenie* (angl. *split*), pre rôzne spojky.

### Jednotný zápis označených formúl typu $\beta$

**Definícia 5.10** (Jednotný zápis označených formúl typu  $\beta$ ).

Označená formula  $B^+$  je typu  $\beta$  vtt má jeden z tvarov v ľavom stĺpci tabuľky pre nejaké formuly  $X$  a  $Y$ . Takéto formuly budeme označovať písmenom  $\beta$ ;  $\beta_1$  bude označovať príslušnú označenú formulu zo stredného stĺpca,  $\beta_2$  príslušnú formulu z pravého stĺpca.

	$\beta$	$\beta_1$	$\beta_2$
	$\mathbf{F}(X \wedge Y)$	$\mathbf{F}X$	$\mathbf{F}Y$
	$\mathbf{T}(X \vee Y)$	$\mathbf{T}X$	$\mathbf{T}Y$
	$\mathbf{T}(X \rightarrow Y)$	$\mathbf{F}X$	$\mathbf{T}Y$

**Pozorovanie 5.11** (Stručne vďaka jednotnému zápisu). *Nech  $v$  je ľubovoľné ohodnotenie pre jazyk  $\mathcal{L}$  výrokovologickej časti logiky prvého rádu. Potom  $v \models_p \beta$  vtt  $v \models_p \beta_1$  alebo  $v \models_p \beta_2$ .*

### Označovanie označených formúl a ich množín

Čo vlastne dokazujeme v našom príklade? To, že predpoklad existencie ohodnotenia  $v$ , v ktorom sú pravdivé všetky prvky množiny označených formúl

$$S^+ = \{ \begin{array}{l} \mathbf{T}(p(A) \rightarrow (p(B) \wedge p(C))), \\ \mathbf{T}((p(B) \vee p(D)) \rightarrow p(E)), \\ \mathbf{T}(p(F) \rightarrow \neg p(E)), \\ \mathbf{F}(p(A) \rightarrow \neg p(F)) \end{array} \}$$

vedie k sporu, teda že  $S^+$  je *nesplniteľná*.

*Dohoda 5.12.* Pre označené formuly budeme používať veľké písmená zo začiatku a konca abecedy s horným indexom + a prípadne s dolnými indexmi, napr.  $A^+$ ,  $X_7^+$ .

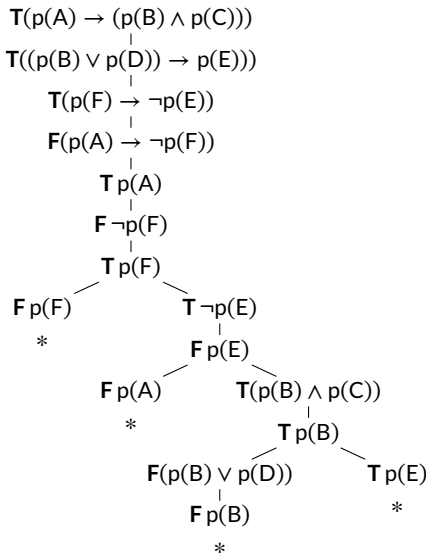
Pre množiny označených formúl budeme používať písmená  $S$ ,  $T$  s horným indexom + a prípadne s dolnými indexmi, napr.  $S^+$ ,  $T_3^+$ .

*Príklad 5.12* (Párty 2 · tablo).

1.	$\mathbf{T}(p(A) \rightarrow (p(B) \wedge p(C)))$	$S^+$								
2.	$\mathbf{T}((p(B) \vee p(D)) \rightarrow p(E))$	$S^+$								
3.	$\mathbf{T}(p(F) \rightarrow \neg p(E))$	$S^+$								
4.	$\mathbf{F}(p(A) \rightarrow \neg p(F))$	$S^+$								
5.	$\mathbf{T} p(A)$	$\alpha 4$								
6.	$\mathbf{F} \neg p(F)$	$\alpha 4$								
7.	$\mathbf{T} p(F)$	$\alpha 6$								
8.	$\mathbf{F} p(F) \quad \beta 3$ *7, 8	<table><tr><td>9.</td><td><math>\mathbf{T} \neg p(E) \quad \beta 3</math></td></tr><tr><td>10.</td><td><math>\mathbf{F} p(E) \quad \alpha 9</math></td></tr></table>	9.	$\mathbf{T} \neg p(E) \quad \beta 3$	10.	$\mathbf{F} p(E) \quad \alpha 9$				
9.	$\mathbf{T} \neg p(E) \quad \beta 3$									
10.	$\mathbf{F} p(E) \quad \alpha 9$									
	11. $\mathbf{F} p(A) \quad \beta 1$ *5, 11	<table><tr><td>12.</td><td><math>\mathbf{T}(p(B) \wedge p(C)) \quad \beta 1</math></td></tr><tr><td>13.</td><td><math>\mathbf{T} p(B) \quad \alpha 12</math></td></tr></table>	12.	$\mathbf{T}(p(B) \wedge p(C)) \quad \beta 1$	13.	$\mathbf{T} p(B) \quad \alpha 12$				
12.	$\mathbf{T}(p(B) \wedge p(C)) \quad \beta 1$									
13.	$\mathbf{T} p(B) \quad \alpha 12$									
		<table><tr><td>14.</td><td><math>\mathbf{F}(p(B) \vee p(D)) \quad \beta 2</math></td><td>15.</td><td><math>\mathbf{T} p(E) \quad \beta 2</math></td></tr><tr><td>16.</td><td><math>\mathbf{F} p(B) \quad \alpha 14</math> *13, 16</td><td></td><td>*10,15</td></tr></table>	14.	$\mathbf{F}(p(B) \vee p(D)) \quad \beta 2$	15.	$\mathbf{T} p(E) \quad \beta 2$	16.	$\mathbf{F} p(B) \quad \alpha 14$ *13, 16		*10,15
14.	$\mathbf{F}(p(B) \vee p(D)) \quad \beta 2$	15.	$\mathbf{T} p(E) \quad \beta 2$							
16.	$\mathbf{F} p(B) \quad \alpha 14$ *13, 16		*10,15							

## Štruktúra tabla

Čo je teda tablo? Aká „dátová štruktúra“? Čo v nej musí platiť?



**Definícia 5.13** (Tablo pre množinu označených formúl [Smullyan, 1979]). *Analytické tablo pre množinu označených formúl  $S^+$*  (skrátene *tablo pre  $S^+$* ) je binárny strom, ktorého vrcholy obsahujú označené formuly a ktorý je skonštruovaný podľa nasledovných indukčných pravidiel:

- Strom s jediným vrcholom (koreňom) obsahujúcim niektorú označenú formulu  $A^+$  z  $S^+$  je tablom pre  $S^+$ .
- Nech  $\mathcal{T}$  je tablo pre  $S^+$  a  $y$  je nejaký jeho list. Potom tablom pre  $S^+$  je aj každé *priame rozšírenie*  $\mathcal{T}$  ktorýmkoľvek z pravidiel:

$\alpha$ : Ak sa na vetve  $\pi_y$  (ceste z koreňa do  $y$ ) vyskytuje nejaká označená formula  $\alpha$ , tak ako jediné dieťa  $y$  pripojíme nový vrchol obsahujúci  $\alpha_1$  alebo  $\alpha_2$ .

$\beta$ : Ak sa na vetve  $\pi_y$  (ceste z koreňa do  $y$ ) vyskytuje nejaká označená formula  $\beta$ , tak ako deti  $y$  pripojíme *dva* nové vrcholy, pričom ľavé dieťa bude obsahovať  $\beta_1$  a pravé  $\beta_2$ .

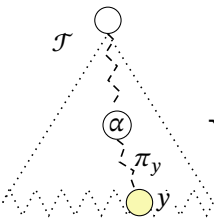
$S^+$ : Ako jediné dieťa  $y$  pripojíme nový vrchol obsahujúci ľubovoľnú označenú formulu  $A^+ \in S^+$ .



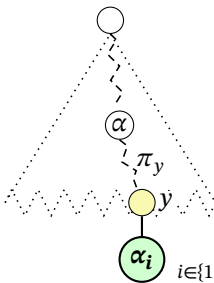
Nič iné nie je tablom pre  $S^+$ .

**Tablá a tablové pravidlá**

**Pôvodné tablo**   **Možné priame rozšírenie**   **Pravidlá a označené formuly v nich**



$\rightsquigarrow$



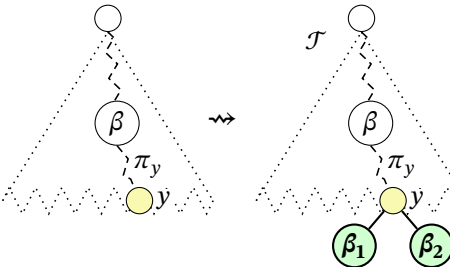
$i \in \{1, 2\}$

$$\frac{\alpha}{\alpha_1} \quad \frac{\alpha}{\alpha_2}$$

$\alpha$	$\alpha_1$	$\alpha_2$
$T(X \wedge Y)$	$TX$	$TY$
$F(X \vee Y)$	$FX$	$FY$
$F(X \rightarrow Y)$	$TX$	$FY$
$T \neg X$	$FX$	$FX$
$F \neg X$	$TX$	$TX$

$$\frac{\beta}{\beta_1 \mid \beta_2}$$

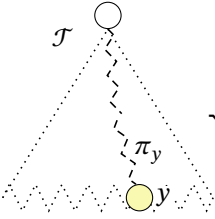
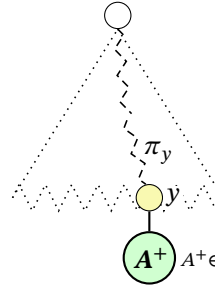
$\beta$	$\beta_1$	$\beta_2$
$F(X \wedge Y)$	$FX$	$FY$
$T(X \vee Y)$	$TX$	$TY$
$T(X \rightarrow Y)$	$FX$	$TY$



*Legenda:*  $y$  je list v table  $\mathcal{T}$ ,  $\pi_y$  je cesta od koreňa k  $y$

**Tablá a tablové pravidlá (pokračovanie)**

**Pôvodné tablo**   **Možné priame rozšírenie**   **Pravidlá a označené formuly v nich**

	$\rightsquigarrow$		$\frac{}{A^+}$	$A^+ \in S^+$
---	--------------------	---	----------------	---------------

*Legenda:*  $y$  je list v table  $\mathcal{T}$ ,  $\pi_y$  je cesta od koreňa k  $y$

## Uzavretosť a otvorenosť vetvy a tabla

**Definícia 5.14.** *Vetvou tabla  $\mathcal{T}$  je každá cesta od koreňa  $\mathcal{T}$  k niektorému listu  $\mathcal{T}$ .*

Označená formula  $X^+$  sa vyskytuje na vetve  $\pi$  v  $\mathcal{T}$  vtt  $X^+$  sa nachádza v niektorom vrchole na  $\pi$ . Skrátene to budeme zapisovať  $X^+ \in \text{formulas}(\pi)$ .

Tablo  $\sim$  dôkaz sporom. Vetvenie  $\sim$  rozbor možných prípadov.  $\implies$  Spor musí nastať vo všetkých vetvách.

**Definícia 5.15.** *Vetva  $\pi$  tabla  $\mathcal{T}$  je uzavretá vtt na  $\pi$  sa súčasne vyskytujú označené formuly  $\mathbf{F}X$  a  $\mathbf{T}X$  pre nejakú formulu  $X$ . Inak je  $\pi$  otvorená.*

*Tablo  $\mathcal{T}$  je uzavreté vtt každá jeho vetva je uzavretá. Naopak,  $\mathcal{T}$  je otvorené vtt aspoň jedna jeho vetva je otvorená.*

### Príklad — vetvy a uzavretosť

*Príklad 5.16 (Vetvy a uzavretosť).* Určme vetvy v table a zistíme, či sú uzavreté a či je uzavreté tablo:

1.	$\mathbf{T}(p(A) \rightarrow (p(B) \wedge p(C)))$	$S^+$																
2.	$\mathbf{T}((p(B) \vee p(D)) \rightarrow p(E))$	$S^+$																
3.	$\mathbf{T}(p(F) \rightarrow \neg p(E))$	$S^+$																
4.	$\mathbf{F}(p(A) \rightarrow \neg p(F))$	$S^+$																
5.	$\mathbf{T} p(A)$	$\alpha 4$																
6.	$\mathbf{F} \neg p(F)$	$\alpha 4$																
7.	$\mathbf{T} p(F)$	$\alpha 6$																
8.	$\mathbf{F} p(F) \quad \beta 3$ *7, 8	<table><tr><td>9.</td><td><math>\mathbf{T} \neg p(E) \quad \beta 3</math></td></tr><tr><td>10.</td><td><math>\mathbf{F} p(E) \quad \alpha 9</math></td></tr><tr><td>11.</td><td><math>\mathbf{F} p(A) \quad \beta 1</math> *5, 11</td><td><table><tr><td>12.</td><td><math>\mathbf{T}(p(B) \wedge p(C)) \quad \beta 1</math></td></tr><tr><td>13.</td><td><math>\mathbf{T} p(B) \quad \alpha 12</math></td></tr><tr><td>14.</td><td><math>\mathbf{F}(p(B) \vee p(D)) \quad \beta 2</math></td><td><table><tr><td>15.</td><td><math>\mathbf{T} p(E) \quad \beta 2</math> *10,15</td></tr></table></td></tr></table></td></tr></table>	9.	$\mathbf{T} \neg p(E) \quad \beta 3$	10.	$\mathbf{F} p(E) \quad \alpha 9$	11.	$\mathbf{F} p(A) \quad \beta 1$ *5, 11	<table><tr><td>12.</td><td><math>\mathbf{T}(p(B) \wedge p(C)) \quad \beta 1</math></td></tr><tr><td>13.</td><td><math>\mathbf{T} p(B) \quad \alpha 12</math></td></tr><tr><td>14.</td><td><math>\mathbf{F}(p(B) \vee p(D)) \quad \beta 2</math></td><td><table><tr><td>15.</td><td><math>\mathbf{T} p(E) \quad \beta 2</math> *10,15</td></tr></table></td></tr></table>	12.	$\mathbf{T}(p(B) \wedge p(C)) \quad \beta 1$	13.	$\mathbf{T} p(B) \quad \alpha 12$	14.	$\mathbf{F}(p(B) \vee p(D)) \quad \beta 2$	<table><tr><td>15.</td><td><math>\mathbf{T} p(E) \quad \beta 2</math> *10,15</td></tr></table>	15.	$\mathbf{T} p(E) \quad \beta 2$ *10,15
9.	$\mathbf{T} \neg p(E) \quad \beta 3$																	
10.	$\mathbf{F} p(E) \quad \alpha 9$																	
11.	$\mathbf{F} p(A) \quad \beta 1$ *5, 11	<table><tr><td>12.</td><td><math>\mathbf{T}(p(B) \wedge p(C)) \quad \beta 1</math></td></tr><tr><td>13.</td><td><math>\mathbf{T} p(B) \quad \alpha 12</math></td></tr><tr><td>14.</td><td><math>\mathbf{F}(p(B) \vee p(D)) \quad \beta 2</math></td><td><table><tr><td>15.</td><td><math>\mathbf{T} p(E) \quad \beta 2</math> *10,15</td></tr></table></td></tr></table>	12.	$\mathbf{T}(p(B) \wedge p(C)) \quad \beta 1$	13.	$\mathbf{T} p(B) \quad \alpha 12$	14.	$\mathbf{F}(p(B) \vee p(D)) \quad \beta 2$	<table><tr><td>15.</td><td><math>\mathbf{T} p(E) \quad \beta 2</math> *10,15</td></tr></table>	15.	$\mathbf{T} p(E) \quad \beta 2$ *10,15							
12.	$\mathbf{T}(p(B) \wedge p(C)) \quad \beta 1$																	
13.	$\mathbf{T} p(B) \quad \alpha 12$																	
14.	$\mathbf{F}(p(B) \vee p(D)) \quad \beta 2$	<table><tr><td>15.</td><td><math>\mathbf{T} p(E) \quad \beta 2</math> *10,15</td></tr></table>	15.	$\mathbf{T} p(E) \quad \beta 2$ *10,15														
15.	$\mathbf{T} p(E) \quad \beta 2$ *10,15																	

### Korektnosť tablového kalkulu

**Veta 5.17** (Korektnosť tablového kalkulu [Smullyan, 1979]). *Nech  $S^+$  je množina označených formúl a  $\mathcal{T}$  je uzavreté tablo pre  $S^+$ . Potom je množina  $S^+$  nesplniteľná.*

**Dôsledok 5.18.** *Nech  $S$  je výrokovologická teória,  $X$  je výrokovologická formula a nech  $X$  je výrokovologicky dokázateľná z  $S$  (skrát.  $S \vdash_p X$ ), t.j., nech existuje uzavreté tablo pre množinu označených formúl  $\{\mathbf{T} A \mid A \in S\} \cup \{\mathbf{F} X\}$ . Potom z  $S$  výrokovologicky vyplýva  $X$  ( $S \models_p X$ ).*

**Dôsledok 5.19.** *Nech  $X$  je výrokovologická formula a nech  $X$  je výrokovologicky dokázateľná (skrát.  $\vdash_p X$ ), t.j., nech existuje uzavreté tablo pre množinu označených formúl  $\{\mathbf{F} X\}$ . Potom  $X$  je tautológia ( $\models_p X$ ).*

### ***Spomeňte si 5.1***

1. Má každé tablo *aspoň* jedno priame rozšírenie?
2. Má každé tablo *najviac* jedno priame rozšírenie?

## 5. prednáška

# Korektnosť a úplnosť výrokovologických tabiel

---

### Rekapitulácia a plán

Minulý týždeň:

- Sformalizovali sme dôkazy sporom pomocou tabiel.
- Vyslovili, ale nedokázali tvrdenie o *korektnosti tabiel*: *uzavreté tablo* dokazuje výrokovologickú *nesplniteľnosť*
- a dôsledky pre dokazovanie vyplývania a tautológií.

Dnes:

- *Dokážeme* korektnosť tabiel.
- Preskúmame, čo vedia tablá povedať o *splniteľnosti*.
- *Dokážeme* úplnosť tabiel.

### 5.3 Korektnosť tabiel

#### Korektnosť — idea dôkazu

Aby sme dokázali korektnosť tabiel, teda vetu 5.17, dokážeme postupne dve lemy:

K1: Ak máme tablo pre splniteľnú množinu  $S^+$  s aspoň jednou splniteľnou vetvou, tak každé jeho *priame rozšírenie* má tiež splniteľnú vetvu.

K2: Každé tablo pre splniteľnú množinu  $S^+$  má aspoň jednu splniteľnú vetvu.

Z toho ľahko sporom dokážeme, že množina, pre ktorú sme našli uzavreté tablo je nesplniteľná.

### Korektnosť — pravdivosť priameho rozšírenia tabla

Všimnime si:

Vetva sa správa ako konjunkcia svojich označených formúl — všetky musia byť naraz pravdivé.

Tablo sa správa ako disjunkcia vetiev — niektorá musí byť pravdivá.

**Definícia 5.20.** Nech  $S^+$  je množina označených formúl v jazyku  $\mathcal{L}$ , nech  $\mathcal{T}$  je tablo pre  $S^+$ , nech  $\pi$  je vetva tabla  $\mathcal{T}$  a nech  $v$  je výrokovologické ohodnotenie pre  $\mathcal{L}$ . Potom:

- *vetva  $\pi$  je pravdivá vo  $v$  ( $v \models_p \pi$ ) vtt vo  $v$  sú pravdivé všetky označené formuly vyskytujúce sa na vetve  $\pi$ .*
- *tablo  $\mathcal{T}$  je pravdivé vo  $v$  ( $v \models_p \mathcal{T}$ ) vtt niektorá vetva v table  $\mathcal{T}$  je pravdivá.*

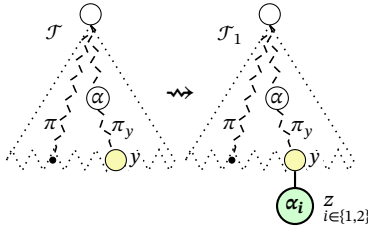
### Korektnosť — pravdivosť priameho rozšírenia tabla

Pomocou predchádzajúcej definície sformulujeme lemu K1 takto:

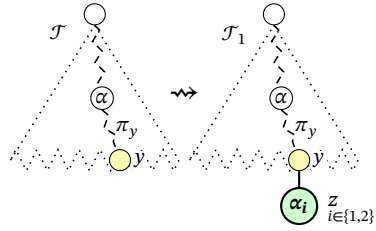
**Lema 5.21 (K1).** *Nech  $S^+$  je množina označených formúl v jazyku  $\mathcal{L}$ , nech  $\mathcal{T}$  je tablo pre  $S^+$  a nech  $v$  je výrokovologické ohodnotenie pre  $\mathcal{L}$ . Ak  $S^+$  a  $\mathcal{T}$  sú pravdivé vo  $v$ , tak aj každé priame rozšírenie  $\mathcal{T}$  je pravdivé vo  $v$ .*

*Dôkaz lemy K1.* Nech  $S^+$  je množina označených formúl,  $\mathcal{T}$  je tablo pre  $S^+$  a  $v$  je ohodnotenie. Nech  $v \models_p S^+$  a nech  $\mathcal{T}$  je pravdivé vo  $v$ . Potom je pravdivá niektorá vetva v  $\mathcal{T}$ . Zoberme jednu takú vetvu a označme ju  $\pi$ . Nech  $\mathcal{T}_1$  je priame rozšírenie  $\mathcal{T}$ . Nastáva jeden z prípadov:

- $\mathcal{T}_1$  vzniklo z  $\mathcal{T}$  pravidlom  $\alpha$ , pridaním nového dieťaťa z nejakému listu  $y$  v  $\mathcal{T}$ , pričom  $z$  obsahuje  $\alpha_1$  alebo  $\alpha_2$  pre nejakú formulu  $\alpha$  na vetve  $\pi_y$ .

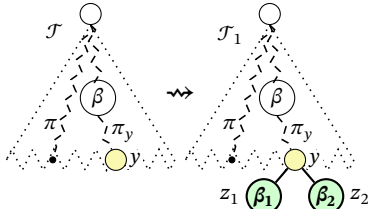


Ak  $\pi \neq \pi_y$ , tak  $\mathcal{T}_1$  obsahuje  $\pi$ , a teda aj  $\mathcal{T}_1$  je pravdivé vo  $v$ .

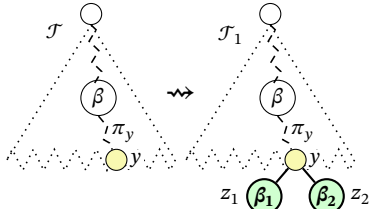


Ak  $\pi = \pi_y$ , tak  $\alpha$  je pravdivá vo  $v$ , pretože  $\alpha$  je na  $\pi$ . Potom aj  $\alpha_1$  a  $\alpha_2$  sú pravdivé vo  $v$  (pozorovanie 5.8). Vetva  $\pi_z$  v table  $\mathcal{T}_1$  rozširuje vetvu  $\pi$  pravdivú vo  $v$  o vrchol  $z$  obsahujúci ozn. formulu  $\alpha_1$  alebo  $\alpha_2$  pravdivú vo  $v$ . Preto  $\pi_z$  je pravdivá vo  $v$ , a teda aj tablo  $\mathcal{T}_1$  je pravdivé vo  $v$ .

- $\mathcal{T}_1$  vzniklo z  $\mathcal{T}$  pravidlom  $\beta$ , pridaním detí  $z_1$  a  $z_2$  nejakému listu  $y$  v  $\mathcal{T}$ , pričom  $z_1$  obsahuje  $\beta_1$  a  $z_2$  obsahuje  $\beta_2$  pre nejakú formulu  $\beta$  na vetve  $\pi_y$ .

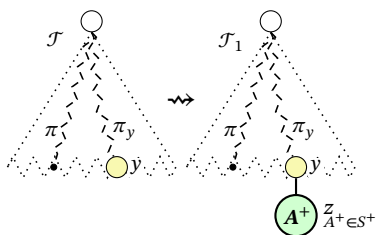


Ak  $\pi \neq \pi_y$ , tak  $\mathcal{T}_1$  obsahuje  $\pi$ , a teda aj  $\mathcal{T}_1$  je pravdivé vo  $v$ .

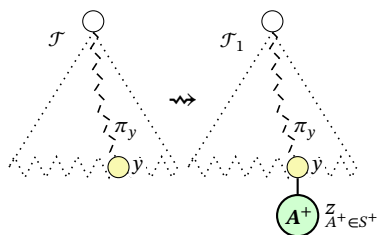


Ak  $\pi = \pi_y$ , tak  $v \models_p \beta$ , pretože  $\beta$  je na  $\pi$ . Potom  $v \models_p \beta_1$  alebo  $v \models_p \beta_2$  (poz. 5.11). Ak  $v \models_p \beta_1$ , tak  $v \models_p \pi_{z_1}$ , a teda  $v \models_p \mathcal{T}_1$ . Ak  $v \models_p \beta_2$ , tak  $v \models_p \pi_{z_2}$ , a teda  $v \models_p \mathcal{T}_1$ .

- $\mathcal{T}_1$  vzniklo z  $\mathcal{T}$  pravidlom  $S^+$ , pridaním nového dieťaťa z nejakému listu  $y$  v  $\mathcal{T}$ , pričom  $z$  obsahuje formulu  $A^+ \in S^+$ .



Ak  $\pi \neq \pi_y$ , tak  $\mathcal{T}_1$  obsahuje  $\pi$ , a teda aj  $\mathcal{T}_1$  je pravdivé vo  $v$ .



Ak  $\pi = \pi_y$ , tak  $\pi_z$  v table  $\mathcal{T}_1$  je pravdivá vo  $v$ , pretože je rozšírením vetvy  $\pi$  pravdivej vo  $v$  o vrchol  $z$  obsahujúci formulu  $A^+$  pravdivú vo  $v$  (pretože  $v \models_p S^+$  a  $A^+ \in S^+$ ). Preto tablo  $\mathcal{T}_1$  je pravdivé vo  $v$ .  $\square$

## Korektnosť — pravdivosť množiny a tabla pre ňu

**Lema 5.22 (K2).** *Nech  $S^+$  je množina označených formúl v jazyku  $\mathcal{L}$ , nech  $\mathcal{T}$  je tablo pre  $S^+$  a nech  $v$  je ohodnotenie pre  $\mathcal{L}$ . Ak  $S^+$  je pravdivá vo  $v$ , tak aj  $\mathcal{T}$  je pravdivé vo  $v$ .*

*Dôkaz lemy K2.* Nech  $S^+$  je množina označených formúl, nech  $v$  je ohodnotenie a nech  $v \models_p S^+$ . Úplnou indukciou na počet vrcholov tabla  $\mathcal{T}$  dokážeme, že vo  $v$  je pravdivé každé tablo  $\mathcal{T}$  pre  $S^+$ .

Ak má  $\mathcal{T}$  jediný vrchol, tento vrchol obsahuje formulu  $A^+ \in S^+$ , ktorá je pravdivá vo  $v$ . Preto je pravdivá jediná vetva v  $\mathcal{T}$ , teda aj  $\mathcal{T}$ .

Ak  $\mathcal{T}$  má viac ako jeden vrchol, je priamym rozšírením nejakého tabla  $\mathcal{T}_0$ , ktoré má o 1 alebo o 2 vrcholy menej ako  $\mathcal{T}$ . Podľa indukčného predpokladu je  $\mathcal{T}_0$  pravdivé vo  $v$ . Podľa lemy K1 je potom vo  $v$  pravdivé aj  $\mathcal{T}$ .  $\square$

## Korektnosť — dôkaz

*Dôkaz vety o korektnosti 5.17.* Nech  $S^+$  je množina označených formúl a  $\mathcal{T}$  je uzavreté tablo pre  $S^+$ . Sporom: Predpokladajme, že existuje ohodnotenie, v ktorom je  $S^+$  pravdivá. Označme ho  $v$ . Potom podľa lemy K2 je vo  $v$  pravdivé tablo  $\mathcal{T}$ , teda vo  $v$  je pravdivá niektorá vetva  $\pi$  v  $\mathcal{T}$ . Pretože  $\mathcal{T}$  je uzavreté, aj vetva  $\pi$  je uzavretá. Na  $\pi$  sa teda nachádzajú označené formuly **TX**

a  $\mathbf{F}X$  pre nejakú formulu  $X$ . Pretože  $\pi$  je pravdivá vo  $v$ , musia byť vo  $v$  pravdivé všetky formuly na nej. Ale  $v \models_p \mathbf{T}X$  vtt  $v \models_p X$  a  $v \models_p \mathbf{F}X$  vtt  $v \not\models_p X$ . Teda  $\mathbf{T}X$  a  $\mathbf{F}X$  nemôžu byť obe pravdivé, čo je spor.  $\square$

## 5.4 Testovanie nespľniteľnosti, splniteľnosti a falzifikovateľnosti

### Úplná vetva a tablo

*Príklad 5.23.* Zistíme tablom, či

$$\{((\text{rychly}(p) \vee \text{spravny}(p)) \wedge (\text{citatelny}(p) \vee \text{rychly}(p)))\} \\ \models_p (\text{rychly}(p) \wedge (\text{spravny}(p) \vee \text{citatelny}(p))).$$

Vybudujeme tablo pre množinu označených formúl:

$$S^+ = \{\mathbf{T}((\text{rychly}(p) \vee \text{spravny}(p)) \wedge (\text{citatelny}(p) \vee \text{rychly}(p))), \\ \mathbf{F}(\text{rychly}(p) \wedge (\text{spravny}(p) \vee \text{citatelny}(p)))\}$$

Podarí sa nám ho uzavrieť?

### Úplná vetva a tablo

Nech v príklade tablové pravidlá používame akokoľvek,

- *nenájdeme uzavreté* tablo, ale
- ak pravidlá nepoužívame opakovane na rovnakú formulu v rovnakej vetve, po čase *vybudujeme úplné a otvorené* tablo.

**Definícia 5.24** (Úplná vetva a úplné tablo). Nech  $S^+$  je množina označených formúl a  $\mathcal{T}$  je tablo pre  $S^+$ .

*Vetva  $\pi$  v table  $\mathcal{T}$  je úplná* vtt má všetky nasledujúce vlastnosti:

- pre každú označenú formulu  $\alpha$ , ktorá sa vyskytuje na  $\pi$ , sa *obidve* označené formuly  $\alpha_1$  a  $\alpha_2$  vyskytujú na  $\pi$ ;
- pre každú označenú formulu  $\beta$ , ktorá sa vyskytuje na  $\pi$ , sa *aspoň jedna* z označených formúl  $\beta_1, \beta_2$  vyskytuje na  $\pi$ ;
- *každá*  $X^+ \in S^+$  sa vyskytuje na  $\pi$ .

*Tablo  $\mathcal{T}$  je úplné* vtt každá jeho vetva je *úplná alebo uzavretá*.



## Otvorené tablo a splniteľnosť

Z otvoreného a úplného tabla pre  $S^+$  môžeme vytvoriť ohodnotenie  $v$ :

1. nájdeme otvorenú vetvu  $\pi$ ,
2. pre každý atóm  $A$ 
  - ak sa na  $\pi$  nachádza  $\mathbf{T} A$ , definujeme  $v(A) = t$ ;
  - ak sa na  $\pi$  nachádza  $\mathbf{F} A$ , definujeme  $v(A) = f$ ;
  - inak definujeme  $v(A)$  ľubovoľne.

V tomto  $v$  je pravdivá  $\pi$ , a preto je v ňom *pravdivá aj*  $S^+$  (všetky formuly z  $S^+$  sa vyskytujú na  $\pi$ , lebo  $\pi$  je úplná).

Otázka.

- Dá sa vždy nájsť úplné tablo pre  $S^+$ ?
- Naozaj sa z úplného otvoreného tabla dá vytvoriť model  $S^+$ ?

## Existencia úplného tabla

**Lema 5.25** (o existencii úplného tabla). *Nech  $S^+$  je konečná množina označených formúl. Potom existuje úplné tablo pre  $S^+$ .*

*Dôkaz.* Vybudujme tablo  $\mathcal{T}_0$  pre  $S^+$  tak, že do koreňa vložíme niektorú formulu z  $S^+$  a opakovaním spravidla  $S^+$  postupne doplníme ostatné.

Potom tablo postupne rozširujeme tak, že vyberieme ľubovoľný list  $y$  tabla  $\mathcal{T}_i$ , ktorého vetva  $\pi_y$  je otvorená a nie je úplná. Potom nastane aspoň jedna z možností:

- Na  $\pi_y$  sa nachádza nejaká formula  $\alpha$ , ale nenachádza sa *niektorá* z formúl  $\alpha_1$  a  $\alpha_2$ .
- Na  $\pi_y$  sa nachádza nejaká formula  $\beta$ , ale nenachádza sa *ani jedna* z formúl  $\beta_1$  a  $\beta_2$ .

Ak platí prvá alebo obe možnosti, aplikujeme pravidlo  $\alpha$ . Ak platí iba druhá možnosť, aplikujeme pravidlo  $\beta$ . Získame tablo  $\mathcal{T}_{i+1}$ , s ktorým proces opakujeme.

Tento proces po konečnom počte krokov (prečo?) vytvorí nejaké tablo  $\mathcal{T}_n$ , v ktorom už neexistuje vetva, ktorá by bola otvorená a nebola úplná. Teda každá vetva v  $\mathcal{T}_n$  je buď uzavretá alebo úplná, čiže  $\mathcal{T}_n$  je úplné.  $\square$

## 5.5 Úplnosť

### Nadol nasýtené množiny a Hintikkova lemma

**Definícia 5.26.** Množina označených formúl  $S^+$  sa nazýva *nadol nasýtená* vtt platí:

$H_0$ : v  $S^+$  sa nevyskytujú naraz  $\mathbf{T} A$  a  $\mathbf{F} A$  pre žiaden predikátový atóm  $A$ ;

$H_1$ : ak  $\alpha \in S^+$ , tak  $\alpha_1 \in S^+$  a  $\alpha_2 \in S^+$ ;

$H_2$ : ak  $\beta \in S^+$ , tak  $\beta_1 \in S^+$  alebo  $\beta_2 \in S^+$ .

**Pozorovanie 5.27.** *Nech  $\pi$  je úplná otvorená vetva nejakého tabla  $\mathcal{T}$ . Potom množina všetkých označených formúl na  $\pi$  je nadol nasýtená.*

**Lema 5.28** (Hintikkova). *Každá nadol nasýtená množina  $S^+$  je splniteľná.*

*Dôkaz Hintikkovej lemy.* Chceme dokázať, že existuje ohodnotenie  $v$ , v ktorom sú pravdivé všetky označené formuly z  $S^+$ . Definujme  $v$  pre každý predikátový atóm  $A$  takto:

$$v(A) = \begin{cases} t, & \text{ak } \mathbf{T} A \in S^+; \\ f, & \text{ak } \mathbf{F} A \in S^+; \\ t, & \text{ak ani } \mathbf{T} A \text{ ani } \mathbf{F} A \text{ nie sú v } S^+. \end{cases}$$

$v$  je korektne definované vďaka  $H_0$  (každému atómu priradí  $t$  alebo  $f$ , žiadnemu nepriradí obe).

Indukciou na stupeň formuly dokážeme, že vo  $v$  sú pravdivé všetky formuly z  $S^+$ :

1° Všetky označené predikátové atómy (formuly stupňa 0) z  $S^+$  sú pravdivé vo  $v$ .

2° Nech  $X^+ \in S^+$  a nech platí IP: Vo  $v$  sú pravdivé všetky formuly z  $S^+$  nižšieho stupňa ako  $X^+$ .  $X^+$  je buď  $\alpha$  alebo  $\beta$ :

Ak  $X^+$  je  $\alpha$ , potom obidve  $\alpha_1, \alpha_2 \in S^+$  ( $H_1$ ), sú nižšieho stupňa ako  $X^+$ , a teda podľa indukčného predpokladu sú pravdivé vo  $v$ , preto (podľa poz. 5.8) je v ňom pravdivá aj  $\alpha$ .

Ak  $X^+$  je  $\beta$ , potom aspoň jedna z  $\beta_1, \beta_2$  je v  $S^+$  ( $H_2$ ). Nech je to ktorákoľvek, má nižší stupeň ako  $X^+$ , teda podľa IP je pravdivá vo  $v$ , a preto (podľa poz. 5.11) je vo  $v$  pravdivá aj  $\beta$ .  $\square$

## Úplnosť

*Úplnosť kalkulu neformálne:* Ak je nejaké tvrdenie pravdivé, tak existuje jeho dôkaz v kalkule.

**Veta 5.29** (o úplnosti tablového kalkulu [Smullyan, 1979]). *Nech  $S^+$  je konečná nesplniteľná množina označených formúl. Potom existuje uzavreté tablo pre  $S^+$ .*

**Dôsledok 5.30.** *Nech  $S$  je konečná teória a  $X$  je formula. Ak  $S \models_p X$ , tak  $S \vdash_p X$ .*

**Dôsledok 5.31.** *Nech  $X$  je formula. Ak  $\models_p X$ , tak  $\vdash_p X$ .*

Úplnosť platí aj pre nekonečné množiny, ale dôkaz je ťažší.

## Úplnosť — dôkaz

*Dôkaz vety o úplnosti.* Zoberme ľubovoľnú konečnú nesplniteľnú množinu označených formúl  $S^+$ .

Podľa lemy o existencii úplného tabla vieme pre  $S^+$  nájsť úplné tablo  $\mathcal{T}$ , teda také, že každá vetva je buď uzavretá alebo úplná.

Ak by niektorá vetva bola otvorená, potom musí byť úplná, a teda nadol nasýtená. Podľa Hintikkovej lemy by bola splniteľná. Pretože obsahuje všetky formuly z  $S^+$ , bola by aj  $S^+$  splniteľná, čo je spor s nesplniteľnosťou  $S^+$ .

Preto musia byť všetky vetvy tabla  $\mathcal{T}$  uzavreté. □

## 5.6 Nové korektné pravidlá

### Problémy so základnými pravidlami

Základné tablové pravidlá sú jednoduché, ľahko overiteľné a analytické — z (ne)pravdivosti zloženej formuly odvodzujú (ne)pravdivosť jej priamych podformúl.

Nie sú ale úplne pohodlné ani prirodzené, hlavne  $\beta$ .

*Príklad 5.32.* Dokážme, že pre všetky formuly  $A, B, C, X, Y, Z$ :

$$\{(A \rightarrow C), (B \rightarrow C), (C \rightarrow X), (C \rightarrow Y), ((X \wedge Y) \rightarrow Z)\} \\ \vdash_p ((A \vee B) \rightarrow Z)$$

Všimnime si:

- časté použitia pravidla  $\beta$  na implikáciu, kde sa jedna vetva ihneď uzavrie;
- opakovanie jedného podstromu dôkazu.

### Riešenie príkladu 5.32

Tablo pre

$$S^+ = \{ \mathbf{T}(A \rightarrow C), \mathbf{T}(B \rightarrow C), \mathbf{T}(C \rightarrow X), \mathbf{T}(C \rightarrow Y), \mathbf{T}((X \wedge Y) \rightarrow Z), \\ \mathbf{F}((A \vee B) \rightarrow Z) \}$$

<div>1. <math>\mathbf{T}(A \rightarrow C)</math> <math>S^+</math> 2. <math>\mathbf{T}(B \rightarrow C)</math> <math>S^+</math> 3. <math>\mathbf{T}(C \rightarrow X)</math> <math>S^+</math> 4. <math>\mathbf{T}(C \rightarrow Y)</math> <math>S^+</math> 5. <math>\mathbf{T}((X \wedge Y) \rightarrow Z)</math> <math>S^+</math> 6. <math>\mathbf{F}((A \vee B) \rightarrow Z)</math> <math>S^+</math> 7. <math>\mathbf{T}(A \vee B)</math> <math>\alpha 6</math> 8. <math>\mathbf{F}Z</math> <math>\alpha 6</math></div>									
9. $\mathbf{F}(X \wedge Y) \beta 5$									

### Riešenie príkladu 5.32 s modus ponens a modus tolens

1. $T(A \rightarrow C)$	$S^+$
2. $T(B \rightarrow C)$	$S^+$
3. $T(C \rightarrow X)$	$S^+$
4. $T(C \rightarrow Y)$	$S^+$
5. $T((X \wedge Y) \rightarrow Z)$	$S^+$
6. $F((A \vee B) \rightarrow Z)$	$S^+$
7. $T(A \vee B)$	$\alpha 6$
8. $FZ$	$\alpha 6$
9. $F(X \wedge Y)$	MT 5, 8
<hr/>	
10. $TA$	$\beta 7$
11. $TC$	MP 1, 10
12. $TX$	MP 3, 11
13. $TY$	MP 4, 11
<hr/>	
14. $FX$	$\beta 9$
* 12, 14	
15. $FY$	$\beta 9$
* 13, 15	
20. $FX$	$\beta 9$
* 18, 20	
21. $FY$	$\beta 9$
* 19, 21	

### Riešenie príkladu 5.32 s rezom, modus ponens a modus tolens

1. $T(A \rightarrow C)$	$S^+$
2. $T(B \rightarrow C)$	$S^+$
3. $T(C \rightarrow X)$	$S^+$
4. $T(C \rightarrow Y)$	$S^+$
5. $T((X \wedge Y) \rightarrow Z)$	$S^+$
6. $F((A \vee B) \rightarrow Z)$	$S^+$
7. $T(A \vee B)$	$\alpha 6$
8. $FZ$	$\alpha 6$
9. $F(X \wedge Y)$	MT 5, 8
<hr/>	
10. $TC$	cut
11. $TX$	MP 3, 10
12. $TY$	MP 4, 10
<hr/>	
13. $FX$	$\beta 9$
* 11, 13	
14. $FY$	$\beta 9$
* 12, 14	
<hr/>	
15. $FC$	cut
16. $TA$	$\beta 7$
17. $TC$	MP 1, 16
* 15, 17	
18. $TB$	$\beta 7$
19. $FB$	MT 2, 15
* 18, 19	

### Ingrediencie korektnosti a úplnosti tabiel

Všimnite si:

Na dokázanie korektnosti tablového kalkulu stačilo, aby mali pravidlá vlastnosť:

$$\frac{\alpha}{\alpha_1} \quad \frac{\alpha}{\alpha_2} \quad \frac{\beta}{\beta_1 \mid \beta_2} \quad \frac{A^+}{A^+} \quad A^+ \in S^+$$

Nech  $v$  je ľubovoľné ohodnotenie, v ktorom je pravdivá  $S^+$ . Ak je  $vo$  v prav-

divá premisa, tak je vo  $v$  pravdivý aspoň jeden záver.

- Vďaka tejto vlastnosti zo splniteľnej množiny  $S^+$  skonštruujeme iba splniteľné tablá.
- Netreba opačnú implikáciu (ak je vo  $v$  pravdivý aspoň jeden záver, tak je vo  $v$  pravdivá premisa).

Na dôkaz *úplnosti* stačili pravidlá ( $S^+$ ),  $\alpha$ ,  $\beta$ , pretože stačia na vybudovanie úplného tabla.

### Nové pravidlo

Čo sa stane, ak pridáme nové pravidlo, napríklad modus ponens:

$$\frac{\mathbf{T}(X \rightarrow Y) \quad \mathbf{T}X}{\mathbf{T}Y} \quad ? \quad (\text{MP})$$

Upravíme definíciu priameho rozšírenia:

### Úprava definície 5.13

... Nech  $\mathcal{T}$  je tablo pre  $S^+$  a  $y$  je nejaký jeho list. Potom tablom pre  $S^+$  je aj každé *priame rozšírenie*  $\mathcal{T}$  ktorýmkoľvek z pravidiel:

$\alpha$ : ...

:

**MP:** Ak sa na vetve  $\pi_y$  nachádzajú *obe* formuly  $\mathbf{T}(X \rightarrow Y)$  a  $\mathbf{T}X$ , tak ako jediné dieťa  $y$  pripojíme nový vrchol obsahujúci  $\mathbf{T}Y$ .

### Nové pravidlo vs. korektnosť a úplnosť

*Korektnosť* tabiel s MP:

Pri dôkaze lemy K1 (5.21)

Nech  $S^+$  je množina označených formúl v jazyku  $\mathcal{L}$ , nech  $\mathcal{T}$  je tablo pre  $S^+$  a  $v$  je ohodnotenie pre  $\mathcal{L}$ . Ak sú  $S^+$  a  $\mathcal{T}$  pravdivé vo  $v$ , tak je vo  $v$  pravdivé aj každé priame rozšírenie tabla  $\mathcal{T}$ .

využijeme

**Tvrdenie 5.33** (Korektnosť pravidla MP). *Nech  $X$  a  $Y$  sú ľubovoľné formuly a  $v$  je ľubovoľné ohodnotenie. Ak sú vo  $v$  pravdivé  $\mathbf{T}(X \rightarrow Y)$  a  $\mathbf{T}X$ , tak je vo  $v$  pravdivá  $\mathbf{T}Y$ .*

*Dôkaz.* Keďže  $v \models_p \mathbf{T}(X \rightarrow Y)$ , tak  $v \models_p (X \rightarrow Y)$ , teda  $v \not\models_p X$  alebo  $v \models_p Y$ . Pretože ale  $v \models_p \mathbf{T}X$ , tak  $v \models_p X$ . Takže  $v \models_p Y$ , a teda  $v \models_p \mathbf{T}Y$ .  $\square$

Dôkaz lemy K2 (5.22) a samotnej vety o korektnosti (5.17) – bez zmeny.  
*Úplnosť* – bez zmeny, úplné tablo vybudujú základné pravidlá.

### Tablové pravidlá vo všeobecnosti – problém

Zadefinovať vo všeobecnosti, čo je pravidlo a kedy je korektné, nie je také jednoduché.

Potrebuje zachytiť, že pravidlo:

- má premisy, ktoré *nejaký tvar a zdieľajú nejaké podformuly*, napr. moduls tolens (MT) má premisy  $\mathbf{T}(X \rightarrow Y)$  a  $\mathbf{F}Y$ ;
- odvodzuje z nich závery, ktoré tiež zdieľajú podformuly s premisami, napr.  $\mathbf{F}X$  (alebo medzi sebou v prípade rezu).

pre všetky možné zdieľané podformuly, v našom prípade  $X$  a  $Y$ .

### Tablové pravidlá vo všeobecnosti – vzor

Pravidlo sa dá predstaviť nasledovne:

Pravidlo má *vzor* – dvojicu tvorenú vzormi premís a záverov, kde spoločné podformuly predstavujú *konkrétne atómy*, napr. vzor pravidla MT:

$$\frac{\mathbf{T}(p(c) \rightarrow q(c)) \quad \mathbf{F}q(c)}{\mathbf{F}p(c)}$$

### Tablové pravidlá vo všeobecnosti – inštancia

Každý konkrétny prípad — *inštancia* pravidla vznikne *substitúciou* ľubovoľných formúl za atómy vo vzore:

$$\frac{\frac{\mathbf{T}(p(c) \rightarrow q(c))[p(c)|(sedan(a) \wedge biely(a)), q(c)|kupi(B, a)]}{\mathbf{F} q(c)[p(c)|(sedan(a) \wedge biely(a)), q(c)|kupi(B, a)]}}{\mathbf{F} p(c)[p(c)|(sedan(a) \wedge biely(a)), q(c)|kupi(B, a)]}} = \frac{\mathbf{T}((sedan(a) \wedge biely(a)) \rightarrow kupi(B, a))}{\mathbf{F} kupi(B, a)} = \frac{\mathbf{F} kupi(B, a)}{\mathbf{F}(sedan(a) \wedge biely(a))}$$

### Tablové pravidlá vo všeobecnosti — pravidlo

Samotné pravidlo je množina všetkých inšancií vzoru:

$$MT = \left\{ \frac{\mathbf{T}(p(c) \rightarrow q(c))[p(c)|X, q(c)|Y]}{\frac{\mathbf{F} q(c)[p(c)|X, q(c)|Y]}{\mathbf{F} p(c)[p(c)|X, q(c)|Y]}} \mid X, Y \in \mathcal{E}_{\mathcal{L}} \right\}$$

Samozrejme, *konkrétne* pravidlo vieme zapísať aj bez substitúcie:

$$MT = \left\{ \frac{\mathbf{T}(X \rightarrow Y) \quad \mathbf{F} Y}{\mathbf{F} X} \mid X, Y \in \mathcal{E}_{\mathcal{L}} \right\}$$

### Tablové pravidlá vo všeobecnosti

**Definícia 5.34** (Vzor tablového pravidla). Nech  $n \geq 0$  a  $k > 0$  sú prirodzené čísla, nech  $P_1^+, \dots, P_n^+, C_1^+, \dots, C_k^+$  sú označené formuly.

Dvojicu tvorenú  $n$ -ticou  $(P_1^+, \dots, P_n^+)$  a  $k$ -ticou  $(C_1^+, \dots, C_k^+)$  a zapisovanú

$$\frac{P_1^+ \quad \dots \quad P_n^+}{C_1^+ \mid \dots \mid C_k^+}$$

nazývame *vzorom tablového pravidla*.

Označené formuly  $P_1^+, \dots, P_n^+$  nazývame *vzory premís*, označené formuly  $C_1^+, \dots, C_k^+$  nazývame *vzory záverov*.



## Tablové pravidlá vo všeobecnosti

**Definícia 5.35** (Tablové pravidlo a jeho inštancia). Nech

$$\frac{P_1^+ \quad \dots \quad P_n^+}{C_1^+ \quad \dots \quad C_k^+}$$

je vzor tablového pravidla a  $a_1, \dots, a_m$  sú všetky atómy, ktoré sa vyskytujú v označených formulách  $P_1^+, \dots, P_n^+, C_1^+, \dots, C_k^+$ .

Tablové pravidlo  $R$  je množina

$$R = \left\{ \frac{P_1^+_{[a_1|X_1, \dots, a_m|X_m]} \quad \dots \quad P_n^+_{[a_1|X_1, \dots, a_m|X_m]}}{C_1^+_{[a_1|X_1, \dots, a_m|X_m]} \quad \dots \quad C_k^+_{[a_1|X_1, \dots, a_m|X_m]}} \mid X_1, \dots, X_m \in \mathcal{E}_{\mathcal{L}} \right\},$$

Každý prvok množiny  $R$  nazývame *inštanciou* pravidla  $R$ .

## Nové pravidlá vo všeobecnosti

Keď už vieme, čo je pravidlo, môžeme povedať, kedy je korektné:

**Definícia 5.36** (Tablové pravidlo a jeho korektnosť). Tablové pravidlo  $R$  je *korektné* vtt pre každú inštanciu pravidla  $R$

$$\frac{P_1^+ \quad \dots \quad P_n^+}{C_1^+ \quad \dots \quad C_k^+}$$

a pre každé ohodnotenie  $v$  platí, že ak sú vo  $v$  pravdivé *všetky* premisy  $P_1^+, \dots, P_n^+$ , tak je vo  $v$  pravdivý *niektorý* záver  $C_1^+, \dots, C_k^+$ .

## Nové pravidlá vo všeobecnosti

### Úprava definície 5.13

...

- ...
- Nech  $\mathcal{T}$  je tablo pre  $S^+$  a  $y$  je nejaký jeho list. Potom tablom pre  $S^+$  je aj každé *priame rozšírenie*  $\mathcal{T}$  ktorýmkoľvek z pravidiel:

⋮

**R:** Ak sa pre nejakú inštanciu pravidla  $R$

$$\frac{P_1^+ \quad \dots \quad P_n^+}{C_1^+ \mid \dots \mid C_k^+}$$

na vetve  $\pi_y$  nachádzajú všetky premisy  $P_1^+, \dots, P_n^+$ , tak k uzlu  $y$  pripojíme  $k$  nových vrcholov obsahujúcich postupne závery  $C_1^+, \dots, C_k^+$ .

### Príklad: korektnosť rezu

To, že rez

$$\frac{}{TX \mid FX}$$

je korektné pravidlo, dokážeme veľmi ľahko:

**Tvrdenie 5.37** (Korektnosť pravidla rezu). *Nech  $X$  je ľubovoľná formula a  $v$  je ľubovoľné ohodnotenie. Potom je vo  $v$  pravdivý niektorý zo záverov pravidla rezu  $TX$  alebo  $FX$ .*

*Dôkaz.* Formula  $X$  je vo  $v$  buď pravdivá alebo nepravdivá. V prvom prípade  $v \models_p TX$ . V druhom prípade  $v \models_p FX$ . Teda v oboch prípadoch platí, že vo  $v$  je pravdivý niektorý zo záverov  $TX$  alebo  $FX$  pravidla rezu.  $\square$

### Príklad: zložitejšie pravidlá

Príklady zložitejších pravidiel:

- Viacnásobné pravidlá  $\beta$  :

$$\frac{T(A_1 \vee A_2 \vee \dots \vee A_n)}{TA_1 \mid TA_2 \mid \dots \mid TA_n} \qquad \frac{F(A_1 \wedge A_2 \wedge \dots \wedge A_n)}{FA_1 \mid FA_2 \mid \dots \mid FA_n}$$

- Pravidlo konštruktívnej dilemy:

$$\frac{TP \rightarrow Q \quad TR \rightarrow S \quad TP \vee R}{TQ \mid TS}$$

Zistite, či sú tieto pravidlá korektné.

## 5.7 Iné dokazovacie systémy

**i** *Materiál z nasledujúcich slajdov slúži na ilustráciu historických súvislostí a rozšírenie všeobecného prehľadu. Ak ho nepreberáme aj inde, netreba sa ho učiť na skúšku ani na písomky.*

Ukážeme si niekoľko iných dokazovacích systémov pre výrovkovú logiku a porovnáme ich navzájom.

### Hilbertov kalkul

Schémy axióm:

$$A1. \varphi \rightarrow (\psi \rightarrow \varphi)$$

$$A2. (\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$$

$$A3. ((\neg\psi \rightarrow \neg\varphi) \rightarrow (\varphi \rightarrow \psi))$$

Odvodzovacie pravidlo:

$$\frac{\varphi \quad \varphi \rightarrow \psi}{\psi} \quad \text{modus ponens, MP}$$

**Korektnosť:** Všetky axiómy sú tautológie, MP je korektné pravidlo. **FOL:** Áno, pridať 2 axiómy a 2 pravidlá pre kvantifikátory.

### Hilbertov kalkul: príklad dôkazu

Dokážeme  $p \rightarrow p$ .

$$1. (p \rightarrow ((p \rightarrow p) \rightarrow p)) \rightarrow ((p \rightarrow (p \rightarrow p)) \rightarrow (p \rightarrow p)) \quad A2$$

$$2. p \rightarrow ((p \rightarrow p) \rightarrow p) \quad A1$$

$$3. (p \rightarrow (p \rightarrow p)) \rightarrow (p \rightarrow p) \quad \text{MP}(2,1)$$

$$4. p \rightarrow (p \rightarrow p) \quad A1$$

$$5. p \rightarrow p \quad \text{MP}(4,3)$$

- Dôkaz je neprimerane dlhý a zahŕňa formuly podstatne zložitejšie ako dokazovaná. Odkiaľ ich vziať?

## Hilbertov kalkul vs. tablá

- *Korektnosť Hilbertovho kalkulu:* Každá formula v dôkaze je splnená v každej štruktúre (lebo je to tautológia či dôsledok MP). Korektnosť je tak zjavná, priam triviálna.
- Naopak pri tabľách je korektnosť komplikovaná: tablo dáva zmysel len ako celok, jednotlivé formuly v ňom sú bezvýznamné, nevieme nič povedať o ich splniteľnosti (dokonca niektoré vetvy obsahujú spor, ale to je opäť vlastnosť celej vety, nie jednotlivých formúl na nej).
- *Korektnosť tablového kalkulu:* Množina  $S^+$  je splniteľná vtt existuje vetva, v ktorej sú v nejakej štruktúre splnené všetky formuly naraz.
- Tieto (červene vyznačené) invarianty sa zachovávajú aj po pridaní kvantifikátorov.

## Hilbertov kalkul vs. tablá

Pri úplnosti je to zase naopak:

- Pre tablá je pomerne zjavná — tablo rozbiže formulu na atómy a ak by pre tautológiu  $X$  nebol v nejakej vetve tabla pre  $\mathbf{F}X$  spor, priamo si z vetvy prečítame ohodnotenie atómov, v ktorom by  $X$  nebola splnená.
- Pre Hilbertov kalkul vôbec nevidno, prečo by práve uvedené axiómy boli postačujúce. Príklad podobnej sady axióm, ktorá nie je úplná:

$$A1. \varphi \rightarrow (\psi \rightarrow \varphi)$$

$$A2. \neg\varphi \rightarrow (\varphi \rightarrow \psi)$$

$$A3. \neg\neg\varphi \rightarrow \varphi$$

## Sekventový kalkul

**Sekvent** je zápis v tvare  $\Gamma \vdash \Delta$ . Intuitívne: ak platia všetky formuly z  $\Gamma$ , potom platí aspoň jedna formula z  $\Delta$ .

### Základné pravidlá:

- Identita (Ax):  $\varphi \vdash \varphi$

- Oslabenie:  $\frac{\Gamma \vdash \Delta}{\Gamma, \varphi \vdash \Delta}, \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, \varphi}$
- Kontrakcia:  $\frac{\Gamma, \varphi, \varphi \vdash \Delta}{\Gamma, \varphi \vdash \Delta}, \quad \frac{\Gamma \vdash \Delta, \varphi, \varphi}{\Gamma \vdash \Delta, \varphi}$
- Výmena:  $\frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \psi, \varphi \vdash \Delta}, \quad \frac{\Gamma \vdash \Delta, \varphi, \psi}{\Gamma \vdash \Delta, \psi, \varphi}$
- Pravidlá pre log. spojky, napr.  $\rightarrow_L$ :  $\frac{\Gamma \vdash \varphi \quad \psi, \Gamma' \vdash \Delta}{\Gamma, \Gamma', \varphi \rightarrow \psi \vdash \Delta}$

**Korektnosť:** Všetky pravidlá zachovávajú pravdivosť. **FOL:** Áno, pridať 4 pravidlá pre kvantifikátory.

### Sekventový kalkul: príklad dôkazu

Dokážeme sekvent  $p, p \rightarrow q \vdash q$ .

$$\frac{\frac{}{p \vdash p} (Ax) \quad \frac{}{q \vdash q} (Ax)}{p, p \rightarrow q \vdash q} (\rightarrow_L)$$

Pri použití  $\rightarrow_L$  máme  $\Gamma = p, \Gamma' = \emptyset, \varphi = p, \psi = q, \Delta = q$ .

- Nutnosť pravidiel pre výmenu a kontrakciu naznačuje nepríjemnosti pri používaní.
- Dokazované formuly postupne skladáme z jednoduchších častí, nevyskytuje sa použitie „uhádnutých“ dlhých formúl ako pri Hilbertovom kalkule.
- Dôkazy v niektorých variantoch sekventového kalkulu vyzerajú ako tablo obrátené hore nohami.

### Sekventový kalkul: teória typov

- Sekventový kalkul sa využíva v teórii typov. Množina  $\Gamma$  reprezentuje typy existujúcich premenných, t.j. kontext, v rámci ktorého uvažujeme o typoch nových premenných.

- Ukážka pravidiel:

$$\frac{\Gamma \vdash f : A \rightarrow B \quad \Gamma \vdash a : A}{\Gamma \vdash f(a) : B} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash (\lambda x. t) : A \rightarrow B}$$

- Typy v programovacích jazykoch sú netriviálny problém: napr. šablóny v C++ sú turingovsky úplné (v čase kompilácie).
- Funkcionálne jazyky ako Haskell sú cenené pre ľahkú dokázateľnosť správnosti programov. Súčasťou tých dôkazov je presné a dokázateľne korektné uvažovanie o typoch premenných.

## Rezolvencia

**Rezolvencia** operuje nad klauzulami (disjunkciami literálov) a odvodzuje nové klauzuly, kým nenájde spor (prázdnu klauzulu).

Dve pravidlá:

$$\frac{(A \vee C_1 \vee C_2) \quad (\neg A \vee B_1 \vee B_2)}{C_1 \vee C_2 \vee B_1 \vee B_2} \qquad \frac{A \vee A \vee C_1 \vee C_2}{A \vee C_1 \vee C_2}$$

Rezolvenciou možno znížiť počet klauzúl či boolovských premenných (atómov vo formulách), čo naznačuje, ako postupovať pri dôkaze nesplniteľnosti množiny formúl.

**Korektnosť:** Rezolvenčné pravidlo zachováva pravdivosť. **FOL:** Áno, pomocou skolemizácie a unifikácie.

## Rezolvencia: príklad dôkazu

Dokážeme nesplniteľnosť množiny

$$\{A \vee B, \quad \neg A, \quad \neg B\}.$$

1. Rezolvujeme  $A \vee B$  s  $\neg A$ , dostaneme  $B$ .
  2. Rezolvujeme  $B$  s  $\neg B$ , dostávame prázdnu klauzulu, tá je nesplniteľná.
- Vstup pre rezolvenciu je „umelý“, keďže všetko treba prepísať do klauzúl.
  - Aj dôkaz je tak dosť neprirodzený a nekopíruje originálnu formalizáciu.

## Dokazovacie systémy

Existujú iné typy dokazovacích systémov, najmä dvoch druhov:

- Historické, sú prekonané alebo sa neujali, napr. aristotelovské sylogizmy či existenciálne grafy (obrázkový systém, Peirce 1882). Vlastne každý veľký logik v minulosti vymyslel vlastný systém, keďže sme nemali žiadne štandardy.
- Moderné z posledných cca 20 rokov, budúcnosť nejasná.

Pre *výrokovú* logiku sú všetky dokazovacie systémy úplné (ak je niečo pravda, existuje dôkaz) a dôkaz vieme algoritmicky nájsť.

Pre *predikátovú* logiku (s kvantifikátormi) sa dajú použiť všetky okrem SAT solverov. Stále sú úplné, ale dôkaz nemožno algoritmicky hľadať. Tu je veľký priestor pre umelú inteligenciu.

## Dokazovacie systémy

Používané dokazovacie systémy sa dajú zhruba zhrnúť do dvoch skupín:

- *hilbertovské*: „veľa axióm, málo pravidiel“ (bežne len MP), dôkazy sú dlhé, pôsobia umelo a neintuitívne;
- *gentzenovské*: „veľa pravidiel, málo axióm“ (ideálne žiadne), dôkazy sú kratšie a priamočiarejšie (ale štruktúrne zložitejšie, vetvenie miesto lineárnosti) — tablá, prirodzená dedukcia, sekventový kalkul.

Rezolvencia a SAT solvery sú skôr gentzenovské (nemajú axiómy), ale sú zamerané na výpočtový výkon, nie prirodzenosť, priamočiarosť či prehľadnosť dôkazov.

Pre tréningovanie AI má prehľadnosť a generalizovateľnosť veľkú hodnotu aj z hľadiska počítačového spracovania, nielen pre ľudské pochopenie. Napr. AlphaProof sa učil na miliónoch vygenerovaných dôkazov nevelmi zaujímavých geometrických tvrdení v Lean.

## História

1900	Hilbertov program (hľadanie úplnej sady axióm)
1920s	hilbertovský systém (Hilbert, Ackermann)
1930s	prirodzená dedukcia a sekventový kalkul (Gentzen)
1950s	<i>tablo</i> (Beth, Smullyan)
1965	<i>rezolvencia</i> (Robinson)
1970s	Prolog (SLD-rezolvencia, orientovaná na cieľ)
1980s	tablá pre modálnu logiku (musí/môže byť)
1990s	tablá na vrchole popularity
2000s	ústup tabiel, rezolvencia pre FOL s rovnosťou ( <a href="#">dokazovač Vampire</a> ), <i>SAT solvery</i> pre výrokovú logiku
2010s	SMT solvery (SAT + aritmetika + polia, nie kvantifikátory), dominujú vo verifikácii programov
2020s	Lean (based on type theory), AI proof search



## 6. prednáška

# SAT solvery

---

Časti tejto prednášky sa netreba učiť na skúšku, slúžia len na ilustráciu historických či vecných súvislostí a rozšírenie všeobecného prehľadu. Sú označené slovom „*informatívne*“ na slajde alebo v názve podkapitoly.

## 6 SAT solvery

### 6.1 Problém výrokovologickej splniteľnosti (SAT)

#### Problém SAT

**Definícia 6.1** (Problém SAT). *Problémom výrokovologickej splniteľnosti (SAT)* je problém určenia toho, či je daná množina výrokovologických formúl splniteľná.

- Zvyčajne sa redukuje na problém splniteľnosti *klauzálnej* teórie (teda formuly v CNF).
- *SAT solver* je program, ktorý rieši problém SAT.

*Príklad 6.2.* Nech  $a, b, c$  sú predikátové atómy. Nech  $S = \{(a \vee b), (a \vee \neg b), (\neg a \vee b), (\neg a \vee \neg b \vee \neg c), (\neg a \vee c)\}$ . Je množina klauzúl  $S$  splniteľná?

#### Problém SAT

Súvisiace problémy:

- AllSAT — nájsť všetky ohodnotenia, pre ktoré je formula splnená
- #SAT — zistiť počet ohodnotení, pre ktoré je formula splnená
- MaxSAT — zistiť najväčší možný počet klauzúl formuly v CNF, ktoré je možné splniť súčasne
- weighted MaxSAT — klauzuly majú rôznu váhu a maximalizujeme súčet váh splnených klauzúl

- 3-SAT — klauzuly majú  $\leq 3$  literály (NP-ťažké)
- 2-SAT — klauzuly majú  $\leq 2$  literály (P)

## Problém SAT

Praktické využitie:

- verifikácia hardvéru (Intel i7)
- verifikácia softvéru (Windows 7 device drivers)
- manažment softvérových závislostí (Eclipse plugins, Python Conda)
- konfigurácia produktov (Daimler)
- bioinformatika, kryptológia
- expertné systémy, letová kontrola, rozvrhovanie, ...

## História (*informatívne*)

- výrazný pokrok v rokoch 1996–2001, keď sa SAT solvery stali dostatočne rýchle pre praktické využitie
- od r. 2002 každoročne SAT Competition
- o.i. kategória „Glucose hack“ — modifikácia existujúceho solvera nesmie presiahnuť 1000 znakov
- desiatky SAT solverov s otvoreným zdrojovým kódom
- 2013+ SAT *configuration* competition: pre obmedzený okruh vstupov možno dosiahnuť zrýchlenie typicky 2–10× (4,5× pre verifikáciu hardvéru)

## 6.2 Výpočtová zložitosť: teória a prax (informatívne)

### Výpočtová zložitosť — teória

- *zložitosť algoritmu* — počet krokov výpočtu ako funkcia veľkosti vstupu  $n$  (nezávisí od hardvéru)
- *zložitosť problému* — zložitosť optimálneho algoritmu riešiaceho daný problém; je známa len veľmi výnimočne, napr. triedenie porovnávaním je  $O(n \log n)$
- zložitosť porovnávame za predpokladu  $n$  idúceho do nekonečna

### Výpočtová zložitosť — teória

- od cca 1970 problémy delíme na „ľahké“ (známy polynomiálny algoritmus, trieda P) a „ťažké“ (nik nepozná polynomiálny algoritmus, triedy NP, PSPACE...)
- veľa ťažkých problémov patrí do NP: riešenie je možné overiť v polynomiálnom čase
- napriek rozsiahlemu výskumu vôbec nevieme, či  $P \neq NP$
- niektoré problémy sú nerozhodnuteľné (vieme dokázať, že nemôže existovať algoritmus)

### Výpočtová zložitosť — prax

- $2^n$  je lepšie ako  $n^{100}$  (ale ak sme na niečo našli polynomiálny algoritmus, zväčša sme do pár rokov našli aj prakticky použiteľný polyn. algoritmus)
- asymptotické porovnávanie ignoruje konštanty, a tie sú niekedy podstatné (napr. v quicksorte sa nepoužíva lineárny algoritmus na hľadanie mediánu)
- teoreticky najlepšie algoritmy neraz nie sú implementované — sú výhodné len pre obrovské vstupy (ktoré sa možno ani nezmestia do pamäte)

- hardvér je neraz dôležitejší ako algoritmus (hodinky dnes majú viac výkonu ako niekdajšie superpočítače)
- niekedy je podstatný špecializovaný hardvér (napr. Bitcoin mining, AI)

### Výpočtová zložitosť — prax

- strojový čas je lacnejší ako ľudský; komplexita alg. prináša chyby
- niekedy využívame pravdepodobnostné algoritmy, ktoré napr. negarantujú čas behu v najhoršom prípade, ale „takmer vždy“ sú rýchle
- NP-úplné problémy sú teoreticky ekvivalentné, ale v praxi výrazne odlišné (edge colouring vs. circular edge colouring)
- algoritmy s veľkou zložitosťou občas fungujú prekvapivo dobre, najmä ak sú doplnené efektívnymi heuristikami
- klasická teória zložitosti nezohľadňuje nerovnomernú distribúciu vstupov vyskytujúcich sa v praxi
- pre problém splniteľnosti sú praktické vstupy aj 10–100× väčšie, než naznačuje teória

### Problém SAT

- prvý problém s dokázanou NP-úplnosťou
- teoretická zložitosť najlepších algoritmov cca  $1.3^n$  v najhoršom prípade
- ale v praxi riešiteľný pre tisíce až milióny premenných/atómov

## 6.3 Algoritmy na riešenie problému splniteľnosti

### História (*informatívne*)

- hrubá sila (tabuľka všetkých ohodnotení)
- backtracking
- DPLL [1960]
- CDCL (conflict-driven clause learning) [1996]
- watched literals [2001]
- VSIDS heuristic [2001]
- VSIDS combined with machine learning [Maple 2016+]

### Tabuľková metóda

Tabuľková metóda:

- Skúma *všetky* ohodnotenia predikátových atómov
- Trvá  $O(s \cdot 2^n)$  krokov,
  - $n$  je počet atómov a  $s$  je súčet veľkostí klauzúl
  - $2^n$  ohodnotení, pre každé treba zistiť, či sú všetky klauzuly pravdivé
- Zaberá priestor  $O(k \cdot 2^n)$ 
  - $k$  je počet klauzúl
  - Pamätáme si (píšeme na papier) celú tabuľku
- Tabuľka slúži *aj* ako dôkaz prípadnej nesplniteľnosti
  - kratší dôkaz nesplniteľnosti než kompletný záznam činnosti algoritmu riešiaceho SAT zatiaľ nemáme
  - ak by existoval dôkaz s polynomiálnou dĺžkou, bolo by  $NP = coNP$

## 6.4 Backtracking

### Naivný backtracking v Pythone

```
#!/usr/bin/env python3
```

```
class Sat(object):
    def __init__(self, n, clauses):
        self.n, self.clauses, self.solution = n, clauses, None
    def checkClause(self, v, c):
        return any( ( v[abs(lit)] if lit > 0 else not v[abs(lit)] )
                    for lit in c )
    def check(self, v):
        return all( self.checkClause(v, cl) for cl in self.clauses )
    def solve(self, i, v):
        if i >= self.n: # ohodnotili sme vsetky atomy
            if self.check(v):
                self.solution = v
                return True
            return False
        for b in [True, False]:
            v[i] = b
            if self.solve(i+1, v):
                return True
        return False
Sat(20, [[]]).solve(0, {})
```

Čas:  $O(s \cdot 2^n)$ , priestor:  $O(s+n)$ ;

$n$  – počet atómov,

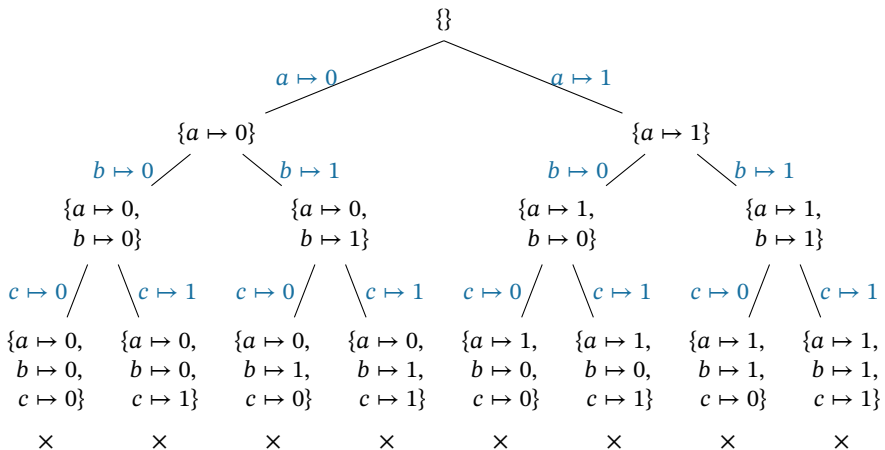
$s$  – súčet veľkostí klauzúl

### Strom prehľadávania ohodnotení

$S = \{(a \vee b), (a \vee \neg b), (\neg a \vee b), (\neg a \vee \neg b \vee \neg c), (\neg a \vee c)\}$

$\times$  znamená  $\not\models_p S$

$f := 0, t := 1$



## Priebežné vyhodnocovanie klauzúl

Strom ohodnotení:

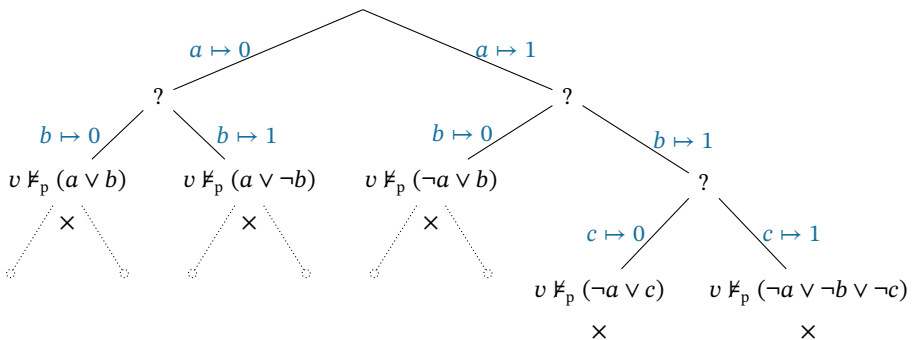
- List — ohodnotenie všetkých premenných
- Každý uzol — čiastočné ohodnotenie
- Ohodnotenie v uzle je *rozšírením* ohodnotenia v rodičovi
- Niektoré klauzuly sa dajú vyhodnotiť aj v čiastočnom ohodnotení
  - V čiastočnom ohodnotení  $v = \{a \mapsto 0, b \mapsto 1\}$  sa dá určiť pravdivosť  $(a \vee b)$ ,  $(a \vee \neg b)$ ,  $(\neg a \vee b)$  z našej  $S$
- Ak nájdeme nepravdivú, môžeme hneď „backtracknúť“ — zastaviť prehľadávanie vetvy a vrátiť sa o úroveň vyššie
  - V čiastočnom ohodnotení  $v = \{a \mapsto 0, b \mapsto 0\}$  je nepravdivá  $(a \vee b)$  z  $S$

## Prehľadávanie s priebežným vyhodnocovaním

$S = \{(a \vee b), (a \vee \neg b), (\neg a \vee b), (\neg a \vee \neg b \vee \neg c), (\neg a \vee c)\}$

$\times$  znamená  $v \not\models_p S$

? znamená zatiaľ žiadna nepravdivá klauzula



## Zjednodušenie množiny klauzúl podľa literálu

Nech  $v$  je čiastočné ohodnotenie, v ktorom  $v(a) = 1$ .

V každom rozšírení ohodnotenia  $v$ :

- sú pravdivé klauzuly obsahujúce  $a$ 
  - $\{a \mapsto 1, \dots\} \models_p (a \vee b)$
  - $\{a \mapsto 1, \dots\} \models_p (a \vee \neg b)$
- je pravdivá klauzula  $(\ell_1 \vee \dots \vee \neg a \vee \dots \vee \ell_n)$  obsahujúca  $\neg a$  vtt je pravdivá *zjednodušená* klauzula  $(\ell_1 \vee \dots \vee \dots \vee \ell_n)$ 
  - $\{a \mapsto 1, \dots\} \models_p (\neg a \vee \neg b \vee \neg c)$  vtt  $\{a \mapsto 1, \dots\} \models_p (\neg b \vee \neg c)$

Takže množinu  $S$  môžeme *zjednodušiť*:

- klauzuly s  $a$  môžeme *vynechať*;
- klauzuly s  $\neg a$  môžeme *zjednodušiť*.

### Zjednodušenie množiny klauzúl podľa literálu

Množinu klauzúl

$$S = \{(a \vee b), (a \vee \neg b), (\neg a \vee b), (\neg a \vee \neg b \vee \neg c), (\neg a \vee c)\}$$

môžeme *zjednodušiť podľa*  $a \mapsto 1$  na

$$S|_{a \mapsto 1} = \{ \quad b, \quad (\neg b \vee \neg c), \quad c \}.$$

Analogicky môžeme  $S$  zjednodušiť podľa  $a \mapsto 0$  na

$$S|_{a \mapsto 0} = \{ \quad b, \quad \neg b \quad \}.$$

### Zjednodušenie množiny klauzúl podľa literálu

**Definícia 6.3.** Nech  $P$  je predikátový atóm,  $S$  je množina klauzúl,  $(t, f)$  je dvojica pravdivostných hodnôt. Potom definujeme

$$S|_P \mapsto f = \{(\ell_1 \vee \dots \vee \dots \vee \ell_n) \mid (\ell_1 \vee \dots \vee P \vee \dots \vee \ell_n) \in S\} \\ \cup \{C \mid C \in S, \text{ v } C \text{ sa nevyskytuje } P \text{ ani } \neg P\}$$

$$S|_P \mapsto t = \{(\ell_1 \vee \dots \vee \dots \vee \ell_n) \mid (\ell_1 \vee \dots \vee \neg P \vee \dots \vee \ell_n) \in S\} \\ \cup \{C \mid C \in S, \text{ v } C \text{ sa nevyskytuje } P \text{ ani } \neg P\}$$

$$S|_{\neg P} \mapsto t = S|_P \mapsto f$$

$$S|_{\neg P} \mapsto f = S|_P \mapsto t$$

**Tvrdenie 6.4.** Nech  $P$  je predikátový atóm,  $S$  je množina klauzúl,  $(t, f)$  dvojica pravdivostných hodnôt. Nech  $b \in \{t, f\}$  a  $v$  je ohodnotenie také, že  $v(P) = b$ . Potom  $v \models_p S$  vtt  $v \models_p S|_P \mapsto b$ .



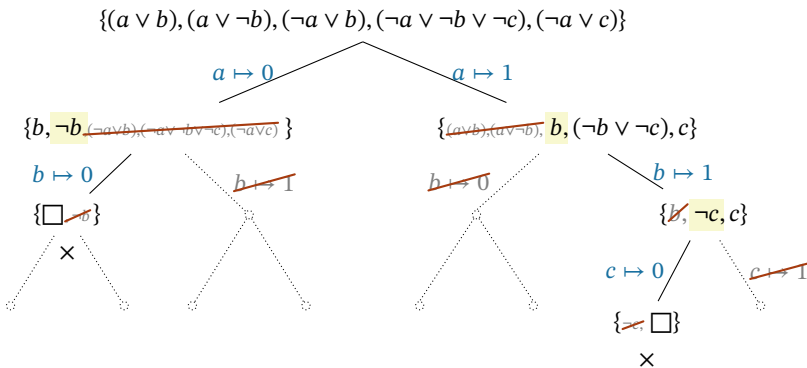
## Propagácia jednotkových klauzúl

Nech  $T = \{(a \vee \neg b), (a \vee b \vee c)\}$ . Začnime zjednodušením podľa  $a \mapsto 0$ :

- $T' := T|_{a \mapsto 0} = \{\neg b, (b \vee c)\}$ 
  - $\neg b$  – jednotková klauzula (unit clause) alebo iba unit
  - $T'$  spĺňajú iba ohodnotenia  $v$ , kde  $v(b) = 0$
  - Takže  $T'$  zjednodušíme podľa  $b \mapsto 0$
- $T'' := T'|_{b \mapsto 0} = \{c\}$ 
  - $c$  – jednotková klauzula
  - $T''$  spĺňajú iba ohodnotenia  $v$ , kde  $v(c) = 1$
  - Takže  $T''$  zjednodušíme podľa  $c$
- $T''' := T''|_{c \mapsto 1} = \{\}$  prázdna, pravdivá v hocijakom ohodnotení.  
Podľa tvrdenia 6.4:
  - $T'''$  je pravdivá v každom ohodnotení, kde  $v(c) = 1$ .
  - $T'$  je pravdivá v každom ohodnotení, kde  $v(b) = 0, v(c) = 1$ .
  - $T$  je pravdivá v ohodnotení  $v = \{a \mapsto 0, b \mapsto 0, c \mapsto 1\}$ .

## Prehľadávanie so zjednodušovaním klauzúl unit propagation

Propagácia jednotkových klauzúl (unit propagation) je proces opakovaného rozširovania ohodnotení podľa jednotkových klauzúl a zjednodušovania.



## Eliminácia nezmiešaných literálov

Všimnime si literál  $P$  v množine klauzúl:

$$T = \{(\neg a \vee \neg b \vee c), (\neg a \vee P), (\neg b \vee P), a, b, \neg c\}$$

Literál  $P$  je *nezmiešaný* (angl. *pure*) v  $T$ :  $P$  sa vyskytuje v  $T$ , ale jeho komplement  $\neg P$  sa tam nevyskytuje.

Nech  $T' := T|_P \mapsto 1 = \{(\neg a \vee \neg b \vee c), a, b, \neg c\}$

- Ak nájdeme ohodnotenie  $v \models_p T'$ , tak  $v_0 := v[P \mapsto 0]$  aj  $v_1 := v[P \mapsto 1]$  sú modelmi  $T'$  a  $v_1$  je navyše modelom  $T$ , teda  $T$  je splniteľná.
- Ak je  $T'$  nesplniteľná, tak je nesplniteľná každá jej nadmnožina, teda aj  $T$ .

Z hľadiska splniteľnosti sú klauzuly obsahujúce  $P$  nepodstatné. Stačí uvažovať  $T|_P \mapsto 1$ .

## Eliminácia nezmiešaných literálov

**Definícia 6.5.** Nech  $P$  je predikátový atóm. Komplementom literálu  $P$  je  $\neg P$ . Komplementom literálu  $\neg P$  je  $P$ .

Komplement literálu  $\ell$  označujeme  $\bar{\ell}$ .

**Definícia 6.6.** Nech  $\ell$  je literál a  $S$  je množina klauzúl. Literál  $\ell$  je *nezmiešaný* (*pure*) v  $S$  vtt  $\ell$  sa vyskytuje v niektorej klauzule z  $S$ , ale jeho komplement  $\bar{\ell}$  sa nevyskytuje v žiadnej klauzule z  $S$ .

**Tvrdenie 6.7.** Nech  $\ell$  je literál a  $S$  je množina klauzúl. Ak  $\ell$  je nezmiešaný v  $S$ , tak  $S$  je splniteľná vtt  $S|_{\ell \mapsto 1}$  je splniteľná.

## 6.5 DPLL a sledované literály

### DPLL

Algoritmus 6.8 (Davis and Putnam [1960], Davis et al. [1962]).

```
1: def DPLL( $\Phi, v$ ):  
2:   if  $\Phi$  obsahuje prázdnu klauzulu:  
3:     return False
```

```

4:   if  $v$  ohodnocuje všetky atómy:
5:     return True
6:   while existuje jednotková (unit) klauzula  $\ell$  vo  $\Phi$ :
7:      $\Phi, v = \text{UNIT-PROPAGATE}(\ell, \Phi, v)$ 
8:   while existuje nezmiešaný (pure) literál  $\ell$  vo  $\Phi$ :
9:      $\Phi, v = \text{PURE-LITERAL-ASSIGN}(\ell, \Phi, v)$ 
10:   $x = \text{CHOOSE-BRANCH-ATOM}(\Phi, v)$ 
11:  return  $\text{DPLL}(\Phi|_x \mapsto t, v(x \mapsto t))$  or  $\text{DPLL}(\Phi|_x \mapsto f, v(x \mapsto f))$ 

```

### Technika sledovaných literálov (watched literals)

Aby sme nemuseli zjednodušovať množinu klauzúl:

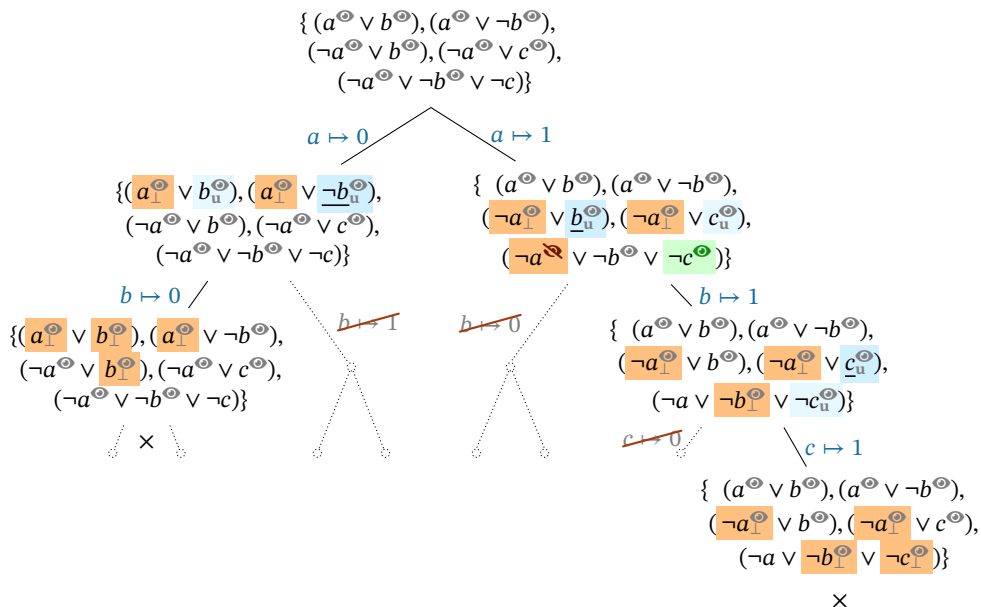
- Pre každú klauzulu vyberieme 2 *sledované literály*.  
 $(\neg a^{\odot} \vee \neg b^{\odot} \vee \neg c)$
- Sledovaný literál musí byť *nenastavený* alebo *true*, ak sa to dá.
- Ak sa sledovaný literál stane *true*: nič nemusíme robiť.  
 $\{a \mapsto 0\} \quad (\neg a^{\odot} \vee \neg b^{\odot} \vee \neg c)$
- Ak sa sledovaný literál stane *false*: musíme nájsť iný.  
 $\{a \mapsto 1\} \quad (\neg a^{\odot} \vee \neg b^{\odot} \vee \neg c^{\odot})$   
 Ak iný nie je, práve sme vyrobili jednotkovú klauzulu  
 (všetky literály okrem druhého sledovaného sú *false*),  
 $\{a \mapsto 1, b \mapsto 1\} \quad (\neg a \vee \neg b^{\odot} \vee \neg c^{\odot})$   
 alebo spor (aj druhý sledovaný je už *false*).  
 $\{a \mapsto 1, b \mapsto 1, c \mapsto 0\} \quad (\neg a^{\odot} \vee \neg c^{\odot})$
- Keď backtrackujeme: nič nemusíme robiť (možno sa niektoré sledované literály stanú *nenastavenými*).

### Technika sledovaných literálov (watched literals)

- netreba v každom kroku prepisovať skúmanú formulu
- pri unit propagation máme priamo odkaz na relevantné klauzuly, nemusíme prepisovať všetky ani hľadať ich vo formule

- žiadna práca pri kroku naspäť
- pre 3-SAT sa ušetrí len málo, preto preferovaná veľkosť klauzúl je výrazne viac ako 3 (dosiahne sa predspracovaním vstupu)
- nezlepšuje asymptotickú zložitosť, ale veľmi užitočné v praxi

## Prehľadávanie s unit propagation a sledovaním

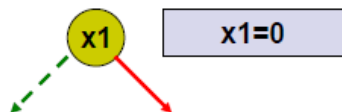


## 6.6 CDCL

### CDCL — conflict-driven clause learning

## Step 1

$x1 + x4$   
 $x1 + x3' + x8'$   
 $x1 + x8 + x12$   
 $x2 + x11$   
 $x7' + x3' + x9$   
 $x7' + x8 + x9'$   
 $x7 + x8 + x10'$   
 $x7 + x10 + x12'$

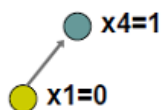
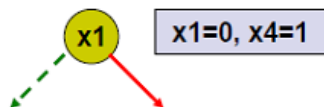


●  $x1=0$

(žltá — rozhodnutie, šedá — unit propagation)

## Step 2

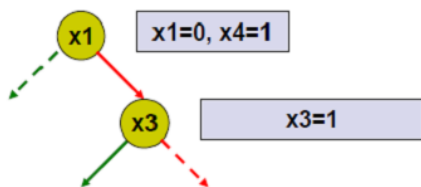
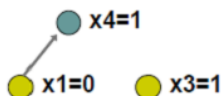
$x1 + x4$   
 $x1 + x3' + x8'$   
 $x1 + x8 + x12$   
 $x2 + x11$   
 $x7' + x3' + x9$   
 $x7' + x8 + x9'$   
 $x7 + x8 + x10'$   
 $x7 + x10 + x12'$



(žltá — rozhodnutie, šedá — unit propagation)

### Step 3

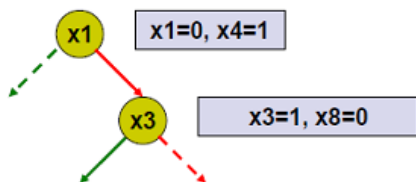
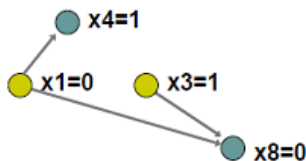
$x1 + x4$   
 $x1 + x3' + x8'$   
 $x1 + x8 + x12$   
 $x2 + x11$   
 $x7' + x3' + x9$   
 $x7' + x8 + x9'$   
 $x7 + x8 + x10'$   
 $x7 + x10 + x12'$



(žltá — rozhodnutie, šedá — unit propagation)

### Step 4

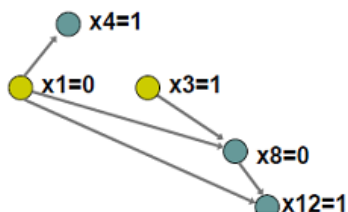
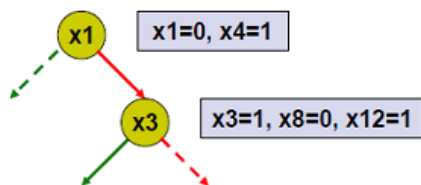
$x1 + x4$   
 $x1 + x3' + x8'$   
 $x1 + x8 + x12$   
 $x2 + x11$   
 $x7' + x3' + x9$   
 $x7' + x8 + x9'$   
 $x7 + x8 + x10'$   
 $x7 + x10 + x12'$



(žltá — rozhodnutie, šedá — unit propagation)

$x1 + x4$   
 $x1 + x3' + x8'$   
 $x1 + x8 + x12$   
 $x2 + x11$   
 $x7' + x3' + x9$   
 $x7' + x8 + x9'$   
 $x7 + x8 + x10'$   
 $x7 + x10 + x12'$

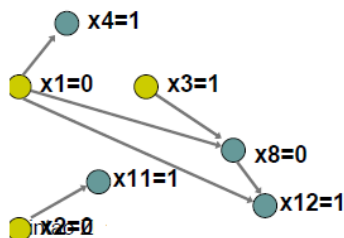
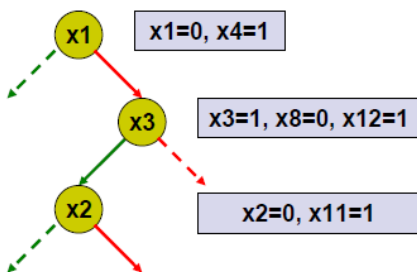
Step 5



(žltá — rozhodnutie, šedá — unit propagation)

$x1 + x4$   
 $x1 + x3' + x8'$   
 $x1 + x8 + x12$   
 $x2 + x11$   
 $x7' + x3' + x9$   
 $x7' + x8 + x9'$   
 $x7 + x8 + x10'$   
 $x7 + x10 + x12'$

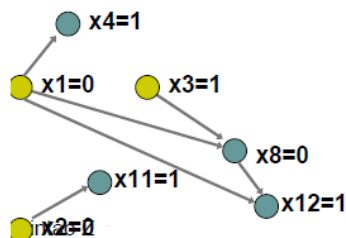
Step 7



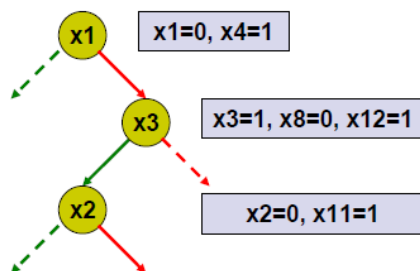
(žltá — rozhodnutie, šedá — unit propagation)

$x1 + x4$   
 $x1 + x3' + x8'$   
 $x1 + x8 + x12$   
 $x2 + x11$   
 $x7' + x3' + x9$   
 $x7' + x8 + x9'$   
 $x7 + x8 + x10'$   
 $x7 + x10 + x12'$

### Step 8

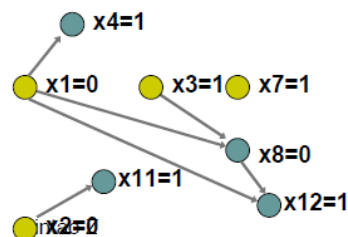


(žltá — rozhodnutie, šedá — unit propagation)

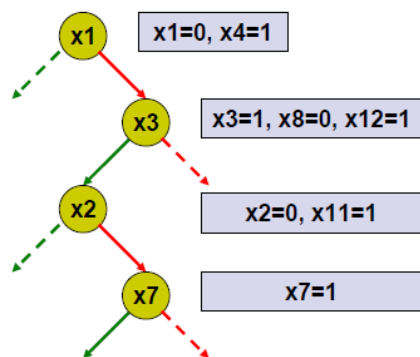


$x1 + x4$   
 $x1 + x3' + x8'$   
 $x1 + x8 + x12$   
 $x2 + x11$   
 $x7' + x3' + x9$   
 $x7' + x8 + x9'$   
 $x7 + x8 + x10'$   
 $x7 + x10 + x12'$

### Step 9



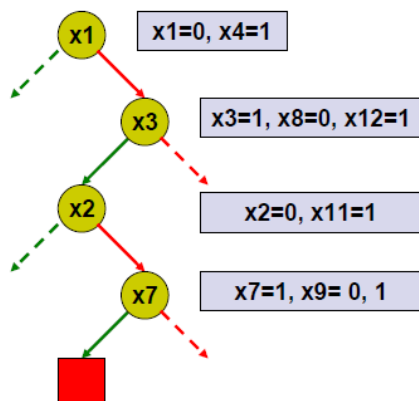
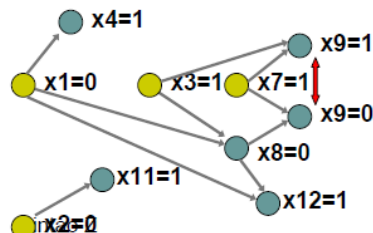
(žltá — rozhodnutie, šedá — unit propagation)





$x1 + x4$   
 $x1 + x3' + x8'$   
 $x1 + x8 + x12$   
 $x2 + x11$   
 $x7' + x3' + x9$   
 $x7' + x8 + x9'$   
 $x7 + x8 + x10'$   
 $x7 + x10 + x12'$

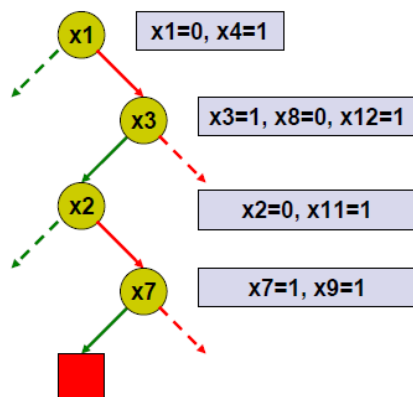
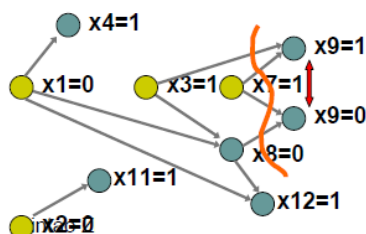
### Step 10



(žltá — rozhodnutie, šedá — unit propagation)

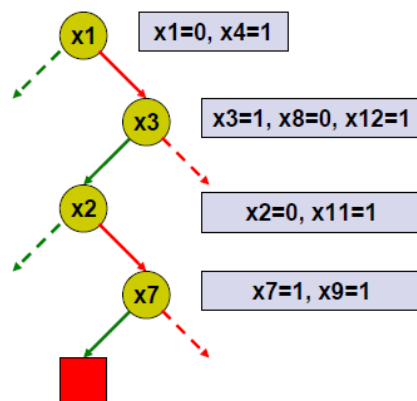
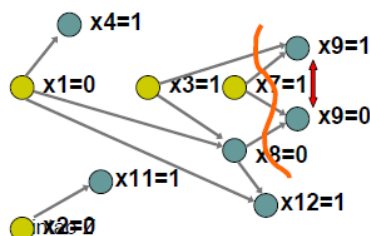
$x1 + x4$   
 $x1 + x3' + x8'$   
 $x1 + x8 + x12$   
 $x2 + x11$   
 $x7' + x3' + x9$   
 $x7' + x8 + x9'$   
 $x7 + x8 + x10'$   
 $x7 + x10 + x12'$

### Step 11



$x1 + x4$   
 $x1 + x3' + x8'$   
 $x1 + x8 + x12$   
 $x2 + x11$   
 $x7' + x3' + x9$   
 $x7' + x8 + x9'$   
 $x7 + x8 + x10'$   
 $x7 + x10 + x12'$

### Step 13



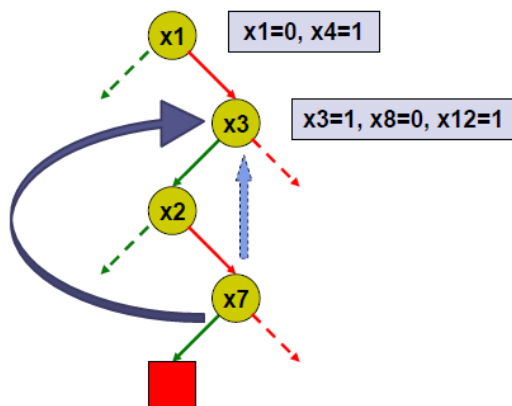
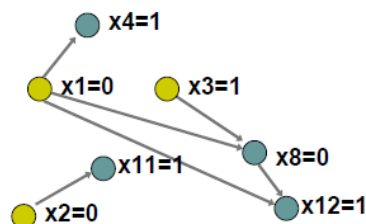
$x3=1 \wedge x7=1 \wedge x8=0 \rightarrow \text{conflict}$

Add conflict clause:  $x3' + x7' + x8$

Uvedený rez nie je jediný, mohli by sme pridať  $x1 \vee \neg x3 \vee \neg x7$ .

$x1 + x4$   
 $x1 + x3' + x8'$   
 $x1 + x8 + x12$   
 $x2 + x11$   
 $x7' + x3' + x9$   
 $x7' + x8 + x9'$   
 $x7 + x8 + x10'$   
 $x7 + x10 + x12'$   
 $x3' + x8 + x7'$

### Step 14

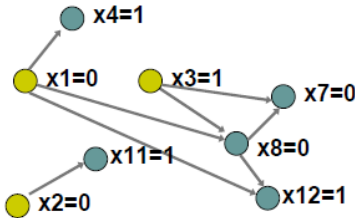
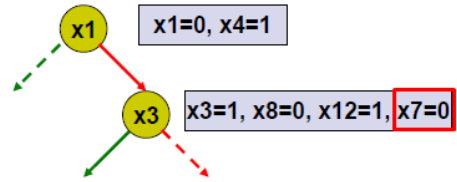


Backtrack to the decision level of  $x3=1$ :  
 $x7 = 0$

Návrat do bodu, kde pridaná klauzula vynúti ohodnotenie jednej doteraz neohodnotenej premennej (čo zabráni vzniku tohto konfliktu kdekoľvek v podstrome).

$x1 + x4$   
 $x1 + x3' + x8'$   
 $x1 + x8 + x12$   
 $x2 + x11$   
 $x7' + x3' + x9$   
 $x7' + x8 + x9'$   
 $x7 + x8 + x10'$   
 $x7 + x10 + x12'$   
 $x3' + x8 + x7'$

Step 15



By Tamkin04iut - asdf, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=25662956>

## CDCL — conflict-driven clause learning

- vytvárame implikačný graf
- keď nájdeme konflikt, zvolíme rez oddelujúci rozhodnutia od konfliktu a odvodíme novú klauzulu, ktorá konfliktu predchádza (*learning*); ak je takých rezov viac, heuristikou niektorý vyberieme
- vrátime sa k predposlednému z rozhodnutí, ktoré viedli ku konfliktu (nie chronologicky — preskočíme rozhodnutia o literáloch nesúvisiacich s konfliktom)

## CDCL — conflict-driven clause learning (*informatívne*)

Problémy (viac v [Zhang]):

- exponenciálne veľa klauzúl, ktoré takto možno odvodiť; ktoré si pamätať a ako dlho? riešenie: rôzne heuristiky, aktívna oblasť výskumu (Kruger et al. [2022])
- čas výpočtu má distribúciu s ťažkým chvostom (fat-tailed — pre niektoré postupnosti rozhodnutí trvá výpočet výrazne dlhšie ako pre iné)

riešenie: občasný reštart backtrackingu (napr. „Luby restarts“, založené na štatistickej analýze náhodných procesov)

### **Ako vybrať nasledujúci literál? (*informatívne*)**

- voľba literálu pre ďalšie rozhodnutie má výrazný efekt na čas výpočtu
- heuristika VSIDS: „additive bumping, multiplicative decay“
- pre každý literál počítame počet jeho výskytov v odvodených klauzulách (t.j. konfliktoch)
- periodicky toto skóre predelíme konštantou (zdôrazníme tak nedávno naučené klauzuly)
- prekvapivo efektívne, využíva sa vo mnohých súčasných solveroch
- heuristika LRB [Maple 2016]: reinforcement learning (multi-armed bandit problem)
- pravidelné prepínanie medzi VSIDS a LRB

## **6.7 Ďalšie aspekty (*informatívne*)**

### **Predspracovanie**

- všetky moderné SAT solvery venujú značnú pozornosť predspracovaniu formuly
- počet premenných je zvyčajne podstatnejší ako veľkosť formuly
- rezolvenciou možno znížiť počet klauzúl (ale narastie ich veľkosť)
- rezolvenciou možno znížiť počet premenných (ale výrazne narastie počet klauzúl)
- poradie klauzúl nemá zásadný vplyv na dĺžku výpočtu
- redundantné klauzuly môžu pomôcť

## Predspracovanie

- desiatky rôznych techník, často doménovo špecifických
- napr. neúplné BDD reprezentácie umožňujú získať klauzuly, ktoré nemožno odvodiť počas CDCL
- cryptominisat akceptuje XOR-klauzuly a pri predspracovaní sa na ne díva ako na sústavu lineárnych rovníc nad  $\mathbb{Z}_2$  a používa Gaussovu elimináciu
- pri „ľahkých“ inštanciách môže predspracovanie zabráť viac času než následné riešenie, treba nájsť vhodný kompromis
- v niektorých prípadoch zase predspracovanie zvyšuje dobu následného riešenia

## Vstupné formuly

- niektoré problémy prirodzene vedú skôr k disjunktívnej normálnej forme, štandardný algoritmus úpravy potom vytvára exponenciálne veľkú CNF
- riešenie: ekvisplniteľné formuly (*equisatisfiable*)

$$\bigvee_i (a_i \wedge b_i \wedge c_i)$$

$$\left( \bigvee_i z_i \right) \wedge \bigwedge_i [(\overline{z_i} \vee a_i) \wedge (\overline{z_i} \vee b_i) \wedge (\overline{z_i} \vee c_i)]$$

(nie ekvivalentné, lebo sú tam premenné navyše, ale jedna je splniteľná práve vtedy, keď druhá)

## Neúplné solvery

- šanca na rýchle objavenie ohodnotenia, v ktorom je formula pravdivá
- neúplné solvery negarantujú dôkaz nesplniteľnosti

- založené na heuristikách (random walks, genetic algorithms, simulated annealing...)
- úspešné využitie metód štatistickej fyziky (napr. survey propagation pre 3-SAT), lebo náhodný SAT vykazuje podobné správanie (*threshold, clustering*)
- automatizované plánovanie: bežne kombinácia neúplného a úplného solvera

### Ďalšie aspekty

- existujúce solvery nie sú dobre paralelizovateľné: vedia použiť mnoho vlákien, ale s otáznym efektom (ak chceme riešiť niekoľko vstupných inštancií, je lepšie riešiť každú v osobitnom vlákne)
- solvery sú konfigurovateľné (lingeling: 300 parametrov); na optimalizáciu na úzkej triede vstupov možno použiť strojové učenie
- zmeny parametrov vedú typicky k zrýchleniu 2–10×

## 6.8 Verifikácia hardvéru (*informatívne*)

### Ukážka aplikácie SAT solverov

- verifikácia hardvéru je azda najvýznamnejšia oblasť využitia — bez moderných procesorov nevieme robiť žiadne iné výpočty
- softvér sa vymení ľahko, vymieňať hardvér je prakticky nemožné alebo neekonomické; nedá sa opraviť časť procesora
- pri desiatkach miliónov tranzistorov nemáme inú dostatočne výkonnú alternatívu

### Metódy verifikácie hardvéru a softvéru

#### 1. Simulácia

- užitočná, ale nič nezaručuje

- je ťažké až nemožné zachytiť všetky možné stavy, v ktorých sa má systém používať

## 2. Formálna verifikácia

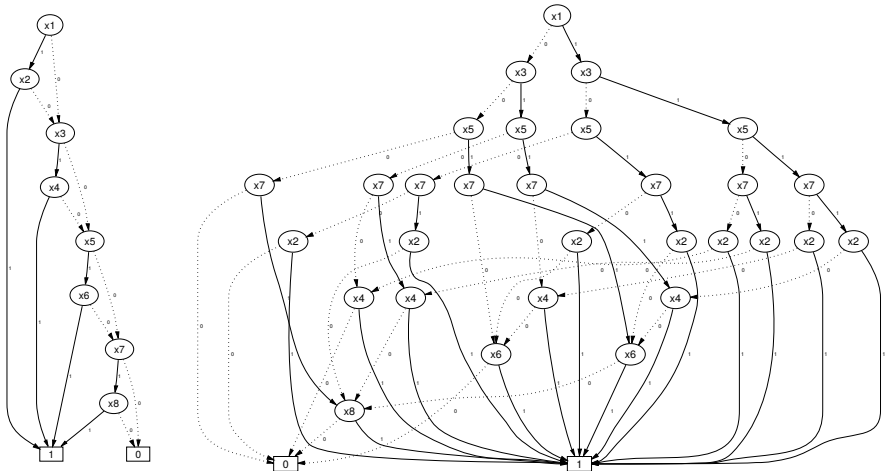
- v princípe úplný matematický dôkaz správnosti
- nedá sa použiť pre fyzickú vrstvu, ale ideálna pre logickú
- používa sa zriedka — drahá a vyžaduje vysokú odbornosť
- atómové elektrárne, vesmírne lety, veľké série procesorov

## Verifikácia hardvéru

- ekvivalencia booleovských výpočtových okruhov (napr. po optimalizácii)
- dôkaz invariantov
- *safety*: is a state reachable?
- *liveness*: is a state  $T$  always reached after  $S$ ?

## Binary decision diagrams (BDDs)

$$f(x_1, \dots, x_8) = x_1x_2 + x_3x_4 + x_5x_6 + x_7x_8$$



## Verifikácia hardvéru: BDDs

- využívané desaťročia
- nevýhody:
  - poradie premenných musí byť vo všetkých vetvách rovnaké
  - poradie má významný efekt na veľkosť diagramu
  - diagramy môžu byť exponenciálne veľké

## Bounded model checking (BMC)

- vyjadrenie verifikačných problémov cez splniteľnosť výrokovologických formúl [Biere et al. 1999]
- rozvineme  $k$  krokov výpočtu, skontrolujeme neporušenosť invariantov, zvýšime  $k$
- vyjadrenie v CNF, rieši sa SAT solverom
- riešiteľné vstupy: 0.4M premenných, 7M klauzúl [2004]

## 6.9 Kombinatorické problémy (*informatívne*)

### Kombinatorické problémy

- pre mnoho problémov v diskkrétnej matematike je SAT solver jediné v súčasnosti použiteľné riešenie
- pre určité problémy sú špecializované solvery rýchlejšie (napr. TSP)
- pre riešiteľné inštancie sú neraz rýchlejšie špecifické heuristiky
- SAT solverom sa darí výborne, ak počet premenných rastie s veľkosťou problému lineárne (napr. farbenia grafov)



### 3-edge-colouring of cubic graphs

Rôzne spôsoby vyjadrenia regulárnosti hranového 3-farbenia pre graf, ktorý má všetky vrcholy stupňa 3:

1. incidentné hrany majú navzájom rôzne farby
2. v každom vrchole je každá farba použitá na práve jednej hrane
3. farby hrán incidentných s každým jedným vrcholom tvoria trojicu z povoleného pevného zoznamu trojíc
4. formula založená na combinatorial nullstellensatz

### 3-edge-colouring of cubic graphs

- $O(n)$  premenných,  $O(n)$  klauzúl, veľkosť formuly  $O(n)$
- čas výpočtu pre rôzne solvery a ich konfigurácie: nízka variácia, cca 4×
- voľne koreluje s veľkosťou formuly
- všetky možnosti fungujú lepšie ako formulácia cez ILP (integer linear programming) a riešenie GLPK či Gurobi
- pre grafy do 50–100 vrcholov vyhráva backtracking (najmä ak sú za-farbitelné)
- SAT solvery fungujú aj pre tisíce vrcholov
- absencia krátkych kružníc v grafe (čiže lokálne vyzerá ako strom) pre-dlžuje výpočet

### Symetria

- Ak je vstupná formula symetrická (má mnoho automorfizmov), solver bude opakovane preverovať „rovnaké“ časti priestoru možných riešení.

- Oplatí sa symetriu odstraňovať, napr. odstránime permutácie farieb, predpíšeme na fixnú hranu konkrétnu farbu, prvý vrchol na hľadanej hamiltonovskej kružnici apod.
- Symetrické premenné možno zoradiť lexikograficky a pridať klauzuly popisujúce  $x_1 \geq x_2 \geq x_3 \geq \dots$  (čiže ak by pre niektoré spĺňajúce ohodnotenie boli niektoré  $x_i$  pravdivé, vyberieme také ohodnotenie, že to postupne budú  $x_1, x_2, \dots$ ).
- Existujú špecializované knižnice práve na tento účel (napr. BreakID), je za tým 30 rokov výskumu.

### Hamiltonovské kružnice

1. premenné  $x_{v,i}$  — true ak  $v$  je  $i$ -ty vrchol kružnice
  2. každá pozícia na kružnici má priradený vrchol
  3. žiadne dva vrcholy nemajú priradenú tú istú pozíciu
  4. každý vrchol je použitý najviac raz
  5. každé dva po sebe idúce vrcholy na kružnici sú spojené hranou
- $O(n^2)$  premenných,  $O(n^3)$  klauzúl!
  - pre kubické grafy funguje do 40–50 vrcholov, podobne ako backtracking
  - ak má byť redukcia na SAT naozaj efektívna, potrebujeme cca lineárne veľa premenných
  - premenných aspoň  $n \log n$ , lebo kódujeme permutácie do bitov (ale neplatí pre postupy, ktoré kružnicu priamo nehľadajú, len akosi inak rozhodujú o existencii)

## Hamiltonovské kružnice

- hamiltonovská kružnica je súvislý 2-faktor (podgraf s vrcholmi stupňa práve 2)
- pomocou booleovskej formuly možno stupeň vrcholu ľahko popísať lokálne
- CNF s veľkosťou  $O(n)$
- stačí overiť súvislosť každého 2-faktora — AllSAT
- málo dostupných solverov: clasp, BDD\_MINISAT\_ALL
- pre skúmané grafy rýchlejšie ako redukcia na SAT, ale počet 2-faktorov rastie exponenciálne
- pre kubické grafy funguje do cca 40 vrcholov (milióny 2-faktorov)

## AllSAT

- z každého SAT solvera možno spraviť AllSAT solver (stačí po každom nájdenom riešení pridať na vstup klauzulu, ktorá ho zakazuje)
- neefektívne, klauzuly objavené CDCL zakaždým zahodíme, keď na novo štartuje výpočet po objavení riešenia
- vstupná formula narastá príliš rýchlo (napr. pre CryptoMiniSat možno takto realisticky nájsť desaťtisíce riešení, ale nie milióny)
- iná možnosť: obmedziť skoky pre non-chronological backtracking

## AllSAT

- doteraz najlepšie riešenie: *formula-BDD caching* [Toda 2015]
- dá sa pridať k akémukoľvek backtrackingu, ak vopred zafixujeme poradie premenných

- zruší prínosy VSIDS (vyberáme si len hodnotu, nie premennú, ktorú ideme ohodnotiť)
- BDD\_MINISAT\_ALL je náročný na pamäť, ale ako jediný dokáže pracovať s miliardami riešení
- prekvapivo vie v niektorých prípadoch dokázať nespĺniteľnosť rýchlejšie ako SAT solvery

## 6.10 SMT solvery (*informatívne*)

### Motivácia

- SAT solvery sú veľmi rýchle pre booleovskú logiku (log. spojky).
- Booleovské formuly však ponúkajú len obmedzené vyjadrovacie možnosti, napr. aritmetika sa dá robiť bitovo, ale je to pomalé, zložité a vzdialené od pôvodných pojmov, ktoré modelujeme.
- Problémy reálneho sveta si často vyžadujú bohatšie teórie, napr. s poliami a funkciami.
- SMT solvery rozširujú SAT tak, aby riešili tieto komplexné požiadavky.

### Čo je SMT solver?

- *SMT* — Satisfiability Modulo Theories.
- SMT solver rozhoduje splniteľnosť formuly s výrokovologickou štruktúrou, ale zároveň berie do úvahy doplnkové teórie (*background theories*), ktoré sa vymykajú výrokovej logike:
  - Linear Integer Arithmetic (LIA)
  - Uninterpreted Functions (UF)
  - polia, bitové vektory, dátové typy
- Background theories sú fragmentmi prvorádovej logiky, ktoré zahŕňajú len špecifické predikáty so štandardnou interpretáciou a vyhýbajú sa tak algoritmickej nerozhodnuteľnosti.

## SMT formuly

SMT formuly:

- Obsahujú premenné rôznych typov, napr. celé čísla, reálne čísla, polia.
- Kombinujú logickú štruktúru ( $\wedge, \vee, \neg$ ) s atómami pochádzajúcimi z doplnkových teórií.
- Príklad (LIA + UF + Boolean):

$$\underbrace{(a + b \leq 5)}_{\text{LIA}} \vee \underbrace{(f(a) = g(f(b)))}_{\text{UF}} \wedge \underbrace{(a \geq 0)}_{\text{LIA}} \wedge \underbrace{(c = \text{True})}_{\text{Boolean}}$$

- SAT solver nerozumie vnútornej štruktúre atómov, len vie, ktorý podriadený solver daný atóm rozozná.

## Algoritmus DPLL(T)

1. **SAT Solver:** navrhuje ohodnotenia atómov.
2. **Theory Solver:** overuje konzistentnosť týchto ohodnotení vzhľadom na doplnkové teórie.
3. Interakcia:
  - SAT solver vykonáva DPLL/CDCL.
  - Ak SAT solver nájde spĺňajúce ohodnotenie, zavolá podriadený solver pre príslušnú teóriu, aby skontroloval, či je priradenie prípustné.
  - Ak podriadený solver odhalí nepripustnosť, SAT solver sa naučí novú klauzulu — *theory learn step*.

## Algoritmus DPLL(T): príklad

- Vstupná formula obsahuje LIA atómy

$$A_1 = (x = 4), \quad A_2 = (x \geq 0), \quad A_3 = (y = x+2), \quad A_4 = (y > 5).$$

- SAT solver navrhne, že  $A_1, A_2, A_3$  budú true,  $A_4$  false.
- LIA solver nájde nesúlad pre  $A_1, A_3, \neg A_4$ .
- SAT solver pridá klauzulu  $\neg A_1 \vee \neg A_3 \vee A_4$  (*theory learn step*).

## Aplikácie

Využitie:

- automatizované dokazovanie tvrdení,
- formálna verifikácia softvéru,
- overovanie hardvéru,
- testovanie a ladenie softvéru.

Napr. Microsoft vyvinul Z3 a niekoľko ďalších nástrojov založených na SMT; použil ich v systémoch ako SAGE na odhalenie viac ako 1000 kritických bezpečnostných chýb vo Windows a Office. Amazon používa internú knižnicu TLS s názvom s2n, navrhnutú špeciálne tak, aby bola malá a dala sa formálne verifikovať; denne spúšťajú miliardy výpočtov SMT na všetko od bootovacieho kódu AWS a operačných systémov IoT až po kryptografické knižnice a sieťové konfigurácie ([link](#)).

## 7. prednáška

# Kvantifikátory

---

## 7 Kvantifikátory

### 7.1 Kvantifikácia

#### Prívlastky

Doteraz sme sa stretávali s prívlastkami, ktoré vyjadrovali vlastnosti alebo vzťahy *konkrétnych jednotlivých* objektov.

- Jurko kŕmi *veľkú Vierkinu* myš Ňufka.  
(kŕmi(Jurko, Ňufko)  $\wedge$  veľký(Ňufko)  $\wedge$  patrí(Ňufko, Vierka)  $\wedge$  myš(Ňufko))

#### Kvantifikované tvrdenia

V slovenských vetách sa ale používajú aj prívlastky ako *každý*, *nejaká*, *tri*, *tí*, *všetky*, *žiadny*, *nijaké* (gramaticky sú to zámená a číslovky).

- všetky veľké Vierkine myši; nejaké dieťa; traja muži v člne; žiadny Košičan; väčšina škrečkov; tá skriňa v kúte; ...

Nevyjadrujú vlastnosť konkrétnych objektov.

Vyjadrujú počet (*kvantitu*) objektov, ktoré majú nejaké vlastnosti alebo sú v nejakých vzťahoch.

Tvrdeniam, ktoré obsahujú tieto prívlastky, sa preto v logike hovorí *kvantifikované tvrdenia*.

#### Kvantifikácia a logické dôsledky

Kvantifikujúci prívlastok výrazne mení logické vlastnosti tvrdenia:

<i>Všetky</i> myši sú sivé. Ňufko je myš.	<i>Väčšina</i> myši je sivá. Ňufko je myš.	<i>Žiadne</i> myši nie sú sivé. Ňufko je myš.
Ňufko je sivý.	Ňufko je sivý.	Ňufko je sivý.
Je logický dôsledok.	Nie je log. dôsledkom, ale je prijateľné.	Nie je log. dôsledkom, ani prijateľné. Opak je pravdou.

Kvantifikácia sa nespráva ako funkcia na pravdivostných hodnotách — na rozdiel od logických spojok.

Vyjadruje vzťah súborov (tried, množín) objektov (tých objektov, ktoré sú myšami, a tých objektov, ktoré sú sivé).

### Skrytá kvantifikácia

Niektoré spojky a vzťahy implicitne vyjadrujú kvantifikáciu:

- Jurko kŕmi Ňufka, iba *keď* je noc.  
Jurko kŕmi Ňufka *vždy* v noci.  
V *každej* chvíli, v ktorej Jurko kŕmi Ňufka, je noc.
- V pondelok cvičí Klárka hru na flautu.  
V *každý* deň, ktorý je pondelkom, cvičí Klárka hru na flautu.
- Z *P* logicky vyplýva *Q*.  
V *každom* stave sveta, v ktorom je pravdivé *P*, je pravdivé aj *Q*.

## 7.2 Kvantifikátory a premenné

### Kvantifikátory logiky prvého rádu

Logika prvého rádu má iba dva symboly kvantifikátorov:  $\forall$  a  $\exists$ .

Zodpovedajú zámenám *všetko* a *niečo*.

S pomocou predikátov, výrokovologických spojok a rovnosti ale dokážu vyjadriť napr. kvantifikácie:

- všetky veľké Vierkine myši; nejaké dieťa; traja muži v člne; žiadny Košičan; zakaždým, keď.

Nedokážeme však nimi vyjadriť:



- väčšina škrekčov; málo študentov; nekonečne veľa prvočísel.

Kvantifikované *premenné označujú výhradne objekty z domény*, nemožno kvantifikovať množiny objektov ani nič zložitejšie.

## Premenné

Na vyjadrenie toho, na ktoré argumenty predikátov sa vzťahuje kvantifikátor, sa používajú individuové premenné.

*Individuová premenná*

- môže byť argumentom predikátu, *podobne* ako individuová konštanta;
- neoznačuje konkrétny objekt, *na rozdiel* od individuovej konštanty, ale prepája argumenty predikátov, na ktoré sa vzťahuje ten istý kvantifikátor.

V každom prvorádovom jazyku s kvantifikátormi je *nekonečne veľa* premenných — väčšinou malé písmená z konca abecedy, podľa potreby s dolnými indexmi:  $u, v_4, w, x, y_{37}, z_{123}$ .

## Termy a atómy

Možné argumenty predikátov a rovnosti, teda premenné a konštanty, súhrnne nazývame *termy*.

*Atomickými formulami* logiky prvého rádu s kvantifikátormi sú potom

- predikátové atómy  $\text{predikát}(term_1, \dots, term_k)$ , kde  $k$  je arita predikátu;
- rovnostné atómy  $term_1 \doteq term_2$ .

## Všeobecný kvantifikátor

*Všeobecný kvantifikátor*  $\forall$  zodpovedá obratom *všetko, každý/ktorýkoľvek/akýkoľvek/hociktorý/lubovoľný objekt, všetky objekty*.

Vždy *viaže* premennú uvedenú bezprostredne za ním.

Postupnosť  $\forall x$  čítame „*pre každý objekt  $x$* “ (alebo trocha nepresne „*pre každé  $x$* “).

*Oblasť platnosti* všeobecného kvantifikátora — *najkratšia ucelená formula nasledujúca bezprostredne za viazanou premennou* — vyjadruje vlastnosť, ktorú prisudzujeme všetkým objektom, napr.:

- $\forall x \text{ doma}(x)$  — Pre každý objekt  $x$  je pravda, že  $x$  je doma. (Všetko je doma.) Veta „ $x$  je doma“ je *výroková forma*, nie výrok. Jej pravdivosť sa dá jednoznačne určiť, iba keď poznáme hodnotu  $x$ .
- $\forall x(\text{človek}(x) \rightarrow \text{doma}(x))$  — Pre každý objekt  $x$  je pravda, že ak  $x$  je človek, tak  $x$  je doma. (Každý človek je doma.)

### Existenčný kvantifikátor

*Existenčný kvantifikátor  $\exists$  zodpovedá obratom niečo, nejaký/niektorý/akýsi/ aspoň jeden objekt, je/existuje taký objekt.*

Vždy viaže premennú uvedenú bezprostredne za ním.

Postupnosť  $\exists x$  čítame „pre nejaký objekt  $x$ “ (alebo trochu nepresne „pre nejaké  $x$ “).

*Oblasť platnosti existenčného kvantifikátora — najkratšia ucelená formula nasledujúca bezprostredne za viazanou premennou — vyjadruje vlastnosť, o ktorej tvrdíme, že ju má aspoň jeden objekt:*

- $\exists x \text{ doma}(x)$  — Pre nejaký objekt  $x$  je pravda, že  $x$  je doma. (Niečo je doma.)
- $\exists x(\text{človek}(x) \wedge \text{doma}(x))$  — Pre nejaký objekt  $x$  je pravda, že  $x$  je človek a  $x$  je doma. (Nekajý človek je doma.)

## 7.3 Syntax relačnej logiky prvého rádu

### Symbody jazyka relačnej logiky prvého rádu

**Definícia 7.1.** *Symbolmi jazyka  $\mathcal{L}$  relačnej logiky prvého rádu sú:*

*individuové premenné z nejakej nekonečnej spočítateľnej množiny  $\mathcal{V}_{\mathcal{L}}$ ;*

*mimologické symboly, ktorými sú*

*individuové konštanty z nejakej spočítateľnej množiny  $\mathcal{C}_{\mathcal{L}}$ ;*

*predikátové symboly z nejakej spočítateľnej množiny  $\mathcal{P}_{\mathcal{L}}$ ;*

*logické symboly, ktorými sú*

*logické spojky: unárna  $\neg$ , binárne  $\wedge, \vee, \rightarrow$ ,*

symbol rovnosti  $\doteq$ ,

kvantifikátory: existenčný  $\exists$  a všeobecný  $\forall$ ;

pomocné symboly  $(, )$  a  $(\text{'}, \text{'})$  (ľavá, pravá zátvorka a čiarka).

Množiny  $\mathcal{V}_{\mathcal{L}}, \mathcal{C}_{\mathcal{L}}, \mathcal{P}_{\mathcal{L}}$  sú vzájomne disjunktné. Logické a pomocné symboly sa nevyskytujú v symboloch z  $\mathcal{V}_{\mathcal{L}}, \mathcal{C}_{\mathcal{L}}, \mathcal{P}_{\mathcal{L}}$ .

Každému symbolu  $P \in \mathcal{P}_{\mathcal{L}}$  je priradená arita  $\text{ar}(P) \in \mathbb{N}^+$ .

### Označovanie symbolov rôznych druhov

Keď budeme hovoriť o *ľubovoľnom* jazyku  $\mathcal{L}$ , často budeme potrebovať nejak označiť niektoré jeho konštanty alebo predikáty, aj keď nebudeme vedieť, aké konkrétne symboly to sú.

Na označenie symbolov použijeme *meta premenné*: premenné v (matematickej) slovenčine, pomocou ktorých budeme hovoriť o (po grécky *meta*) týchto symboloch.

*Dohoda 7.2.* Individuové premenné budeme spravidla označovať meta premennými  $u, v, w, x, \dots, z$  s prípadnými dolnými indexmi.

Individuové konštanty budeme spravidla označovať meta premennými  $a, b, c, d$  s prípadnými dolnými indexmi.

Predikátové symboly budeme spravidla označovať meta premennými  $P, Q, R$  s prípadnými dolnými indexmi.

### Atomické formuly relačnej logiky prvého rádu

**Definícia 7.3** (Term). Nech  $\mathcal{L}$  je jazyk relačnej logiky prvého rádu. Individuové premenné z  $\mathcal{V}_{\mathcal{L}}$  a konštanty z  $\mathcal{C}_{\mathcal{L}}$  súhrnne nazývame *termy* jazyka  $\mathcal{L}$ .

**Definícia 7.4** (Atomické formuly). Nech  $\mathcal{L}$  je jazyk relačnej logiky prvého rádu.

*Rovnostný atóm* jazyka  $\mathcal{L}$  je každá postupnosť symbolov  $t_1 \doteq t_2$ , kde  $t_1$  a  $t_2$  sú termy jazyka  $\mathcal{L}$ .

*Predikátový atóm* jazyka  $\mathcal{L}$  je každá postupnosť symbolov  $P(t_1, \dots, t_n)$ , kde  $P$  je predikátový symbol s aritou  $n$  a  $t_1, \dots, t_n$  sú termy jazyka  $\mathcal{L}$ .

*Atomickými formulami* (skrátene *atómami*) jazyka  $\mathcal{L}$  súhrnne nazývame všetky rovnostné a predikátové atómy jazyka  $\mathcal{L}$ .

Množinu všetkých atómov jazyka  $\mathcal{L}$  označujeme  $\mathcal{A}_{\mathcal{L}}$ .

## Formuly jazyka relačnej logiky prvého rádu

**Definícia 7.5.** Množina  $\mathcal{E}_{\mathcal{L}}$  všetkých *formúl* jazyka relačnej logiky prvého rádu  $\mathcal{L}$  je *najmenšia* množina postupností symbolov jazyka  $\mathcal{L}$ , ktorá spĺňa všetky nasledujúce podmienky:

1. Každý atóm z  $\mathcal{A}_{\mathcal{L}}$  je formulou z  $\mathcal{E}_{\mathcal{L}}$ . Inak povedané,  $\mathcal{A}_{\mathcal{L}} \subseteq \mathcal{E}_{\mathcal{L}}$ .
2. Ak  $A$  patrí do  $\mathcal{E}_{\mathcal{L}}$ , tak aj postupnosť symbolov  $\neg A$  patrí do  $\mathcal{E}_{\mathcal{L}}$  a nazývame ju *negácia* formuly  $A$ .
3. Ak  $A$  a  $B$  sú v  $\mathcal{E}_{\mathcal{L}}$ , tak aj postupnosti symbolov  $(A \wedge B)$ ,  $(A \vee B)$  a  $(A \rightarrow B)$  patria do  $\mathcal{E}_{\mathcal{L}}$  a nazývame ich postupne *konjunkcia*, *disjunkcia* a *implikácia* formúl  $A$  a  $B$ .
4. Ak  $x$  je individuová premenná a  $A$  patrí do  $\mathcal{E}_{\mathcal{L}}$ , tak aj postupnosti symbolov  $\exists x A$  a  $\forall x A$  patria do  $\mathcal{E}_{\mathcal{L}}$  a nazývame ich postupne *existenčná* a *všeobecná kvantifikácia* formuly  $A$  vzhľadom na  $x$ .

Každý prvok  $A$  množiny  $\mathcal{E}_{\mathcal{L}}$  nazývame *formulou* jazyka  $\mathcal{L}$ .

## Príklady formúl

*Príklad 7.6.* Nech  $\mathcal{L}$  je prvorádový jazyk, v ktorom  $\mathcal{V}_{\mathcal{L}} = \{x, y, x_1, y_1, x_2, y_2, \dots\}$ ,  $\mathcal{C}_{\mathcal{L}} = \{\text{Jurko, Vierka, Ľufko}\}$   $\mathcal{P}_{\mathcal{L}} = \{\text{myš}^1, \text{škrečok}^1, \text{biely}^1, \text{patrí}^2\}$ .

Formulami v jazyku  $\mathcal{L}$  sú napríklad:  $\text{myš}(\text{Jurko})$ ,  $\text{myš}(x)$ ,  $\text{patrí}(y_2, \text{Vierka})$ ,  $\text{patrí}(x, y)$ ,  $(\text{myš}(x) \wedge \text{biely}(x))$ ,

$\exists y \text{patrí}(x, y)$ ,

$((\text{myš}(x) \wedge \text{biely}(x)) \rightarrow \exists y \text{patrí}(x, y))$ ,

$\forall x((\text{myš}(x) \wedge \text{biely}(x)) \rightarrow \exists y \text{patrí}(x, y))$

## Označovanie formúl a skratka ekvivalencie

Stále platia doterajšie dohody:

*Dohoda 7.7.* Formuly označujeme meta premennými  $A, B, C, X, Y, Z$ , s prípadnými dolnými indexmi.

*Dohoda 7.8.* Pre každú dvojicu formúl  $A, B \in \mathcal{E}_{\mathcal{L}}$  je zápis  $(A \leftrightarrow B)$  skratka za formulu  $((A \rightarrow B) \wedge (B \rightarrow A))$ .

## Oblasť platnosti kvantifikátora

*Dohoda 7.9.* Nech  $\mathcal{L}$  je ľubovoľný jazyk logiky prvého rádu. Všetky symboly, termy a formuly v nasledujúcich definíciách a tvrdeniach sú v jazyku  $\mathcal{L}$ .

**Definícia 7.10** (Oblasť platnosti kvantifikátora). Nech  $A$  je postupnosť symbolov, nech  $B$  je formula, nech  $Q \in \{\forall, \exists\}$ , nech  $x$  je premenná. V postupnosti  $A = \dots Qx B \dots$  sa výskyt formuly  $Qx B$  nazýva *oblasť platnosti kvantifikátora*  $Qx$  v  $A$ .

*Príklad 7.11.* Vyznačme všetky oblasti platnosti kvantifikátora  $\forall x$  vo formule

$$((\forall x M(x) \wedge P(x, x)) \rightarrow (\forall x (P(x, y) \wedge \exists y M(y)) \vee \forall y M(y))).$$

*Riešenie.*  $((\forall x M(x) \wedge P(x, x)) \rightarrow (\forall x (P(x, y) \wedge \exists y M(y)) \vee \forall y M(y)))$

## Voľné a viazané výskyty premenných

**Definícia 7.12** (Voľné a viazané výskyty premenných). Nech  $A$  je postupnosť symbolov, nech  $x$  je premenná.

Výskyt premennej  $x$  v  $A$  je **viazaný** vtt sa *nachádza* v *niektorej* oblasti platnosti kvantifikátora  $\forall x$  alebo  $\exists x$  v  $A$ .

Výskyt premennej  $x$  v  $A$  je **voľný** vtt sa *nenachádza* v *žiadnej* oblasti platnosti kvantifikátora  $\forall x$  ani  $\exists x$  v  $A$ .

*Príklad 7.13.*

$$\begin{aligned} & \neg P(\underline{x}, \underline{y}) \wedge K(\underline{y}, \underline{x}) \\ & \neg P(\underline{x}, \underline{y}) \wedge \exists \underline{y} K(\underline{y}, \underline{x}) \\ & \exists \underline{y} (\neg P(\underline{x}, \underline{y}) \wedge K(\underline{y}, \underline{x})) \\ & \forall \underline{x} \exists \underline{y} (\neg P(\underline{x}, \underline{y}) \wedge K(\underline{y}, \underline{x})) \\ & \forall \underline{x} (\neg P(\underline{x}, \underline{y}) \wedge \exists \underline{y} K(\underline{y}, \underline{x})) \end{aligned}$$

## Volné a viazané premenné

**Definícia 7.14** (Volné a viazané premenné). Nech  $A$  je formula alebo term, nech  $x$  je premenná.

Premenná  $x$  je *viazaná* v  $A$  vtt  $x$  sa vyskytuje v  $A$  a všetky výskyty  $x$  v  $A$  sú viazané.

Premenná  $x$  je *voľná* v  $A$  vtt  $x$  má v  $A$  aspoň jeden voľný výskyt.

Množinu voľných premenných formuly  $A$  označíme  $\text{free}(A)$ .

Príklad 7.15.

$$\begin{aligned}\text{free}(\neg P(x, y) \wedge K(y, z)) &= \{x, y, z\} \\ \text{free}(\neg P(x, y) \wedge \exists y K(y, z)) &= \{x, y, z\} \\ \text{free}(\exists y (\neg P(x, y) \wedge K(y, z))) &= \{x, z\} \\ \text{free}(\exists y (\neg P(x, y) \wedge \forall z K(y, z))) &= \{x\} \\ \text{free}(\exists y \exists z (\forall x \neg P(x, y) \wedge K(y, z))) &= \{\}\end{aligned}$$

## Volné a viazané premenné

**Tvrdenie 7.16.** Pre každú individuovú premennú  $x$ , každý symbol konštanty  $a$ , každú aritu  $n > 0$ , každý predikátový symbol  $P$  s aritou  $n$ , všetky termy  $t_1, t_2, \dots, t_n$  a všetky formuly  $A, B$  platí:

$$\begin{aligned}\text{free}(x) &= \{x\} \\ \text{free}(a) &= \{\} \\ \text{free}(t_1 \doteq t_2) &= \text{free}(t_1) \cup \text{free}(t_2) \\ \text{free}(P(t_1, \dots, t_n)) &= \text{free}(t_1) \cup \dots \cup \text{free}(t_n) \\ \text{free}(\neg A) &= \text{free}(A) \\ \text{free}(A \wedge B) &= \text{free}(A \vee B) = \text{free}(A \rightarrow B) = \\ &= \text{free}(A) \cup \text{free}(B) \\ \text{free}(\forall x A) &= \text{free}(\exists x A) = \text{free}(A) \setminus \{x\}\end{aligned}$$

## Uzavreté formuly a teórie

**Definícia 7.17** (Uzavretá formula, teória). Formula  $A$  jazyka  $\mathcal{L}$  je *uzavretá* vtt žiadna premenná nie je voľná v  $A$  (teda  $\text{free}(A) = \emptyset$ ).

*Teóriou* v jazyku  $\mathcal{L}$  je každá spočítateľná množina uzavretých formúl jazyka  $\mathcal{L}$ .

*Príklad 7.18.* Ktoré z týchto formúl sú uzavreté?

- $\exists x P(x, x)$  uzavretá,
- $\exists y P(x, y)$  otvorená,  $x$  je voľná,
- $((M(x) \wedge B(x)) \rightarrow \exists y P(x, y))$  otvorená,  $x$  je voľná,
- $\forall x ((M(x) \wedge B(x)) \rightarrow \exists y P(x, y))$  uzavretá.

## 7.4 Sémantika relačnej logiky prvého rádu

### Štruktúra

**Definícia 7.19.** Nech  $\mathcal{L}$  je jazyk relačnej logiky prvého rádu. *Štruktúrou* pre jazyk  $\mathcal{L}$  nazývame dvojicu  $\mathcal{M} = (D, i)$ , kde  $D$  je ľubovoľná *neprázdna* množina nazývaná *doména* štruktúry  $\mathcal{M}$ ;  $i$  je zobrazenie, nazývané *interpretácia* štruktúry  $\mathcal{M}$ , ktoré

- každej individuovej konštante  $c$  jazyka  $\mathcal{L}$  priraduje prvok  $i(c) \in D$ ;
- každému predikátovému symbolu  $P$  jazyka  $\mathcal{L}$  s aritou  $n$  priraduje množinu  $i(P) \subseteq D^n$ .

*Dohoda 7.20.* Štruktúry označujeme veľkými *písanými* písmenami  $\mathcal{M}, \mathcal{N}, \dots$

## Ohodnotenie individuových premenných

**Definícia 7.21.** Nech  $\mathcal{M} = (D, i)$  je štruktúra pre jazyk  $\mathcal{L}$ . *Ohodnotenie individuových premenných* je ľubovoľná funkcia  $e : \mathcal{V}_{\mathcal{L}} \rightarrow D$  (priraduje premenným prvky domény).

Nech ďalej  $x$  je individuová premenná z  $\mathcal{L}$  a  $d$  je prvok  $D$ . Zápisom  $e(x/d)$  označíme ohodnotenie individuových premenných, ktoré premennej  $x$  priraduje hodnotu  $d$  a všetkým ostatným premenným rovnakú hodnotu ako im priraduje  $e$ , čiže

$$e(x/d)(y) = \begin{cases} d, & \text{ak } y = x, \\ e(y), & \text{ak } y \neq x, \end{cases}$$

alebo množinovo zapísané  $e(x/d) = e \setminus \{x \mapsto e(x)\} \cup \{x \mapsto d\}$ .

### Príklad ohodnotenia individuových premenných

Nech

$$\mathcal{V}_{\mathcal{L}} = \{x_1, x_2, y, \dots\}$$

$$D = \{\text{Alica}, \text{Bonifác}, \text{Cyril}\}.$$

Ohodnotením (individuových) premenných je napríklad

$$e = \{x_1 \mapsto \text{Bonifác}, x_2 \mapsto \text{Alica}, y \mapsto \text{Bonifác}, \dots\}$$

Potom

$$e(y/\text{Cyril}) = \{x_1 \mapsto \text{Bonifác}, x_2 \mapsto \text{Alica}, y \mapsto \text{Cyril}, \dots\}$$

### Hodnota termov

**Definícia 7.22.** Nech  $\mathcal{M} = (D, i)$  je štruktúra,  $e$  je ohodnotenie premenných. *Hodnotou termu  $t$  v štruktúre  $\mathcal{M}$  pri ohodnotení premenných  $e$*  je prvok  $t^{\mathcal{M}}[e]$  z  $D$  určený nasledovne:

- $t^{\mathcal{M}}[e] = e(x)$ , ak  $t$  je premenná  $x \in \mathcal{V}_{\mathcal{L}}$ ,
- $t^{\mathcal{M}}[e] = i(a)$ , ak  $t$  je konštanta  $a \in \mathcal{C}_{\mathcal{L}}$ .



## Splnenie atomickej formuly v štruktúre

Určenie významu *atomickej* formuly, napr.  $\text{patrí}(x, \text{Vierka})$ , v danej štruktúre, napr.  $\mathcal{M} = (D, i)$ , kde

$$D = \{i_{\text{Viera}}, i_{\text{Juraj}}, i_{\text{Eva}}, \text{🐕}, \text{🐈}, \text{🐾}, \text{🐶}, \text{🐱}\}$$

$$i(\text{Vierka}) = i_{\text{Viera}},$$

$$i(\text{biele}) = \{\text{🐾}, \text{🐶}, \text{🐱}\}$$

$$i(\text{patrí}) = \{(\text{🐶}, i_{\text{Viera}}), (\text{🐾}, i_{\text{Eva}})\}$$

$$i(\text{Jurko}) = i_{\text{Juraj}},$$

$$i(\text{Ňufko}) = \text{🐾}$$

pri ohodnotení premenných, napr.  $e = \{x \mapsto \text{🐾}, y \mapsto i_{\text{Eva}}, \dots\}$ :

1. vyhodnotíme termy, ktoré sú argumentmi predikátu:

$$x^{\mathcal{M}}[e] = e(x) = \text{🐾}$$

$$\text{Vierka}^{\mathcal{M}}[e] = i(\text{Vierka}) = i_{\text{Viera}},$$

2. zistíme, či  $(\text{🐾}, i_{\text{Viera}}) \in i(\text{patrí})$ : *nie*

Štruktúra  $\mathcal{M}$  **nesplňa** formulu  $\text{patrí}(x, \text{Vierka})$  **pri ohodnotení**  $e$   
 $\mathcal{M} \not\models \text{patrí}(x, \text{Vierka})[e]$

## Splnenie existenčne kvantifikovanej formuly

$\mathcal{M} \models \exists y \text{patrí}(y, \text{Vierka})[e]$ ?

1. Vyskúšame *všetky* varianty ohodnotenia  $e$ , ktoré priradujú kvantifikovanej premennej  $y$  jednotlivé prvky domény:

$e(y/d)$	$\mathcal{M} \models^? \text{patrí}(y, \text{Vierka})[e(y/d)]$
$\{x \mapsto \text{🐾}, y \mapsto i_{\text{Viera}}, \dots\}$	?
$\vdots$	$\vdots$
$\{x \mapsto \text{🐾}, y \mapsto \text{🐾}, \dots\}$	?
$\{x \mapsto \text{🐾}, y \mapsto \text{🐶}, \dots\}$	?
$\{x \mapsto \text{🐾}, y \mapsto \text{🐱}, \dots\}$	✗

2.  $\mathcal{M} \models \exists y \text{patrí}(y, \text{Vierka})[e]$

vtt **pre aspoň jedno**  $d \in D$  máme  $\mathcal{M} \models \text{patrí}(y, \text{Vierka})[e(y/d)]$ .

Pravá strana je **pravdivá** pre  $d = \text{🐶}$  – *svedok*.

Takže  $\mathcal{M} \models \exists y \text{patrí}(y, \text{Vierka})[e]$ .

$$\begin{aligned}
D &= \{\text{Viera}, \text{Juraj}, \text{Eva}, \\
&\quad \text{Jurko}, \text{Nufko}, \\
&\quad \text{biele}, \text{patrí}\} \\
i(\text{Vierka}) &= \text{Viera}, \\
i(\text{Jurko}) &= \text{Juraj}, \\
i(\text{Nufko}) &= \text{Nufko}, \\
i(\text{biele}) &= \{\text{biele}, \\
&\quad \text{patrí}\} \\
i(\text{patrí}) &= \{(\text{biele}, \text{Viera}), \\
&\quad (\text{patrí}, \text{Eva})\} \\
e &= \{x \mapsto \text{biele}, \\
&\quad y \mapsto \text{Eva}, \dots\}
\end{aligned}$$

### Splnenie všeobecne kvantifikovanej formuly

$$\mathcal{M} \models \forall x(\text{biele}(x) \rightarrow \text{patrí}(x, y)) [e] \quad \text{skrátíme: } B = \text{biele}, P = \text{patrí}$$

1. Vyskúšame *všetky* varianty ohodnotenia  $e$ , ktoré priradujú kvantifikovanej premennej  $x$  jednotlivé prvky domény:

$e(x/d)$	$\mathcal{M} \models^? (B(x) \rightarrow P(x, y)) [e(x/d)]$
$\{x \mapsto \text{Viera}, y \mapsto \text{Eva}, \dots\}$	?
$\vdots$	$\vdots$
$\{x \mapsto \text{Jurko}, y \mapsto \text{Eva}, \dots\}$	?
$\{x \mapsto \text{Nufko}, y \mapsto \text{Eva}, \dots\}$	?
$\{x \mapsto \text{biele}, y \mapsto \text{Eva}, \dots\}$	?
$\{x \mapsto \text{patrí}, y \mapsto \text{Eva}, \dots\}$	$\neq$ lebo $\mathcal{M} \models B(x) [e(x/d)]$ a $\mathcal{M} \not\models P(x, y) [e(x/d)]$

2.  $\mathcal{M} \models \forall x(\text{biele}(x) \rightarrow \text{patrí}(x, y)) [e]$

vtt **pre všetky**  $d \in D$  máme  $\mathcal{M} \models (\text{biele}(x) \rightarrow \text{patrí}(x, y)) [e(x/d)]$ .

Pravá strana je **nepravdivá** pre  $d = \text{biele}$  a  $d = \text{patrí}$  – *kontrapríklady*.

Takže  $\mathcal{M} \not\models \forall x(\text{biele}(x) \rightarrow \text{patrí}(x, y)) [e]$ .

$$\begin{aligned}
D &= \{\text{Viera}, \text{Juraj}, \text{Eva}, \\
&\quad \text{Jurko}, \text{Nufko}, \\
&\quad \text{biele}, \text{patrí}\}
\end{aligned}$$

$$\begin{aligned}
 i(\text{Vierka}) &= \text{🐕}_{\text{Viera}}, \\
 i(\text{Jurko}) &= \text{🐕}_{\text{Juraj}}, \\
 i(\text{Ňufko}) &= \text{🐶}, \\
 i(\text{biele}) &= \{\text{🐶}, \\
 &\quad \text{🐕}, \\
 &\quad \text{🐶}\}, \\
 i(\text{patrí}) &= \{(\text{🐕}, \text{🐕}_{\text{Viera}}), \\
 &\quad (\text{🐶}, \text{🐕}_{\text{Eva}})\}, \\
 e &= \{x \mapsto \text{🐶}, \\
 &\quad y \mapsto \text{🐕}_{\text{Eva}}, \dots\}
 \end{aligned}$$

### Splnenie všeobecne kvantifikovanej implikácie

⚠ Naša  $\mathcal{M}$  spĺňa implikáciu  $(\text{biele}(x) \rightarrow \text{patrí}(x, y))$  pri  $e(x/d)$  pre väčšinu  $d \in D$  preto, že jej antecedent,  $\text{biele}(x)$ , je nesplnený.

To zodpovedá čítaniu formuly  $\forall x(\text{biele}(x) \rightarrow \text{patrí}(x, y))$  ako výroku „všetko biele patrí  $y$ “:

- Objekty, ktoré *nie sú biele*, *neovplyvňujú* pravdivosť tohto výroku ani pravdivosť implikácie  $(\text{biele}(x) \rightarrow \text{patrí}(x, y))$ .
- Výrok aj implikácia sú nepravdivé *iba* vtedy, keď nejaký biely objekt nepatrí  $y$ .

Ak by nič nebolo biele, teda by  $\mathcal{M} \not\models \text{biele}(x)[e(x/d)]$  pre všetky  $d \in D$ , tak by aj formula  $\forall x(\text{biele}(x) \rightarrow \text{patrí}(x, y))$  aj tvrdenie „všetko biele patrí  $y$ “ boli *triviálne* splnené.

### Nezávislosť od ohodnotenia viazanej premennej

Pri vyhodnocovaní splnenia kvantifikovanej formuly štruktúrou pri danom ohodnotení  $e$

$$\begin{aligned}
 \mathcal{M} &\models \exists y \text{patrí}(y, \text{Vierka}) [e] \\
 \mathcal{M} &\models \forall x(\text{biele}(x) \rightarrow \text{patrí}(x, y)) [e]
 \end{aligned}$$

*nezáleží* na tom, akú hodnotu priraďuje pôvodné ohodnotenie  $e$  *viazanej premennej*.

Priamu podformulu kvantifikovanej formuly vyhodnocujeme pri *nových* ohodnoteniach  $e(y/d)$ , resp.  $e(x/d)$ , cez všetky  $d \in D$ .

### Nezávislosť od ohodnotenia viazanej premennej

Pri vyhodnocovaní splnenia formuly s *jedinou* premennou, ktorá je kvantifikovaná, na pôvodnom ohodnotení vôbec nezáleží.

Mohlo by sa preto zdať, že pre prácu s uzavretými formulami je naša definícia zbytočne zložitá.

Ale ak je premenných viac, ohodnotenie už môže ovplyvniť splňanie. Napr. pri vyhodnocovaní

$$\mathcal{M} \models \exists x \forall y \text{ patrí}(x, y) [e]$$

najprv zvolíme konkrétneho kandidáta na svedka  $d$  pre  $x$  a pri následnom vyhodnocovaní

$$\mathcal{M} \models \forall y \text{ patrí}(x, y) [e(x/d)]$$

hodnotu premennej  $x$  používame, ale už ju nijako nevieme ovplyvniť.

### Splnenie formuly v štruktúre

**Definícia 7.23.** Nech  $\mathcal{M} = (D, i)$  je štruktúra,  $e$  je ohodnotenie premenných. Relácia štruktúra  $\mathcal{M}$  *spĺňa formulu  $A$  pri ohodnotení  $e$*  (skrátene  $\mathcal{M} \models A[e]$ ) má nasledovnú indukčnú definíciu:

- $\mathcal{M} \models t_1 \doteq t_2[e]$  vtt  $t_1^{\mathcal{M}}[e] = t_2^{\mathcal{M}}[e]$ ,
- $\mathcal{M} \models P(t_1, \dots, t_n)[e]$  vtt  $(t_1^{\mathcal{M}}[e], \dots, t_n^{\mathcal{M}}[e]) \in i(P)$ ,
- $\mathcal{M} \models \neg A[e]$  vtt  $\mathcal{M} \not\models A[e]$ ,
- $\mathcal{M} \models (A \wedge B)[e]$  vtt  $\mathcal{M} \models A[e]$  a zároveň  $\mathcal{M} \models B[e]$ ,
- $\mathcal{M} \models (A \vee B)[e]$  vtt  $\mathcal{M} \models A[e]$  alebo  $\mathcal{M} \models B[e]$ ,
- $\mathcal{M} \models (A \rightarrow B)[e]$  vtt  $\mathcal{M} \not\models A[e]$  alebo  $\mathcal{M} \models B[e]$ ,
- $\mathcal{M} \models \exists x A[e]$  vtt pre nejaký prvok  $d \in D$  máme  $\mathcal{M} \models A[e(x/d)]$ ,
- $\mathcal{M} \models \forall x A[e]$  vtt pre každý prvok  $d \in D$  máme  $\mathcal{M} \models A[e(x/d)]$ ,

pre všetky arity  $n > 0$ , všetky predikátové symboly  $P$  s aritou  $n$ , všetky termy  $t_1, t_2, \dots, t_n$ , všetky premenné  $x$  a všetky formuly  $A, B$ .

## Splnenie formuly v štruktúre pri ohodnotení • príklad

Príklad 7.24. Nech  $\mathcal{M} = (D, i)$ , kde

$D = \{1, 2, 3, 4, 5\}$

$i(\text{Jurko}) = 1$	$i(\text{myš}) = \{3, 4\}$	$i(\text{patrí}) = \{(3, 2),$
$i(\text{Vierka}) = 2$	$i(\text{škrekčok}) = \{5\}$	$(4, 2),$
$i(\text{Ňufko}) = 4$	$i(\text{biele}) = \{4, 5\}$	$(5, 1)\}$ .

Nech  $e = \{x \mapsto 3, y \mapsto 5, \dots\}$ .

Zistime, či

- $\mathcal{M} \models ((\text{myš}(x) \wedge \text{biele}(x)) \rightarrow \exists y \text{patrí}(x, y)) [e]$
- $\mathcal{M} \models \forall x((\text{myš}(x) \wedge \text{biele}(x)) \rightarrow \exists y \text{patrí}(x, y)) [e]$

## Pravdivosť uzavretej formuly a teórie

Neuzavreté formuly zodpovedajú *výrokovým formám*. Ich splnenie v štruktúre závisí od ohodnotenia voľných premenných.

Uzavreté formuly zodpovedajú *výrokom*. Ich splnenie v štruktúre *nezávisí od ohodnotenia*. Preto pri nich môžeme hovoriť o *pravdivosti v štruktúre*.

**Definícia 7.25.** Nech  $X$  je uzavretá formula jazyka  $\mathcal{L}$ , nech  $T$  je teória v jazyku  $\mathcal{L}$  a nech  $\mathcal{M}$  je štruktúra pre  $\mathcal{L}$ .

Formula  $X$  je *pravdivá* v štruktúre  $\mathcal{M}$  (skrátene  $\mathcal{M} \models X$ ) vtt  $\mathcal{M}$  spĺňa formulu  $X$  pri každom ohodnotení  $e$ . Vtedy tiež hovoríme, že  $\mathcal{M}$  je *modelom* formuly  $X$ .

Teória  $T$  je *pravdivá* v štruktúre  $\mathcal{M}$  (skrátene  $\mathcal{M} \models T$ ) vtt každá formula  $X$  z  $T$  je pravdivá v  $\mathcal{M}$ . Vtedy tiež hovoríme, že  $\mathcal{M}$  je *modelom* teórie  $T$ .

## Prvorádová ekvivalentnosť formúl

Na cvičeniach budeme potrebovať pojem ekvivalentnosti formúl v logike prvého rádu.

**Definícia 7.26.** Nech  $X$  a  $Y$  sú formuly jazyka  $\mathcal{L}$ .

Formula  $X$  je *prvorádovo ekvivalentná* s formulou  $Y$  (skr.  $X \Leftrightarrow Y$ ) vtt pre každú štruktúru  $\mathcal{M}$  a každé ohodnotenie  $e$  máme  $\mathcal{M} \models X[e]$  vtt  $\mathcal{M} \models Y[e]$ .

*Dohoda 7.27.* Príslovku „prvorádovo“ budeme zvyčajne vynechávať.

Na dokazovanie ekvivalentnosti nemáme zatiaľ pohodlnú metódu, ale na jej vyvrátenie (dokázanie  $X \not\leftrightarrow Y$ ) stačí nájsť takú štruktúru  $\mathcal{M}$  a ohodnotenie  $e$ , že...

## 7.5 Neexistencia

### Neexistencia — úskalía negácie

Neexistenciu v slovenčine zvyčajne vyjadruje *dvojitý zápor*: negatívne zámeno (nikto/nič/žiadne) a negatívne tvrdenie, napr.

„Nič *nie* je dokonalé“.

Na jej správne vyjadrenie musíme tvrdenie *parafrázovať* tak, aby sme negáciu vyjadrili *iba* pomocou „*nie je pravda, že*“. Sú dve možnosti:

1. „*Nie je pravda, že niečo je dokonalé.*“  
„*Nie je pravda, že existuje* taký objekt  $x$ , že  $x$  je dokonalé.“

►  $\neg \exists x \text{ dokonalé}(x)$

⚠ Negácia sa vzťahuje na *celú* kvantifikovanú formulu!

2. „*Všetko je nedokonalé*“  
„*Pre každý* objekt  $x$  *nie je pravda, že*  $x$  je dokonalé.“

►  $\forall x \neg \text{dokonalé}(x)$

⚠ Pri parafrázovaní *nepoužívajte* „*neexistuje*“ — je *zavádzajúce*! (Presvedčíme sa o tom na budúci týždeň.)

## 7.6 Aristotelovské formy

### Štyri aristotelovské formy

Dávno pred kodifikáciou logiky prvého rádu sa kvantifikovanými tvrdeniami zaoberal staroveký grécky filozof Aristoteles.

Študoval najmä tvrdenia v tvaroch:

- Všetky  $P$  sú  $Q$ .
- Niektoré  $P$  sú  $Q$ .

- Žiadne  $P$  nie sú  $Q$ .
- Niektoré  $P$  nie sú  $Q$ .

ktorým dnes hovoríme obmedzená alebo ohraničená kvantifikácia.

Matematici ju radi zapisujú skrátene, napr.

$$\forall \epsilon > 0 \exists \delta > 0 \forall x \in \mathbb{R} (0 < |x - a| < \delta \rightarrow |f(x) - b| < \epsilon)$$

Logika prvého rádu dokáže tieto formy vyjadriť pomocou základných (neobmedzených) kvantifikátorov a logických spojok.

### Všetky $P$ sú $Q$

Formu „Všetky  $P$  sú  $Q$ “ (napr. „Všetky myši sú sivé“) formalizujeme

$$\forall x (P(x) \rightarrow Q(x)) \quad \checkmark$$

teda „Pre každý objekt  $x$  je pravda, že ak  $x$  má vlastnosť  $P$ , tak  $x$  má vlastnosť  $Q$ .“, ekvivalentne „Pre každý objekt  $x$  je pravda, že  $x$  nemá vlastnosť  $P$  alebo  $x$  má vlastnosť  $Q$ .“

Študenti túto formu niekedy *nesprávne* sformalizujú ako

$$\forall x (P(x) \wedge Q(x)) \quad \times$$

Pritom táto formalizácia neprejde jednoduchou skúškou — *stačí si ju prečítať*: „Každý objekt  $x$  má súčasne vlastnosť  $P$  aj vlastnosť  $Q$ ,“ prirodzenejšie „Všetko je  $P$  aj  $Q$ “ (napr. „Všetko je myš a je to sivé“).

### Všetky $P$ sú $Q$ — varianty

Forma „Všetky  $P$  sú  $Q$ “ sa v prirodzených vetách niekedy rozpoznáva ťažšie, napríklad keď je  $P$  alebo  $Q$  vzťah:

- Všetky myši kŕmi Jurko.  
Všetky myši sú také, že ich kŕmi Jurko.  
 $\forall x (\text{myš}(x) \rightarrow \text{kŕmi}(\text{Jurko}, x))$
- Jurko kŕmi iba myši.  
Všetko, čo Jurko kŕmi, sú myši.  
 $\forall x (\text{kŕmi}(\text{Jurko}, x) \rightarrow \text{myš}(x)).$

## Niektoré $P$ sú $Q$

Formu „Niektoré  $P$  sú  $Q$ “ (napr. „Niektoré myši sú biele“) formalizujeme

$$\exists x (P(x) \wedge Q(x)) \quad \checkmark$$

teda „Existuje aspoň taký jeden objekt  $x$ , že  $x$  má vlastnosť  $P$  a  $x$  má vlastnosť  $Q$ .“

Študenti túto formu niekedy *nesprávne* sformalizujú ako

$$\exists x (P(x) \rightarrow Q(x)) \quad \times$$

Ani táto formalizácia neprejde čítacou skúškou: „Existuje objekt  $x$ , ktorý ak má vlastnosť  $P$ , tak má vlastnosť  $Q$ ,“ čo je podľa definície splnenia v ohodnotení ekvivalentné s „Existuje objekt  $x$ , ktorý nemá vlastnosť  $P$  alebo má vlastnosť  $Q$ ,“ (napr. „Niečo nie je myš alebo je to biele“ — je pravdivé vo svete, kde sú všetky myši sivé a je tam jeden človek).

## Niektoré $P$ sú $Q$ — varianty

Forma „Niektoré  $P$  sú  $Q$ “ sa v prirodzených vetách niekedy rozpoznáva ťažšie, napríklad keď je  $P$  alebo  $Q$  vzťah.

- Jurko kŕmi nejaké myši.  
Jurko kŕmi (nejakú) myš.  
Niečo z toho, čo Jurko kŕmi, sú myši.  
 $\exists x (\text{kŕmi}(\text{Jurko}, x) \wedge \text{myš}(x))$

Niektorých študentov prekvapuje, že pri tejto forme nezáleží na poradí  $P$  a  $Q$ .

- Nejaké myši kŕmi Jurko.  
Niektoré myši sú také, že ich kŕmi Jurko.  
 $\exists x (\text{kŕmi}(\text{Jurko}, x) \wedge \text{myš}(x))$

Je ale *vernejšie* poradie pri formalizácii zachovať:

$$\exists x (\text{myš}(x) \wedge \text{kŕmi}(\text{Jurko}, x))$$



### Žiadne $P$ nie sú $Q$

Formu „Žiadne  $P$  nie sú  $Q$ “ (napr. „Žiadne myši nie sú červené“) parafrázujeme ako

„Nie je pravda, že existuje taký objekt  $x$ , že  $x$  má vlastnosť  $P$  a  $x$  má vlastnosť  $Q$ .“

a formalizujeme

$$\neg \exists x (P(x) \wedge Q(x)) \quad \checkmark$$

alebo parafrázujeme ako

„Pre každý objekt  $x$  je pravda, že ak  $x$  má vlastnosť  $P$ , tak  $x$  nemá vlastnosť  $Q$ ,“ teda „Každé  $P$  nie je  $Q$ ,“

a formalizujeme ako

$$\forall x (P(x) \rightarrow \neg Q(x)) \quad \checkmark$$

Ani pri tejto forme nezáleží na poradí  $P$  a  $Q$ , ale je *vernejšie* ho pri formalizácii zachovať.

### Niektoré $P$ nie sú $Q$

Formu „Niektoré  $P$  nie sú  $Q$ “ (napr. „Niektoré myši nie sú sivé“) formalizujeme

$$\exists x (P(x) \wedge \neg Q(x)) \quad \checkmark$$

teda „Pre nejaký objekt  $x$  je pravda, že  $x$  má vlastnosť  $P$  a  $x$  nemá vlastnosť  $Q$ .“

## 7.7 Zamlčané a zdanlivo opačné kvantifikátory

### Zamlčaný všeobecný kvantifikátor

Niekedy kvantifikátor nie je explicitne vyjadrený príslušným zámenom.

Použitie všeobecného podstatného mena (zvyčajne, ale nie nutne v množnom čísle) v úlohe *podmetu* sa zvyčajne chápe ako *všeobecná* kvantifikácia:

- Myši sú sivé.

$$\forall x (\text{myš}(x) \rightarrow \text{sivý}(x))$$

- Myš je hlodavec.  
 $\forall x(\text{myš}(x) \rightarrow \text{hlodavec}(x))$
- Kto prešiel Logiku, ten odovzdával domáce úlohy.  
 $\forall x(\text{prešiel}(x, \text{Logiku}) \rightarrow \text{odovzdával\_DÚ}(x))$

### Zamlčaný existenčný kvantifikátor

Použitie všeobecného podstatného mena v úlohe *predmetu pozitívneho* prísudku sa *zvyčajne* chápe ako *existenčná* kvantifikácia:

- Jurko kŕmi myš.  $\exists x(\text{kŕmi}(\text{Jurko}, x) \wedge \text{myš}(x))$
- Bonifác si kúpil syr.  $\exists x(\text{kúpil}(\text{Bonifác}, x) \wedge \text{syr}(x))$

### Zamlčaná neexistencia

Použitie všeobecného podstatného mena v úlohe *predmetu negatívneho* prísudku sa *zvyčajne* chápe ako vyjadrenie *neexistencie*:

- Bonifác si nekúpil syr.  
 $\neg \exists x(\text{kúpil}(\text{Bonifác}, x) \wedge \text{syr}(x))$   
 $\forall x(\text{kúpil}(\text{Bonifác}, x) \rightarrow \neg \text{syr}(x))$
- Jurko nekŕmi myši.  
 $\forall x(\text{myš}(x) \rightarrow \neg \text{kŕmi}(\text{Jurko}, x))$   
 $\forall x(\text{kŕmi}(\text{Jurko}, x) \rightarrow \neg \text{myš}(x))$   
 $\neg \exists x(\text{kŕmi}(\text{Jurko}, x) \wedge \text{myš}(x))$

### Opatrnosti nikdy nie je nazvyš

Ľudia neraz veci kvantifikujú nesprávne alebo neúplne (zvlášť pozor na ľudí pôsobiacich v inom odvetví, kde nepoznáte konvencie).

Ako interpretovať tvrdenie „chlieb predávajú v potravinách“?

- každý chlieb predávajú len v potravinách a nikde inde
- existuje druh chleba, ktorý predávajú len v miestnej predajni potravín
- existuje druh chleba, ktorý predávajú v každých potravinách

- existujú potraviny, ktoré predávajú aspoň jeden chlieb
- každé potraviny predávajú aspoň jeden chlieb
- ...

### Zdanlivá existencia

V podmienkach sa občas vyskytujú neurčité zámená (niekto/niečo/niektoré/...), na ktoré sa ale odkazujeme v podmienenej vete:

- Keď *niekto* cvičí, tak je (on/ona) zdravý/-á.
- Ak sa Jurko o *niečo* stará, tak to *to* rád.

Také tvrdenie nezodpovedá implikácii s existenčným kvantifikátorom:

✗  $(\exists x \text{ doma}(x) \rightarrow \text{zodpovedný}(x))$

✗  $\exists x(\text{doma}(x) \rightarrow \text{zodpovedný}(x))$

ale zodpovedá všeobecne kvantifikovanej implikácii:

✓  $\forall x(\text{cvičí}(x) \rightarrow \text{zdravý}(x))$

✓  $\forall x(\text{stará\_sa\_o}(\text{Jurko}, x) \rightarrow \text{má\_rád}(\text{Jurko}, x))$

## 7.8 Nutné a postačujúce podmienky

### Nutné a postačujúce podmienky

Tvrdenia so (zamlčanou) všeobecnou kvantifikáciou majú často formu podradovacích súvetí:

1. Logiku prejde *každý/-á, kto* odovzdáva domáce úlohy.
2. Logiku prejde *iba tá/ten, kto* odovzdáva domáce úlohy.

pričom

- hlavná veta („Logiku prejde ...“) vyjadruje nejakú *vlastnosť*,
- vedľajšia veta („kto odovzdáva domáce úlohy“) vyjadruje *podmienku*, ktorá súvisí s touto vlastnosťou.

Aký je rozdiel medzi týmito podmienkami?

### Postačujúca podmienka

Výrok „Logiku prejde *každý/-á, kto* odovzdáva domáce úlohy“:

- je nepravdivý, keď existuje niekto, kto odovzdáva domáce úlohy, ale Logiku neprejde;
- teda hovorí, že na to, aby nejaká osoba prešla Logiku, *postačuje*, aby odovzdávala domáce úlohy.
- Odovzdávať domáce úlohy je teda *postačujúcou* podmienkou prejdania Logiky.
- Inak povedané: Pre každého/-ú platí, že
  - prejde Logiku, ak odovzdáva domáce úlohy;
  - ak odovzdáva domáce úlohy, tak prejde Logiku.
- Formalizácia je teda:

$$\forall x(\text{odovzdáva\_DÚ}(x) \rightarrow \text{prejde}(x, \text{Logika}))$$

### Nutná podmienka

Výrok „Logiku prejde *iba tá/ten, kto* odovzdáva domáce úlohy.“:

- je nepravdivý, keď existuje niekto, kto neodovzdáva domáce úlohy, ale Logiku predsa prejde;
- teda hovorí, že na to, aby nejaká osoba prešla Logiku, je *nutné*, aby odovzdávala domáce úlohy.
- Odovzdávať domáce úlohy je teda *nutnou* podmienkou prejdania Logiky.
- Ekvivalentne: Pre každého/-u platí, že
  - prejde Logiku, *iba* ak odovzdáva domáce úlohy.
  - ak *neodovzdáva* domáce úlohy, tak *neprejde* Logiku.
  - ak prejde Logiku, tak odovzdáva domáce úlohy.
- Formalizácia je teda:

$$\forall x(\text{prejde}(x, \text{Logika}) \rightarrow \text{odovzdáva\_DÚ}(x))$$

## 7.9 Zložené kvantifikované vlastnosti

### Zložené kvantifikované vlastnosti

Často potrebujeme kvantifikovať objekty, ktoré majú zložité vlastnosti:

1. Nejaká Jankina biela myš je sýta.
2. Všetky Jankine biele myši sú sýte.

Prvý druh kvantifikácií je zrejme existenčný a už vieme, že sa spravidla spája s konjunkciou.

Druhý druh kvantifikácií je zrejme všeobecný a vieme, že sa spravidla spája s implikáciou.

Použitie spojok ale závisí od pozície kvantifikácie vo vete.

### Zložené existenčne kvantifikované vlastnosti ako podmet

Nejaká Jankina biela myš je sýta.

- Výrok má formu „Niektoré  $P$  sú  $Q$ ,“ teda prekladáme ho ako  $\exists x( P(x) \wedge Q(x) )$ . Pričom ale  $P$  je zložená vlastnosť.
- Vlastnosť  $P$  opisuje objekt, ktorý má zrejme byť súčasne Jankin, biely a má to byť myš. Preto  $P$  vytvoríme z jednotlivých predikátov konjunkciou.

$$\exists x( ( \text{patrí}(x, \text{Janka}) \wedge \text{biely}(x) \wedge \text{myš}(x) ) \wedge \text{sýty}(x) )$$

### Zložené všeobecne kvantifikované vlastnosti ako podmet

Všetky Jankine biele myši sú sýte.

- Výrok má formu „Všetky  $P$  sú  $Q$ ,“ teda prekladáme ho ako  $\forall x( P(x) \rightarrow Q(x) )$ , pričom  $P$  je zložená vlastnosť.
- Vlastnosť  $P$  opäť opisuje objekty, ktoré majú byť súčasne Jankine, biele a myši. Preto aj teraz  $P$  vytvoríme konjunkciou.

$$\forall x( ( \text{patrí}(x, \text{Janka}) \wedge \text{biely}(x) \wedge \text{myš}(x) ) \rightarrow \text{sýty}(x) )$$

## Zložené existenčne kvantifikované vlastnosti ako predmet

Jurko má nejakú sýtu bielu myš.

- Aby sme zistili, ktorú aristotelovskú formu má veta, musíme ju preformulovať:  
Nejaká sýta biela myš patrí Jurkovi.
- Veta má formu „Niektoré  $P$  sú  $Q$ ,“ teda prekladáme ju ako  $\exists x(P(x) \wedge Q(x))$ , pričom  $P$  je zložená vlastnosť.

$$\exists x((\text{sýty}(x) \wedge \text{myš}(x) \wedge \text{biely}(x)) \wedge \text{patrí}(x, \text{Jurko}))$$

## Zložené všeobecne kvantifikované vlastnosti ako predmet

Jurko má všetky sýte biele myši.

- Aj túto vetu musíme preformulovať:  
Všetky sýte biele myši sú Jurkove.
- Veta má formu „Všetky  $P$  sú  $Q$ ,“ teda prekladáme ju ako  $\forall x(P(x) \rightarrow Q(x))$ , pričom  $P$  je zložená vlastnosť.

$$\forall x((\text{sýty}(x) \wedge \text{myš}(x) \wedge \text{biely}(x)) \rightarrow \text{patrí}(x, \text{Jurko}))$$

## Viacnásobné všeobecne kvantifikované prívlastky

Jurko má všetky myši a škrečky.

- Preformulujeme: Všetky myši a škrečky sú Jurkove.
- Veta má formu „Všetky  $P$  sú  $Q$ ,“ teda prekladáme ju ako  $\forall x(P(x) \rightarrow Q(x))$ .
- $P$  je zložená vlastnosť. Ale ako je zložená?
- ✖ Keď „myši a škrečky“ sformalizujeme ( $\text{myš}(x) \wedge \text{škrečok}(x)$ ),  $\forall x(P(x) \rightarrow Q(x))$  bude znamenať „Pre každé  $x$ , ak  $x$  je myš a zároveň  $x$  je škrečok, tak  $x$  patrí Jurkovi.“
- Vieme ale, že nič nie je naraz myš aj škrečok, takže podmienke (v našom svete) nevyhovuje žiaden objekt, takže Jurkovi nemusí nič patriť.

- ✔ Intuitívny význam („a“ ako množinové zjednotenie) zachováme, keď „myši a škrečky“ sformalizujeme ( $\text{myš}(x) \vee \text{škrečok}(x)$ ).

$$\forall x((\text{myš}(x) \vee \text{škrečok}(x)) \rightarrow \text{patrí}(x, \text{Jurko}))$$

## 7.10 Konverzačné implikatury

### Triviálne pravdivé všeobecne kvantifikované implikácie

Nie všetkým sa zdá intuitívne, že formula

$$\forall x(\text{myš}(x) \rightarrow \text{biela}(x))$$

je *pravdivá* vo svetoch, kde *nie sú žiadne myši*.

Dobry spôsob, ako to pochopiť je, že uvedomiť si, že vo svete, kde nie sú myši, *neexistuje kontrapriklad* pre túto formulu — myš, ktorá by nebola biela.

Hovoríme, že v takom svete je táto formula *triviálne pravdivá*.

Podobne je vo svetoch bez myši triviálne pravdivá ešte prekvapujúcejšia formula:

$$\forall x(\text{myš}(x) \rightarrow \text{človek}(x))$$

### Triviálne pravdivé všeobecne kvantifikované implikácie

Tvrdenie „Každý prvák, ktorý si zapísal logiku, z nej dostal A,“ v sebe nesie implikaturu (domnelý dôsledok), že takí prváci existujú.

Ak je takéto tvrdenie nutne triviálne pravdivé, lebo objekty z predpokladu neexistujú (napr. prváci si logiku nemôžu zapisovať), intuitívne ho považujeme zavádzajúce.

Nič to ale nemení na fakte, že je pravdivé.

Existencia prváka, ktorý si zapísal logiku, je skutočne iba implikatura.

Dodatok „Ale žiadny prvák si ju nikdy nezapísal“ (negácia implikatury), nie je s tvrdením v spore, ale objasňuje, že je triviálne pravdivé.

### „Niektoré“ neimplikuje „nie všetky“

Ďalšia implikatura sa spája s tvrdeniami: „Niektoré  $P$  sú  $Q$ .“

Niekomu sa môže „Niektoré  $P$  sú  $Q$ “ zdať sporné s „Všetky  $P$  sú  $Q$ .“ — Prečo by sme hovorili „niektoré  $P$ “, keď to platí pre všetky  $P$ ? Takýto človek

považuje tvrdenie „Nie všetky  $P$  sú  $Q$ “ za dôsledok tvrdenia „Niektoré  $P$  sú  $Q$ “. Aj to je však iba implikatúra.

Keď ale na otázku „Dostal niekto Ačko?“ odpovieme „Áno, niektorí študenti Ačko dostali. *Vlastne ho dostali všetci*,“ druhá veta prvú dopĺňa, ale neprotirečí jej, hoci je negáciou implikatúry.

Ak chceme jasne vyjadriť domnelý význam, povieme „Niektorí študenti Ačko dostali, *ale nie všetci*,“ čo formalizujeme formulou v tvare  $(\exists x(P(x) \wedge Q(x)) \wedge \neg \forall x(P(x) \rightarrow Q(x)))$ .

## 7.11 Vyhybanie sa chybám

### Bežné chyby

Ďalšie bežné chyby:

- použitie predikátu v rovnostnom atóme (na niektorej strane rovnosti miesto termu)
- reťazenie predikátov (jeden je argumentom druhého)
- nepoužitie potrebného predikátu
- voľná premenná vo formule
- zámena ekvivalencie a implikácie

### Kontrola

Techniky na kontrolu:

- Prejdem si zoznam funkčných aj predikátových symbolov, vylúčim tie, čo nesúvisia s daným tvrdením, a overím, že všetky ostatné sú vo formule obsiahnuté.
- Mechanicky prepíšem zapísanú formulu (napr. použitím de Morganových pravidiel, presunutím negácie dovnútra kvantifikátora či von, odstránením implikácií); prečítam si a pochopím upravenú formulu — hovorí to, čo má?



## Kontrola

- Zapišem to isté iným spôsobom (nie vždy mi nejaký napadne ľahko, ale vždy viem zapísať opak, čiže negáciu, napr. úvahou cez protipríklad — či už pre celú formulu, alebo pre jej časť).
  - Potom mechanicky upravujem dve formuly na ekvivalentné a snažím sa ich previesť do tvaru, kde už je jasné, či sú rovnaké alebo rôzne (napr. do negačnej normálnej formy).
  - Alebo prečítam alternatívny zápis, preložím si ho do prirodzeného jazyka a porovnám, či sa zhoduje s pôvodným.
- Overím, či niekde nemám zápis  $\forall x(P(x) \wedge \dots)$  — takmer nikdy nechcete povedať niečo typu: „Všetky objekty sú počítače.“
- Overím, či niekde nemám zápis  $\exists x(P(x) \rightarrow Q(x))$  — takmer nikdy nechcete povedať niečo typu: „Existuje objekt, ktorý *nie* je počítač *alebo* je pokazený.“

## 8. prednáška

# Tablá pre kvantifikátory.

## Viackvantifikátorové tvrdenia

---

### 8 Tablá s kvantifikátormi

#### 8.1 Logické vlastnosti a vzťahy v logike prvého rádu

##### Logické vlastnosti a vzťahy v logike prvého rádu

Minulý týždeň sme zadefinovali, kedy je *uzavretá* formula a teória (množina uzavretých formúl) *pravdivá* v danej štruktúre ( $\mathcal{M} \models A$ ,  $\mathcal{M} \models T$ ).

Použili sme pomocný induktívne definovaný vzťah *štruktúra spĺňa formulu pri ohodnotení* ( $\mathcal{M} \models X[e]$ ). Je definovaný pre *všetky* formuly (otvorené aj uzavreté).

Pomocou štruktúr a pravdivosti môžeme pre relačnú logiku prvého rádu skonkretizovať *logické vlastnosti a vzťahy*, ktoré už poznáme z výrokovologickej časti logiky prvého rádu:

- splniteľnosť a nesplniteľnosť,
- „vždy pravdivé“ formuly (vo výrokovom prípade sa volali tautológie),
- vyplývanie/logický dôsledok.

##### Splniteľnosť a nesplniteľnosť

Ako sme sa dohodli minule, predpokladáme, že sme si pevne zvolili ľubovoľný jazyk relačnej logiky prvého rádu  $\mathcal{L}$ . Všetky definície platia pre symboly, termy, atómy, formuly, teórie, atď. v tomto jazyku a štruktúry a ohodnotenia individuových premenných pre tento jazyk. Pretože  $\mathcal{L}$  je ľubovoľný, dajú sa definície aplikovať na všetky jazyky relačnej logiky prvého rádu.

**Definícia 8.1.** Nech  $X$  je uzavretá formula a  $T$  je teória. Formula  $X$  je *prvorádovo splniteľná* vtt  $X$  je pravdivá v *nejakej* štruktúre (ekvivalentne: *existuje*

štruktúra  $\mathcal{M}$  taká, že  $\mathcal{M} \models X$ ). Teória  $T$  je *prvorádovo splniteľná* vtt  $T$  má model (ekvivalentne:  $T$  je pravdivá v *nejakej* štruktúre; *existuje* štruktúra  $\mathcal{M}$  taká, že  $\mathcal{M} \models T$ ).

Formula resp. teória je *prvorádovo nesplniteľná* vtt nie je prvorádovo splniteľná.

### Splniteľnosť — príklad

*Príklad 8.2.* Teória  $\{\forall x(\text{človek}(x) \vee \text{myš}(x)), \forall x(\text{človek}(x) \rightarrow \neg \text{myš}(x))\}$  je prvorádovo *splniteľná*.

Je to tak preto, že je *pravdivá v štruktúre* (teda jej modelom je)  $\mathcal{M} = (D, i)$ , kde  $D = \{1, 2\}$ ,  $i(\text{človek}) = \{1\}$  a  $i(\text{myš}) = \{2\}$ .

Samozrejme je pravdivá v mnohých iných štruktúrach.

### Platné formuly

Formulám, ktoré sú výrokovologicky pravdivé (pravdivé v každom výrokovologickom ohodnotení atómov), sme hovorili tautológie.

Pre formuly, ktoré sú prvorádovo pravdivé (pravdivé v každej štruktúre), sa používa iný pojem:

**Definícia 8.3.** Nech  $X$  je uzavretá formula. Formula  $X$  je *platná* (skrátene  $\models X$ ) vtt  $X$  je pravdivá v *každej* štruktúre (teda pre *každú* štruktúru  $\mathcal{M}$  máme  $\mathcal{M} \models X$ ).

Samozrejme, formula *nie je platná* vtt je nepravdivá v *aspoň jednej* štruktúre.

Platnosť sa ale *nedá overiť* *vymenovaním* všetkých štruktúr, lebo tých je nekonečne veľa.

### Platné formuly — príklad

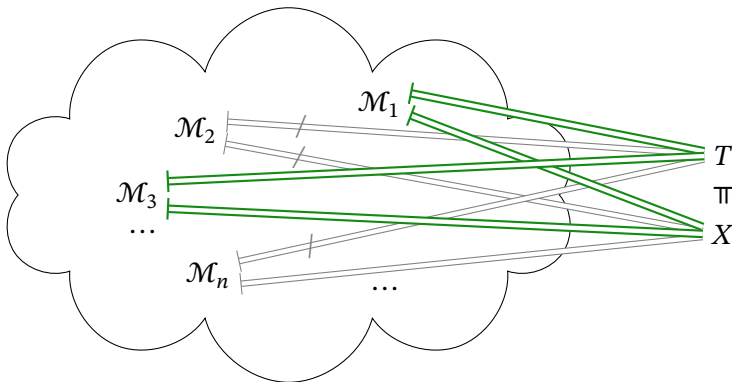
*Príklad 8.4.* Formula  $X = (\forall x \text{doma}(x) \rightarrow \text{doma}(\text{Jurko}))$  je platná.

*Dôkaz.* Predpokladajme, že by  $X$  nebola platná, teda by bola nepravdivá v nejakej štruktúre  $\mathcal{M} = (D, i)$ . Potom by v  $\mathcal{M}$  pri nejakom ohodnotení splnený antecedent  $\forall x \text{doma}(x)$ , ale nesplnený konzekvent  $\text{doma}(\text{Jurko})$ , teda  $i(\text{Jurko}) \notin i(\text{doma})$ . Ak je ale splnené  $\forall x \text{doma}(x)$ , tak pre každé  $m \in D$  máme  $m \in i(\text{doma})$ . Preto aj  $i(\text{Jurko}) \in i(\text{doma})$ , čo je spor.

Preto  $X$  je platná.

## Prvorádové vyplývanie, prvorádový logický dôsledok

**Definícia 8.5.** Z teórie  $T$  *prvorádovo logicky vyplýva* uzavretá formula  $X$  (tiež  $X$  je *prvorádovým logickým dôsledkom*  $T$ , skrátene  $T \models X$ ) vtt  $X$  je pravdivá v každom modeli  $T$  (ekvivalentne podrobnejšie: pre každú štruktúru  $\mathcal{M}$  platí, že ak je v  $\mathcal{M}$  pravdivá  $T$ , tak je v  $\mathcal{M}$  pravdivá  $X$ ).



Nezabúdajme, že význam „ $\models$ “ závisí od *typu* objektu na *ľavej* strane (a od prítomnosti ohodnotenia indiv. premenných):

**pravdivosť:** *štruktúra*  $\models$  formula/teória

**splnenie:** *štruktúra*  $\models$  formula [*ohodnotenie*]

**vyplývanie:** *teória*  $\models$  formula

### Prvorádové vyplývanie — príklad

Prvorádové vyplývanie sa *nedá overiť* vymenovaním všetkých štruktúr, rovnako ako platnosť.

**Príklad 8.6.** Vyplýva prvorádovo z teórie  $T = \{ \forall x(\text{opraví}(\text{TServis}, x) \rightarrow \text{tlačiareň}(x)), \neg \text{tlačiareň}(\text{môjMobil}) \}$

formula  $X = \neg \text{opraví}(\text{TServis}, \text{môjMobil})$ ?

Áno, vyplýva. Presvedčíme sa o tom podobnou úvahou ako v príklade platnej formuly.

### Prvorádové nevyplývanie a príklad

Samozrejme, formula  $X$  *nevyplýva* z teórie  $T$  vtt  $X$  nie je pravdivá v *aspoň jednom* modeli  $T$ . Tento model je *kontrapríkladom* vyplývania.

*Príklad 8.7.* Vyplýva prvorádovo z teórie  $T = \{ \forall x(\text{opraví}(\text{TServis}, x) \rightarrow \text{tlačiareň}(x)), \text{tlačiareň}(\text{mojaHP}) \}$

formula  $X = \text{opraví}(\text{TServis}, \text{mojaHP})$ ?

Nie, nevyplýva. Napríklad štruktúra  $\mathcal{M} = (D, i)$ , kde  $D = \{1, 2, 3\}$ ,

$$\begin{array}{ll} i(\text{TServis}) = 1, & i(\text{tlačiareň}) = \{2, 3\}, \\ i(\text{mojaHP}) = 2, & i(\text{opraví}) = \{(1, 3)\} \end{array}$$

je kontrapríkladom toho, že  $T \models X$ , pretože  $\mathcal{M} \models T$ , ale  $\mathcal{M} \not\models X$ .

### Výrokovologické, prvorádové a logické vyplývanie

Podobne ako výrokovologické vyplývanie, aj prvorádové vyplývanie je *špeciálny prípad* logického vyplývania v prirodzenom jazyku.

Logické vyplývanie v prirodzenom jazyku je *bohatšie* ako prvorádové vyplývanie. Tvrdenie zodpovedajúce formule  $X$  logicky vyplýva z tvrdení v  $T$  — keď rozumieme vzťahu „väčší“.

Logika prvého rádu ale „nevidí“ význam predikátov. Pozerá sa na ne len pomocou formúl, v ktorých vystupujú.

*Dohoda 8.8.* Nateraz budeme *stručne ale nepresne* hovoriť „logický dôsledok“ a „vyplývanie“ namiesto „prvorádový logický dôsledok“ a „prvorádové logické vyplývanie“.

Viac o vzťahu výrokovologického, prvorádového a logického vyplývania neskôr.

### Platnosť a vyplývanie

Medzi platnými formulami a prvorádovým vyplývaním je podobný vzťah ako medzi tautológiami a výrokovologickým vyplývaním.

**Tvrdenie 8.9.** *Nech  $X$  je uzavretá formula. Nasledujúce tvrdenia sú vzájomne ekvivalentné:*

- $X$  je platná ( $\models X$ );
- $X$  vyplýva z prázdnej teórie ( $\emptyset \models X$ );
- $X$  vyplýva z každej teórie (pre každú teóriu  $T$  máme  $T \models X$ ).

**Tvrdenie 8.10.** *Nech  $T = \{A_1, \dots, A_n\}$  je konečná teória a nech  $X$  je uzavretá formula. Nasledujúce tvrdenia sú vzájomne ekvivalentné:*

- *formula  $(\bigwedge_{i=1}^n A_i \rightarrow X)$  je platná (t.j.,  $\models (\bigwedge_{i=1}^n A_i \rightarrow X)$ );*
- *$X$  vyplýva z teórie  $T$  (t.j.,  $T \models X$ ).*

## Platnosť, vyplývanie a ekvivalencia

**Tvrdenie 8.11.** *Nech  $X$  a  $Y$  sú uzavreté formuly. Nasledujúce tvrdenia sú vzájomne ekvivalentné:*

- *$X$  je prvorádovo ekvivalentná s  $Y$  (skr.  $X \Leftrightarrow Y$ );*
- *formula  $(X \leftrightarrow Y)$  je platná (skr.  $\models (X \leftrightarrow Y)$ );*
- *z  $\{X\}$  prvorádovo vyplýva  $Y$  a z  $\{Y\}$  prvorádovo vyplýva  $X$  (skr.  $\{X\} \models Y$  a  $\{Y\} \models X$ ).*

## 8.2 Dokazovanie s kvantifikátormi

### Dôkazy a tablá pre logiku prvého rádu

Dôkazy s kvantifikovanými formulami sformalizujeme pomocou rozšírenia tabiel na logiku prvého rádu.

Tablá budú obsahovať označené formuly prvého rádu.

V tabľách dovoľíme aj *otvorené* formuly.

Tablové pravidlá budú zachovávať splniteľnosť tabla.

### Označené formuly logiky prvého rádu

Podobne ako vo výrokovej logike môžeme zaviesť označovanie formúl logiky prvého rádu znamienkami **T** a **F**.

**Definícia 8.12.** Nech  $\mathcal{M}$  je štruktúra,  $e$  je ohodnotenie individuových premenných a  $X$  je formula. Potom

- *$\mathcal{M}$  spĺňa označenú formulu **T**  $X$  pri ohodnotení  $e$  vtt  $\mathcal{M}$  spĺňa formulu  $X$  pri ohodnotení  $e$ , skráteno:  $\mathcal{M} \models \mathbf{T}X[e]$  vtt  $\mathcal{M} \models X[e]$ ;*
- *$\mathcal{M}$  spĺňa označenú formulu **F**  $X$  pri ohodnotení  $e$  vtt  $\mathcal{M}$  nespĺňa formulu  $X$  pri ohodnotení  $e$ , skráteno:  $\mathcal{M} \models \mathbf{F}X[e]$  vtt  $\mathcal{M} \not\models X[e]$ .*

$\mathcal{M}$  spĺňa množinu označených formúl  $S^+$  pri ohodnotení  $e$  vtt  $\mathcal{M}$  spĺňa každú označenú formulu  $A^+$  z  $S^+$  pri ohodnotení  $e$ , skrátene:  $\mathcal{M} \models S^+[e]$  vtt pre každú  $A^+ \in S^+$  máme  $\mathcal{M} \models A^+[e]$ .

## Splniteľnosť označených formúl a ich množín

**Definícia 8.13** (Splniteľnosť označených formúl a ich množín). Ozn. formula  $X^+$  je *splniteľná* vtt pre nejakú štruktúru  $\mathcal{M}$  a nejaké ohodnotenie individuových premenných  $e$  máme  $\mathcal{M} \models X^+[e]$ .

Množina ozn. formúl  $S^+$  je *splniteľná* vtt pre nejakú štruktúru  $\mathcal{M}$  a nejaké ohodnotenie individuových premenných  $e$  máme  $\mathcal{M} \models S^+[e]$ .

*Príklad 8.14* (Dôkaz s pozitívnou všeobecnou kvantifikáciou). Dokážme neformálne, ale *veľmi podrobne*, že z teórie  $T = \{ \quad (1) \forall x(\text{opraví}(\text{TServis}, x) \rightarrow \text{tlačiareň}(x)), \quad (2) \neg \text{tlačiareň}(\text{môjMobil}) \}$  prvorádovo vyplýva (3)  $\neg \text{opraví}(\text{TServis}, \text{môjMobil})$ .

*Sporom:* Nech sú formuly (1) a (2) pravdivé v nejakej štruktúre  $\mathcal{M} = (D, i)$ , ale (3) je v nej nepravdivá, teda nie je splnená pri nejakom ohodnotení  $e$ .

Potom (4)  $\text{opraví}(\text{TServis}, \text{môjMobil})$  je splnená v  $\mathcal{M}$  pri  $e$ . Navyše (5)  $\text{tlačiareň}(\text{môjMobil})$  je nespĺnená. Pretože podľa prvého predpokladu (1) je formula  $(\text{opraví}(\text{TServis}, x) \rightarrow \text{tlačiareň}(x))$  splnená pri  $e(x/d)$  pre každé  $d \in D$ , musí byť splnená aj pre objekt  $i(\text{môjMobil})$ . Teda (6)  $(\text{opraví}(\text{TServis}, \text{môjMobil}) \rightarrow \text{tlačiareň}(\text{môjMobil}))$  je splnená pri  $e$ . Pretože už vieme, že ľavá strana je pravdivá (4), musí byť aj pravá strana (7)  $\text{tlačiareň}(\text{môjMobil})$  pravdivá. To je ale v spore so skorším zistením (5), že táto formula je nepravdivá.  $\square$

## Tablo pre dôkaz

Na väčšinu krokov v predchádzajúcom dôkaze stačia doterajšie tablové pravidlá.

1. $\mathbf{T} \forall x(\text{opraví}(\text{TServis}, x) \rightarrow \text{tlačiareň}(x))$	$S^+$
2. $\mathbf{T} \neg \text{tlačiareň}(\text{môjMobil})$	$S^+$
3. $\mathbf{F} \neg \text{opraví}(\text{TServis}, \text{môjMobil})$	$S^+$
4. $\mathbf{T} \text{opraví}(\text{TServis}, \text{môjMobil})$	$\alpha 3$
5. $\mathbf{F} \text{tlačiareň}(\text{môjMobil})$	$\alpha 2$
6. $\mathbf{T} (\text{opraví}(\text{TServis}, \text{môjMobil}) \rightarrow \text{tlačiareň}(\text{môjMobil}))$	?1
7. $\mathbf{T} \text{tlačiareň}(\text{môjMobil})$	MP4, 6
* 5, 7	

### Špeciálny prípad pravdivej všeobecne kvantifikovanej formuly

Doterajšie pravidlá ale nestačia na kľúčový krok, v ktorom sme z *pravdivej všeobecne kvantifikovanej formuly* (1)

$$\forall x(\text{opraví}(\text{TServis}, x) \rightarrow \text{tlačiareň}(x))$$

odvodili jej špeciálny prípad (*inštanciu*) (6) pre konštantu môjMobil:

$$(\text{opraví}(\text{TServis}, \text{môjMobil}) \rightarrow \text{tlačiareň}(\text{môjMobil}))$$

Táto formula, ale aj každá iná, ktorá vznikne analogicky dosadením hocijakého termu za premennú  $x$ , je logickým dôsledkom formuly (1).

### Pravidlo pre pravdivé všeobecne kvantifikované formuly

Na tento krok potrebujeme nové pravidlo:

$$\frac{\text{T } \forall x A}{\text{T } A\{x \mapsto t\}} \gamma$$

pre každú formulu  $A$ , každú premennú  $x$  a každý *term*  $t$ , ak spĺňajú dôležitú dodatočnú podmienku — viac o nej neskôr.

Zápis  $\{x \mapsto t\}$  označuje *substitúciu* — zobrazenie premenných na termy (v tomto prípade je toto zobrazenie iba jednoprvkové).

Zápis  $A\{x \mapsto t\}$  označuje *aplikáciu* substitúcie  $\{x \mapsto t\}$  na formulu  $A$  — je to formula, ktorá vznikne z formuly  $A$  nahradením *všetkých voľných výskytov* premennej  $x$  termom  $t$ .

### Špeciálny prípad nepravdivej existenčne kvantifikovanej formuly

Veľmi podobná situácia nastáva pre *nepravdivú existenčne kvantifikovanú formulu*, napr.

$$\text{F } \exists x(\text{opraví}(\text{TServis}, x) \wedge \text{mobil}(x)).$$

Inštancia

$$\text{F}(\text{opraví}(\text{TServis}, \text{Chrumko}) \wedge \text{mobil}(\text{Chrumko}))$$

je logickým dôsledkom pôvodnej označenej formuly.



Rovnako je jej logickým dôsledkom každá iná inštancia a môžeme sformulovať pravidlo:

$$\frac{\mathbf{F} \exists x A}{\mathbf{F} A\{x \mapsto t\}} \gamma$$

pre každú formulu  $A$ , každú premennú  $x$  a každý *term*  $t$ , ak (opäť) spĺňajú dôležitú dodatočnú podmienku.

### Dôkaz s $\mathbf{T} \forall x A$ a $\mathbf{F} \exists x A$

Pomocou nových pravidiel môžeme dokázať napr.  $\{\forall x(\text{opraví}(\text{TServis}, x) \rightarrow \text{tlačiareň}(x)), \forall x(\text{mobil}(x) \rightarrow \neg \text{tlačiareň}(x)), \text{mobil}(\text{môjMobil})\} \models \exists x(\text{mobil}(x) \wedge \neg \text{opraví}(\text{TServis}, x))$ :

1. $\mathbf{T} \forall x(\text{opraví}(\text{TServis}, x) \rightarrow \text{tlačiareň}(x))$	$S^+$
2. $\mathbf{T} \forall x(\text{mobil}(x) \rightarrow \neg \text{tlačiareň}(x))$	$S^+$
3. $\mathbf{T} \text{mobil}(\text{môjMobil})$	$S^+$
4. $\mathbf{F} \exists x(\text{mobil}(x) \wedge \neg \text{opraví}(\text{TServis}, x))$	$S^+$
5. $\mathbf{T}(\text{mobil}(\text{môjMobil}) \rightarrow \neg \text{tlačiareň}(\text{môjMobil}))$	$\gamma 2\{x \mapsto \text{môjMobil}\}$
6. $\mathbf{T} \neg \text{tlačiareň}(\text{môjMobil})$	MP5, 3
7. $\mathbf{F} \text{tlačiareň}(\text{môjMobil})$	$\alpha 6$
8. $\mathbf{T}(\text{opraví}(\text{TServis}, \text{môjMobil}) \rightarrow \text{tlačiareň}(\text{môjMobil}))$	$\gamma 1\{x \mapsto \text{môjMobil}\}$
9. $\mathbf{F} \text{opraví}(\text{TServis}, \text{môjMobil})$	MT8, 7
10. $\mathbf{F}(\text{mobil}(\text{môjMobil}) \wedge \neg \text{opraví}(\text{TServis}, \text{môjMobil}))$	$\gamma 4\{x \mapsto \text{môjMobil}\}$
11. $\mathbf{F} \text{mobil}(\text{môjMobil}) \quad \beta 10$ * 3, 11	12. $\mathbf{F} \neg \text{opraví}(\text{TServis}, \text{môjMobil}) \quad \beta 10$ 13. $\mathbf{T} \text{opraví}(\text{TServis}, \text{môjMobil}) \quad \alpha 12$ * 9, 13

### Dôkaz s pozitívnou existenčnou kvantifikáciou

**Príklad 8.15.** Dokážme neformálne, že z teórie  $T = \{ \quad (1) \forall x(\text{opraví}(\text{TServis}, x) \rightarrow \text{tlačiareň}(x)), \quad (2) \exists x \neg \text{tlačiareň}(x) \}$  prvorádovo vyplýva  $(3) \exists x \neg \text{opraví}(\text{TServis}, x)$

*Sporom:* Nech sú formuly (1) a (2) pravdivé v nejakej štruktúre  $\mathcal{M} = (D, i)$ , ale (3) je v nej nepravdivá, teda nie je splnená pri nejakom ohodnotení  $e$ .

Podľa (2) existuje v doméne objekt  $d$  taký, že  $\neg \text{tlačiareň}(x)$  je splnená pri ohodnotení  $e(x/d)$ . *Zoberme si jeden z takýchto objektov a označme ho napríklad premennou  $z$ .* Pri  $e(z/d)$  potom je (4)  $\neg \text{tlačiareň}(z)$  splnená, a teda (5)  $\text{tlačiareň}(z)$  je nespĺnená. Podľa (1) je formula (6)  $(\text{opraví}(\text{TServis}, z) \rightarrow \text{tlačiareň}(z))$  splnená. Pretože už vieme, že pravá strana je nespĺnená (5), je aj ľavá strana (7)  $\text{opraví}(\text{TServis}, z)$  nespĺnená. Podľa predpokladu dôkazu

sporom (3) je však aj jeho inštancia (8)  $\neg \text{opraví}(\text{TServis}, z)$  nesplnená, teda (9) je splnená opraví(TServis, z), čo je v spore so skorším zistením (7).  $\square$

### Pozitívna existenčná kvantifikácia a jej vlastná premenná

Kľúčovým krokom v predchádzajúcom dôkaze je označenie objektu (*svedka*), ktorý existuje podľa *pozitívnej existenčne* kvantifikovanej formuly

$$\mathbf{T} \exists x \neg \text{tlačiareň}(x),$$

*dočasným menom* — voľnou premennou  $z$  a odvodenie:

$$\mathbf{T} \neg \text{tlačiareň}(z).$$

⚠ Táto premenná sa *predtým na vetve nesmie vyskytovať voľná*. ⚠

Musí to byť *nová, vlastná* premenná pre formulu  $\mathbf{T} \exists x \neg \text{tlačiareň}(x)$ .

Vo všeobecnosti:

$$\frac{\mathbf{T} \exists x A}{\mathbf{T} A\{x \mapsto y\}} \delta$$

pre každú formulu  $A$ , každú premennú  $x$  a každú *novú premennú*  $y$ , ak (opäť) spĺňajú dôležitú dodatočnú podmienku.

### Prečo vlastná premenná?

Prečo potrebuje každá pozitívna existenčná formula vlastnú premennú?

Pravidlá *musia zachovávať splniteľnosť* vetiev v table. Konštanty a iné voľné premenné v table môžu označovať objekty s konfliktnými vlastnosťami. Ich dosadením za existenčne kvantifikovanú premennú by sme dospeli k *falošnému* sporu.

### Prečo vlastná premenná? — príklad

Vetva

n+1.  $\mathbf{T} \text{tlačiareň}(x)$


n+2.  $\mathbf{T} \exists x \neg \text{tlačiareň}(x)$

je *splniteľná* (napr. je splnená štruktúrou  $\mathcal{M} = (\{1, 2\}, i)$ ,  $i(\text{tlačiareň}) = \{1\}$  pri ohodnotení  $e = \{x \mapsto 1, \dots\}$ ).

Vetva

n+1.  $\neg$  tlačiareň( $x$ )

n+2.  $\neg \exists x \neg$  tlačiareň( $x$ )


n+3.  $\neg$  tlačiareň( $z$ )   $\delta 2\{x \mapsto z\}$

je *splniteľná* (napr. je splnená štruktúrou  $\mathcal{M} = (\{1, 2\}, i)$ ,  $i(\text{tlačiareň}) = \{1\}$  pri ohodnotení  $e = \{x \mapsto 1, z \mapsto 2, \dots\}$ ).

Chybná vetva

n+1.  $\neg$  tlačiareň( $x$ )

n+2.  $\neg \exists x \neg$  tlačiareň( $x$ )

n+3.  $\neg$  tlačiareň( $x$ )  „ $\delta 2\{x \mapsto x\}$

by bola *nesplniteľná*.

## Negatívna všeobecná kvantifikácia a jej vlastná premenná



*Negatívna všeobecne kvantifikovaná formula*

$\neg \forall x \text{ tlačiareň}(x)$ ,

znamená, že pre niektorý objekt  $x$  (*kontrapríklad*) je jej priama podformula tlačiareň( $x$ ) nepravdivá.

Tento objekt teda môžeme opäť označiť novou *vlastnou premennou* formule  $\neg \forall x \text{ tlačiareň}(x)$ , napríklad  $u$ , a môžeme odvodiť:

$\neg \text{tlačiareň}(u)$ .

 Táto premenná sa *predtým na vetve nesmie vyskytovať voľná*. 

Vo všeobecnosti:

$$\frac{\neg \forall x A}{\neg A\{x \mapsto y\}} \delta$$

pre každú formulu  $A$ , každú premennú  $x$  a každú *novú premennú*  $y$ , ak (opäť) spĺňajú dôležitú dodatočnú podmienku.

## Dôkaz s pravidlami pre kvantifikátory

$\{ \exists x \forall y (\text{opraví}(x, y) \rightarrow \text{tlačiareň}(y)),$

$\forall x (\text{mobil}(x) \rightarrow \neg \text{tlačiareň}(x)) \}$

$\models \forall x (\text{mobil}(x) \rightarrow \exists y \neg \text{opraví}(y, x))$ :

1.  $\mathsf{T} \exists x \forall y (\text{opraví}(x, y) \rightarrow \text{tlačiareň}(y))$   $S^+$
  2.  $\mathsf{T} \forall x (\text{mobil}(x) \rightarrow \neg \text{tlačiareň}(x))$   $S^+$
  3.  $\mathsf{F} \forall x (\text{mobil}(x) \rightarrow \exists y \neg \text{opraví}(y, x))$   $S^+$
  4.  $\mathsf{F} (\text{mobil}(u) \rightarrow \exists y \neg \text{opraví}(y, u))$   $\delta 3\{x \mapsto u\}$
  5.  $\mathsf{T} \text{mobil}(u)$   $\alpha 4$
  6.  $\mathsf{F} \exists y \neg \text{opraví}(y, u)$   $\alpha 4$
  7.  $\mathsf{T} \forall y (\text{opraví}(z, y) \rightarrow \text{tlačiareň}(y))$   $\delta 1\{x \mapsto z\}$
  8.  $\mathsf{T} (\text{mobil}(u) \rightarrow \neg \text{tlačiareň}(u))$   $\gamma 2\{x \mapsto u\}$
  9.  $\mathsf{T} \neg \text{tlačiareň}(u)$   $\text{MP} 8, 5$
  10.  $\mathsf{F} \text{tlačiareň}(u)$   $\alpha 9$
  11.  $\mathsf{T} (\text{opraví}(z, u) \rightarrow \text{tlačiareň}(u))$   $\gamma 7\{y \mapsto u\}$
  12.  $\mathsf{F} \text{opraví}(z, u)$   $\text{MT} 11, 10$
  13.  $\mathsf{F} \neg \text{opraví}(z, u)$   $\gamma 6\{y \mapsto z\}$
  14.  $\mathsf{T} \text{opraví}(z, u)$   $\alpha 13$
- \* 12, 14

## Tablové pravidlá pre logiku prvého rádu

**Definícia 8.16.** Pravidlami tablového kalkulu pre logiku prvého rádu sú pravidlá typu  $\alpha$  a  $\beta$  pre výrokovú logiku a pravidlá:

$$\begin{array}{ll}
 \gamma & \frac{\mathsf{T} \forall x A}{\mathsf{T} A\{x \mapsto t\}} \quad \frac{\mathsf{F} \exists x A}{\mathsf{F} A\{x \mapsto t\}} \quad \text{jednotne: } \frac{\gamma(x)}{\gamma_1(t)} \\
 \delta & \frac{\mathsf{F} \forall x A}{\mathsf{F} A\{x \mapsto y\}} \quad \frac{\mathsf{T} \exists x A}{\mathsf{T} A\{x \mapsto y\}} \quad \text{jednotne: } \frac{\delta(x)}{\delta_1(y)}
 \end{array}$$

kde  $A$  je formula,  $x$  je premenná,  $t$  je term *substituovateľný* za  $x$  v  $A$  a  $y$  je premenná *substituovateľná* za  $x$  v  $A$ .

Pri operácii rozšírenia vetvy tabla  $\pi$  o dôsledok niektorého z pravidiel typu  $\delta$  navyše musí platiť, že **premenná  $y$  nemá voľný výskyt v žiadnej formule na vetve  $\pi$ .**

Substituovateľnosť vysvetlíme nižšie.

## Korektnosť pravidiel $\gamma$ a $\delta$

**Tvrdenie 8.17** (Korektnosť pravidiel  $\gamma$  a  $\delta$ ). *Nech  $S^+$  je množina označených formúl v jazyku  $\mathcal{L}$ , nech  $x$  a  $y$  sú premenné, nech  $t$  je term.*

- Ak  $\gamma(x) \in S^+$  a  $t$  je substituovateľný za  $x$  v  $\gamma_1(x)$ , tak  $S^+$  je splniteľná vtt  $S^+ \cup \{\gamma_1(t)\}$  je splniteľná.
- Ak  $\delta(x) \in S^+$ ,  $y$  je substituovateľná za  $x$  v  $\delta_1(x)$  a  $y$  sa nemá voľný výskyt v  $S^+$ , tak  $S^+$  je splniteľná vtt  $S^+ \cup \{\delta_1(y)\}$  je splniteľná.

## Tablový kalkul pre logiku prvého rádu

Princíp tablových dôkazov ostáva nezmenený:

- Ak chceme dokázať, že formula  $X$  je platná, hľadáme uzavreté tablo pre  $S^+ = \{FX\}$ . Predpokladáme teda, že v nejakej štruktúre a nejakom ohodnotení je  $X$  nesplnená a ukážeme spor.
- Podobne pre prvorádové vyplývanie  $T \models X$  predpokladáme, že v nejakej štruktúre a nejakom ohodnotení sú splnené všetky formuly z  $T$  ( $TA$  pre  $A \in T$ ), ale  $X$  je nesplnená ( $FX$ ) a ukážeme spor, teda hľadáme uzavreté tablo pre  $S^+ = \{TA \mid A \in T\} \cup \{FX\}$ .

## Častá chyba pri pravidlách $\gamma$ a $\delta$ : aplikácia na podformuly

Vetva:

1.  $F \text{ mobil}(u)$
2.  $T \text{ tlačiareň}(u)$
3.  $T (\forall x \text{ tlačiareň}(x) \rightarrow \forall y \text{ mobil}(y))$

je *splniteľná*

(je splnená napr. štruktúrou  $\mathcal{M} = (\{1, 2\}, i)$ , kde  $i(\text{mobil}) = \{1\}$ ,  $i(\text{tlačiareň}) = \{2\}$  pri ohodnotení  $e = \{u \mapsto 2, \dots\}$ ).

V table:

1. $F \text{ mobil}(u)$	
2. $T \text{ tlačiareň}(u)$	
3. $T (\forall x \text{ tlačiareň}(x) \rightarrow \forall y \text{ mobil}(y))$	
4. $F \forall x \text{ tlačiareň}(x)$ ✓ $\beta 3$	5. $T \forall y \text{ mobil}(y)$ ✓ $\beta 3$
6. $F \text{ tlačiareň}(v)$ ✓ $\delta 4$	7. $T \text{ mobil}(u)$ ✓ $\gamma 3$
	* 7, 1

je ľavá vetva *splniteľná*

(je splnená napr. tou istou štruktúrou  $\mathcal{M}$  ako pôvodná vetva pri ohodnotení  $e = \{u \mapsto 2, v \mapsto 1 \dots\}$ ).

Chybná vetva:

1.  $\mathbf{F}$  mobil( $u$ )
2.  $\mathbf{T}$  tlačiareň( $u$ )
3.  $\mathbf{T} (\forall x \text{ tlačiareň}(x) \rightarrow \forall y \text{ mobil}(y))$
4.  $\mathbf{T} (\text{tlačiareň}(u) \rightarrow \forall y \text{ mobil}(y))$  ✖ „ $\gamma 3$ “
5.  $\mathbf{T} \forall y \text{ mobil}(y)$  MP4, 2
6.  $\mathbf{T} \text{ mobil}(u)$   $\gamma 5$

je *nesplniteľná*.

Tablové pravidlá sa *nikdy neaplikujú na podformuly* označenej formuly v uzle.

### 8.3 Substitúcia a substituovateľnosť

#### Substitúcia

**Definícia 8.18** (Substitúcia). *Substitúciou* (v jazyku  $\mathcal{L}$ ) nazývame každé zobrazenie  $\sigma : V \rightarrow \mathcal{T}_{\mathcal{L}}$  z nejakej množiny individuových premenných  $V \subseteq \mathcal{V}_{\mathcal{L}}$  do termov jazyka  $\mathcal{L}$ .

*Príklad 8.19.* Keď  $\mathcal{V}_{\mathcal{L}} = \{u, v, \dots, z, u_1, \dots\}$ ,  $\mathcal{C}_{\mathcal{L}} = \{\text{Klárka, Jurko}\}$ , napríklad  $\sigma_1 = \{x \mapsto \text{Klárka}, y \mapsto u, z \mapsto x\}$  je substitúcia.

#### Problém so substitúciou

Vetva

- n+1.  $\mathbf{T} \forall x \neg \text{pozná}(x, x)$
- n+2.  $\mathbf{T} \neg \text{pozná}(y, y)$   $\gamma 1\{x \mapsto y\}$
- n+3.  $\mathbf{T} \forall x \exists y \text{ pozná}(x, y)$

je *splniteľná* (napr. je splnená štruktúrou  $\mathcal{M} = (\{1, 2\}, i)$ ,  $i(\text{pozná}) = \{(1, 2), (2, 1)\}$  pri ohodnotení  $e = \{y \mapsto 1, \dots\}$ ).

Ale vetva

- n+1.  $\mathbf{T} \forall x \neg \text{pozná}(x, x)$
- n+2.  $\mathbf{T} \neg \text{pozná}(y, y)$   $\gamma 1\{x \mapsto y\}$
- n+3.  $\mathbf{T} \forall x \exists y \text{ pozná}(x, y)$
- n+4.  $\mathbf{T} \exists y \text{ pozná}(y, y)$  ✖ „ $\gamma 3\{x \mapsto y\}$ “

je *nesplniteľná*. Oprava: Vetva

- n+1.  $\top \forall x \neg \text{pozná}(x, x)$   
 n+2.  $\top \neg \text{pozná}(z, z) \quad \gamma 1 \{x \mapsto z\}$   
 n+3.  $\top \forall x \exists y \text{ pozná}(x, y)$   
 n+4.  $\top \exists y \text{ pozná}(z, y) \quad \checkmark \gamma 3 \{x \mapsto z\}$

je *splniteľná*.

**Definícia 8.20** (Substituovateľnosť, aplikovateľnosť substitúcie). Nech  $A$  postupnosť symbolov (term alebo formula), nech  $t, t_1, \dots, t_n$  sú termy a  $x, x_1, \dots, x_n$  sú premenné.

Term  $t$  je *substituovateľný* za premennú  $x$  v  $A$  vtt *nie je pravda*, že pre niektorú premennú  $y$  vyskytujúcu sa v  $t$  platí, že v nejakej oblasti platnosti kvantifikátora  $\exists y$  alebo  $\forall y$  vo formule  $A$  sa premenná  $x$  vyskytuje voľná.

Substitúcia  $\{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$  je *aplikovateľná* na  $A$  vtt term  $t_i$  je substituovateľný za  $x_i$  v  $A$  pre každé  $i \in \{1, \dots, n\}$ .

**Príklad 8.21.** Nech  $A = \exists y \text{ pozná}(x, y)$ .

- Substitúcia  $\{x \mapsto y, z \mapsto \text{Jurko}\}$  *nie je aplikovateľná* na  $A$ , lebo term  $y$  *nie je substituovateľný* za premennú  $x$  v  $A$ .
- Substitúcia  $\{x \mapsto z, y \mapsto \text{Jurko}, z \mapsto y\}$  je *aplikovateľná* na  $A$ .

## Substitúcia do postupnosti symbolov

**Definícia 8.22** (Substitúcia do postupnosti symbolov). Nech  $A$  je postupnosť symbolov, nech  $\sigma = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$  je substitúcia. Ak  $\sigma$  je aplikovateľná na  $A$ , tak  $A\sigma$  je postupnosť symbolov, ktorá vznikne *súčasným* nahradením každého *voľného* výskytu premennej  $x_i$  v  $A$  termom  $t_i$ .

**Príklad 8.23.** Nech  $A = \exists y \text{ pozná}(x, y)$  a  $\sigma = \{x \mapsto z, y \mapsto u, z \mapsto y\}$ .

Substitúcia  $\sigma$  je *aplikovateľná* na  $A$ . V  $A$  je *voľná* iba premenná  $x$ , dosadíme za ňu term  $z$ , ktorý neobsahuje viazanú premennú  $y$ . Všetky výskyty  $y$  sú *viazané*, za ne sa nedosádza. Premenná  $z$  sa v  $A$  nevyskytuje, nie je za čo dosadzovať.

$$A\sigma = \exists y \text{ pozná}(z, y)$$

## Substitúcia do termov a formúl rekurzívne

**Tvrdenie 8.24.** Pre každú substitúciu  $\sigma = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$ , každú premennú  $y \in \mathcal{V}_{\mathcal{L}} \setminus \{x_1, \dots, x_n\}$ , každý symbol konštanty  $a \in \mathcal{C}_{\mathcal{L}}$ , každý predikátový symbol  $P^k \in \mathcal{P}_{\mathcal{L}}$ , každé  $i \in \{1, \dots, n\}$ , každú spojku  $\diamond \in \{\wedge, \vee, \rightarrow\}$ , všetky formuly  $A$  a  $B$  a všetky termy  $s_1, s_2, \dots, s_k \in \mathcal{T}_{\mathcal{L}}$  platí:

$$\begin{array}{lll}
 x_i \sigma = t_i & y \sigma = y & a \sigma = a \\
 (s_1 \doteq s_2) \sigma = (s_1 \sigma \doteq s_2 \sigma) & (P(s_1, \dots, s_k)) \sigma = P(s_1 \sigma, \dots, s_k \sigma) & \\
 (\neg A) \sigma = \neg(A \sigma) & ((A \diamond B)) \sigma = (A \sigma \diamond B \sigma) & \\
 (\forall y A) \sigma = \forall y (A \sigma) & (\exists y A) \sigma = \exists y (A \sigma) & \\
 (\forall x_i A) \sigma = \forall x_i (A \sigma_i) & (\exists x_i A) \sigma = \exists x_i (A \sigma_i), & 
 \end{array}$$

kde  $\sigma_i = \sigma \setminus \{x_i \mapsto t_i\}$ , za predpokladu, že  $\sigma$  je v danom prípade aplikovateľná.

## 9 Formalizácia s viacerými kvantifikátormi

### Viacnásobné použitie rovnakého kvantifikátora

Použitím jedného kvantifikátora vo formule sme minulý týždeň dokázali vyjadriť pomerne komplikované tvrdenia.

Ale už v príklade tabiel sme videli, že niektoré tvrdenia zodpovedajú viacerým kvantifikátorom vo formule.

Rozoberme si niekoľko typických prípadov.

### 9.1 Rovnaký kvantifikátor

#### Viacnásobné použitie rovnakého kvantifikátora

Najjednoduchšie sú opakované použitia rovnakého kvantifikátora na začiatku formuly:

- $\exists x \exists y ((\text{človek}(x) \wedge \text{škrečok}(y)) \wedge \text{krími}(x, y))$
- $\forall x \forall y ((\text{človek}(x) \wedge \text{škrečok}(y)) \rightarrow \text{krími}(x, y))$

Význam je ľahké uhádnuť, aj keď je možno zrejmejší v alternatívnej forme, ktorá priamo zodpovedá aristotelovským formám obmedzenej kvantifikácie:



- $\exists x(\text{človek}(x) \wedge \exists y(\text{škrekčok}(y) \wedge \text{kŕmi}(x, y)))$  Nejaký človek (má vlastnosť, že) kŕmi nejakého škrekča.
- $\forall x(\text{človek}(x) \rightarrow \forall y(\text{škrekčok}(y) \rightarrow \text{kŕmi}(x, y)))$  Každý človek kŕmi každého škrekča.

### Prenexové vs. hlbšie vnorené formy

Dve uvedené formy každého typu tvrdenia sú vzájomne ekvivalentné, majú rovnaký význam.

Prvé formy sú *prenexové* — kvantifikátory sú na začiatku formuly.

⚠ Nie je vždy dobré snažiť sa o prenexovú formu, v zložitejších prípadoch môže byť zavádzajúca.

### Rôznosť objektov označených premennými — všeobecný prípad

Tento typ tvrdení je väčšinou bezproblémový až na jeden prípad:

$$\forall x \forall y ((\text{zvieratko}(x) \wedge \text{zvieratko}(y)) \rightarrow (\text{väčší}(x, y) \vee \text{menší}(x, y)))$$

nezodpovedá tvrdeniu: *Pre každé zvieratká  $x$  a  $y$  platí, že  $x$  je väčšie od  $y$  alebo  $x$  je menšie od  $y$ .*

Slovenské *každé (dve) zvieratká  $x$  a  $y$*  znamená, že  $x$  a  $y$  označujú naozaj viacero zvieratiek. Ale v logike prvého rádu je každá premenná kvantifikovaná samostatne a *rôzne premenné môžu označovať ten istý objekt*. Rôznosť musíme zapísať explicitne:

$$\forall x \forall y ((\text{zvieratko}(x) \wedge \text{zvieratko}(y) \wedge x \neq y) \rightarrow (\text{väčší}(x, y) \vee \text{menší}(x, y)))$$

Pre ľubovoľné termý  $s, t$  je  $s \neq t$  je skratka za  $\neg s \doteq t$ .

### Rôznosť objektov označených premennými — existenčný prípad

Podobne formula

$$\exists x \exists y (\text{zvieratko}(x) \wedge \text{zvieratko}(y))$$

*neznamená, že existujú aspoň dve zvieratká (je ekvivalentná s  $\exists x \text{zvieratko}(x)$ ).*

Existenciu aspoň dvoch zvieratiek zabezpečí formula:

$$\exists x \exists y (\text{zvieratko}(x) \wedge \text{zvieratko}(y) \wedge x \neq y)$$

Podľa dohody zo 4. prednášky do seba vnorené vľavo uzátvorkované konjunkcie skrátene zapisujeme bez vnútorných zátvoriek. Teda  $(\text{zvieratko}(x) \wedge \text{zvieratko}(y) \wedge x \neq y)$  je skrátenejší zápis  $((\text{zvieratko}(x) \wedge \text{zvieratko}(y)) \wedge x \neq y)$ . Podobne skrácujeme do seba vnorené disjunkcie.

## 9.2 Alternácia kvantifikátorov

### Existencia pre všetky

Časté formuly, v ktorých sa vyskytujú oba kvantifikátory, sú ako

$$\forall x (\text{zvieratko}(x) \rightarrow \exists y (\text{človek}(y) \wedge \text{kŕmi}(y, x)))$$

Hovorí, že *každé zvieratko má vlastnosť, že nejaký človek ho kŕmi*, teda *každé zvieratko niekto kŕmi*.

Ekvivalentne sa to dá vyjadriť aj (v menej vernej) prenexovej forme:

🗨️  $\forall x \exists y (\text{zvieratko}(x) \rightarrow (\text{človek}(y) \wedge \text{kŕmi}(y, x)))$

### Poradie kvantifikátorov

Pri rovnakých kvantifikátoroch v prenexovej forme na ich poradí nezáleží:

- $\forall x \forall y \text{ má\_réd}(x, y)$  je ekvivalentné  $\forall y \forall x \text{ má\_réd}(x, y)$ ;
- $\exists x \exists y \text{ má\_réd}(x, y)$  je ekvivalentné  $\exists y \exists x \text{ má\_réd}(x, y)$ .

Pri rôznych kvantifikátoroch zmena poradia vážne mení význam:

- $\forall x \exists y \text{ má\_réd}(x, y)$  — *Každý/-á má rád/rada niekoho.*
- $\exists x \forall y \text{ má\_réd}(x, y)$  — *Niektor má rád/rada všetkých.*

## Poradie kvantifikovaných premenných

Záleží aj na tom, ako sa kvantifikované premenné použijú vo formule, ktorá je kvantifikovaná.

Porovnajme:

- $\forall x \exists y \text{ má\_réd}(\underline{x}, y) - \underline{\text{Každý/-á má rád/rada niekoho.}}$
- $\forall x \exists y \text{ má\_réd}(y, \underline{x}) - \underline{\text{Každú/-ého má niekto rád.}}$

a

- $\exists x \forall y \text{ má\_réd}(\underline{x}, y) - \underline{\text{Nieкто má rada/rád všetkých.}}$
- $\exists x \forall y \text{ má\_réd}(y, \underline{x}) - \underline{\text{Nieкого majú radi všetci.}}$

Cvičenie: Pre každú z týchto štyroch formúl nájdite štruktúru, v ktorej je pravdivá, ale ostatné sú nepravdivé.

## Unikátna existencia

Kombináciou oboch kvantifikátorov s rovnosťou môžeme vyjadriť existenciu *práve jedného* (unikátneho) objektu s danou vlastnosťou:

$$\exists x(\text{škrekčok}(x) \wedge \forall y(\text{škrekčok}(y) \rightarrow x \doteq y))$$

Neformálne: *Nejaký škrekčok je jediným škrekčkom.*

Podobne sa dá vyjadriť existencia práve  $k$  objektov pre každé prirodzené číslo  $k$ .

## 9.3 Postupná formalizácia a parafrázovanie

### Postupná formalizácia

Na formalizáciu zložitých tvrdení je najlepšie ísť postupne.

Sformalizujme: *Každého škrekča kŕmi nejaké dieťa.*

1. Rozpoznáme, že tvrdenie má tvar *Všetky P sú Q*, pričom  $P$  je atomická vlastnosť. Môžeme ho teda čiastočne sformalizovať na:

$$\forall x(\text{škrekčok}(x) \rightarrow \text{nejaké dieťa kŕmi } x)$$

2. Sformalizujeme *nejaké dieťa kŕmi x*: Má formu: *Nejaké P je Q*:


$$\exists y(\text{dieťa}(y) \wedge \text{kŕmi}(y, x))$$

3. Dosadíme:


$$\forall x(\text{škrekčok}(x) \rightarrow \exists y(\text{dieťa}(y) \wedge \text{kŕmi}(y, x)))$$


Systematickým prístupom sa dajú správne sformalizovať aj veľmi zložité tvrdenia.

### Viacnásobná negácia — nesprávne možnosti

 Opatrnosť je potrebná pri formalizácii tvrdení s viacnásobnou negáciou, napríklad: „*Nijaké dieťa nechová žiadnu vretenicu.*“

Tu sa ľahko stane, že pri *neopatrnej* postupnej formalizácii skončíme s chybnou formulou:

  $\neg \exists x(\text{dieťa}(x) \wedge \neg \exists y(\text{vretenicu}(y) \wedge \text{chová}(x, y)))$  — *Nie je pravda, že nejaké dieťa nemá vlastnosť, že chová nejakú vretenicu, teda Každé dieťa má vlastnosť, že chová nejakú vretenicu, teda Každé dieťa chová nejakú vretenicu.*

  $\neg \exists x(\text{dieťa}(x) \wedge \neg \exists y(\text{vretenicu}(y) \wedge \neg \text{chová}(x, y)))$  — *Nie je pravda, že nejaké dieťa nemá vlastnosť, že nechová nejakú vretenicu, teda Každé dieťa nechová nejakú vretenicu (ale môže chovať iné).*

### Viacnásobná negácia — parafráza a správna formalizácia

Na správne sformalizovanie „*Žiadne dieťa nechová žiadnu vretenicu*“ je lepšie toto tvrdenie *parafrázovať*:

- *Nie je pravda, že nejaké dieťa je také, že chová nejakú vretenicu.*

  $\neg \exists x(\text{dieťa}(x) \wedge \exists y(\text{vretenicu}(y) \wedge \text{chová}(x, y)))$

- *Pre každé dieťa nie je pravda, že chová nejakú vretenicu.*

  $\forall x(\text{dieťa}(x) \rightarrow \neg \exists y(\text{vretenicu}(y) \wedge \text{chová}(x, y)))$

- *Pre každé dieťa x je pravda, že pre každú vretenicu y je pravda, že x nechová y.*

  $\forall x(\text{dieťa}(x) \rightarrow \forall y(\text{vretenicu}(y) \rightarrow \neg \text{chová}(x, y)))$

## Viacnásobná negácia — nesprávna parafráza

- ⚠ Podľa našich skúseností študenti/-ky najčastejšie výroky „*Žiadne dieťa nechová žiadnu vretenicu*“, „*nesprávne parafrázujú ako*“
- ✗ „*Neexistuje dieťa  $x$  a neexistuje vretenica  $y$ , pre ktoré je pravda, že  $x$  chová  $y$* “, čo navyše následne sformalizujú v prenexovom tvare ako

$$\text{✗ } \neg \exists x \neg \exists y (\text{dieťa}(x) \wedge \text{vretenica}(y) \wedge \text{chová}(x, y))$$

„Každý objekt v doméne je dieťa, ktoré chová nejakú vretenicu.“

Pri parafrázovaní preto:

- ✗ nepoužívajte „*neexistuje ...*“,
- ✓ namiesto toho používajte: „*nie je pravda, že existuje ...*“;
- ✗ nespájajte kvantifikátory spojkami „*a*“, „*alebo*“,
- ✓ za kvantifikátorom začnite vedľajšiu vetu: „*Existuje  $x$  také, že  $x$  je dieťa a existuje  $y$  také, že ...*“

## Postupná formalizácia, negácia a databázy

Niekedy sa oplatí pozrieť na tvrdenie cez jeho negáciu.

Užitočné napríklad pri formalizácii do databázových jazykov — dopyty v datalogu či SQL (bez agregácie) možno vyjadriť formulami prvorádovej logiky, ale *nemožno použiť všeobecný kvantifikátor*.

Skúste schematicky zakresliť situáciu k výrokovej forme „*človek, ktorý pozná všetkých známych svojich známych*“.

$$\text{človek}(x) \wedge \forall y (\text{pozná}(x, y) \rightarrow \forall z (\text{pozná}(y, z) \rightarrow \text{pozná}(x, z)))$$

Opak: človek, čo nepozná niektorého známeho svojho známeho.

$$\text{človek}(x) \wedge \neg \exists y \exists z (\text{pozná}(x, y) \wedge \text{pozná}(y, z) \wedge \neg \text{pozná}(x, z))$$

## Odkaz z konzekventu — o sedliakoch a osloch

Už minule sme rozoberali zdanlivo existenčné tvrdenia typu:

*Ak nejaký prvák navštevuje LPI, tak (on) je bystrý.*

Postupnou formalizáciou by sme mohli dospieť k nesprávnej otvorenej formule:

$$\times \quad (\exists x(\text{prvák}(x) \wedge \text{navštevuje}(x, \text{LPI})) \rightarrow \text{bystrý}(x)).$$

$$\checkmark \quad \forall x((\text{prvák}(x) \wedge \text{navštevuje}(x, \text{LPI})) \rightarrow \text{bystrý}(x)).$$

Vyskytujú sa aj v zložitejších kombináciách. Úderným príkladom je:

*Každý sedliak, ktorý vlastní nejakého osla, ho bije.*

Na existenčné tvrdenie *vlastní nějakého osla* v antecedente odkazuje zámeno *ho* v konzekvente.

### Odkaz z konzekventu — nesprávne možnosti

Postupnou formalizáciou by sme mohli dostať nesprávnu formulu:

$$\times \quad \forall x((\text{sedliak}(x) \wedge \exists y(\text{osol}(y) \wedge \text{vlastní}(x, y))) \rightarrow \text{bije}(x, y))$$

Keby sme sa ju pokúsili „zachrániť“ tým, že zaviazeme premennú  $y$ , mohlo by to dopadnúť rôzne, ale stále neprávne:

$$\times \quad \forall x(\text{sedliak}(x) \wedge \exists y(\text{osol}(y) \wedge \text{vlastní}(x, y) \wedge \text{bije}(x, y)))$$

— *Všetko je sedliak, který vlastní osla, kterého bije.*

$$\times \quad \forall x(\text{sedliak}(x) \rightarrow \exists y(\text{osol}(y) \wedge \text{vlastní}(x, y) \wedge \text{bije}(x, y)))$$

— *Každý sedliak určite vlastní osla, kterého bije.*

Existenčný kvantifikátor teda nefunguje.

### Odkaz z konzekventu — parafráza a správna formalizácia

Na správne sformalizovanie je tvrdenie *Každý sedliak, ktorý vlastní nejakého osla, ho bije*, potrebné *parafrázovať* na

- *Každý sedliak bije každého osla, kterého vlastní.*
- *Pre každého osla je pravda, že každý sedliak, ktorý ho vlastní, ho bije.*

Z parafráz už ľahko dostaneme správne formalizácie:

- ✓  $\forall x(\text{sedliak}(x) \rightarrow \forall y((\text{osol}(y) \wedge \text{vlastní}(x, y)) \rightarrow \text{bije}(x, y)))$
- ✓  $\forall x(\text{osol}(x) \rightarrow \forall y((\text{sedliak}(y) \wedge \text{vlastní}(y, x)) \rightarrow \text{bije}(y, x)))$

## 9.4 Závislosť od kontextu

### Nejednoznačné tvrdenia

Každú minútu v New Yorku prepadne jedného človeka. Dnes  
nám poskytne rozhovor. — SNL

Vtip spočíva v potenciálnej nejednoznačnosti prvej vety. Pravdepodobne  
ste ju pochopili („slabé“ čítanie)

$$\forall x(\text{minúta}(x) \rightarrow \exists y(\text{človek}(y) \wedge \text{prepadnutýPočas}(x, y)))$$

Ale druhá veta vyzdvihla menej pravdepodobný alternatívny význam („silné“  
čítanie):

$$\exists y(\text{človek}(y) \wedge \forall x(\text{minúta}(x) \rightarrow \text{prepadnutýPočas}(x, y)))$$

Závisí od situácie, ktoré z čítaní je správne. Formalizácia je teda *kontextovo závislá*.

## 9.5 Dodatky k formalizácii s jedným kvantifikátorom

### Enumerácia — vymenovanie objektov s vlastnosťou

Niekedy potrebujeme vymenovať objekty s nejakou vlastnosťou:

- Na bunke č. 14 bývajú Ad'a, Biba, Ciri, Dada.

$$(\text{býva\_v}(\text{Ad'a}, \text{bunka14}) \wedge \dots \wedge \text{býva\_v}(\text{Dada}, \text{bunka14}))$$

Ekvivalentne: Každá z Ad'a, Biba, Ciri, Dada býva v bunke č. 14.

$$\forall x((x \doteq \text{Ad'a} \vee \dots \vee x \doteq \text{Dada}) \rightarrow \text{býva\_v}(x, \text{bunka14}))$$

- Na bunke č. 14 bývajú iba Ad'a, Biba, Ciri, Dada.

Každý, kto býva v bunke č. 14, je jedna z Ad'a, Biba, Ciri, Dada.

$$\forall x(\text{býva\_v}(x, \text{bunka14}) \rightarrow (x \doteq \text{Ad'a} \vee \dots \vee x \doteq \text{Dada}))$$

## Výnimky a implikátúra

Tvrdenia s výnimkami niekedy vyznievajú silnejšie, ako naozaj sú.

*Mám rád všetko ovocie, okrem jablák.*

Toto tvrdenie zodpovedá aristotelovskej forme: *Každé P je Q*, kde *P* = ovocie a *nie jablko* a *Q* = *také, že ho mám rád*, teda formálne:

$$\forall x((\text{ovocie}(x) \wedge \neg \text{jablko}(x)) \rightarrow \text{mám\_rád}(x))$$

Je *veľmi* lákavé z tohto tvrdenia usúdiť, že navyše znamená: *Jablká nemám rád*, ale je to iba implikátúra (zdanlivý dôsledok).

K *Mám rád všetko ovocie, okrem jablák* môžeme síce prekvapivo, ale *bez sporu* dodať:

- *Jablká milujem.*
- *Z jablák mám rád iba červené.*

V spore s pôvodným tvrdením by bol dodatok: *Ale slivky nemám rád*, pretože slivky sú ovocie a nie sú jablká, takže podľa pôvodného tvrdenia ich mám rád.

## Literatúra

Martin Davis and Hillary Putnam. A computing procedure for quantification theory. *J. Assoc. Comput. Mach.*, 7:201–215, 1960.

Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, 1962.

T. Kruger et al. Too much information: Why CDCL solvers need to forget learned clauses. *Plos One*, 2022. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9417043/>.

Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994. ISBN 978-0-201-53082-7.

Raymond M. Smullyan. *Logika prvého rádu*. Alfa, 1979. Z angl. orig. *First-Order Logic*, Berlin-Heidelberg: Springer-Verlag, 1968 preložil Svätoslav Mathé.



L. Zhang. SAT-Solving: From Davis-Putnam to Zchaff and beyond. [Online]  
[https://www.inf.ed.ac.uk/teaching/courses/propm/papers/Zhang/sat\\_course1.pdf](https://www.inf.ed.ac.uk/teaching/courses/propm/papers/Zhang/sat_course1.pdf),  
[https://www.inf.ed.ac.uk/teaching/courses/propm/papers/Zhang/sat\\_course2.pdf](https://www.inf.ed.ac.uk/teaching/courses/propm/papers/Zhang/sat_course2.pdf),  
[https://www.inf.ed.ac.uk/teaching/courses/propm/papers/Zhang/sat\\_course3.pdf](https://www.inf.ed.ac.uk/teaching/courses/propm/papers/Zhang/sat_course3.pdf).