

Logika pre informatikov a Úvod do matematickej logiky

Poznámky z prednášok

Ján Kl'uka, Ján Mazák, Jozef Šiška

Letný semester 2024/2025

Posledná aktualizácia: 13. mája 2025

Obsah

P1	Úvod. Atomické formuly a štruktúry	6
0	Úvod	6
0.1	O logike	6
0.2	O kurzoch LPI a UdML	15
1	Atomické formuly a štruktúry	16
1.1	Syntax atomických formúl	22
1.2	Štruktúry	26
1.3	Sémantika atomických formúl	29
1.4	Zhrnutie	30

P2	Výrokovologické spojky a ohodnotenia	32
2	Výrokovologické spojky a ohodnotenia	32
2.1	Boolovské spojky	33
2.2	Implikácia	39
2.3	Ekvivalencia	41
2.4	Správnosť a vernosť formalizácie	42
2.5	Syntax výrokovologických formúl	45
2.6	Sémantika výrokovologických formúl	54
2.7	Teórie a ich modely	56
2.8	Výrokovologické ohodnotenia	57
P3	Výrokovologické vyplývanie, sémantické vlastnosti formúl a ekvivalencia	64
3	Výrokovologické vyplývanie	64
3.1	Výrokovologické teórie a modely	65
3.2	Vyplývanie, nezávislosť a nesplniteľnosť	66
4	Sémantické vlastnosti a vzťahy formúl	73
4.1	Tautológie, splniteľné, falzifikovateľné a nesplniteľné formuly	73
4.2	Hľadanie ohodnotení	78
4.3	Ekvivalencia	80
4.4	Vzťah tautológií, vyplývania a ekvivalencie	84
4.5	Ekvivalentné úpravy a CNF	86
4.6	CNF vs. XOR	92
P4	Dôkazy a výrokovologické tablá	94
5	Dôkazy a výrokovologické tablá	94
5.1	Druhy dôkazov	97
5.2	Výrokovologické tablá	99

P5	Korektnosť a úplnosť výrokovologických tabiel	109
5.3	Korektnosť tabiel	109
5.4	Testovanie nesplniteľnosti, splniteľnosti a falzifikovateľnosti	113
5.5	Úplnosť	115
5.6	Nové korektné pravidlá	116
5.7	Iné dokazovacie systémy	124
P6	SAT solvery	130
6	SAT solvery	130
6.1	Problém výrokovologickej splniteľnosti (SAT)	130
6.2	Výpočtová zložitosť: teória a prax (<i>informatívne</i>)	132
6.3	Algoritmy na riešenie problému splniteľnosti	134
6.4	Backtracking	135
6.5	DPLL a sledované literály	139
6.6	CDCL	141
6.7	Ďalšie aspekty (<i>informatívne</i>)	149
6.8	Verifikácia hardvéru (<i>informatívne</i>)	151
6.9	Kombinatorické problémy (<i>informatívne</i>)	153
6.10	SMT solvery (<i>informatívne</i>)	157
P7	Kvantifikátory	160
7	Kvantifikátory	160
7.1	Kvantifikácia	160
7.2	Kvantifikátory a premenné	161
7.3	Syntax relačnej logiky prvého rádu	163
7.4	Sémantika relačnej logiky prvého rádu	168
7.5	Neexistencia	175
7.6	Aristotelovské formy	176
7.7	Zamĺčané a zdanlivo opačné kvantifikátory	179
7.8	Nutné a postačujúce podmienky	181
7.9	Zložené kvantifikované vlastnosti	182
7.10	Konverzačné implikatúry	184

7.11	Vyhýbanie sa chybám	186
P8	Tablá pre kvantifikátory. Viackvantifikátorové tvrdenia	188
8	Tablá s kvantifikátormi	188
8.1	Logické vlastnosti a vzťahy v logike prvého rádu	188
8.2	Dokazovanie s kvantifikátormi	192
8.3	Substitúcia a substituovateľnosť	200
9	Formalizácia s viacerými kvantifikátormi	202
9.1	Rovnaký kvantifikátor	202
9.2	Alternácia kvantifikátorov	204
9.3	Postupná formalizácia a parafrázovanie	205
9.4	Závislosť od kontextu	209
9.5	Dodatky k formalizácii s jedným kvantifikátorom	209
P9	Funkčné symboly. Tablá s rovnosťou	211
10	Logika prvého rádu	211
10.1	Funkčné symboly	211
10.2	Syntax logiky prvého rádu	216
10.3	Sémantika logiky prvého rádu	220
11	Tablá pre logiku prvého rádu	224
11.1	Vlastnosti rovnosti	225
11.2	Tablové pravidlá pre rovnosť	226
11.3	Tablá pre logiku prvého rádu	229
12	Vlastnosti kvantifikátorov	231

P10 Korektnosť prvorádových tabiel. Explicitné definície. Unifikácia	234
13 Korektnosť tablového kalkulu pre logiku prvého rádu	234
13.1 Vlastnosti ohodnotení a substitúcie	234
13.2 Korektnosť tabiel	235
13.3 Ďalšie korektné pravidlá	238
14 Explicitné definície	238
15 Unifikácia termov	246
P11 Rezolvenca	253
16 Rezolvenca	253
16.1 Rezolvenca vo výrokovkej logike	253
16.2 Prevod do klauzálnej teórie a skolemizácia	260
16.3 Rezolvenca v logike prvého rádu	267
P12 Nerozhodnuteľnosť, neúplnosť a logika 2. rádu	272
17 Logika 1. a 2. rádu	272
18 Nerozhodnuteľnosť platnosti formuly	275
19 Deskriptívna zložitosť	278
20 Neúplnosť	279
21 Dôkazy ako programy	283
22 Nadväzujúce predmety	288

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 kládeme 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 práčne) 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 príklade: 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 zložené 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:

- $\text{all}(x > m \text{ for } x \text{ in } \text{arr})$,

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 dvojtretinovou 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 vecí, 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?

$$\rightsquigarrow k = 3 \cdot m$$

$$k + m = 12$$

Stretli ste sa už aj s formálnym jazykom výrokovkej logiky.

Príklad 0.6. Sformalizujme náš párty 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

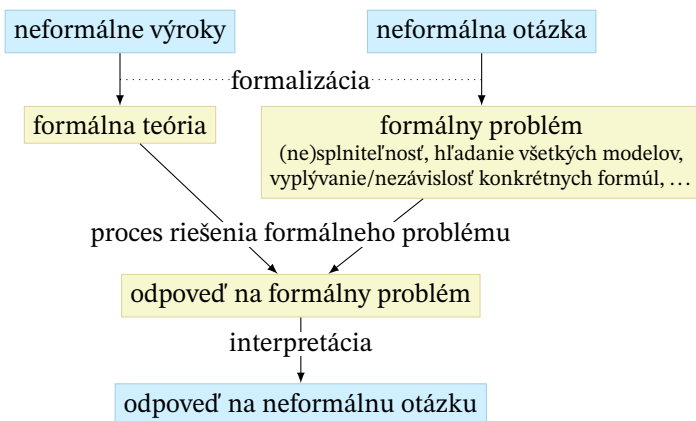
Nevýhodou výpočtov založených na dátach je chýbajúca kontrola nad smerovaním výpočtu a nemožnosť pochopenia/overenia.

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 *neoddeluje 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



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]

Odteraz až navždy každú sekundu vznikne jadro hélia.

Problémy s výroky

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 výroky

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 výroky

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

$\forall x \in \mathbb{R} : x > 0 \rightarrow x$ 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 Čaputová. |
| ✓ 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_Čaputová, sobota, π , ...

Individuová konštanta:

- 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);
 - *nemusí* 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^1 , vidí^2 , dal^4 , $<^2$.

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 mladší² môže označovať vzťah „*x* je mladší ako *y*“ presne.

Predikát mladý¹ 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 Bobyho.

↪ ~~$d(\text{Jarka})$~~ ~~$\text{dalBobyho}(\text{Jarka}, \text{Milo})$~~ $\text{dal}(\text{Jarka}, \text{Milo}, \text{Boby})$

A_2 : Evka dostala Bobyho od Mila.

↪ ~~$\text{dalBobyho}(\text{Milo}, \text{Evka})$~~ $\text{dal}(\text{Milo}, \text{Evka}, \text{Boby})$

A_3 : Evka dala Jarke Čilku.

↪ ~~$\text{dalČilku}(\text{Evka}, \text{Jarka})$~~ $\text{dal}(\text{Evka}, \text{Jarka}, \text{Čilka})$

A_4 : Boby je pes.

↪ $\text{pes}(\text{Boby})$

Návrh jazyka pri formalizácii

Minimalizujeme počet predikátov, uprednostňujeme flexibilnejšie, viacúčelovejšie (dal^3 pred dalBobyho^2 a 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 zadefinovať 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* $ar_{\mathcal{L}}(P) \in \mathbb{N}^+$.

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 *ľubovolnom* 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}_{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

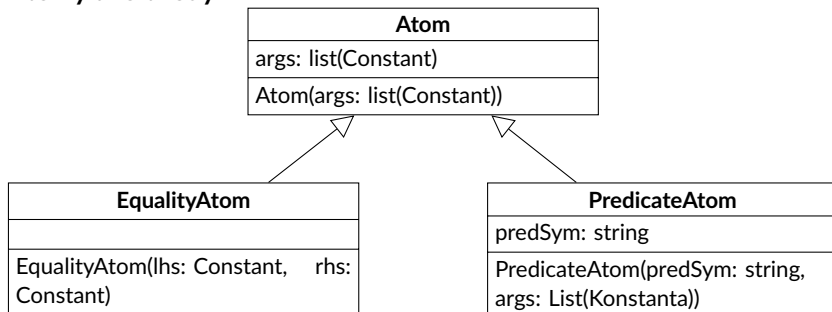
a predikátové atómy:

pes(Cilka)

dal(Cilka, Milo, Boby)

dal(Jarka, Evka, Milo).

Atómy ako triedy



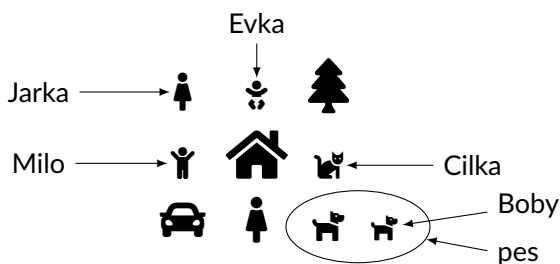
1.2 Štruktúry

Vyhodnotenie atomickej formuly

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

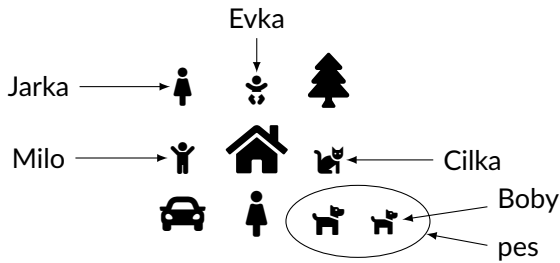
Definícia 1.14. Nech \mathcal{L} je jazyk atomických formúl logiky prvého rádu. *Štruktúrou* pre jazyk \mathcal{L} (niekedy *interpretáciou* jazyka \mathcal{L}) nazývame dvojicu $\mathcal{M} = (D, i)$, kde D je ľubovoľná neprázdna množina nazývaná *doména* štruktúry \mathcal{M} ; i je zobrazenie, nazývané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

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

Štruktúra ako informatický objekt

Štruktúru sme definovali pomocou *matematických* objektov.

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

Predikátové symboly jazyka ~ zjednodušená databázová schéma (arita ~ počet stĺpcov)

Interpretácia predikátových symbolov ~ konkrétne tabuľky s dátami (doména ~ dátový typ)

$i(\text{pes}^1)$
1

$i(\text{dal}^3)$		
1	2	3

$i(\text{clovek}^2)$	
Meno	Rodné číslo
Petra	1234
Michal	5678

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}* skrátene 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).

$$\mathcal{M} = (D, i), \quad D = \left\{ \text{ľudia, strom, dom, pes, auto, žena, mačka} \right\}$$

$$i(\text{Boby}) = \text{ľudia} \quad i(\text{Cilka}) = \text{mačka}$$

$$i(\text{Evka}) = \text{ľudia} \quad i(\text{Jarka}) = \text{ľudia} \quad i(\text{Milo}) = \text{ľudia}$$

$$i(\text{pes}) = \{ \text{ľudia, mačka} \}$$

$$i(\text{dal}) = \left\{ (\text{ľudia, strom, mačka}), (\text{ľudia, žena, mačka}), (\text{ľudia, žena, mačka}) \right\}$$

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 $\{ \text{ľudia, mačka} \} = i(\text{pes})$.

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 $(i(\text{Evka}), i(\text{Jarka}), i(\text{Cilka})) = (\text{ľudia, ľudia, mačka}) \in i(\text{dal})$.

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 neprázdnom 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. □

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> .
$(\neg \text{doma}(\text{Karol}))$	nesprávna
$\neg(\text{doma}(\text{Karol}))$	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.
(doma(Jarka) \wedge doma(Karol))
- Jarka je v škole, *no* Karol je doma.
(v_škole(Jarka) \wedge doma(Karol))
- *Ani* Jarka nie je doma, *ani* Karol tam nie je.
(\neg doma(Jarka) \wedge \neg doma(Karol))
- *Nielen* Jarka je chorý, *ale aj* Karol je chorý.
(chorý(Jarka) \wedge chorý(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 pesík.*
((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 spojкам *alebo*, či v *inkluzívnom* význame (môžu nastať aj obe možnosti). Inkluzívnu disjunkciu vyjadruje tiež „*alebo aj/i*“ a častice *respektíve*, *eventuálne*, *popripade*, *prípadne*.

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. (kúpe(Jarka, Boby) \vee kúpe(Karol, 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. (doma(Jarka) \vee doma(Karol))
- Jarka je doma alebo v škole. (doma(Jarka) \vee v_škole(Jarka))
- Jarka dostala Bobyho od mamy alebo otca. (dostal(Jarka, Boby, mama) \vee dostal(Jarka, Boby, otec))
- Boby je čierny či tmavohnedý psík. ((čierny(Boby) \vee tmavohnedý(Boby)) \wedge pes(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:

$$\begin{aligned} & ((\text{vybitá}(\text{batéria}) \vee \text{svieti}(\text{kontrolka})) \wedge \\ & \neg(\text{vybitá}(\text{batéria}) \wedge \text{svieti}(\text{kontrolka}))) \end{aligned}$$

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 spojkami 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.
Niekto 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 Boby šťastný. Aj Karol je doma, aj Jarka je doma, alebo je Boby 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, Bobyho).

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 (príde(Kim) \rightarrow príde(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 (príde(Jim) \rightarrow príde(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 prejdenie 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 Nemeč, ako je Stirlitz Rus. ($\text{Nemeč}(\text{Müller}) \leftrightarrow \text{Rus}(\text{Stirlitz})$)

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 Nemeč a je Rus*, ktorá v úlohe nebola explicitne uvedená.

Uprednostňujeme ich vyjadrovanie *samostatnými formulami*.

Rovnaké dôvody ako pre vernosť.

Skutočné súčasti významu a konverzačné implikatúry

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 implikatúry

Časť významu tvrdenia, ktorú *môžeme poprieť* dodatkami bez sporu s pôvodným tvrdením, sa nazýva *konverzačná implikatúra* (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 implikatúra.

- 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 žiaľbohu a niekedy je to aj chvalabohu, že žiaľbohu, na Slovensku preda 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 $\mathcal{C}_{\mathcal{L}} = \{\text{Kim, Jim, Sarah}\}$ 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 spojku, napr. $(\neg\text{príde}(\text{Kim}) \vee \text{príde}(\text{Sarah}))$.
- Ale negovať a spájať spojками 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?

Induktívnu 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 skrakuje „vtedy a len vtedy, keď“.

Výrokovologické formuly by sa dali alternatívne zdefinovať 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))

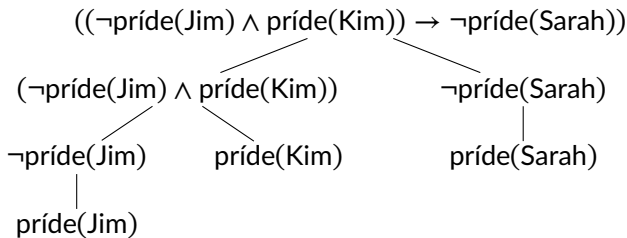
ale

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

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:

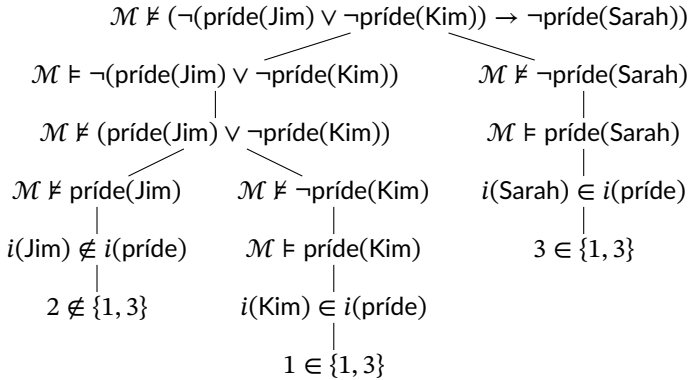
- $\mathcal{M} \models c_1 \doteq c_2$ vtt $i(c_1) = i(c_2)$,
- $\mathcal{M} \models P(c_1, \dots, c_n)$ vtt $(i(c_1), \dots, i(c_n)) \in i(P)$,
- $\mathcal{M} \models \neg A$ vtt $\mathcal{M} \not\models A$,
- $\mathcal{M} \models (A \wedge B)$ vtt $\mathcal{M} \models A$ a zároveň $\mathcal{M} \models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

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

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 štruktúr je nekonečne veľa a ak má teória jeden model, má aj nekonečne veľa ďalších:

$\mathcal{M}_1 = (\{k, j, s\}, i_1)$	$\mathcal{M}'_1 = (\{k, j, s, 0, 1\}, i'_1)$	$\mathcal{M}''_1 = (\{2, 4, 6\}, i''_1)$	\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_{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_{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 ohodnotení

Ako vyhodnotíme, či je formula pravdivá v nejakom ohodnotení?

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* ($v \vDash_p A$) definujeme *induktívne* pre všetky predikátové atómy a a všetky výrokovologické formuly A, B jazyka \mathcal{L} nasledovne:

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

kde vtt skrakuje *vtedy* a len vtedy a $v \not\vDash_p A$ 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	\vDash_p	\vDash_p	$\not\vDash_p$	$\not\vDash_p$	\vDash_p	$\not\vDash_p$

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

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

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:*

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

je výrokovologické ohodnotenie zhodné s \mathcal{M} .

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

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

(\Rightarrow) Nepriamo: Ak $\mathcal{M} \not\vDash A$, 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:

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

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 $v \models_{\mathcal{P}} X$ 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 $v \models_{\mathcal{P}} X$ 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 $v \vDash_p \neg X$ vtt (podľa def. \vDash_p) $v \not\vDash_p X$ vtt (podľa IP) $\mathcal{M} \not\vDash X$ vtt (podľa def. \vDash) $\mathcal{M} \vDash \neg X$. 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 $v \vDash_p (X \rightarrow Y)$ vtt $v \not\vDash_p X$ alebo $v \vDash_p Y$ vtt (podľa IP) vtt $\mathcal{M} \not\vDash X$ alebo $\mathcal{M} \vDash Y$ vtt $\mathcal{M} \vDash (X \rightarrow Y)$.

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

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

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

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

ale nie

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

Príklad výrokovologického modelu

Príklad 3.3 (Výrokovologický model teórie o party).

$$\begin{array}{l} v = \{\text{príde}(\text{Kim}) \mapsto t, \text{príde}(\text{Jim}) \mapsto t, \text{príde}(\text{Sarah}) \mapsto f\} \\ v \vDash_p ((\text{príde}(\text{Kim}) \vee \text{príde}(\text{Jim})) \vee \text{príde}(\text{Sarah})) \\ v \vDash_p (\text{príde}(\text{Kim}) \rightarrow \neg \text{príde}(\text{Sarah})) \\ v \vDash_p (\text{príde}(\text{Jim}) \rightarrow \text{príde}(\text{Kim})) \\ v \vDash_p (\text{príde}(\text{Sarah}) \rightarrow \text{príde}(\text{Jim})) \end{array} \left. \vphantom{\begin{array}{l} v = \{\text{príde}(\text{Kim}) \mapsto t, \text{príde}(\text{Jim}) \mapsto t, \text{príde}(\text{Sarah}) \mapsto f\} \\ v \vDash_p ((\text{príde}(\text{Kim}) \vee \text{príde}(\text{Jim})) \vee \text{príde}(\text{Sarah})) \\ v \vDash_p (\text{príde}(\text{Kim}) \rightarrow \neg \text{príde}(\text{Sarah})) \\ v \vDash_p (\text{príde}(\text{Jim}) \rightarrow \text{príde}(\text{Kim})) \\ v \vDash_p (\text{príde}(\text{Sarah}) \rightarrow \text{príde}(\text{Jim})) \end{array}} \right\} v \vDash_p T_{\text{party}}$$

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 , skrátene $v \models_p T$, 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 , skrátene $v \not\models_p T$, vtt T nie je pravdivá vo v .

Zrejme $v \not\models_p T$ 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_{party} je evidentne splniteľná.

3.2 Vyplyvanie, nezávislosť a nespľniteľnosť

Výrokovologické vyplyvanie

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 *vyplyva* z T a píšeme $T \models_p X$.

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

Príklad výrokovologickeho vyplývania

Príklad 3.8. Vyplýva príde(Kim) výrokovologiccky z T_{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_{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) výrokovologiccky vyplýva z T_{party} .

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

Príklad nezávislosti

Príklad 3.9. Vyplýva príde(Jim) výrokovologiccky z T_{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_{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_{party} .

Výrokovologická nezávislosť

Vzťahu medzi príde(Jim) a T_{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 $v_0 \vDash_p T$ aj $v_1 \vDash_p T$, ale $v_0 \not\vDash_p X$ a $v_1 \vDash_p X$.

Príklad 3.11 (pokračovanie príkladu 3.9). *Logický záver*: Formula príde(Jim) je *nezávislá* od T_{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 príde(Sarah) výrokovologickým dôsledkom T_{party} alebo nezávislá od T_{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_{party}	$p(S)$
	$p(K)$	$p(J)$	$p(S)$						
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	$\not\vDash_p$
v_7	<i>t</i>	<i>t</i>	<i>t</i>	\vDash_p	$\not\vDash_p$			$\not\vDash_p$	

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

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

Vzťahy teórií a formulí

Medzi ohodnotením a formulou sú iba dva vzájomne vylučné vzťahy:

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

Medzi teóriou a formulou je viac možných vzťahov:

	existuje v také, že $v \vDash_p T$ a $v \vDash_p X$	existuje v také, že $v \vDash_p T$ a $v \not\vDash_p X$	pre všetky v , ak $v \vDash_p T$, tak $v \vDash_p X$
existuje v také, že $v \vDash_p T$ a $v \vDash_p X$	X je nezávislá od T $T \not\vDash_p X$ a $T \not\vDash_p \neg X$	$T \vDash_p \neg X$ a $T \not\vDash_p X$	
pre všetky v , ak $v \vDash_p T$, tak $v \vDash_p X$	$T \vDash_p X$ a $T \not\vDash_p \neg X$	T je nesplniteľná $T \vDash_p X$ aj $T \vDash_p \neg X$	

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'_{party}
	$p(K)$	$p(J)$	$p(S)$						
u_0	f	f	f	$\not\vDash_p$					$\not\vDash_p$
u_1	f	f	t	\vDash_p	\vDash_p	\vDash_p	$\not\vDash_p$		$\not\vDash_p$
u_2	f	t	f	\vDash_p	\vDash_p	$\not\vDash_p$			$\not\vDash_p$
u_3	f	t	t	\vDash_p	\vDash_p	$\not\vDash_p$			$\not\vDash_p$
u_4	t	f	f	\vDash_p	\vDash_p	\vDash_p	\vDash_p	$\not\vDash_p$	$\not\vDash_p$
u_5	t	f	t	\vDash_p	$\not\vDash_p$				$\not\vDash_p$
u_6	t	t	f	\vDash_p	\vDash_p	\vDash_p	\vDash_p	$\not\vDash_p$	$\not\vDash_p$
u_7	t	t	t	\vDash_p	$\not\vDash_p$				$\not\vDash_p$

Logický záver: T'_{party} je nespílniteľná, vyplýva z nej každá formula.

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

Vyplývanie a nespílniteľnosť

Nespílniteľnosť 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 $T \cup \{\neg X\}$ je výrokovologicky nespílniteľná.

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

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

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 $\text{atoms}(X)$ $\Downarrow v_1 \vDash_p X$ vtt $v_2 \vDash_p X$

Báza: X je atóm.

- (3) X predikátový atóm podľa 1
- (4) $v_1 \vDash_p X$ vtt $v_1(X) = t$ def. pravdivosti
- (5) $v_2 \vDash_p X$ vtt $v_2(X) = t$ def. pravdivosti
- (6) $v_1(X) = v_2(X)$ podľa 2
- $v_1 \vDash_p X$ vtt $v_2 \vDash_p X$ 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 $\text{atoms}(Z)$ $\Downarrow v_1 \vDash_p Z \text{ vtt } v_2 \vDash_p Z$
-

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) $v_1 \vDash_p X \text{ vtt } v_2 \vDash_p X$ 3, IP pre $Z = X$
 - (5) $v_1 \vDash_p \neg X \text{ vtt } v_1 \not\vDash_p X$ def. \vDash_p
 - (6) $v_2 \vDash_p \neg X \text{ vtt } v_2 \not\vDash_p X$ def. \vDash_p
 - (7) $v_1 \not\vDash_p X \text{ vtt } v_2 \not\vDash_p X$ 4, def. $\not\vDash_p$
 - $v_1 \vDash_p \neg X \text{ vtt } v_2 \vDash_p \neg X$ 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 $\text{atoms}(Z)$ $\Downarrow v_1 \vDash_p Z \text{ vtt } v_2 \vDash_p Z$
-

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

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 $v_1 \vDash_p X$ vtt $v_2 \vDash_p X$. Preto $v_1 \vDash_p \neg X$ vtt (def. \vDash_p) $v_1 \not\vDash_p X$ vtt (IP) $v_2 \not\vDash_p X$ vtt (def. \vDash_p) $v_2 \vDash_p \neg X$.

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 $v_1 \vDash_p X$ vtt $v_2 \vDash_p X$; tiež sa zhodujú na $\text{atoms}(Y)$, a teda podľa IP $v_1 \vDash_p Y$ vtt $v_2 \vDash_p Y$. Preto $v_1 \vDash_p (X \wedge Y)$ vtt (def. \vDash_p) $v_1 \vDash_p X$ a $v_1 \vDash_p Y$ vtt (IP) $v_2 \vDash_p X$ a $v_2 \vDash_p Y$ vtt (def. \vDash_p) $v_2 \vDash_p (X \wedge Y)$.

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	\vDash_p	$\not\vdash_p$	\vDash_p
v_1	f	t	\vDash_p	$\not\vdash_p$	\vDash_p
v_2	t	f	$\not\vdash_p$	\vDash_p	\vDash_p
v_3	t	t	\vDash_p	\vDash_p	\vDash_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 $\vDash_p X$) vtt X je *pravdivá v každom* výrokovologickom ohodnotení v pre \mathcal{L} (teda *pre každé* výrokovologické ohodnotenie v pre \mathcal{L} platí $v \vDash_p X$).

	v_i				
	A_1	A_2	\dots	X	
v_0	f	f	\dots	\vDash_p	Definícia vyžaduje preveriť <i>všetky možné</i> ohodno-
v_1	f	f	\dots	\vDash_p	
		\dots			
v_k	t	f	\dots	\vDash_p	
		\dots			

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

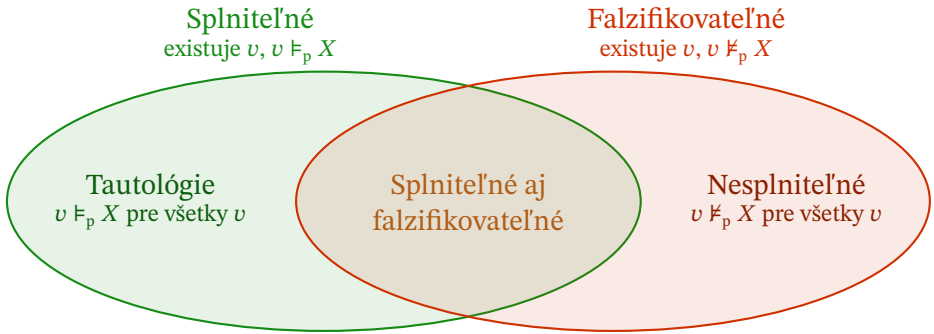
Tvrdenie 4.4 (Tautológie, vyplývanie a jeho monotónnosť). Nech \mathcal{L} je jazyk výrokovologickej časti logiky prvého rádu. Nech A je výrokovologická formula v \mathcal{L} . Nech T_1 a T_2 sú výrokovologické teórie v \mathcal{L} . Potom:

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

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: $A \wedge \neg A \wedge B_1 \wedge B_2 \wedge \dots \wedge B_{100}$.

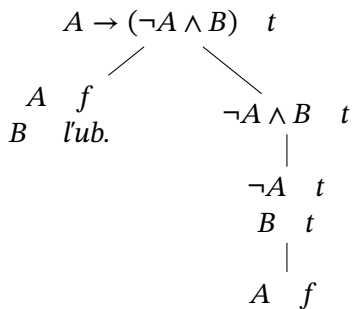
Ohodnotenia pre danú formulu

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.

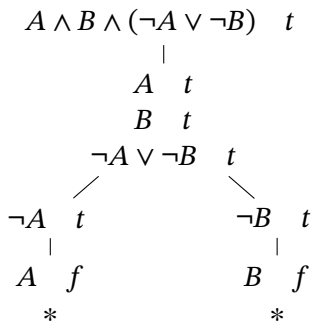
Hľadanie ohodnotení



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

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

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 výrokovologickej 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 $Y = (A \wedge B)$ výrokovologickej ekvivalentné?

Preskúmajme všetky výrokovologické ohodnotenia atómov A a B :

	v_i			$(A \rightarrow \neg B)$	$\neg(A \rightarrow \neg B)$	X	Y
	A	B	$\neg B$				
v_0	f	f	\vDash_p	\vDash_p	\vDash_p	\vDash_p	\vDash_p
v_1	f	t	\vDash_p	\vDash_p	\vDash_p	\vDash_p	\vDash_p
v_2	t	f	\vDash_p	\vDash_p	\vDash_p	\vDash_p	\vDash_p
v_3	t	t	\vDash_p	\vDash_p	\vDash_p	\vDash_p	\vDash_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*, skrátene $X \Leftrightarrow_p Y$ 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 skrátene 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 ozrejmime 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 výrokovologicke 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 $v \vDash_p A$ a $v \vDash_p B$, $v \vDash_p A$ a $v \not\vDash_p B$, $v \not\vDash_p A$ a $v \vDash_p B$, $v \not\vDash_p A$ a $v \not\vDash_p B$)
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:

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

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 $v \vDash_p (A \rightarrow B)$ vtt $v \vDash_p (\neg A \vee B)$. Preto môžeme konštatovať, že bez ohľadu na to, ktorý prípad nastáva, v ohodnotení v platí, že $v \vDash_p (A \rightarrow B)$ vtt $v \vDash_p (\neg A \vee B)$.

Pretože ohodnotenie v bolo *ľubovoľné*, môžeme toto konštatovanie *zovšeobecniť* 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	$\not\vdash_p$	$\not\vdash_p$	\vdash_p	\vdash_p
v	$\not\vdash_p$	\vdash_p	\vdash_p	\vdash_p
v	\vdash_p	$\not\vdash_p$	$\not\vdash_p$	$\not\vdash_p$
v	\vdash_p	\vdash_p	\vdash_p	\vdash_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 $v \not\vdash_p (A_1 \wedge A_2)$ alebo $v \vdash_p B$, čiže keď v každom ohodnotení v , v ktorom $v \vdash_p (A_1 \wedge A_2)$, máme aj $v \vdash_p B$ teda keď v každom ohodnotení v , v ktorom $v \vdash_p A_1$ a $v \vdash_p A_2$, máme aj $v \vdash_p B$, teda keď $z \{A_1, A_2\}$ výrokovologicky vyplýva B .

Vzťahy výrokovologického vyplývania a tautológií

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

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 $v \not\models_p A$ 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:

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

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 $v \vDash_p (X \rightarrow Y)$ a $v \vDash_p (Y \rightarrow X)$, vtt v každom ohodnotení v máme buď $v \vDash_p X$ alebo $v \vDash_p Y$ a zároveň buď $v \vDash_p Y$ alebo $v \vDash_p X$, vtt v každom ohodnotení v platí, že ak $v \vDash_p X$, tak $v \vDash_p Y$, a ak $v \vDash_p Y$, tak $v \vDash_p X$, vtt v každom ohodnotení v máme $v \vDash_p X$ vtt $v \vDash_p Y$, 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: $\vDash_p (X \leftrightarrow Y)$ 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 $v \vDash_p X$ vtt $v \vDash_p Z$.

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

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

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

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 $b \in \{\wedge, \vee, \rightarrow\}$:

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

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 skrátene 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 nakopirovali 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. $z \oplus b \oplus c$ 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

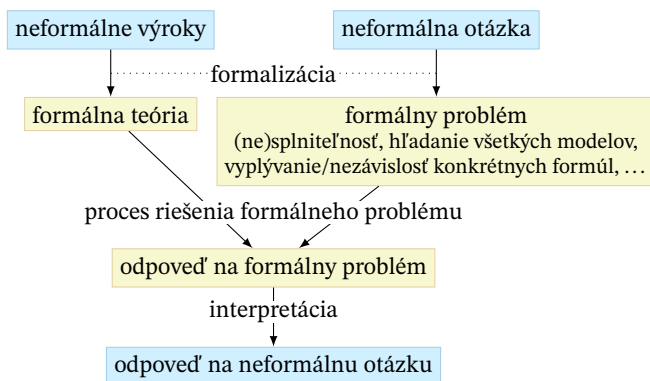
Minulý týždeň sme sa zaoberali:

- 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* 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 *spochybníme 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 zjednodušať.

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.

vyplýva 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) $v \vDash_p (p(A) \rightarrow (p(B) \wedge p(C)))$,

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

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

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

Podľa definície pravdivosti v ohodnotení, potom máme:

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

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

(7) $v \vDash_p p(F)$ z (6). Ďalej

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

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

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

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

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

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

spor s (13);

Tablový kalkúl

Z takýchto dôkazov sporom vychádza *tablový kalkúl* — 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 $v \vDash_p \dots$ a $v \not\vDash_p \dots$.

Definícia 5.4. Nech \mathcal{L} je jazyk výrokovologickej časti logiky prvého rádu. Nech X je výrokovologická formula jazyka \mathcal{L} . Postupnosti symbolov \mathbf{TX} a \mathbf{FX} 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 \mathbf{TX} (skrátene $v \vDash_p \mathbf{TX}$) vtt vo v je pravdivá X ;

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

Príklad 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} výrokovo-logickej časti logiky prvého rádu. Nech X a Y sú ľubovoľné formuly \mathcal{L} :*

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

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 pozerat' ako na *špeciálne prípady jedného pravidla*, ktorému sa hovorí α , *zjednodušenie* alebo *sploštenie* (angl. *flattening*), pre rôzne spojky.

Jednotný zápis označených formúl typu α

Definícia 5.7 (Jednotný zápis označených formúl typu α).

Označená formula A^+ je *typu* α vtt má jeden z tvarov v ľavom stĺpci tabuľky pre nejaké formuly X a Y . Takéto formuly budeme označovať písmenom α ; α_1 bude označovať príslušnú označenú formulu zo stredného stĺpca, α_2 príslušnú formulu z pravého stĺpca.

α	α_1	α_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 $v \vDash_p \alpha$ vtt $v \vDash_p \alpha_1$ a $v \vDash_p \alpha_2$.*

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

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 β , *vetvenie* alebo *rozdelenie* (angl. *split*), pre rôzne spojky.

Jednotný zápis označených formúl typu β

Definícia 5.10 (Jednotný zápis označených formúl typu β).

Označená formula B^+ je typu β vtt má jeden z tvarov v ľavom stĺpci tabuľky pre nejaké formuly X a Y . Takéto formuly budeme označovať písmenom β ; β_1 bude označovať príslušnú označenú formulu zo stredného stĺpca, β_2 príslušnú formulu z pravého stĺpca.

β	β_1	β_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 $v \vDash_p \beta$ vtt $v \vDash_p \beta_1$ alebo $v \vDash_p \beta_2$.*

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

$$S^+ = \{ \mathbf{T}(p(A) \rightarrow (p(B) \wedge p(C))), \\ \mathbf{T}((p(B) \vee p(D)) \rightarrow p(E)), \\ \mathbf{T}(p(F) \rightarrow \neg p(E)), \\ \mathbf{F}(p(A) \rightarrow \neg p(F)) \}$$

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^+ .

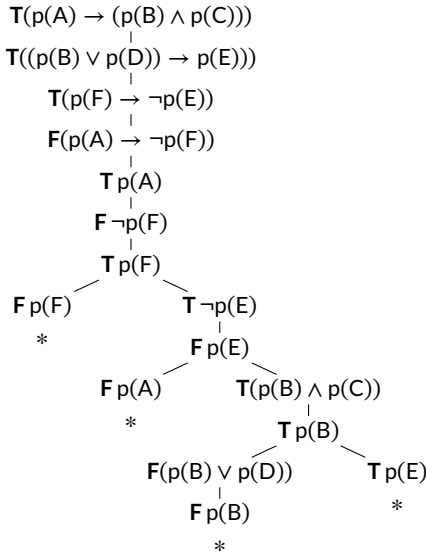
Pre množiny označených formúl budeme používať písmená S, T s horným indexom + a prípadne s dolnými indexmi, napr. S^+, T_3^+ .

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. $\mathbf{F} p(F) \beta 3$ *7, 8	9. $\mathbf{T} \neg p(E) \beta 3$ 10. $\mathbf{F} p(E) \alpha 9$
11. $\mathbf{F} p(A) \beta 1$ *5, 11	12. $\mathbf{T}(p(B) \wedge p(C)) \beta 1$ 13. $\mathbf{T} p(B) \alpha 12$
14. $\mathbf{F}(p(B) \vee p(D)) \beta 2$ 16. $\mathbf{F} p(B) \alpha 14$ *13, 16	15. $\mathbf{T} p(E) \beta 2$ *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 induktívnych 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:

α : Ak sa na vetve π_y (ceste z koreňa do y) vyskytuje nejaká označená formula α , tak ako jediné dieťa y pripojíme nový vrchol obsahujúci α_1 alebo α_2 .

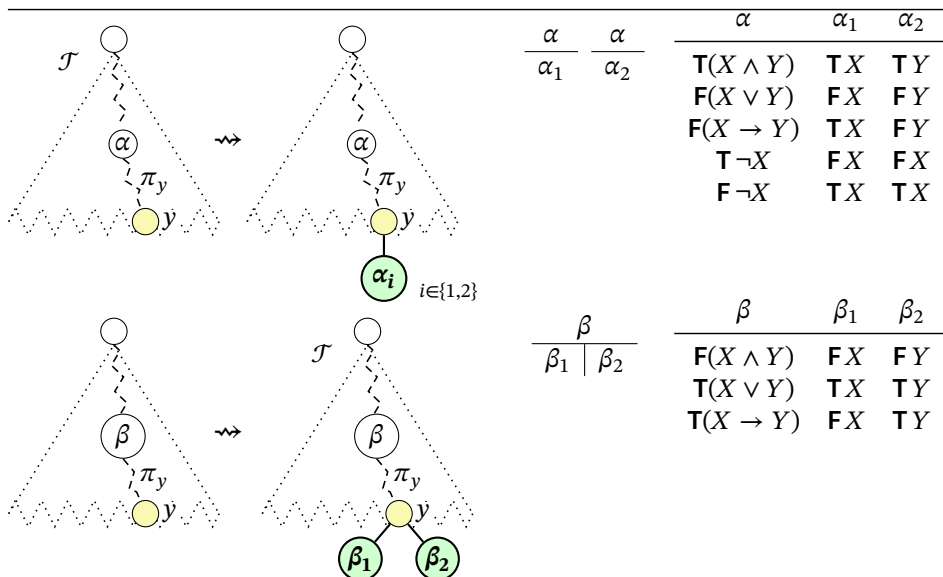
β : Ak sa na vetve π_y (ceste z koreňa do y) vyskytuje nejaká označená formula β , tak ako deti y pripojíme *dva* nové vrcholy, pričom ľavé dieťa bude obsahovať β_1 a pravé β_2 .

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

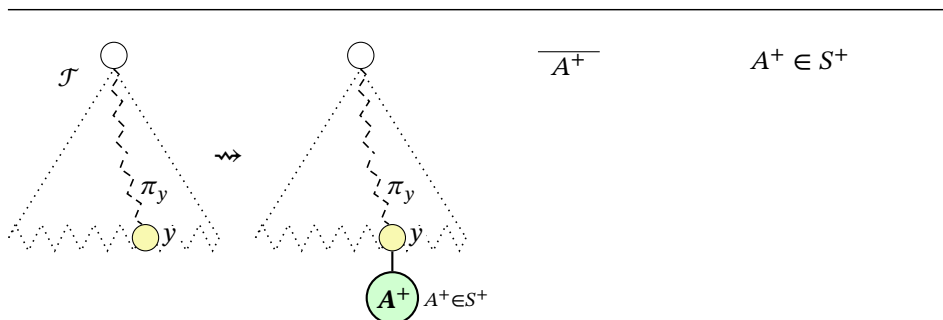
Pôvodné tablo Možné priame rozšírenie Pravidlá a označené formuly v nich



Legenda: y je list v table \mathcal{T} , π_y je cesta od koreňa k y

Tablá a tablové pravidlá (pokračovanie)

Pôvodné tablo Možné priame rozšírenie Pravidlá a označené formuly v nich



Legenda: y je list v table \mathcal{T} , π_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 π v \mathcal{T} vtt X^+ sa nachádza v niektorom vrchole na π . 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 π tabla \mathcal{T} je uzavretá vtt na π sa súčasne vyskytujú označené formuly $\mathbf{F}X$ a $\mathbf{T}X$ pre nejakú formulu X . Inak je π 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	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	12. $\mathbf{T}(p(B) \wedge p(C)) \quad \beta 1$ 13. $\mathbf{T} p(B) \quad \alpha 12$
14. $\mathbf{F}(p(B) \vee p(D)) \quad \beta 2$	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 výrokovologicnú *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 π je vetva tabla \mathcal{T} a nech v je výrokovologické ohodnotenie pre \mathcal{L} . Potom:

- *vetva π je pravdivá vo v ($v \models_p \pi$) vtt vo v sú pravdivé všetky označené formuly vyskytujúce sa na vetve π .*
- *tablo \mathcal{T} je pravdivé vo v ($v \models_p \mathcal{T}$) vtt niektorá vetva v table \mathcal{T} je pravdivá.*

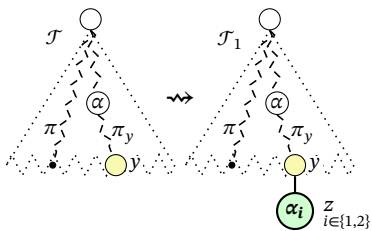
Korektnosť — pravdivosť priameho rozšírenia tabla

Pomocou predchádzajúcej definície sformulujeme lemu K1 takto:

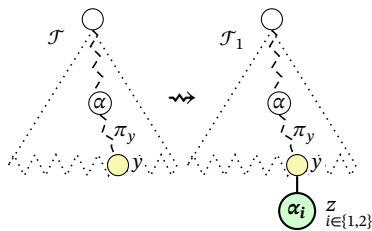
Lema 5.21 (K1). *Nech S^+ je množina označených formúl v jazyku \mathcal{L} , nech \mathcal{T} je tablo pre S^+ a nech v je výrokovologické ohodnotenie pre \mathcal{L} . Ak S^+ a \mathcal{T} sú pravdivé vo v , tak aj každé priame rozšírenie \mathcal{T} je pravdivé vo v .*

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 π . Nech \mathcal{T}_1 je priame rozšírenie \mathcal{T} . Nastáva jeden z prípadov:

- \mathcal{T}_1 vzniklo z \mathcal{T} pravidlom α , pridaním nového dieťaťa z nejakému listu y v \mathcal{T} , pričom y obsahuje α_1 alebo α_2 pre nejakú formulu α na vetve π_y .

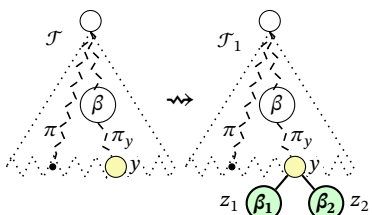


Ak $\pi \neq \pi_y$, tak \mathcal{T}_1 obsahuje π , a teda aj \mathcal{T}_1 je pravdivé vo v .

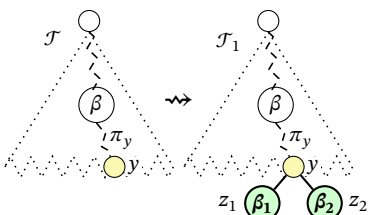


Ak $\pi = \pi_y$, tak α je pravdivá vo v , pretože α je na π . Potom aj α_1 a α_2 sú pravdivé vo v (pozorovanie 5.8). Vetva π_z v table \mathcal{T}_1 rozširuje vetvu π pravdivú vo v o vrchol z obsahujúci ozn. formulu α_1 alebo α_2 pravdivú vo v . Preto π_z je pravdivá vo v , a teda aj tablo \mathcal{T}_1 je pravdivé vo v .

- \mathcal{T}_1 vzniklo z \mathcal{T} pravidlom β , pridaním detí z_1 a z_2 nejakému listu y v \mathcal{T} , pričom z_1 obsahuje β_1 a z_2 obsahuje β_2 pre nejakú formulu β na vetve π_y .

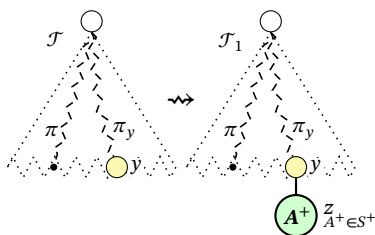


Ak $\pi \neq \pi_y$, tak \mathcal{T}_1 obsahuje π , a teda aj \mathcal{T}_1 je pravdivé vo v .

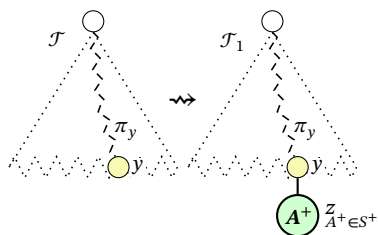


Ak $\pi = \pi_y$, tak $v \models_p \beta$, pretože β je na π . Potom $v \models_p \beta_1$ alebo $v \models_p \beta_2$ (poz. 5.11). Ak $v \models_p \beta_1$, tak $v \models_p \pi_{z_1}$, a teda $v \models_p \mathcal{T}_1$. Ak $v \models_p \beta_2$, tak $v \models_p \pi_{z_2}$, a teda $v \models_p \mathcal{T}_1$.

- \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 π , a teda aj \mathcal{T}_1 je pravdivé vo v .



Ak $\pi = \pi_y$, tak π_z v table \mathcal{T}_1 je pravdivá vo v , pretože je rozšírením vetvy π 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 π v \mathcal{T} . Pretože \mathcal{T} je uzavreté, aj vetva π je uzavretá. Na π sa teda nachádzajú označené formuly **TX**

a $\mathbf{F}X$ pre nejakú formulu X . Pretože π 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 π v table \mathcal{T} je úplná vtt má všetky nasledujúce vlastnosti:

- pre každú označenú formulu α , ktorá sa vyskytuje na π , sa *obidve* označené formuly α_1 a α_2 vyskytujú na π ;
- pre každú označenú formulu β , ktorá sa vyskytuje na π , sa *aspoň jedna* z označených formúl β_1, β_2 vyskytuje na π ;
- *každá* $X^+ \in S^+$ sa vyskytuje na π .

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 π ,
2. pre každý atóm A
 - ak sa na π nachádza $\mathbf{T}A$, definujeme $v(A) = t$;
 - ak sa na π nachádza $\mathbf{F}A$, definujeme $v(A) = f$;
 - inak definujeme $v(A)$ ľubovoľne.

V tomto v je pravdivá π , a preto je v ňom *pravdivá aj S^+* (všetky formuly z S^+ sa vyskytujú na π , lebo π 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 π_y je otvorená a nie je úplná. Potom nastane aspoň jedna z možností:

- Na π_y sa nachádza nejaká formula α , ale nenachádza sa *niektorá* z formúl α_1 a α_2 .
- Na π_y sa nachádza nejaká formula β , ale nenachádza sa *ani jedna* z formúl β_1 a β_2 .

Ak platí prvá alebo obe možnosti, aplikujeme pravidlo α . Ak platí iba druhá možnosť, aplikujeme pravidlo β . 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 π je úplná otvorená vetva nejakého tabla \mathcal{T} . Potom množina všetkých označených formúl na π 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 α alebo β :

Ak X^+ je α , 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 α .

Ak X^+ je β , potom aspoň jedna z β_1, β_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 β . \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 β .

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 β na implikáciu, kde sa jedna vetva ihneď uzavrie;
- opakovanie jedného podstromu dôkazu.

Riešenie príkladu 5.32

Tablo pre

$$S^+ = \{T(A \rightarrow C), T(B \rightarrow C), T(C \rightarrow X), T(C \rightarrow Y), T((X \wedge Y) \rightarrow Z), F((A \vee B) \rightarrow Z)\}$$

				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. $T(A \vee B)$	α_6						
				8. FZ	α_6						
				9. $F(X \wedge Y)$		β_5					
10. TA				β_7						28. TZ	β_5
										* 8, 28	
11. FA				β_1							
* 10, 11											
13. FC				β_3							
* 12, 13											
15. FC				β_4							
* 12, 15											
17. FX				β_9							
* 14, 17											
18. FY				β_9							
* 16, 18											
				20. FB		β_2					
				* 19, 20							
				22. FC		β_3					
				* 21, 22							
				24. FC		β_4					
				* 21, 24							
				26. FX		β_9					
				* 23, 26							
				27. FY		β_9					
				* 25, 27							

Odstránenie problémov — nové pravidlá

Keby tablový kalkul obsahoval napríklad veľmi prirodzené pravidlá *modus ponens*, *modus tolens* a *rez*:

$$\frac{T(X \rightarrow Y) \quad TX}{TY} \quad (\text{MP})$$

$$\frac{T(X \rightarrow Y) \quad FY}{FX} \quad (\text{MT})$$

$$\frac{}{TX \mid FX} \quad (\text{cut})$$

dôkaz v príklade by sa dal sprehľadniť a odstrániť by sa duplicita.

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. $F(X \wedge Y)$ $MT 5, 8$

10. TA $\beta 7$	16. TB $\beta 7$
11. TC $MP 1, 10$	17. TC $MP 2, 16$
12. TX $MP 3, 11$	18. TX $MP 3, 17$
13. TY $MP 4, 11$	19. TY $MP 4, 17$
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. $F(X \wedge Y)$ $MT 5, 8$

10. TC cut	15. FC cut	
11. TX $MP 3, 10$	16. TA $\beta 7$	18. TB $\beta 7$
12. TY $MP 4, 10$	17. TC $MP 1, 16$ * 15, 17	19. FB $MT 2, 15$ * 18, 19
13. FX $\beta 9$ * 11, 13	14. FY $\beta 9$ * 12, 14	

Ingrediencie korektnosti a úplnosti tabiel

Všimnite si:

Na dokázanie korektnosti tablového kalkulu stačilo, aby mali pravidlá vlastnosť:

$$\frac{\alpha}{\alpha_1} \quad \frac{\alpha}{\alpha_2}$$

$$\frac{\beta}{\beta_1 \mid \beta_2} \quad \frac{A^+}{A^+} \quad A^+ \in S^+$$

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^+), α , β , 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:

α : ...

:

MP: Ak sa na vetve π_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 príklade 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:

$$\frac{\begin{array}{l} \mathbf{T}(p(c) \rightarrow q(c))[p(c)|(sedan(a) \wedge biely(a)), q(c)|kupi(B, a)] \\ \mathbf{F} q(c)[p(c)|(sedan(a) \wedge biely(a)), q(c)|kupi(B, a)] \\ \hline \mathbf{F} p(c)[p(c)|(sedan(a) \wedge biely(a)), q(c)|kupi(B, a)] \end{array}}{\mathbf{T}((sedan(a) \wedge biely(a)) \rightarrow kupi(B, a))} = \frac{\mathbf{F} kupi(B, a)}{\mathbf{F}(sedan(a) \wedge biely(a))}$$

Tablové pravidlá vo všeobecnosti — pravidlo

Samotné pravidlo je množina všetkých inšancií vzoru:

$$MT = \left\{ \frac{\begin{array}{l} \mathbf{T}(p(c) \rightarrow q(c))[p(c)|X, q(c)|Y] \\ \mathbf{F} q(c)[p(c)|X, q(c)|Y] \\ \hline \mathbf{F} p(c)[p(c)|X, q(c)|Y] \end{array}}{\left| X, Y \in \mathcal{E}_{\mathcal{L}} \right.} \right\}$$

Samozrejme, *konkrétne* pravidlo vieme zapísať aj bez substitúcie:

$$MT = \left\{ \frac{\begin{array}{l} \mathbf{T}(X \rightarrow Y) \quad \mathbf{F} Y \\ \hline \mathbf{F} X \end{array}}{\left| X, Y \in \mathcal{E}_{\mathcal{L}} \right.} \right\}$$

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

$$\frac{P_1^+ \quad \dots \quad P_n^+}{C_1^+ \quad | \quad \dots \quad | \quad C_k^+}$$

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

$$\frac{P_1^+ \quad \dots \quad P_n^+}{C_1^+ \quad | \quad \dots \quad | \quad C_k^+}$$

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

$$R = \left\{ \frac{P_1^+_{[a_1|X_1, \dots, a_m|X_m]} \quad \dots \quad P_n^+_{[a_1|X_1, \dots, a_m|X_m]}}{C_1^+_{[a_1|X_1, \dots, a_m|X_m]} \quad | \quad \dots \quad | \quad C_k^+_{[a_1|X_1, \dots, a_m|X_m]}} \mid X_1, \dots, X_m \in \mathcal{E}_{\mathcal{L}} \right\},$$

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

$$\frac{P_1^+ \quad \dots \quad P_n^+}{C_1^+ \quad | \quad \dots \quad | \quad C_k^+}$$

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

$$\frac{P_1^+ \quad \dots \quad P_n^+}{C_1^+ \quad | \quad \dots \quad | \quad C_k^+}$$

na vetve π_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

$$\frac{}{\mathbf{TX} \quad | \quad \mathbf{FX}}$$

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 \mathbf{TX} alebo \mathbf{FX} .*

Dôkaz. Formula X je vo v buď pravdivá alebo nepravdivá. V prvom prípade $v \models_p \mathbf{TX}$. V druhom prípade $v \models_p \mathbf{FX}$. Teda v oboch prípadoch platí, že vo v je pravdivý niektorý zo záverov \mathbf{TX} alebo \mathbf{FX} pravidla rezu. \square

Príklad: zložitejšie pravidlá

Príklady zložitejších pravidiel:

- Viacnásobné pravidlá β :

$$\frac{\mathbf{T}(A_1 \vee A_2 \vee \dots \vee A_n)}{\mathbf{T}A_1 \quad | \quad \mathbf{T}A_2 \quad | \quad \dots \quad | \quad \mathbf{T}A_n} \qquad \frac{\mathbf{F}(A_1 \wedge A_2 \wedge \dots \wedge A_n)}{\mathbf{F}A_1 \quad | \quad \mathbf{F}A_2 \quad | \quad \dots \quad | \quad \mathbf{F}A_n}$$

- Pravidlo konštruktívnej dilemy:

$$\frac{\mathbf{T}(P \rightarrow Q) \quad \mathbf{T}(R \rightarrow S) \quad \mathbf{T}(P \vee R)}{\mathbf{T}Q \quad | \quad \mathbf{T}S}$$

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 rozbije formulu na atómy a ak by pre tautológiu X nebol v nejakej vetve tabla pre FX 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 Γ , potom platí aspoň jedna formula z Δ .

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 kalkulus: príklad dôkazu

Dokážeme sekvent $p, p \rightarrow q \vdash q$.

$$\frac{\frac{\text{---}}{p \vdash p} (Ax) \quad \frac{\text{---}}{q \vdash q} (Ax)}{p, p \rightarrow q \vdash q} (\rightarrow_L)$$

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 kalkulus: teória typov

- Sekventový kalkulus sa využíva v teórii typov. Množina Γ 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 *tablo* (Beth, Smullyan)
- 1965 *rezolvenca* (Robinson)
- 1970s Prolog (SLD-rezolvenca, orientovaná na cieľ)
- 1980s tablá pre modálnu logiku (musí/môže byť)
- 1990s tablá na vrchole popularity
- 2000s ústup tabiel, rezolvenca pre FOL s rovnosťou (**dokazovač** Vampire),
SAT solvery 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 ≤ 3 literály (NP-ťažké)
- 2-SAT — klauzuly majú ≤ 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 nespľniteľnosti
 - kratší dôkaz nespľniteľnosti 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,

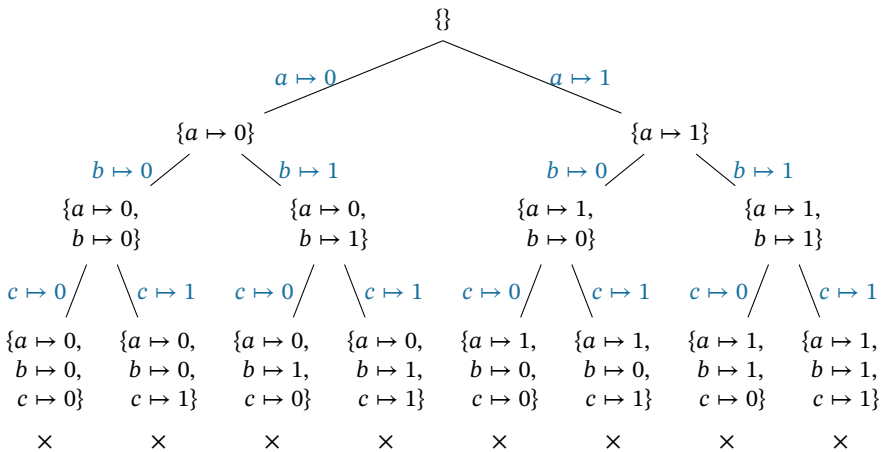
s – súčet veľkostí klauzúl

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 $v \notin_p S$

$f := 0, t := 1$



Priebežné vyhodnocovanie klauzúl

Strom ohodnotení:

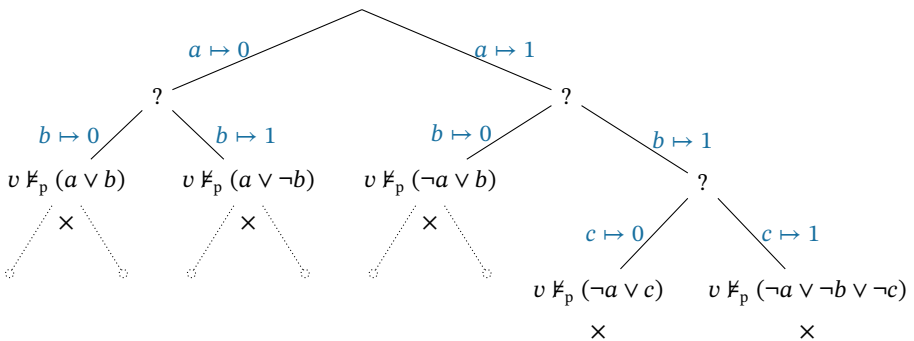
- List – ohodnotenie všetkých premenných
- Každý uzol – čiastočné ohodnotenie
- Ohodnotenie v uzle je *rozšírením* ohodnotenia v rodičovi
- Niektoré klauzuly sa dajú vyhodnotiť aj v čiastočnom ohodnotení
 - V čiastočnom ohodnotení $v = \{a \mapsto 0, b \mapsto 1\}$ sa dá určiť pravdivosť $(a \vee b)$, $(a \vee \neg b)$, $(\neg a \vee b)$ 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 $v \notin_p S$

? 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
 - $\{a \mapsto 1, \dots\} \vDash_p (a \vee b)$
 - $\{a \mapsto 1, \dots\} \vDash_p (a \vee \neg b)$
- je pravdivá klauzula $(\ell_1 \vee \dots \vee \neg a \vee \dots \vee \ell_n)$ obsahujúca $\neg a$ vtt je pravdivá zjednodušená klauzula $(\ell_1 \vee \dots \vee \ell_n)$
 - $\{a \mapsto 1, \dots\} \vDash_p (\neg a \vee \neg b \vee \neg c)$ vtt $\{a \mapsto 1, \dots\} \vDash_p (\neg b \vee \neg c)$

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

$$S|_{a \mapsto 1} = \{ \quad \quad \quad b, \quad (\neg b \vee \neg c), \quad c \}.$$

Analogicky môžeme S zjednodušiť podľa $a \mapsto 0$ na

$$S|_{a \mapsto 0} = \{ \quad b, \quad \neg b \quad \quad \quad \}.$$

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

$$S|_{P \mapsto f} = \{(\ell_1 \vee \dots \vee \ell_n) \mid (\ell_1 \vee \dots \vee P \vee \dots \vee \ell_n) \in S\} \\ \cup \{C \mid C \in S, \text{ v } C \text{ sa nevyskytuje } P \text{ ani } \neg P\}$$

$$S|_{P \mapsto t} = \{(\ell_1 \vee \dots \vee \ell_n) \mid (\ell_1 \vee \dots \vee \neg P \vee \dots \vee \ell_n) \in S\} \\ \cup \{C \mid C \in S, \text{ v } C \text{ sa nevyskytuje } P \text{ ani } \neg P\}$$

$$S|_{\neg P \mapsto t} = S|_{P \mapsto f}$$

$$S|_{\neg P \mapsto f} = S|_{P \mapsto t}$$

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 $v \vDash_p S$ vtt $v \vDash_p S|_{P \mapsto b}$.

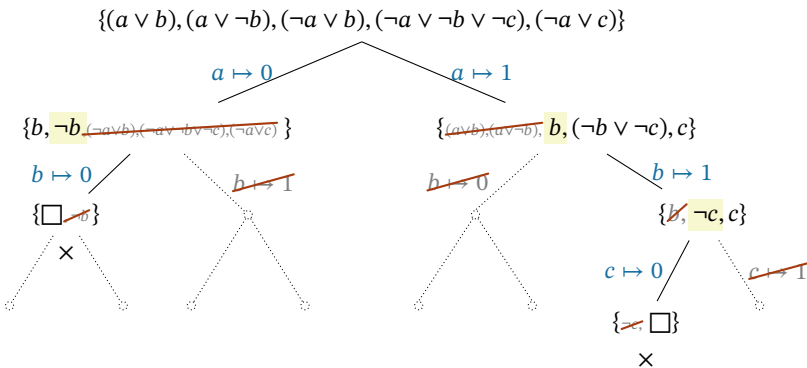
Propagácia jednotkových klauzúl

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 ℓ označujeme $\bar{\ell}$.

Definícia 6.6. Nech ℓ je literál a S je množina klauzúl. Literál ℓ je *nezmiešaný* (*pure*) v S vtt ℓ sa vyskytuje v niektorej klauzule z S , ale jeho komplement $\bar{\ell}$ sa nevyskytuje v žiadnej klauzule z S .

Tvrdenie 6.7. *Nech ℓ je literál a S je množina klauzúl. Ak ℓ 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(Φ, v):
- 2: **if** Φ 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.

$$(\neg a^{\circ} \vee \neg b^{\circ} \vee \neg c)$$

- Sledovaný literál musí byť *nenastavený* alebo *true*, ak sa to dá.

- Ak sa sledovaný literál stane *true*: nič nemusíme robiť.

$$\{a \mapsto 0\} \quad (\neg a^{\circ} \vee \neg b^{\circ} \vee \neg c)$$

- Ak sa sledovaný literál stane *false*: musíme nájsť iný.

$$\{a \mapsto 1\} \quad (\neg a^{\circ} \vee \neg b^{\circ} \vee \neg c^{\circ})$$

Ak iný nie je, práve sme vyrobili jednotkovú klauzulu (všetky literály okrem druhého sledovaného sú *false*),

$$\{a \mapsto 1, b \mapsto 1\} \quad (\neg a \vee \neg b^{\circ} \vee \neg c^{\circ})$$

alebo spor (aj druhý sledovaný je už *false*).

$$\{a \mapsto 1, b \mapsto 1, c \mapsto 0\} \quad (\neg a^{\circ} \vee c^{\circ})$$

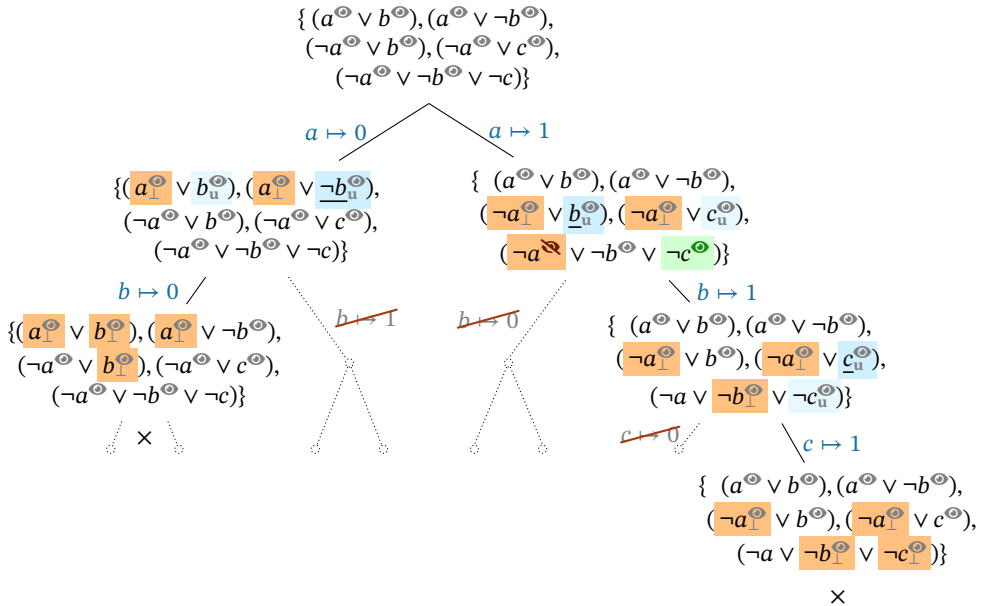
- Keď backtrackujeme: nič nemusíme robiť (možno sa niektoré sledované literály stanú *nenastavenými*).

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

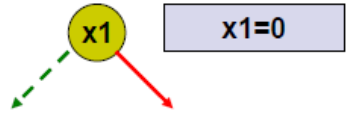


6.6 CDCL

CDCL – conflict-driven clause learning

$x1 + x4$
 $x1 + x3' + x8'$
 $x1 + x8 + x12$
 $x2 + x11$
 $x7' + x3' + x9$
 $x7' + x8 + x9'$
 $x7 + x8 + x10'$
 $x7 + x10 + x12'$

Step 1

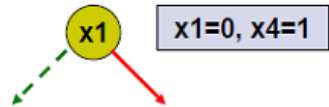


$x1=0$

(žltá – rozhodnutie, šedá – unit propagation)

$x1 + x4$
 $x1 + x3' + x8'$
 $x1 + x8 + x12$
 $x2 + x11$
 $x7' + x3' + x9$
 $x7' + x8 + x9'$
 $x7 + x8 + x10'$
 $x7 + x10 + x12'$

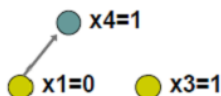
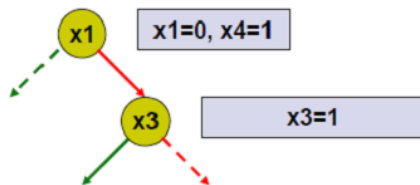
Step 2



(žltá – rozhodnutie, šedá – unit propagation)

Step 3

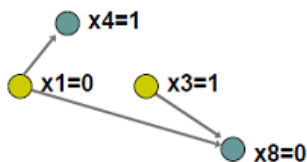
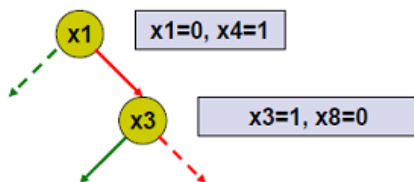
$x1 + x4$
 $x1 + x3' + x8'$
 $x1 + x8 + x12$
 $x2 + x11$
 $x7' + x3' + x9$
 $x7' + x8 + x9'$
 $x7 + x8 + x10'$
 $x7 + x10 + x12'$



(žltá – rozhodnutie, šedá – unit propagation)

Step 4

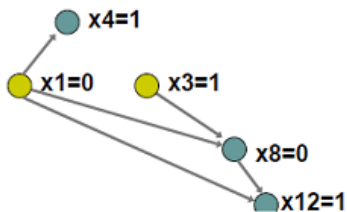
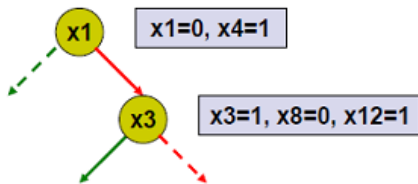
$x1 + x4$
 $x1 + x3' + x8'$
 $x1 + x8 + x12$
 $x2 + x11$
 $x7' + x3' + x9$
 $x7' + x8 + x9'$
 $x7 + x8 + x10'$
 $x7 + x10 + x12'$



(žltá – rozhodnutie, šedá – unit propagation)

$x1 + x4$
 $x1 + x3' + x8'$
 $x1 + x8 + x12$
 $x2 + x11$
 $x7' + x3' + x9$
 $x7' + x8 + x9'$
 $x7 + x8 + x10'$
 $x7 + x10 + x12'$

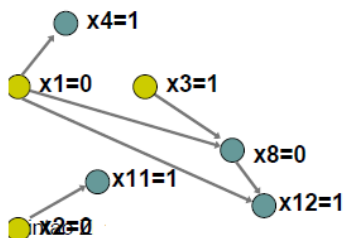
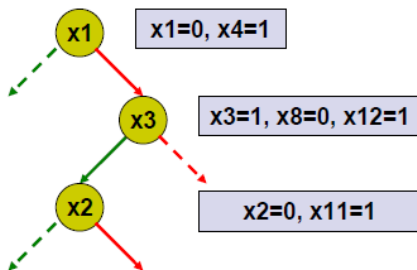
Step 5



(žltá – rozhodnutie, šedá – unit propagation)

$x1 + x4$
 $x1 + x3' + x8'$
 $x1 + x8 + x12$
 $x2 + x11$
 $x7' + x3' + x9$
 $x7' + x8 + x9'$
 $x7 + x8 + x10'$
 $x7 + x10 + x12'$

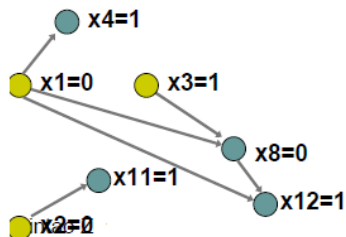
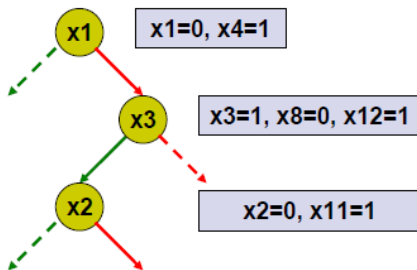
Step 7



(žltá – rozhodnutie, šedá – unit propagation)

$x1 + x4$
 $x1 + x3' + x8'$
 $x1 + x8 + x12$
 $x2 + x11$
 $x7' + x3' + x9$
 $x7' + x8 + x9'$
 $x7 + x8 + x10'$
 $x7 + x10 + x12'$

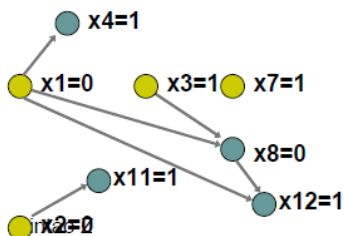
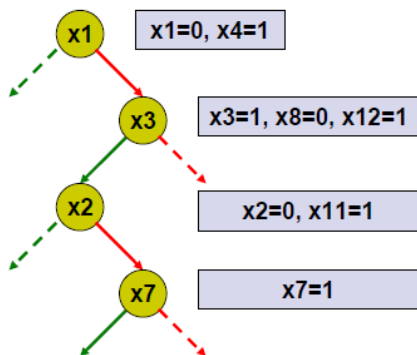
Step 8



(žltá – rozhodnutie, šedá – unit propagation)

$x1 + x4$
 $x1 + x3' + x8'$
 $x1 + x8 + x12$
 $x2 + x11$
 $x7' + x3' + x9$
 $x7' + x8 + x9'$
 $x7 + x8 + x10'$
 $x7 + x10 + x12'$

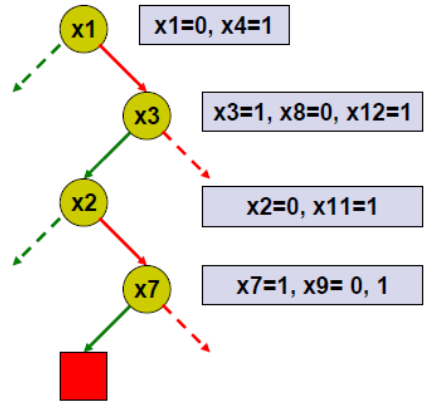
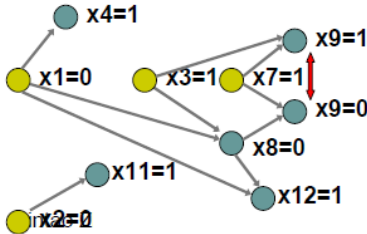
Step 9



(žltá – rozhodnutie, šedá – unit propagation)

- $x1 + x4$
- $x1 + x3' + x8'$
- $x1 + x8 + x12$
- $x2 + x11$
- $x7' + x3' + x9$
- $x7' + x8 + x9'$
- $x7 + x8 + x10'$
- $x7 + x10 + x12'$

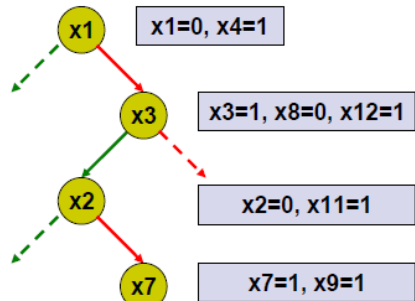
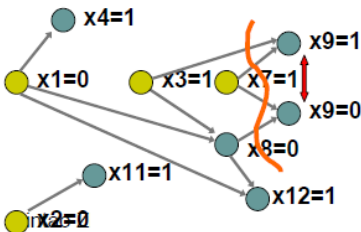
Step 10



(žltá – rozhodnutie, šedá – unit propagation)

- $x1 + x4$
- $x1 + x3' + x8'$
- $x1 + x8 + x12$
- $x2 + x11$
- $x7' + x3' + x9$
- $x7' + x8 + x9'$
- $x7 + x8 + x10'$
- $x7 + x10 + x12'$

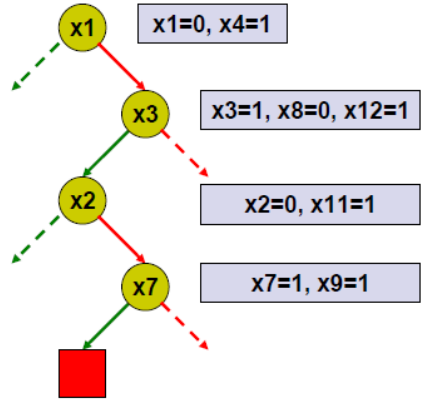
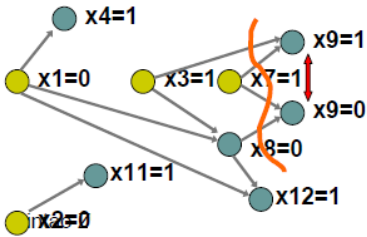
Step 11



$x3=1 \wedge x7=1 \wedge x8=0 \rightarrow \text{conflict}$

$x1 + x4$
 $x1 + x3' + x8'$
 $x1 + x8 + x12$
 $x2 + x11$
 $x7' + x3' + x9$
 $x7' + x8 + x9'$
 $x7 + x8 + x10'$
 $x7 + x10 + x12'$

Step 13

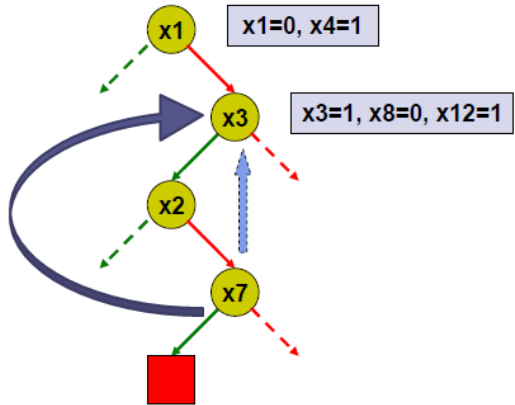
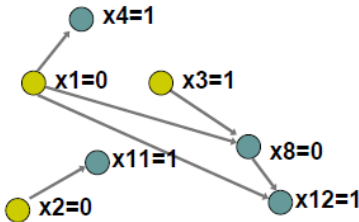


$x3=1 \wedge x7=1 \wedge x8=0 \rightarrow \text{conflict}$
 Add conflict clause: $x3' + x7' + x8$

Uvedený rez nie je jediný, mohli by sme pridať $x_1 \vee \neg x_3 \vee \neg x_7$.

$x1 + x4$
 $x1 + x3' + x8'$
 $x1 + x8 + x12$
 $x2 + x11$
 $x7' + x3' + x9$
 $x7' + x8 + x9'$
 $x7 + x8 + x10'$
 $x3' + x8 + x7'$

Step 14

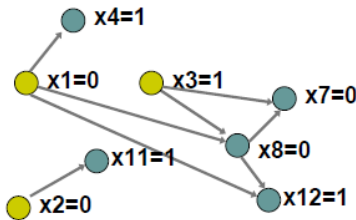
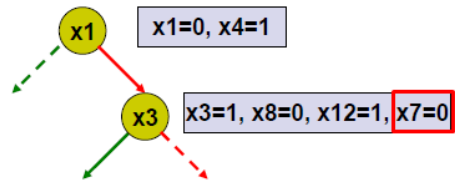


Backtrack to the decision level of $x3=1$:
 $x7 = 0$

Návrat do bodu, kde pridaná klauzula vynúti ohodnotenie jednej doteraz neohodnotenej premennej (čo zabráni vzniku tohto konfliktu kdekoľvek v podstrome).

$x_1 + x_4$
 $x_1 + x_3' + x_8'$
 $x_1 + x_8 + x_{12}$
 $x_2 + x_{11}$
 $x_7' + x_3' + x_9$
 $x_7' + x_8 + x_9'$
 $x_7 + x_8 + x_{10}'$
 $x_7 + x_{10} + x_{12}'$
 $x_3' + x_8 + x_7'$

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 oddeľujúci rozhodnutia od konfliktu a odvodíme novú klauzulu, ktorá konfliktu predchádza (*learning*); ak je takých rezov viac, heuristikou niektorý vyberieme
- vrátíme 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 klauzúlach (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*)

$$\bigvee_i (a_i \wedge b_i \wedge c_i)$$

$$\left(\bigvee_i z_i \right) \wedge \bigwedge_i [(\bar{z}_i \vee a_i) \wedge (\bar{z}_i \vee b_i) \wedge (\bar{z}_i \vee c_i)]$$

(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

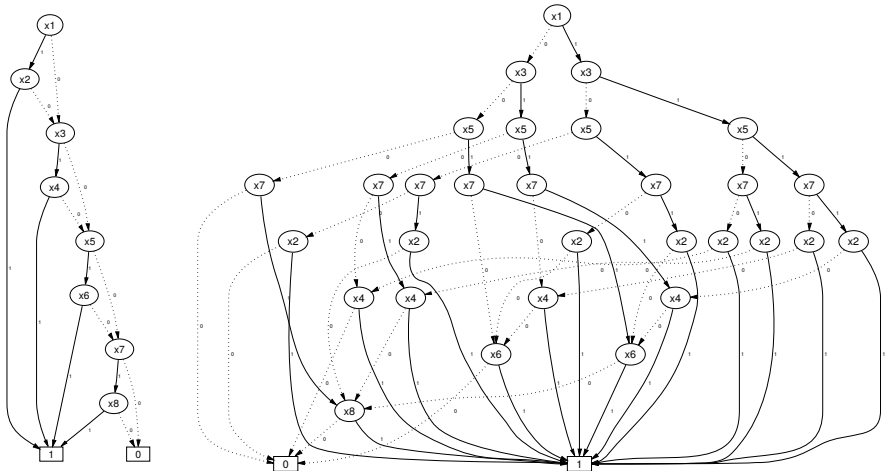
- v princípe úplný matematický dôkaz správnosti
- nedá sa použiť pre fyzickú vrstvu, ale ideálna pre logickú
- používa sa zriedka — drahá a vyžaduje vysokú odbornosť
- atómové elektrárne, vesmírne lety, veľké série procesorov

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-farbitel'né)
- SAT solvery fungujú aj pre tisíce vrcholov
- absencia krátkych kružníc v grafe (čiže lokálne vyzerá ako strom) predlžuje 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 vrcholov 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 znovu 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 nespľniteľnosť 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 neprípustnosť, 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 nesúlاد 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 patrií(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. Je logický dôsledok.	Ňufko je sivý. Nie je log. dôsledkom, ale je prijateľné.	Ňufko je sivý. 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 škrečkov; 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 *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/lubovol'ný 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 trocha 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.)

Relačná logika prvého rádu

Logike prvého rádu, ktorá zahŕňa kvantifikátory, ale nie funkčné symboly, hovoríme *relačná* — predikáty popisujú vzťahy, čiže relácie.

Kvantifikátory prinášajú skokový nárast algoritmickej zložitosti: problém splniteľnosti formuly v relačnej logike je vo všeobecnosti nerozhodnuteľný.

Ak by sme povolili len predikáty arity najviac 1 (*monadická* logika), bola by splniteľnosť ešte rozhodnuteľná.

7.3 Syntax relačnej logiky prvého rádu

Symboly 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 $(\text{'}$ avá, 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 $\mathcal{V}_{\mathcal{L}} = \{x, y, x_1, y_1, x_2, y_2, \dots\}$, $\mathcal{C}_{\mathcal{L}} = \{\text{Jurko, Vierka, Ňufko}\}$ $\mathcal{P}_{\mathcal{L}} = \{\text{myš}^1, \text{škrečok}^1, \text{biely}^1, \text{patrí}^2\}$.

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.

$$\begin{aligned} & \neg P(x, y) \wedge K(y, x) \\ & \neg P(x, y) \wedge \exists y K(y, x) \\ & \exists y (\neg P(x, y) \wedge K(y, x)) \\ \forall x \exists y (\neg P(x, y) \wedge K(y, x)) \\ \forall x (\neg P(x, y) \wedge \exists y K(y, x)) \end{aligned}$$

Voľné a viazané premenné

Definícia 7.14 (Voľné 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.

$$\begin{aligned} \text{free}(\neg P(x, y) \wedge K(y, z)) &= \{x, y, z\} \\ \text{free}(\neg P(x, y) \wedge \exists y K(y, z)) &= \{x, y, z\} \\ \text{free}(\exists y (\neg P(x, y) \wedge K(y, z))) &= \{x, z\} \\ \text{free}(\exists y (\neg P(x, y) \wedge \forall z K(y, z))) &= \{x\} \\ \text{free}(\exists y \exists z (\forall x \neg P(x, y) \wedge K(y, z))) &= \{\} \end{aligned}$$

Voľné 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:

$$\text{free}(x) = \{x\}$$

$$\text{free}(a) = \{\}$$

$$\text{free}(t_1 \doteq t_2) = \text{free}(t_1) \cup \text{free}(t_2)$$

$$\text{free}(P(t_1, \dots, t_n)) = \text{free}(t_1) \cup \dots \cup \text{free}(t_n)$$

$$\text{free}(\neg A) = \text{free}(A)$$

$$\begin{aligned} \text{free}(A \wedge B) &= \text{free}(A \vee B) = \text{free}(A \rightarrow B) = \\ &= \text{free}(A) \cup \text{free}(B) \end{aligned}$$

$$\text{free}(\forall x A) = \text{free}(\exists x A) = \text{free}(A) \setminus \{x\}$$

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 *interpretáčnou funkciou* 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

$$\begin{aligned} \mathcal{V}_{\mathcal{L}} &= \{x_1, x_2, y, \dots\} \\ D &= \{\text{Alica}, \text{Bonifác}, \text{Cyril}\}. \end{aligned}$$

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

$$D = \{i_{\text{Viera}}, i_{\text{Juraj}}, i_{\text{Eva}}, \text{b}, \text{w}, \text{z}, \text{r}, \text{u}\}$$

$$i(\text{Vierka}) = i_{\text{Viera}},$$

$$i(\text{biele}) = \{\text{z}, \text{r}, \text{u}\},$$

$$i(\text{patrí}) = \{(r, i_{\text{Viera}}),$$

$$i(\text{Jurko}) = i_{\text{Juraj}},$$

$$\text{b},$$

$$(z, i_{\text{Eva}})\}$$

$$i(\text{Ňufko}) = \text{z}$$

$$\text{u}\}$$

pri ohodnotení premenných, napr. $e = \{x \mapsto \text{z}, y \mapsto i_{\text{Eva}}, \dots\}$:

1. vyhodnotíme termy, ktoré sú argumentmi predikátu:

$$x^{\mathcal{M}}[e] = e(x) = \text{z}$$

$$\text{Vierka}^{\mathcal{M}}[e] = i(\text{Vierka}) = i_{\text{Viera}},$$

2. zistíme, či $(\text{z}, i_{\text{Viera}}) \in i(\text{patrí})$: *nie*

Štruktúra \mathcal{M} **nesplní**a 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:

$e(y/d)$	$\mathcal{M} \models \text{patrí}(y, \text{Vierka}) [e(y/d)]$
$\{x \mapsto \text{človek}, y \mapsto \text{človek}_{\text{Viera}}, \dots\}$?
\vdots	\vdots
$\{x \mapsto \text{človek}, y \mapsto \text{človek}_{\text{Jurko}}, \dots\}$?
$\{x \mapsto \text{človek}, y \mapsto \text{človek}_{\text{Nufko}}, \dots\}$?
$\{x \mapsto \text{človek}, y \mapsto \text{človek}_{\text{Eva}}, \dots\}$	⊥

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 $d = \text{človek}_{\text{Nufko}}$ – *svedok*.

Takže $\mathcal{M} \models \exists y \text{patrí}(y, \text{Vierka}) [e]$.

$$D = \{\text{človek}_{\text{Viera}}, \text{človek}_{\text{Juraj}}, \text{človek}_{\text{Eva}}, \\ \text{človek}_{\text{Nufko}}, \text{človek}_{\text{Jurko}}, \text{človek}_{\text{Eva}}\}$$

$$i(\text{Vierka}) = \text{človek}_{\text{Viera}},$$

$$i(\text{Jurko}) = \text{človek}_{\text{Juraj}},$$

$$i(\text{Nufko}) = \text{človek}_{\text{Nufko}},$$

$$i(\text{biele}) = \{\text{človek}_{\text{Nufko}}, \\ \text{človek}_{\text{Jurko}}, \\ \text{človek}_{\text{Eva}}\}$$

$$i(\text{patrí}) = \{(\text{človek}_{\text{Nufko}}, \text{človek}_{\text{Viera}}), \\ (\text{človek}_{\text{Nufko}}, \text{človek}_{\text{Eva}})\}$$

$$e = \{x \mapsto \text{človek}_{\text{Nufko}}, \\ y \mapsto \text{človek}_{\text{Eva}}, \dots\}$$

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
$\{x \mapsto \text{Juraj}, y \mapsto \text{Eva}, \dots\}$?
$\{x \mapsto \text{Nufko}, y \mapsto \text{Eva}, \dots\}$?
$\{x \mapsto \text{Eva}, y \mapsto \text{Eva}, \dots\}$?
$\{x \mapsto \text{Eva}, y \mapsto \text{Eva}, \dots\}$	\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 $d = \text{Juraj}$ a $d = \text{Eva}$ – *kontrapriklady*.

Takže $\mathcal{M} \not\models \forall x (\text{biele}(x) \rightarrow \text{patrí}(x, y)) [e]$.

$D = \{\text{Viera}, \text{Juraj}, \text{Eva},$

$\text{Nufko}, \text{Eva}\}$

$i(\text{Vierka}) = \text{Viera},$

$i(\text{Jurko}) = \text{Juraj},$

$i(\text{Nufko}) = \text{Nufko}$

$i(\text{biele}) = \{\text{Nufko}, \text{Eva}\}$

$i(\text{patrí}) = \{(\text{Juraj}, \text{Viera}), (\text{Eva}, \text{Eva})\}$

$e = \{x \mapsto \text{Eva}, y \mapsto \text{Eva}, \dots\}$

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 $\mathcal{M} \vDash \text{biele}(x)[e(x/d)]$ pre všetky $d \in D$, tak by aj formula $\forall x(\text{biele}(x) \rightarrow \text{patr}(x, y))$ aj tvrdenie „všetko biele patrí y“ boli *triviálne* splnené.

Nezávislosť od ohodnotenia viazanej premennej

Pri vyhodnocovaní splnenia kvantifikovanej formuly štruktúrou pri danom ohodnotení e

$$\mathcal{M} \vDash \exists y \text{ patr}(y, \text{Vierka}) [e]$$

$$\mathcal{M} \vDash \forall x(\text{biele}(x) \rightarrow \text{patr}(x, y)) [e]$$

nezáleží na tom, akú hodnotu priraduje 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

$$\mathcal{M} \vDash \exists x \forall y \text{ patr}(x, y) [e]$$

najprv zvolíme konkrétneho kandidáta na svedka d pre x a pri následnom vyhodnocovaní

$$\mathcal{M} \vDash \forall y \text{ patr}(x, y) [e(x/d)]$$

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 $\mathcal{M} \vDash A[e]$) 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 $e = \{x \mapsto 3, y \mapsto 5, \dots\}$.

Zistíme, 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 — úskalia 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! (Pre-svedčíme 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 Zamľčané a zdanlivo opačné kvantifikátory

Zamľčaný 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.
 $\forall x(\text{prešiel}(x, \text{Logiku}) \rightarrow \text{odovzdával_DÚ}(x))$

Zamľčaný 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. $\exists x(\text{kúpil}(\text{Bonifác}, x) \wedge \text{syr}(x))$

Zamľčaná 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

Ludia neraz veci kvantifikujú nesprávne alebo neúplne (zvlášť pozor na ľudí pôsobiacich v inom odvetví, kde nepoznáte konvencie).

Ako chápať 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
- ...

Formalizácia do FOL nás núti vybrať si jedno konkrétne pochopenie, formuly FOL nepripúšťajú žiadnu nejednoznačnosť.

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* 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 podrad'ovacích 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 prejdenia 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 implikatúry

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*.

Dobrý 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 myší 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 implikáciu (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 implikácia.

Dotatok „Ale žiadny prvák si ju nikdy nezapísal“ (negácia implikácie), nie je s tvrdením v spore, ale objasňuje, že je triviálne pravdivé.

„Niektoré“ neimplikuje „nie všetky“

Ďalšia implikácia 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 implikácia.

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 implikácie.

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

- Zapíšem 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 porovnam, 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 nespľniteľnosť,
- „vždy pravdivé“ formuly (vo výrokovom prípade sa volali tautológie),
- vyplývanie/logický dôsledok.

Splniteľnosť a nespľniteľnosť

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 $\{\forall x(\text{človek}(x) \vee \text{mys}(x)), \forall x(\text{človek}(x) \rightarrow \neg \text{mys}(x))\}$ 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 $i(\text{mys}) = \{2\}$.

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

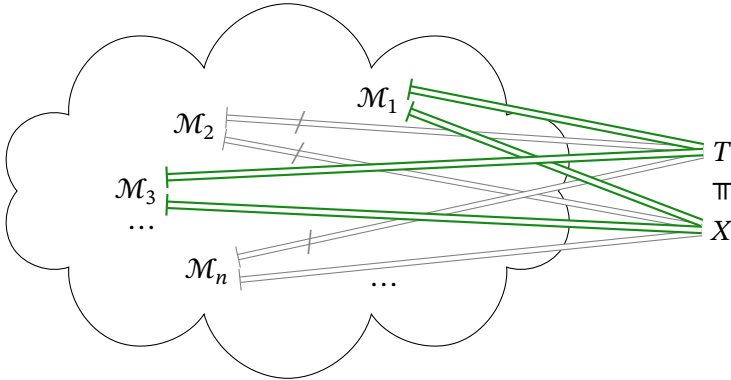
Príklad 8.4. Formula $X = (\forall x \text{ doma}(x) \rightarrow \text{doma}(\text{Jurko}))$ je platná.

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. Napr. pre vzťah „väčší“ v prirodzenom jazyku zvyčajne potichu predpokladáme tranzitívnosť.

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 $\left(\bigwedge_{i=1}^n A_i \rightarrow X\right)$ je platná – t.j. $\models \left(\bigwedge_{i=1}^n A_i \rightarrow X\right)$;
- 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 výrokovkej 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 , skrátene: $\mathcal{M} \models \mathbf{T}X[e]$ 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 , skrátene: $\mathcal{M} \models \mathbf{F}X[e]$ vtt $\mathcal{M} \not\models X[e]$.

\mathcal{M} splňa množinu označených formúl S^+ pri ohodnotení e vtt \mathcal{M} splňa každú označenú formulu A^+ z S^+ pri ohodnotení e , skrátene: $\mathcal{M} \vDash S^+[e]$ vtt pre každú $A^+ \in S^+$ máme $\mathcal{M} \vDash A^+[e]$.

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 $\mathcal{M} \vDash X^+[e]$.

Množina ozn. formúl S^+ je *splniteľná* vtt pre nejakú štruktúru \mathcal{M} a nejaké ohodnotenie individuových premenných e máme $\mathcal{M} \vDash S^+[e]$.

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) $\neg \text{opraví}(\text{TServis}$

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 nesplnená. 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. $\text{T} \forall x(\text{opraví}(\text{TServis}, x) \rightarrow \text{tlačiareň}(x))$	S^+
2. $\text{T} \neg \text{tlačiareň}(\text{môjMobil})$	S^+
3. $\text{F} \neg \text{opraví}(\text{TServis}, \text{môjMobil})$	S^+
4. $\text{T} \text{opraví}(\text{TServis}, \text{môjMobil})$	$\alpha 3$
5. $\text{F} \text{tlačiareň}(\text{môjMobil})$	$\alpha 2$
6. $\text{T} (\text{opraví}(\text{TServis}, \text{môjMobil}) \rightarrow \text{tlačiareň}(\text{môjMobil}))$?1
7. $\text{T} \text{tlačiareň}(\text{môjMobil})$	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:

$$\frac{\top \forall x A}{\top A\{x \mapsto t\}} \gamma$$

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 volných 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. $\mathbf{T} \neg \text{tlačiareň}(\text{môjMobil})$	$\text{MP}5, 3$
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. $\mathbf{F} \text{opraví}(\text{TServis}, \text{môjMobil})$	$\text{MT}8, 7$
10. $\mathbf{F}(\text{mobil}(\text{môjMobil}) \wedge \neg \text{opraví}(\text{TServis}, \text{môjMobil}))$	$\gamma 4\{x \mapsto \text{môjMobil}\}$
11. $\mathbf{F} \text{mobil}(\text{môjMobil})$ $\beta 10$ * 3, 11	12. $\mathbf{F} \neg \text{opraví}(\text{TServis}, \text{môjMobil})$ $\beta 10$ 13. $\mathbf{T} \text{opraví}(\text{TServis}, \text{môjMobil})$ $\alpha 12$ * 9, 13

Dôkaz s pozitívnou existenčnou kvantifikáciou

Príklad 8.15. Dokážme neformálne, že z teórie $T = \{ (1) \forall x(\text{opraví}(\text{TServis}, x) \rightarrow \text{tlačiareň}(x)), (2) \exists x \neg \text{tlačiareň}(x) \}$ 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 opraví(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

$$\top \exists x \neg \text{tlačiareň}(x),$$

dočasným menom — voľnou premennou z a odvodenie:

$$\top \neg \text{tlačiareň}(z).$$

! Táto premenná sa *predtým na vetve nesmie vyskytovať voľná*. **!**

Musí to byť *nová, vlastná* premenná pre formulu $\top \exists x \neg \text{tlačiareň}(x)$.

Vo všeobecnosti:

$$\frac{\top \exists x A}{\top A\{x \mapsto y\}} \delta$$

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. $\top \text{tlačiareň}(x)$

n+2. $\top \exists x \neg \text{tlačiareň}(x)$

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. T tlačiareň(x)

n+2. T $\exists x \neg$ tlačiareň(x)

n+3. T \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. T tlačiareň(x)

n+2. T $\exists x \neg$ tlačiareň(x)

n+3. T \neg tlačiareň(x) ✗ „ $\delta 2\{x \mapsto x\}$

by bola *nesplniteľná*.

Negatívna všeobecná kvantifikácia a jej vlastná premenná

Negatívna všeobecne kvantifikovaná formula

$F \forall x$ 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* formuly $F \forall x$ tlačiareň(x), napríklad u , a môžeme odvodiť:

F tlačiareň(u).

⚠ Táto premenná sa *predtým na vetve nesmie vyskytovať voľná*. ⚠

Vo všeobecnosti:

$$\frac{F \forall x A}{F A\{x \mapsto y\}} \delta$$

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. $\text{T } \exists x \forall y (\text{opraví}(x, y) \rightarrow \text{tlačiareň}(y))$ S^+
 2. $\text{T } \forall x (\text{mobil}(x) \rightarrow \neg \text{tlačiareň}(x))$ S^+
 3. $\text{F } \forall x (\text{mobil}(x) \rightarrow \exists y \neg \text{opraví}(y, x))$ S^+
 4. $\text{F } (\text{mobil}(u) \rightarrow \exists y \neg \text{opraví}(y, u))$ $\delta 3\{x \mapsto u\}$
 5. $\text{T } \text{mobil}(u)$ $\alpha 4$
 6. $\text{F } \exists y \neg \text{opraví}(y, u)$ $\alpha 4$
 7. $\text{T } \forall y (\text{opraví}(z, y) \rightarrow \text{tlačiareň}(y))$ $\delta 1\{x \mapsto z\}$
 8. $\text{T } (\text{mobil}(u) \rightarrow \neg \text{tlačiareň}(u))$ $\gamma 2\{x \mapsto u\}$
 9. $\text{T } \neg \text{tlačiareň}(u)$ $\text{MP} 8, 5$
 10. $\text{F } \text{tlačiareň}(u)$ $\alpha 9$
 11. $\text{T } (\text{opraví}(z, u) \rightarrow \text{tlačiareň}(u))$ $\gamma 7\{y \mapsto u\}$
 12. $\text{F } \text{opraví}(z, u)$ $\text{MT} 11, 10$
 13. $\text{F } \neg \text{opraví}(z, u)$ $\gamma 6\{y \mapsto z\}$
 14. $\text{T } \text{opraví}(z, u)$ $\alpha 13$
- * 12, 14

Tablové pravidlá pre logiku prvého rádu

Definícia 8.16. Pravidlami tablového kalkulu pre logiku prvého rádu sú pravidlá typu α a β pre výrokovú logiku a pravidlá:

$$\gamma \quad \frac{\text{T } \forall x A}{\text{T } A\{x \mapsto t\}} \quad \frac{\text{F } \exists x A}{\text{F } A\{x \mapsto t\}} \quad \text{jednotne: } \frac{\gamma(x)}{\gamma_1(t)}$$

$$\delta \quad \frac{\text{F } \forall x A}{\text{F } A\{x \mapsto y\}} \quad \frac{\text{T } \exists x A}{\text{T } A\{x \mapsto y\}} \quad \text{jednotne: } \frac{\delta(x)}{\delta_1(y)}$$

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 π o dôsledok niektorého z pravidiel typu δ navyše musí platiť, že **premenná y nemá voľný výskyt v žiadnej formule na vetve π .**

Substituovateľnosť vysvetlíme nižšie.

Korektnosť pravidiel γ a δ

Tvrdenie 8.17 (Korektnosť pravidiel γ a δ). *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 kalkulu 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 nespĺnená a ukážeme spor.
- Podobne pre prvorádové vyplývanie $T \vDash X$ predpokladáme, že v nejakej štruktúre a nejakom ohodnotení sú splnené všetky formuly z T (TA pre $A \in T$), ale X je nespĺnená (FX) a ukážeme spor, teda hľadáme uzavreté tablo pre $S^+ = \{TA \mid A \in T\} \cup \{FX\}$.

Častá chyba pri pravidlách γ a δ : 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. F mobil(u)
2. T tlačiareň(u)
3. T ($\forall x$ tlačiareň(x) \rightarrow $\forall y$ mobil(y))
4. T (tlačiareň(u) \rightarrow $\forall y$ mobil(y)) ❌ „ $\gamma 3$ “
5. T $\forall y$ mobil(y) MP4, 2
6. T 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. T $\forall x \neg$ pozná(x, x)
- n+2. T \neg pozná(y, y) $\gamma 1\{x \mapsto y\}$
- n+3. T $\forall x \exists y$ 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. T $\forall x \neg$ pozná(x, x)
- n+2. T \neg pozná(y, y) $\gamma 1\{x \mapsto y\}$
- n+3. T $\forall x \exists y$ pozná(x, y)
- n+4. T $\exists y$ 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 $A = \exists \underline{y} \text{pozná}(\underline{x}, y)$.

- Substitúcia $\{\underline{x} \mapsto \underline{y}, z \mapsto \text{Jurko}\}$ *nie je aplikovateľná* na A , lebo term y *nie je substituovateľný* za premennú x v A .
- Substitúcia $\{\underline{x} \mapsto z, \underline{y} \mapsto \text{Jurko}, z \mapsto \underline{y}\}$ 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 σ 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 $A = \exists \underline{y} \text{pozná}(\underline{x}, \underline{y})$ a $\sigma = \{\underline{x} \mapsto z, \underline{y} \mapsto u, z \mapsto \underline{y}\}$.

Substitúcia σ 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.

$$A\sigma = \exists \underline{y} \text{pozná}(z, \underline{y})$$

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:

$$\begin{aligned}x_i\sigma &= t_i & y\sigma &= y & a\sigma &= a \\(s_1 \doteq s_2)\sigma &= (s_1\sigma \doteq s_2\sigma) & (P(s_1, \dots, s_k))\sigma &= P(s_1\sigma, \dots, s_k\sigma) \\(\neg A)\sigma &= \neg(A\sigma) & ((A \diamond B))\sigma &= (A\sigma \diamond B\sigma) \\(\forall y A)\sigma &= \forall y (A\sigma) & (\exists y A)\sigma &= \exists y (A\sigma) \\(\forall x_i A)\sigma &= \forall x_i (A\sigma_i) & (\exists x_i A)\sigma &= \exists x_i (A\sigma_i),\end{aligned}$$

kde $\sigma_i = \sigma \setminus \{x_i \mapsto t_i\}$, za predpokladu, že σ 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:

- $\exists x(\text{človek}(x) \wedge \exists y(\text{škrekok}(y) \wedge \text{krmí}(x, y)))$ Nejaký človek (má vlastnosť, že) krmí nejakého škrečka.
- $\forall x(\text{človek}(x) \rightarrow \forall y(\text{škrekok}(y) \rightarrow \text{krmí}(x, y)))$ Každý človek krmí každého škrečka.

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:

- $\forall x \forall y \text{ má_rád}(x, y)$ je ekvivalentné $\forall y \forall x \text{ má_rád}(x, y)$;
- $\exists x \exists y \text{ má_rád}(x, y)$ je ekvivalentné $\exists y \exists x \text{ má_rád}(x, y)$.

Pri rôznych kvantifikátoroch zmena poradia vážne mení význam:

- $\forall x \exists y \text{ má_rád}(x, y)$ — *Každý/-á má rád/rada niekoho.*
- $\exists x \forall y \text{ má_rád}(x, y)$ — *Niektor má rád/rada všetkých.*

Poradie kvantifikovaných premenných

Záleží aj na tom, ako sa kvantifikované premenné použijú vo formule, ktorá je kvantifikovaná.

Porovnajme:

- $\forall x \exists y \text{ má_rád}(x, y)$ – Každý/-á má rád/rada niekoho.
- $\forall x \exists y \text{ má_rád}(y, x)$ – Každú/-ého má niekto rád.

a

- $\exists x \forall y \text{ má_rád}(x, y)$ – Niekoľko má rada/rád všetkých.
- $\exists x \forall y \text{ má_rád}(y, x)$ – Niekoho 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:

$$\exists x(\text{škrečok}(x) \wedge \forall y(\text{škrečok}(y) \rightarrow x \doteq y))$$

Neformálne: *Nejaký škrečok je jediným škrečkom.*

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 škrečka kŕmi nejaké dieťa.*

1. Rozpoznáme, že tvrdenie má tvar *Všetky P sú Q*, pričom P je atomická vlastnosť. Môžeme ho teda čiastočne sformalizovať na:

$$\forall x(\text{škrečok}(x) \rightarrow \text{niejaké dieťa kŕmi } x)$$

2. Sformalizujeme *nejaké dieťa kŕmi x*: Má formu: *Nejaké P je Q*:

$$\exists y(\text{dieťa}(y) \wedge \text{kŕmi}(y, x))$$

3. Dosadíme:

$$\forall x(\text{škrekok}(x) \rightarrow \exists y(\text{dieťa}(y) \wedge \text{kŕmi}(y, x)))$$

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 *neopatrnnej* 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 spojками „*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:

$$\times (\exists x(\text{prvák}(x) \wedge \text{navštevuje}(x, \text{LPI})) \rightarrow \text{bystrý}(x)).$$

$$\checkmark \forall x((\text{prvák}(x) \wedge \text{navštevuje}(x, \text{LPI})) \rightarrow \text{bystrý}(x)).$$

Vyskytujú sa aj v zložitejších kombináciách. Úderným príkladom je:

Každý sedliak, ktorý vlastní nejakého osla, ho bije.

Na existenčné tvrdenie *vlastní nejakého osla* v antecedente odkazuje zámeno *ho* v konzekvente.

Odkaz z konzekventu — nesprávne možnosti

Postupnou formalizáciou by sme mohli dostať nesprávnu formulu:

$$\times \forall x((\text{sedliak}(x) \wedge \exists y(\text{osol}(y) \wedge \text{vlastní}(x, y))) \rightarrow \text{bije}(x, y))$$

Keby sme sa ju pokúsili „zachrániť“ tým, že zaviažeme premennú y , mohlo by to dopadnúť rôzne, ale stále neprávne:

$$\times \forall x(\text{sedliak}(x) \wedge \exists y(\text{osol}(y) \wedge \text{vlastní}(x, y) \wedge \text{bije}(x, y)))$$

— *Všetko je sedliak, ktorý vlastní osla, ktorého bije.*

$$\times \forall x(\text{sedliak}(x) \rightarrow \exists y(\text{osol}(y) \wedge \text{vlastní}(x, y) \wedge \text{bije}(x, y)))$$

— *Každý sedliak určite vlastní osla, ktorého 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, ktorého 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 $P = \text{ovocie}$ a $Q = \text{také, že ho mám rád}$, teda formálne:

$$\forall x((\text{ovocie}(x) \wedge \neg \text{jablko}(x)) \rightarrow \text{mám}_\text{rád}(x))$$

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.

9. prednáška

Funkčné symboly. Tablá s rovnosťou

10 Logika prvého rádu

10.1 Funkčné symboly

Vzťahy s jednoznačne určenými objektmi

V niektorých vzťahoch ich predmet vždy *existuje* a je *jednoznačne* určený/-á:

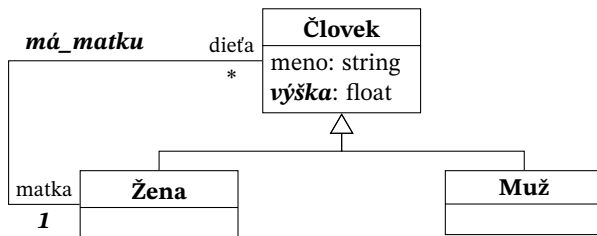
- Každý človek má *práve jednu* biologickú matku.
- Každý človek má (v danej chvíli) *práve jednu* výšku.
- Každé dve čísla majú *práve jeden* súčet (súčin, najväčší spoločný deliteľ, ...).
- Každá neprázdna konečná množina čísel má *práve jeden* maximálny prvok.

Takýto predmet potom jednoznačne pomenávajú menné frázy ako:

- Bonifácova mama; mama Bonifácovej mamy;
- Klárkina výška; výška Jurkovej mamy;
- súčet čísel 2 a 3; súčet čísla 4 a súčinu čísel 2 a 5;
- maximálny prvok množiny {2, 7, 19}.

Vzťahy s jednoznačne určenými cieľmi v UML

UML má na modelovanie takýchto vzťahov dve možnosti:



Vzťah k jednoznačne určenému objektu reprezentuje v UML vzťah s kardinalitou N:1 (*má_matku*).

Vzťah k jednoznačne určenej hodnote reprezentuje v UML atribút (*výška*).

Vzťahy s jednoznačne určenými objektmi — predikát a axiómy

Takéto vzťahy môžeme popísať predikátom a formulou pre existenciu a jednoznačnosť:

- Vzťah medzi dieťaťom a matkou môžeme vyjadriť napríklad predikátom *má_matku* s vlastnosťami existencie a jednoznačnosti:

$$\forall x \exists y \text{ má_matku}(x, y)$$

$$\forall x \forall y_1 \forall y_2 ((\text{má_matku}(x, y_1) \wedge \text{má_matku}(x, y_2)) \rightarrow y_1 \doteq y_2)$$

alebo stručnejšie:

$$\forall x \exists y (\text{má_matku}(x, y) \wedge \forall y_1 (\text{má_matku}(x, y_1) \rightarrow y_1 \doteq y))$$

- Podobne súčet dvoch čísel (ak všetko v doméne sú čísla):

$$\forall x \forall y \exists z (\text{súčet}(x, y, z) \wedge \forall z_1 (\text{súčet}(x, y, z_1) \rightarrow z_1 \doteq z))$$

Vzťahy s jednoznačne určenými objektmi — použitie

Použitie v zložitejších formulách nie je veľmi pohodlné:

- *Bonifácova mama je vedkyňa.*

$$\forall x (\text{má_matku}(\text{Bonifác}, x) \rightarrow \text{vedec}(x))$$

alebo

$$\exists x (\text{má_matku}(\text{Bonifác}, x) \wedge \text{vedec}(x))$$

Výrok hovorí o konkrétnych objektoch (Bonifác a jeho mama), ale vo formule musíme použiť kvantifikátory.

- *Mama Klárkinej mamy má Bobyho.*

$$\forall y \forall z ((\text{má_matku}(\text{Klárka}, y) \wedge \text{má_matku}(y, z)) \rightarrow \text{má}(z, \text{Boby}))$$

- *Ak x delí y a z, delí aj ich súčet.*

$$\forall x \forall y \forall z \forall u ((\text{delí}(x, y) \wedge \text{delí}(x, z) \wedge \text{súčet}(y, z, u)) \rightarrow \text{delí}(x, u))$$

Funkcie a funkčné symboly

Binárne relácie, ktoré sú všade definované a jednoznačné sa nazývajú... zobrazenia alebo *funkcie*.

Keď f je funkcia a $(x, y) \in f$, y sa nazýva *hodnota funkcie f pre x a namiesto y píšeme $f(x)$* .

Reláciám zodpovedajú v logike prvého rádu predikátové symboly. Dalo by sa zdefinovať, ako sa predikáty môžu používať ako funkcie, ale bolo by to komplikované.

Namiesto toho jazyky logiky prvého rádu môžu obsahovať mimologické symboly určené špeciálne na označovanie funkcií — *funkčné symboly*.

Termy s funkčnými symbolmi

Vo formulách sa ani predikátové ani funkčné symboly nedajú použiť samé o sebe — potrebujú argumenty.

Funkčný symbol v jazyku má pevne daný počet argumentov — *aritu* (rovnať ako predikátové symboly).

Postupnosť symbolov

$$\text{funkčný_symbol}(term_1, \dots, term_n)$$

označuje *objekt* — hodnotu funkcie, ktorú označuje *funkčný symbol*, pre n -ticu objektov, ktoré označujú $term_1, \dots, term_n$. Je to teda *term*, nie *formula*.

Funkčné symboly sa teda líšia od predikátových, pretože *predikátový symbol*($term_1, \dots, term_n$) je formula a jej významom je pravdivostná hodnota, nie objekt.

Funkčný symbol namiesto predikátového v atónoch

Napríklad predikát má_matku^2 môžeme nahradiť funkčným symbolom matka^1 .

Term $\text{matka}(\text{Klárka})$ potom označuje objekt — Klárkinu mamu.

Výrok *Klárkina mama je Magda* namiesto predikátového atómu $\text{má_matku}(\text{Klárka})$, vyjadríme rovnostným atómom $\text{matka}(\text{Klárka}) \doteq \text{Magda}$.

Výrok *Bonifácova mama je vedkyňa* namiesto $\forall x(\text{má_matku}(\text{Bonifác}, x) \rightarrow \text{vedec}(x))$ vyjadríme atómom $\text{vedec}(\text{matka}(\text{Bonifác}))$. Ekvivalentne, ale zbytočne nepohodlne: $\forall x(\text{matka}(\text{Bonifác}) \doteq x \rightarrow \text{vedec}(x)) \exists x(\text{matka}(\text{Bonifác}) \doteq x \wedge \text{vedec}(x))$

Podobne, keď súčet³ nahradíme funkčným symbolom +²:

$$\begin{aligned} \text{súčet}(2, 3, 5) &\rightsquigarrow +(2, 3) \doteq 5 \\ \forall x(\text{súčet}(7, 3, x) \rightarrow \text{delí}(5, x)) &\rightsquigarrow \text{delí}(5, +(7, 3)) \end{aligned}$$

Použitie termov s funkčnými symbolmi

Term s funkčným symbolom môžeme použiť všade, kde sme používali doterajšie termy (individuové konštanty a premenné):

- ako argument predikátu alebo rovnosti vo formule:
 - $\forall x \text{ rodič}(\text{matka}(x), x)$ – *Každého mama je jeho rodičom*;
 - $\forall x \neg \text{matka}(x) \doteq x$ – *Nikto nie je sám sebe mamou*;
- ako argument funkčného symbolu:
 - $\text{matka}(\text{matka}(\text{Bonifác}))$ – term označujúci *mamu Bonifárovej mamy* (Bonifácovu starú mamu z maminej strany);
 - $\text{má}(\text{matka}(\text{matka}(\text{Klárka})), \text{Boby})$ – atóm formalizujúci výrok *Klárkina stará mama z maminej strany má Bobyho*;
 - $\exists x \neg >(\text{výška}(\text{matka}(x)), \text{výška}(x))$ – *Niečia mama nie je vyššia ako on/ona*;
 - $\forall x \forall y \forall z((\text{delí}(x, y) \wedge \text{delí}(x, z)) \rightarrow \text{delí}(x, +(y, z)))$ – *Deliteľ sčítancov delí aj ich súčet*.

Infixová notácia

Dohoda 10.1. Atómy s binárnymi predikátovými symbolmi a termy s binárnymi funkčnými symbolmi, ktoré sa skladajú z neabecedných znakov, môžeme skrátene zapisovať *infixovo*. Teda

- Pre každý neabecedný binárny *predikátový symbol* \diamond^2 môžeme *atóm* $\diamond(t_1, t_2)$ skrátiť na $t_1 \diamond t_2$ (bez zátvoriek).
- Pre každý neabecedný *funkčný symbol* \circ^2 môžeme *term* $\circ(t_1, t_2)$ skrátiť na $(t_1 \circ t_2)$ (so zátvorkami).

Posledné dva príklady sa sprehladnia:

- $\exists x \neg \text{výška}(\text{matka}(x)) > \text{výška}(x)$ — *Niečia mama nie je vyššia ako on/ona.*
- $\forall x \forall y \forall z ((x \mid y \wedge x \mid z) \rightarrow x \mid (y + z))$ — *Deliteľ sčítancov delí aj ich súčet.*

Zamýšľaný definičný obor a obor hodnôt funkčných symbolov

Niektoré termy s funkčnými symbolmi môžeme vytvoriť:

výška(výška(Jurko)), matka(výška(Klárka)),

ale nemusia dávať intuitívny zmysel.

Zamýšľaný definičný obor a obor hodnôt funkcie označenej funkčným symbolom môžeme vyjadriť formulami:

$\forall x (\text{človek}(x) \rightarrow (\text{človek}(\text{matka}(x)) \wedge \text{žena}(\text{matka}(x))))$

$\forall x (\text{človek}(x) \rightarrow \text{dĺžka}(\text{výška}(x)))$

Nič to ale nezmení na tom, že funkcia je *definovaná na celej doméne*.

Funkčné symboly — zhrnutie

	Funkčný symbol	Predikátový symbol
aplikácia na argumenty	$\text{matka}(t)$	$\text{rodič}(t_1, t_2)$
syntaktický typ aplikácie	term	atóm
význam aplikácie	objekt (<i>matka t</i>)	pravdivostná hodnota (výroku <i>t₁ je rodičom t₂</i>)
podmienky použitia	<i>matka(t)</i> existuje a je jednoznačne určená pre každé <i>t</i>	<i>t₂</i> nemusí existovať ani byť jednoznačne určená pre každé <i>t₁</i>
režazenie aplikácií	$\text{matka}(\text{matka}(t))$	$\text{rodič}(t_1, \text{rodič}(t_2, t_3))$ $(\text{rodič}(t_1, t_2) \wedge \text{rodič}(t_2, t_3))$

10.2 Syntax logiky prvého rádu

Definícia syntaxe logiky prvého rádu

Keď do definícií doterajšej *relačnej* logiky prvého rádu zahrnieme funkčné symboly, dostaneme konečne (úplnú) *logiku prvého rádu*.

Musíme:

- pridať funkčné symboly medzi symboly jazyka,
- rozšíriť termy o aplikácie funkčných symbolov a vnáranie.

Atomické formuly a formuly zdefinujeme *zdanlivo* rovnako ako doteraz, ale *využitím nových termov*.

Symboly jazyka logiky prvého rádu

Definícia 10.2. *Symbolmi jazyka logiky prvého rádu \mathcal{L} sú:*

individuové premenné z nejakej nekonečnej spočítateľnej množiny $\mathcal{V}_{\mathcal{L}}$;

mimologické symboly:

individuové konštanty z nejakej spočítateľnej množiny $\mathcal{C}_{\mathcal{L}}$,

funkčné symboly z nejakej spočítateľnej množiny $\mathcal{F}_{\mathcal{L}}$,

predikátové symboly z nejakej spočít. množiny $\mathcal{P}_{\mathcal{L}}$;

logické symboly: logické spojky — unárna \neg a binárne \wedge, \vee a \rightarrow , symbol rovnosti \doteq a kvantifikátory — existenčný \exists a všeobecný \forall ;

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

Množiny $\mathcal{V}_{\mathcal{L}}, \mathcal{C}_{\mathcal{L}}, \mathcal{F}_{\mathcal{L}}, \mathcal{P}_{\mathcal{L}}$ sú vzájomne disjunktné. Logické ani pomocné symboly sa nevyskytujú v symboloch z $\mathcal{V}_{\mathcal{L}}, \mathcal{C}_{\mathcal{L}}, \mathcal{F}_{\mathcal{L}}, \mathcal{P}_{\mathcal{L}}$.

Každému symbolu $s \in \mathcal{P}_{\mathcal{L}} \cup \mathcal{F}_{\mathcal{L}}$ je priradená *arita* $\text{ar}(s) \in \mathbb{N}^+$.

Označovanie symbolov jazyka logiky prvého rádu

Dohoda 10.3. Keď budeme hovoriť o ľubovoľných symboloch jazyka \mathcal{L} , budeme ich zvyčajne označovať nasledovnými meta premennými podľa potreby s dolnými indexmi: individuové premenné budeme označovať malými písmenami z konca abecedy (x, y, z); individuové konštanty malými písmenami zo začiatku abecedy (a, b, c, d, e); funkčné symboly písmenami f, g, h ; predikátové symboly písmenami P, Q, R .

Aritu budeme niekedy písať ako horný index symbolov, konkrétnych aj označených meta premennými: $\text{pes}^1, <^2, P^5, \text{matka}^1, f^2$.

Termy jazyka logiky prvého rádu

Keďže argumentmi funkčných symbolov sú termy, ktoré môžu tiež obsahovať funkčné symboly, musíme termy zdefinovať *induktívne*.

Definícia 10.4. Množina $\mathcal{T}_{\mathcal{L}}$ termov jazyka logiky prvého rádu \mathcal{L} je *najmenšia* množina postupností symbolov jazyka \mathcal{L} , pre ktorú platí:

- i. každá individuová premenná $x \in \mathcal{V}_{\mathcal{L}}$ patrí do $\mathcal{T}_{\mathcal{L}}$ (teda $\mathcal{V}_{\mathcal{L}} \subseteq \mathcal{T}_{\mathcal{L}}$);
- ii. každá individuová konštantá $c \in \mathcal{C}_{\mathcal{L}}$ patrí do $\mathcal{T}_{\mathcal{L}}$ (teda $\mathcal{C}_{\mathcal{L}} \subseteq \mathcal{T}_{\mathcal{L}}$);
- iii. ak f je funkčný symbol s aritou n a t_1, \dots, t_n patria do $\mathcal{T}_{\mathcal{L}}$, tak aj postupnosť symbolov $f(t_1, \dots, t_n)$ patrí do $\mathcal{T}_{\mathcal{L}}$.

Každý prvok $\mathcal{T}_{\mathcal{L}}$ je *term* jazyka \mathcal{L} a nič iné nie je termom jazyka \mathcal{L} .

Dohoda 10.5. Termy označujeme písmenami t, s, r s prípadnými dolnými indexmi.

Príklad 10.6. Nech $\mathcal{C}_{\mathcal{L}} = \{\text{Jurko}, \text{Iveta}\}$, $\mathcal{V}_{\mathcal{L}} = \{u, v, x, y, z, u_1, v_1, x_1, y_1, \dots\}$, $\mathcal{F}_{\mathcal{L}} = \{\text{matka}^1, \text{výška}^1\}$.

Podľa i. a ii. bodu definície sú termami: Jurko, Iveta, u, v, x, \dots

Podľa iii. bodu definície sú termami:

matka(Jurko), matka(Iveta), matka(u), matka(v), ...

výška(Jurko), výška(Iveta), výška(u), výška(v), ...

matka(matka(Jurko)), matka(výška(Jurko)),

výška(matka(Jurko)), výška(výška(Jurko)), ... ,

matka(matka(matka(x))), ...

Atomické formuly jazyka logiky prvého rádu

Definícia 10.7 (Atomické formuly). Nech \mathcal{L} je jazyk 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 $\mathcal{A}_{\mathcal{L}}$.

Znenie tejto definície sa takmer nezmenilo, ale zmenili sa pojmy *term* a *jazyk*, ktoré sa v nej používajú. Definuje preto iné postupnosti symbolov ako doteraz.

Formuly jazyka logiky prvého rádu

Definícia 10.8. Množina $\mathcal{E}_{\mathcal{L}}$ všetkých *formúl* jazyka logiky prvého rádu \mathcal{L} je *najmenšia* množina postupností symbolov jazyka \mathcal{L} , ktorá spĺňa všetky nasledujúce podmienky:

- i. Každý atóm z $\mathcal{A}_{\mathcal{L}}$ patrí do $\mathcal{E}_{\mathcal{L}}$. Inak povedané, $\mathcal{A}_{\mathcal{L}} \subseteq \mathcal{E}_{\mathcal{L}}$.
- ii. 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 .
- iii. 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 .

iv. 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} .

Skracovanie zápisu formúl

Dohoda 10.9. Zápis formúl môžeme skracovať nasledujúcim spôsobom:

- Negáciu rovnostného atómu $\neg s \doteq t$ skrátene zapisujeme $s \neq t$.
- Ak $\circ \in \{\wedge, \vee\}$, tak $((A \circ B) \circ C)$ môžeme skrútiť na $(A \circ B \circ C)$.
- Binárnym spojкам priradíme *prioritu*: *najvyššiu* prioritu má \wedge , *strednú* \vee , *najnižšiu* \rightarrow .

Ak spojka \circ má *vyššiu* prioritu ako \diamond , tak v každej formule môžeme podformulu $((A \circ B) \diamond X)$ skrútiť na $(A \circ B \diamond X)$ a podformulu $(X \diamond (A \circ B))$ skrútiť na $(X \diamond A \circ B)$.

- Vonkajší pár zátvoriek okolo celej formuly môžeme vždy vynechať, napr. $(\forall x(a \doteq x \vee P(x)) \rightarrow P(b))$ skrátíme na $\forall x(a \doteq x \vee P(x)) \rightarrow P(b)$.

! **Neodstraňujeme** (ale ani nepridávame) zátvorky okolo priamych podformúl negácie a kvantifikátorov, ani okolo implikácie vnorenej v implikácii.

Skracovanie zápisu formúl

Príklad 10.10. Formulu

$$\left(\exists x \forall y (S(x) \wedge (P(y) \rightarrow ((\neg Z(x, y) \vee R(x, y)) \vee Q(y)))) \rightarrow \forall x ((U(x) \wedge V(x)) \rightarrow Q(x)) \right)$$

môžeme maximálne skrútiť na

$$\exists x \forall y (S(x) \wedge (P(y) \rightarrow \neg Z(x, y) \vee R(x, y) \vee Q(y))) \rightarrow \forall x (U(x) \wedge V(x) \rightarrow Q(x)).$$

Skrátený zápis

$$P(a, x) \wedge (x \doteq b \vee P(x, b) \vee R(x)) \rightarrow P(f(a), x) \vee b \doteq f(x) \wedge P(a, b)$$

vznikol z formuly

$$((P(a, x) \wedge ((x \doteq b \vee P(x, b)) \vee R(x))) \rightarrow (P(f(a), x) \vee (b \doteq f(x) \wedge P(a, b))))).$$

10.3 Sémantika logiky prvého rádu

Štruktúry

Rozšírime štruktúru tak, aby dávala význam aj funkčným symbolom:

Definícia 10.11. Nech \mathcal{L} je jazyk logiky prvého rádu. Štruktúrou pre jazyk \mathcal{L} nazývame každú dvojicu $\mathcal{M} = (D, i)$, kde

doména D štruktúry \mathcal{M} je ľubovoľná neprázdna množina;

interpretačná funkcia i štruktúry \mathcal{M} je zobrazenie, ktoré

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

Štruktúry – príklad

Príklad 10.12. Nájdime štruktúru pre jazyk \mathcal{L} , v ktorom $\mathcal{V}_{\mathcal{L}} = \{x, y, z, x_1, y_1, \dots\}$, $\mathcal{C}_{\mathcal{L}} = \{\text{Klárka, Jurko}\}$, $\mathcal{F}_{\mathcal{L}} = \{\text{matka}^1\}$, $\mathcal{P}_{\mathcal{L}} = \{\text{rodič}^2, \text{žena}^1\}$.

Riešenie. Štruktúrou pre tento jazyk môže byť napríklad $\mathcal{M} = (D, i)$, kde

$$\begin{aligned} D &= \{\mathbf{I}_M, \mathbf{I}_I, \mathbf{I}_K, \mathbf{Y}_J, \mathbf{I}_T, \mathbf{G}\}, \\ i(\text{Klárka}) &= \mathbf{I}_K, \quad i(\text{Jurko}) = \mathbf{Y}_J \\ i(\text{matka}) &= \{(\mathbf{I}_K, \mathbf{I}_M), (\mathbf{Y}_J, \mathbf{I}_M), (\mathbf{I}_M, \mathbf{I}_I), (\mathbf{I}_I, \mathbf{G}), (\mathbf{I}_T, \mathbf{G}), (\mathbf{G}, \mathbf{G})\} \\ i(\text{žena}) &= \{\mathbf{I}_M, \mathbf{I}_I, \mathbf{I}_K, \mathbf{G}\} \\ i(\text{rodič}) &= \{(\mathbf{I}_M, \mathbf{I}_K), (\mathbf{I}_M, \mathbf{Y}_J), (\mathbf{I}_T, \mathbf{Y}_J), (\mathbf{I}_I, \mathbf{I}_M)\} \end{aligned}$$

Všimnite si, že $i(\text{matka})$ je skutočne funkcia na celej doméne.

Ohodnotenie premenných

Zmena definície štruktúry neovplyvňuje ohodnotenia premenných.

Definícia 10.13. 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 v je individuová premenná z \mathcal{L} a v je prvok D . Zápis $e(x/v)$ označuje ohodnotenie e' individuových premenných, pre ktoré platí:

- $e'(x) = v$;
- $e'(y) = e(y)$, ak y je iná premenná ako x .

Hodnota termu

Termy s funkčnými symbolmi môžu byť vnorené, vyhodnocujeme ich rekurzívne:

Definícia 10.14. Nech $\mathcal{M} = (D, i)$ je štruktúra pre jazyk logiky prvého rádu \mathcal{L} , nech e je ohodnotenie premenných. *Hodnotou termu* t v štruktúre \mathcal{M} pri ohodnotení premenných e je prvok z D označovaný $t^{\mathcal{M}}[e]$ a zadeninovaný indukívne pre všetky premenné x , konštanty a , každú aritu n , všetky funkčné symboly f s aritou n , a všetky termy t_1, \dots, t_n nasledovne:

$$\begin{aligned} x^{\mathcal{M}}[e] &= e(x), \\ a^{\mathcal{M}}[e] &= i(a), \\ (f(t_1, \dots, t_n))^{\mathcal{M}}[e] &= i(f)(t_1^{\mathcal{M}}[e], \dots, t_n^{\mathcal{M}}[e]). \end{aligned}$$

Hodnoty termov

Příklad 10.15. V štruktúre $\mathcal{M} = (\{\mathbf{i}_K, \mathbf{y}_J, \mathbf{i}_M, \mathbf{i}_T, \mathbf{g}\}, i)$, $i(\text{Klárka}) = \mathbf{i}_K$, $i(\text{Jurko}) = \mathbf{y}_J$, $i(\text{matka}) = \{(\mathbf{i}_K, \mathbf{i}_M), (\mathbf{y}_J, \mathbf{i}_M), (\mathbf{i}_M, \mathbf{i}_T), (\mathbf{i}_T, \mathbf{g}), (\mathbf{i}_T, \mathbf{g}), (\mathbf{g}, \mathbf{g})\}$ pri ohodnotení $e = \{x \mapsto \mathbf{y}_J, y \mapsto \mathbf{i}_M, \dots\}$ tieto termy: Klárka, x , matka(Klárka), matka(y), matka(matka(Jurko)).

Riešenie. Pripomeňme si podstatné časti štruktúry z príkladu 10.12:

$$\text{Klárka}^{\mathcal{M}}[e] = i(\text{Klárka}) = \mathfrak{K}$$

$$x^{\mathcal{M}}[e] = e(x) = \mathfrak{J}$$

$$\begin{aligned} (\text{matka}(\text{Klárka}))^{\mathcal{M}}[e] &= i(\text{matka})(\text{Klárka}^{\mathcal{M}}[e]) \\ &= i(\text{matka})(\mathfrak{K}) = \mathfrak{M} \end{aligned}$$

$$\begin{aligned} (\text{matka}(y))^{\mathcal{M}}[e] &= i(\text{matka})(y^{\mathcal{M}}[e]) = i(\text{matka})(e(y)) \\ &= i(\text{matka})(\mathfrak{J}) = \mathfrak{I} \end{aligned}$$

$$(\text{matka}(\text{matka}(\text{Jurko})))^{\mathcal{M}}[e] = i(\text{matka})(i(\text{matka})(i(\text{Jurko}))) = \mathfrak{I}$$

Splnenie formuly v štruktúre

Definícia 10.16. Nech $\mathcal{M} = (D, i)$ je štruktúra pre \mathcal{L} , e je ohodnotenie premenných. Relácia štruktúra \mathcal{M} spĺňa formulu X pri ohodnotení e (skrátene $\mathcal{M} \models X[e]$) 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 $m \in D$ máme $\mathcal{M} \models A[e(x/m)]$,
- $\mathcal{M} \models \forall x A[e]$ vtt pre každý prvok $m \in D$ máme $\mathcal{M} \models A[e(x/m)]$,

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 jazyka \mathcal{L} .

Ďalšie pojmy

Pojmy:

- pravdivosť uzavretej formuly v štruktúre,
- pravdivosť teórie v štruktúre,
- splniteľnosť,
- nesplniteľnosť,
- platná formula,
- prvorádové vyplývanie

definujeme analogicky ako v relačnej logike prvého rádu.

Pravdivosť formúl v štruktúre

Príklad 10.17 (Pravdivosť formúl v štruktúre). V štruktúre $\mathcal{M} = (\{i_M, i_I, i_K, \check{y}_J, i_T, \oplus\})$ kde

$$i(\text{Klárka}) = i_K, \quad i(\text{Jurko}) = \check{y}_J$$

$$i(\text{matka}) = \{(i_K, i_M), (\check{y}_J, i_M), (i_M, i_I), (i_I, \oplus), (i_T, \oplus), (\oplus, \oplus)\}$$

$$i(\text{žena}) = \{i_M, i_I, i_K, \oplus\}$$

$$i(\text{rodič}) = \{(i_M, i_K), (i_M, \check{y}_J), (i_T, \check{y}_J), (i_I, i_M)\}$$

máme napríklad

$$\mathcal{M} \models \forall x \text{žena}(\text{matka}(x))$$

$$\mathcal{M} \models \forall x \forall y (\text{žena}(x) \wedge \text{rodič}(x, y) \rightarrow \text{matka}(y) \doteq x)$$

ale

$$\mathcal{M} \not\models \forall x \text{rodič}(\text{matka}(x), x)$$

11 Tablá pre logiku prvého rádu

Jednotný zápis označených formúl — α a β

Definícia (Jednotný zápis označených formúl typu α)

Označená formula je *typu* α vtt má jeden z tvarov v ľavom stĺpci tabuľky pre nejaké formuly A a B . Takéto formuly označujeme písmenom α ; α_1 označuje príslušnú formulu zo stredného stĺpca a α_2 príslušnú formulu z pravého stĺpca.

α	α_1	α_2
$\mathbf{T}(A \wedge B)$	\mathbf{TA}	\mathbf{TB}
$\mathbf{F}(A \vee B)$	\mathbf{FA}	\mathbf{FB}
$\mathbf{F}(A \rightarrow B)$	\mathbf{TA}	\mathbf{FB}
$\mathbf{T}\neg A$	\mathbf{FA}	\mathbf{FA}
$\mathbf{F}\neg A$	\mathbf{TA}	\mathbf{TA}

Definícia (Jednotný zápis označených formúl typu β)

Označená formula je *typu* β vtt má jeden z tvarov v ľavom stĺpci tabuľky pre nejaké formuly A a B . Takéto formuly označujeme písmenom β ; β_1 označuje príslušnú formulu zo stredného stĺpca a β_2 príslušnú formulu z pravého stĺpca.

β	β_1	β_2
$\mathbf{F}(A \wedge B)$	\mathbf{FA}	\mathbf{FB}
$\mathbf{T}(A \vee B)$	\mathbf{TA}	\mathbf{TB}
$\mathbf{T}(A \rightarrow B)$	\mathbf{FA}	\mathbf{TB}

Jednotný zápis označených formúl — γ a δ

Definícia (Jednotný zápis označených formúl typu γ)

Označená formula je *typu* γ vtt má jeden z tvarov v ľavom stĺpci tabuľky pre nejakú formulu A a individuovú premennú x . Takéto formuly označujeme $\gamma(x)$ a pre ľubovoľný term t substituovateľný za x v A príslušnú formulu z pravého stĺpca označujeme $\gamma_1(t)$.

$\gamma(x)$	$\gamma_1(t)$
$\mathbf{F}\exists x A$	$\mathbf{FA}\{x \mapsto t\}$
$\mathbf{T}\forall x A$	$\mathbf{TA}\{x \mapsto t\}$

Definícia (Jednotný zápis označených formúl typu δ)

Označená formula je *typu* δ vtt má jeden z tvarov v ľavom stĺpci tabuľky pre nejakú formulu A a individuovú premennú x . Takéto formuly označujeme $\delta(x)$ a pre ľubovoľnú premennú y substituovateľnú za x v A príslušnú formulu z pravého stĺpca označujeme $\delta_1(y)$.

$\delta(x)$	$\delta_1(y)$
$\mathbf{T}\exists x A$	$\mathbf{TA}\{x \mapsto y\}$
$\mathbf{F}\forall x A$	$\mathbf{FA}\{x \mapsto y\}$

11.1 Vlastnosti rovnosti

Rovnosť

Pravidlá pre α a β formuly

$$\frac{\alpha}{\alpha_1} \quad \frac{\alpha}{\alpha_2} \quad \frac{\beta}{\beta_1 \mid \beta_2}$$

umožňujú pracovať s logickými spojkami.

Pravidlá pre γ a δ formuly

$$\frac{\gamma(x)}{\gamma_1(t)} \quad \frac{\delta(x)}{\delta_1(y)}$$

umožňujú pracovať s kvantifikátormi.

V jazyku je ešte jeden logický symbol – rovnosť (\doteq).

Žiadne pravidlo s ňou zatiaľ nepracuje.

Čo potrebujeme, aby rovnosť mala očakávané vlastnosti?

Axiomatizácia rovnosti

Rovnosť by sme mohli popísať teóriou – *axiomatizovať* ju.

Rovnosť je reflexívna, symetrická a tranzitívna:

$$\begin{aligned} \forall x \ x \doteq x & \quad \forall x \ \forall y (x \doteq y \rightarrow y \doteq x) \\ \forall x \ \forall y \ \forall z (x \doteq y \wedge y \doteq z \rightarrow x \doteq z) \end{aligned}$$

Navyše má vlastnosť *kongruencie*: Pre každý pár rovnajúcich sa k -tic argumentov je hodnota každého funkčného symbolu f^k je rovnaká:

$$\forall x_1 \ \forall y_1 \ \dots \ \forall x_k \ \forall y_k (x_1 \doteq y_1 \wedge \dots \wedge x_k \doteq y_k \rightarrow f(x_1, \dots, x_k) \doteq f(y_1, \dots, y_k))$$

a každý predikátový symbol P^k je na oboch k -ticiach splnený alebo na oboch nesplnený:

$$\forall x_1 \ \forall y_1 \ \dots \ \forall x_k \ \forall y_k (x_1 \doteq y_1 \wedge \dots \wedge x_k \doteq y_k \rightarrow (P(x_1, \dots, x_k) \leftrightarrow P(y_1, \dots, y_k)))$$

Dôkazy s axiomatizovanou rovnosťou

Skúsme niečo dokázať:

1.	$\mathbf{T} \ m(J) \doteq M$	S^+
2.	$\mathbf{T} \ pd(m(J), O) \doteq K$	S^+
3.	$\mathbf{F} \ pd(M, O) \doteq K$	S^+
4.	$\mathbf{T} \ \forall x_1 \forall y_1 \forall x_2 \forall y_2 (x_1 \doteq y_1 \wedge x_2 \doteq y_2 \rightarrow pd(x_1, x_2) \doteq pd(y_1, y_2))$	Kong
5.	$\mathbf{T} \ \forall y_1 \forall x_2 \forall y_2 (m(J) \doteq y_1 \wedge x_2 \doteq y_2 \rightarrow pd(m(J), x_2) \doteq pd(y_1, y_2))$	$\gamma 4\{x_1 \mapsto m(J)\}$
6.	$\mathbf{T} \ \forall x_2 \forall y_2 (m(J) \doteq M \wedge x_2 \doteq y_2 \rightarrow pd(m(J), x_2) \doteq pd(M, y_2))$	$\gamma 5\{y_1 \mapsto M\}$
7.	$\mathbf{T} \ \forall y_2 (m(J) \doteq M \wedge O \doteq y_2 \rightarrow pd(m(J), O) \doteq pd(M, y_2))$	$\gamma 6\{x_2 \mapsto O\}$
8.	$\mathbf{T} \ m(J) \doteq M \wedge O \doteq O \rightarrow pd(m(J), O) \doteq pd(M, O)$	$\gamma 7\{y_2 \mapsto O\}$

9.	$\mathbf{F} \ m(J) \doteq M \wedge O \doteq O$	$\beta 8$	13.	$\mathbf{T} \ pd(m(J), O) \doteq pd(M, O)$	$\beta 8$
10.	$\mathbf{F} \ O \doteq O$	NCS9, 1		⋮	
11.	$\mathbf{T} \ \forall x \ x \doteq x$	Refl			
12.	$\mathbf{T} \ O \doteq O$	$\gamma 11\{x \mapsto O\}$			
	* 10, 12				

Axiómy či pravidlá?

Doteraz sme mali dokazovací systém s mnohými odvodzovacími pravidlami (α, β, \dots) a žiadnymi axiómami. Po pridaní axióm pre rovnosť máme systém, kde sú aj pravidlá, aj axiómy. Náš dokazovací systém je *korektný* aj *úplný*. Nie je však *jediný* taký.

Alternatívny dokazovací systém pre výrokovú logiku (Hilbert):

- Jediné pravidlo: modus ponens.
- Axiómy (A, B, C sú formuly): $A \rightarrow (B \rightarrow A)$ ($A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$ ($\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B)$

Korektných aj úplných dokazovacích systémov môžeme navrhnúť hocikoľko. Z hľadiska logických vlastností budú „ekvivalentné“, môžu sa však v praxi líšiť výpočtovými vlastnosťami a (ne)intuitívnosťou.

11.2 Tablové pravidlá pre rovnosť

Leibnitzovo pravidlo

Dôkazy s axiómami rovnosti sú práce aj v jednoduchých prípadoch.

Kongruencia sa však dá induktívne zovšeobecniť na ľubovoľné formuly – *Leibnitzovo pravidlo*: V každej formule môžeme nahradiť rovné rovným.

1. $\mathbf{T} m(J) \doteq M \quad S^+$
 2. $\mathbf{T} \text{pd}(m(J), O) \doteq K \quad S^+$
 3. $\mathbf{F} \text{pd}(M, O) \doteq K \quad S^+$
 4. $\mathbf{T} \text{pd}(M, O) \doteq K \quad \text{Leibnitz}1, 2$
- * 3, 4

Ale naozaj?

1. $\mathbf{T} m(J) \doteq x \quad S^+$
2. $\mathbf{T} \exists x \text{pd}(m(J), x) \doteq K \quad S^+$
3. $\mathbf{T} \exists x \text{pd}(x, x) \doteq K \quad \text{Leibnitz?}1, 2 \quad \star$

znova konflikt s viazanou premennou

Leibnitzovo pravidlo presne

Leibnitzovo pravidlo: V každej formule môžeme nahradiť rovné rovným.

- Čo znamená „nahradiť“?
- Kedy môžeme nahrádzať bez ohrozenia vyplývania?

Substitúcia $\{x \mapsto t\}$ nahrádza premennú termom.

Pomocou *substituovateľnosti* sme vylúčili prípady, keď by substitúcia odvodila nesprávne „dôsledky“.

Leibnitzovo pravidlo potrebuje nahradiť jeden term t_1 druhým t_2 . Dá sa to popísať substitúciami? Potom by sme možno nepotrebovali špeciálne podmienky pre korektnosť Leibnitzovho pravidla.

Leibnitzovo pravidlo presne

Podľa rovnosti $m(J) \doteq M$ chceme nahradiť term $t_1 = m(J)$ termom $t_2 = M$ v označenej formule:

$$A_1^+ = \mathbf{T} \exists x \text{pd}(m(J), x) \doteq K$$

1. Predstavíme si, že na mieste **nahrádzaného** termu je **nová premenná q** :

$$A^+ = \mathbf{T} \exists x \text{pd}(q, x) \doteq K$$

2. Pôvodná formula A_1^+ vznikne z A^+ substitúciou t_1 za q :

$$\begin{aligned} A_1^+ &= \mathbf{T} \exists x \text{pd}(m(J), x) \doteq K \\ &= A^+ \{q \mapsto m(J)\} \end{aligned}$$

3. Nová formula A_2^+ vznikne z A^+ substitúciou t_2 za q :

$$\begin{aligned} A_2^+ &= A^+\{q \mapsto M\} \\ &= \mathbf{T} \exists x \text{pd}(M, x) \doteq K \end{aligned}$$

Leibnitzovo pravidlo pomocou substitúcií

Vyjadrenie Leibnitzovho pravidla pomocou substitúcií:

$$\frac{\mathbf{T} t_1 \doteq t_2 \quad A^+\{q \mapsto t_1\}}{A^+\{q \mapsto t_2\}}$$

pre všetky termy t_1 a t_2 , označené formuly A^+ a premenné q také, že t_1 a t_2 sú *substituovateľné* za q v A^+ .

Prečo kladieme podmienku aj na t_1 ? Ak by sa t_1 nachádzal v A^+ na mieste, kde by niektorá jeho premenná mala viazaný výskyt, jeho nahradením by sme mohli zmeniť význam formuly.

Leibnitzovo pravidlo — obmedzenia

Automaticky dostávame *rozumné obmedzenia*:

Nemôžeme nahradiť term $t_1 = m(J)$ termom $t_2 = \underline{x}$ vo formule:

$$\begin{aligned} A_1^+ &= \mathbf{T} \exists \underline{x} \text{pd}(m(J), \underline{x}) \doteq K \\ &= A^+\{q \mapsto m(J)\} \\ A^+ &= \mathbf{T} \exists \underline{x} \text{pd}(q, \underline{x}) \doteq K \end{aligned}$$

lebo \underline{x} *nie je substituovateľná* za q v A^+ (\underline{x} je viazaná v mieste voľného výskytu q).

Vlastnosti rovnosti a Leibnitzovo pravidlo

Leibnitzovým pravidlom odvodíme kongruenciu, nie však reflexivitu. Po pridaní pravidla pre reflexivitu odvodíme aj symetriu a tranzitivitu.

$$\frac{}{\mathbf{T} t_0 \doteq t_0}$$

Symetriu potom odvodíme postupnosťou krokov:

1. $\mathsf{T} t_1 \doteq t_2$
2. $\mathsf{T} t_1 \doteq t_1$ reflexivita $\mathsf{T} q \doteq t_1 \{q \mapsto t_1\}$
3. $\mathsf{T} t_2 \doteq t_1$ Leibnitz 1 a 2 $\mathsf{T} q \doteq t_1 \{q \mapsto t_2\}$

Tranzitivitu odvodíme:

1. $\mathsf{T} t_1 \doteq t_2$ $\mathsf{T} t_1 \doteq q \{q \mapsto t_2\}$
2. $\mathsf{T} t_2 \doteq t_3$
3. $\mathsf{T} t_1 \doteq t_3$ Leibnitz 2 a 1 $\mathsf{T} t_1 \doteq q \{q \mapsto t_3\}$

11.3 Tablá pre logiku prvého rádu

Tablové pravidlá pre logiku prvého rádu

Definícia 11.1. Tablovými pravidlami pre logiku prvého rádu sú:

$$\frac{\alpha}{\alpha_1} \quad \frac{\alpha}{\alpha_2} \qquad \frac{\beta}{\beta_1 \mid \beta_2}$$

$$\frac{\gamma(x)}{\gamma_1(t)} \qquad \frac{\delta(x)}{\delta_1(y)}$$

$$\frac{}{\mathsf{T} t_0 \doteq t_0} \qquad \frac{\mathsf{T} t_1 \doteq t_2 \quad A^+\{x \mapsto t_1\}}{A^+\{x \mapsto t_2\}}$$

pre všetky ozn. formuly α , β , $\gamma(x)$, $\delta(x)$ príslušných typov a všetky im zodpovedajúce α_1 , α_2 , β_1 , β_2 , $\gamma_1(t)$ a $\delta_1(y)$, všetky termy t_0 , všetky ozn. formuly A^+ , všetky termy t_1 a t_2 substituovateľné za x do príslušnej A^+ .

Tablo pre množinu označených formúl

Definícia 11.2. *Analytické tablo pre množinu označených formúl S^+ (skrátene tablo pre S^+)* je binárny strom, ktorého vrcholy obsahujú označené formuly a je skonštruovaný induktívne podľa nasledovných 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 ℓ je nejaký jeho list. Potom tablom pre S^+ je aj každé *priame rozšírenie* \mathcal{T} ktorýmkoľvek z pravidiel:
 - S^+ :** Ako jediné dieťa ℓ pripojíme nový vrchol obsahujúci ľubovoľnú označenú formulu $A^+ \in S^+$.
 - α :** Ak sa na vetve π_ℓ (ceste z koreňa do ℓ) vyskytuje nejaká označená formula α , tak ako jediné dieťa ℓ pripojíme nový vrchol obsahujúci α_1 alebo α_2 .
 - β :** Ak sa na vetve π_ℓ vyskytuje nejaká označená formula β , tak ako deti ℓ pripojíme dva nové vrcholy, pričom ľavé dieťa bude obsahovať β_1 a pravé β_2 .

Tablo pre množinu označených formúl

Definícia 11.2 (pokračovanie).

- γ :** Ak sa na vetve π_ℓ vyskytuje nejaká označená formula $\gamma(x)$, tak ako jediné dieťa ℓ pripojíme nový vrchol obsahujúci $\gamma_1(t)$ pre ľubovoľný term t *substituovateľný* za x v $\gamma_1(x)$.
- δ :** Ak sa na vetve π_ℓ vyskytuje nejaká označená formula $\delta(x)$, tak ako jediné dieťa ℓ pripojíme nový vrchol obsahujúci $\delta_1(y)$ pre ľubovoľnú premennú y , ktorá je *substituovateľná* za x v $\delta_1(x)$ a *nemá voľný výskyt* v žiadnej formule na vetve π_ℓ .
- L:** Ak sa na vetve π_ℓ vyskytuje $\mathbf{T} t_1 \doteq t_2$ pre nejaké termy t_1 a t_2 a označená formula $A^+\{x \mapsto t_1\}$ pre nejakú A^+ , v ktorej sú t_1 a t_2 *substituovateľné* za x , tak ako jediné dieťa ℓ pripojíme nový vrchol obsahujúci $A^+\{x \mapsto t_2\}$.
- R:** Ako jediné dieťa ℓ pripojíme nový vrchol obsahujúci označenú formulu $\mathbf{T} t \doteq t$ pre ľubovoľný term t .

Korektnosť prvorádových tabiel

Veta 11.3 (Korektnosť tablového kalkulu). *Nech S^+ je množina označených formúl. Ak existuje uzavreté tablo \mathcal{T} pre S^+ , tak je množina S^+ nesplniteľná.*

Ak je množina S^+ splniteľná, každé tablo pre S^+ má otvorenú vetvu. [2mm]

Pri prvorádovom table však nemožno hovoriť o úplnosti otvorenej vetvy (pravidlá γ , δ , **L**, **R** umožňujú pridať nekonečne veľa formúl).

Prečo vyžadujeme neprázdnu doménu?

Uvažujme v table formulu $\mathbf{T} \exists x x \doteq x$. Možno na ňu aplikovať pravidlo δ , dostaneme $\mathbf{T} u \doteq u$. Ak je doména vždy neprázdna, táto formula je splnená v ľubovoľnej štruktúre pri ohodnotení $e = \{u \mapsto d\}$, kde d je ľub. prvok domény. Ak však doména môže byť prázdna, formula splnená niekedy nebude: v štruktúrach s prázdnom doménou nevieme zmysluplne hovoriť o žiadnom ohodnotení premenných. To spochybňuje pravidlo δ – sotva je legitímne ho použiť, ak nemožno ohodnotiť novú voľnú premennú, ktorú pridáva do tabla.

Problém tiež nastane pri formule $\forall x P(x) \rightarrow \exists x P(x)$. Táto formula je dokázateľná v mnohých bežných dokazovacích systémoch, ale nebola by splnená v štruktúrach s prázdnom doménou, teda ani platná. Tieto dokazovacie systémy by sme tak museli komplikovane upravovať.

12 Vlastnosti kvantifikátorov

Zoslabenie všeobecného kvantifikátora a premenovanie premenných

Tvrdenie 12.1. *Pre každú formulu A a všetky premenné x a y také, že y je substituovateľná za x v A máme:*

- i. $\forall x A \models \exists x A$
- ii. $\forall x A \models \forall y A\{x \mapsto y\}$
- iii. $\exists x A \models \exists y A\{x \mapsto y\}$
- iv. $\forall x A \models \exists y A\{x \mapsto y\}$

$$v. \models \forall y(\forall x A \rightarrow A\{x \mapsto y\})$$

$$vi. \models \exists y(A\{x \mapsto y\} \rightarrow \forall x A)$$

$$vii. \neg \exists y A\{x \mapsto y\} \models \forall y(\exists x A \rightarrow A\{x \mapsto y\})$$

Prvorádovo ekvivalentné formuly

Definícia 12.2. Formuly X a Y sú *prvorádovo ekvivalentné*, skrátene $X \Leftrightarrow Y$, vtt pre každú štruktúru \mathcal{M} a každé ohodnotenie e máme $\mathcal{M} \models X[e]$ vtt $\mathcal{M} \models Y[e]$.

Tvrdenie 12.3. *Nech X a Y sú formuly a nech $\text{free}(X) \cup \text{free}(Y) = \{x_1, \dots, x_n\}$. Nasledujúce tvrdenia sú ekvivalentné:*

- a) $X \Leftrightarrow Y$;
- b) formula $\forall x_1 \dots \forall x_n (X \leftrightarrow Y)$ je platná;
- c) existuje uzavreté tablo pre $\{\mathbf{F}(X \leftrightarrow Y)\}$.

De Morganove a distributívne zákony pre kvantifikátory

Tvrdenie 12.4. *Pre každú formulu A a každú premennú x máme:*

$$i. \neg \forall x A \Leftrightarrow \exists x \neg A,$$

$$ii. \neg \exists x A \Leftrightarrow \forall x \neg A.$$

Tvrdenie 12.5. *Pre všetky formuly A a B a každú premennú x máme:*

$$i. \exists x(A \vee B) \Leftrightarrow (\exists x A \vee \exists x B)$$

$$ii. \forall x(A \wedge B) \Leftrightarrow (\forall x A \wedge \forall x B)$$

$$iii. \exists x(A \wedge B) \models (\exists x A \wedge \exists x B)$$

$$iv. (\forall x A \vee \forall x B) \models \forall x(A \vee B)$$

Obrátené vyplývania k iii. a iv. neplatia!

Špeciálne distributívne zákony

Tvrdenie 12.6. *Pre každú formulu A , každú premennú x a pre každú formulu C , v ktorej sa x nevyskytuje voľná:*

i. $\exists x(A \vee C) \Leftrightarrow \exists x A \vee C$

ii. $\forall x(A \vee C) \Leftrightarrow \forall x A \vee C$

iii. $\forall x(A \wedge C) \Leftrightarrow \forall x A \wedge C$

iv. $\exists x(A \wedge C) \Leftrightarrow \exists x A \wedge C$

v. $\exists x C \Leftrightarrow C$

vi. $\exists x(A \rightarrow C) \Leftrightarrow (\forall x A \rightarrow C)$

vii. $\forall x(A \rightarrow C) \Leftrightarrow (\exists x A \rightarrow C)$

viii. $\forall x(C \rightarrow A) \Leftrightarrow (C \rightarrow \forall x A)$

ix. $\exists x(C \rightarrow A) \Leftrightarrow (C \rightarrow \exists x A)$

x. $\forall x C \Leftrightarrow C$

10. prednáška

Korektnosť prvorádových tabiel.

Explicitné definície. Unifikácia

13 Korektnosť tablového kalkulu

pre logiku prvého rádu

13.1 Vlastnosti ohodnotení a substitúcie

Voľné premenné a hodnota termu, splnenie formuly, teórie

Tvrdenie 13.1. *Nech \mathcal{M} je štruktúra pre \mathcal{L} , nech e_1 a e_2 sú ohodnotenia, nech t je term, A je formula a S je množina formúl jazyka \mathcal{L} .*

- *Ak sa ohodnotenia e_1 a e_2 zhodujú na (voľných) premenných termu t (teda $e_1(x) = e_2(x)$ pre každú $x \in \text{free}(t)$), tak $t^{\mathcal{M}}[e_1] = t^{\mathcal{M}}[e_2]$.*
- *Ak sa ohodnotenia e_1 a e_2 zhodujú na voľných premenných formuly X , tak $\mathcal{M} \models A[e_1]$ vtt $\mathcal{M} \models A[e_2]$.*
- *Ak sa ohodnotenia e_1 a e_2 zhodujú na voľných premenných všetkých formúl z S , tak $\mathcal{M} \models S[e_1]$ vtt $\mathcal{M} \models S[e_2]$.*

Substitúcia a hodnota termu

Ako súvisí hodnota termu po substitúcii s hodnotou termu, do ktorého sa substituuje?

Príklad 13.2. Zoberme štruktúru $\mathcal{M} = (D, i)$, kde

$$\begin{aligned} D &= \{1, 2, 3, 4, 5\}, \\ i(c) &= 3, \quad i(d) = 4 \\ i(f) &= \{1 \mapsto 2, 2 \mapsto 5, 3 \mapsto 1, 4 \mapsto 1, 5 \mapsto 5\} \end{aligned}$$

Nech $e = \{x \mapsto 3, y \mapsto 4\}$.

$$\begin{aligned}
 ((f(x))\{x \mapsto f(y)\})^{\mathcal{M}}[e] &= (f(f(y)))^{\mathcal{M}}[e] \\
 &= i(f)(i(f)(4)) = i(f)(1) = 2 \\
 &= (f(x))^{\mathcal{M}}[e(x/1)] \\
 &= (f(x))^{\mathcal{M}}[e(x/(f(y)))^{\mathcal{M}}[e]]
 \end{aligned}$$

Substitúcia vs. hodnota termu a splnenie formuly

Hodnota termu $t\sigma$ /splnenie formuly $A\sigma$ po substitúcii $\sigma = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$ pri ohodnotení e sa rovná hodnote termu t /splneniu formuly A pri ohodnotení e' , ktoré

- každej substituovanej premennej x_i priradí hodnotu za ňu substituovaného termu t_i pri ohodnotení e ,
- ostatným premenným priraduje rovnaké hodnoty ako e .

Tvrdenie 13.3. *Nech \mathcal{M} je štruktúra pre jazyk \mathcal{L} , e ohodnotenie ind. premenných a nech $\sigma = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$ je substitúcia.*

- *Nech t je term jazyka \mathcal{L} . Potom $(t\sigma)^{\mathcal{M}}[e] = t^{\mathcal{M}}[e(x_1/t_1^{\mathcal{M}}[e]) \cdots (x_n/t_n^{\mathcal{M}}[e])]$.*
- *Nech A je formula jazyka \mathcal{L} a σ je aplikovateľná na A . Potom $\mathcal{M} \models A\sigma[e]$ vtt $\mathcal{M} \models A[e(x_1/t_1^{\mathcal{M}}[e]) \cdots (x_n/t_n^{\mathcal{M}}[e])]$.*

13.2 Korektnosť tabiel

Korektnosť tablových pravidiel

Tvrdenie 13.4. *Nech S^+ je množina označených formúl v jazyku \mathcal{L} , nech x a y sú premenné, nech s, t sú termy, nech $\alpha, \beta, \gamma, \delta$ sú ozn. formuly príslušného typu, A je ozn. formula.*

- *Ak $\alpha \in S^+$, tak S^+ je splniteľná vtt $S^+ \cup \{\alpha_1, \alpha_2\}$ je splniteľná.*
- *Ak $\beta \in S^+$, tak S^+ je splniteľná vtt $S^+ \cup \{\beta_1\}$ je splniteľná alebo $S^+ \cup \{\beta_2\}$ je splniteľná.*

- Ak $\gamma(x) \in S^+$ a t je term 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 nemá voľný výskyt v S^+ , tak S^+ je splniteľná vtt $S^+ \cup \{\delta_1(y)\}$ je splniteľná.
- S^+ je splniteľná vtt $S^+ \cup \{\mathbf{T} t \doteq t\}$ je splniteľná.
- Ak $\{\mathbf{T} s \doteq t, A^+\{x \mapsto s\}\} \subseteq S^+$, s a t sú substituovateľné za x v A^+ , tak S^+ je splniteľná vtt $S^+ \cup \{A^+\{x \mapsto t\}\}$ je splniteľná.

Korektnosť tablových pravidiel — dôkaz

Dôkaz (čiastočný, pre pravidlo δ v smere \Rightarrow). Zoberme ľubovoľné S^+ , x , y a $\delta(x)$ spĺňajúce predpoklady tvrdenia. Nech S^+ je splniteľná, teda existuje štruktúra $\mathcal{M} = (D, i)$ a ohodnotenie e také, že $\mathcal{M} \vDash S^+[e]$. Preto aj $\mathcal{M} \vDash \delta(x)[e]$. Podľa tvaru $\delta(x)$ môžu nastať nasledujúce dva prípady:

- Ak $\delta(x) = \mathbf{T} \exists x A$ pre nejakú formulu A , tak podľa def. 8.12 $\mathcal{M} \vDash \exists x A[e]$ a podľa def. 10.16 máme nejakého svedka $m \in D$ takého, že $\mathcal{M} \vDash A[e(x/m)]$. Podľa tvr. 13.3 potom $\mathcal{M} \vDash A\{x \mapsto y\}[e(x/m)(y/m)]$. Prem. x nie je voľná v $A\{x \mapsto y\}$, preto podľa tvr. 13.1 $\mathcal{M} \vDash A\{x \mapsto y\}[e(y/m)]$, teda $\mathcal{M} \vDash \mathbf{T} A\{x \mapsto y\}[e(y/m)]$, teda $\mathcal{M} \vDash \delta_1(y)[e(y/m)]$.

Korektnosť tablových pravidiel — dôkaz

Dôkaz (čiastočný, pre pravidlo δ v smere \Rightarrow , pokračovanie). • Ak $\delta(x) = \mathbf{F} \forall x A$ pre nejakú formulu A , tak podľa def. 8.12 $\mathcal{M} \not\vDash \forall x A[e]$ a podľa def. 10.16 neplatí, že $\mathcal{M} \vDash A[e(x/m)]$ pre každé $m \in D$. Preto máme nejaký *kontrapríklad* $m \in D$ taký, že $\mathcal{M} \not\vDash A[e(x/m)]$. Podľa tvr. 13.3 potom $\mathcal{M} \not\vDash A\{x \mapsto y\}[e(x/m)(y/m)]$. Prem. x nie je voľná v $A\{x \mapsto y\}$, preto podľa tvr. 13.1 $\mathcal{M} \not\vDash A\{x \mapsto y\}[e(y/m)]$, teda $\mathcal{M} \vDash \mathbf{F} A\{x \mapsto y\}[e(y/m)]$, čiže $\mathcal{M} \vDash \delta_1(y)[e(y/m)]$.

Navyše y nie je voľná v žiadnej formule z S^+ , preto $\mathcal{M} \vDash S^+[e(y/m)]$. Teda $\mathcal{M} \vDash (S^+ \cup \{\delta_1(y)\})[e(y/m)]$. Preto je $S^+ \cup \{\delta_1(y)\}$ splniteľná. \square

Korektnosť — pravdivosť priameho rozšírenia tabla

Vetva sa správa ako konjunkcia svojich označených formúl — všetky musia byť naraz splnené.

Tablo sa správa ako disjunkcia vetiev — niektorá musí byť splnená.

Definícia 13.5. Nech S^+ je množina označených formúl v jazyku \mathcal{L} , nech \mathcal{T} je tablo pre S^+ , nech π je vetva tabla \mathcal{T} . Nech \mathcal{M} je štruktúra pre \mathcal{L} a e je ohodnotenie individuových premenných. Potom:

- štruktúra \mathcal{M} spĺňa vetvu π pri e vtt \mathcal{M} spĺňa všetky označené formuly vyskytujúce sa na vetve π pri e .
- štruktúra \mathcal{M} spĺňa tablo \mathcal{T} pri e vtt \mathcal{M} spĺňa niektorú vetvu v table \mathcal{T} pri e .

Pomocné tvrdenia pre korektnosť prvorádových tabiel

Lema 13.6 (K1). Nech S^+ je množina ozn. formúl v jazyku \mathcal{L} , nech \mathcal{T} je tablo pre S^+ . Nech \mathcal{M} je štruktúra pre \mathcal{L} a e je ohodnotenie ind. premenných. Ak \mathcal{T} a S^+ sú splnené štruktúrou \mathcal{M} pri e , tak aj každé priame rozšírenie \mathcal{T} a S^+ sú splnené štruktúrou \mathcal{M} pri nejakom ohodnotení e' .

Definícia 13.7. Nech \mathcal{T} je tablo pre nejakú množinu označených formúl. Tablo \mathcal{T} je *splniteľné* vtt existuje štruktúra, ktorá spĺňa \mathcal{T} pri nejakom ohodnotení individuových premenných.

Lema 13.8 (K2). Nech S^+ je množina ozn. formúl v jazyku \mathcal{L} , nech \mathcal{T} je tablo pre S^+ . Ak S^+ je splniteľná, tak aj \mathcal{T} je splniteľné.

Korektnosť prvorádových tabiel

Otvorené a uzavreté vetvy a tablá sú definované rovnako ako pri tabľách pre výrokovú logiku.

Veta 13.9 (Korektnosť tablového kalkulu). Nech S^+ je množina označených formúl. Ak existuje uzavreté tablo \mathcal{T} pre S^+ , tak je množina S^+ nesplniteľná.

Dôkaz (sporom). Nech S^+ je množina označených formúl. Nech existuje uzavreté tablo \mathcal{T} pre S^+ , ale S^+ je splniteľná. Pretože \mathcal{T} je uzavreté, pre každú jeho vetvu π existuje formula X taká, že \mathbf{TX} a \mathbf{FX} sa vyskytuje na π , a teda π je nesplniteľná. Preto \mathcal{T} je nesplniteľné. To je v spore s lemov K2, podľa ktorej je \mathcal{T} splniteľné, pretože S^+ je splniteľná. \square

13.3 Ďalšie korektné pravidlá

Pohodnejšie verzie pravidiel γ a δ

Tvrdenie 13.10. Nasledujúce pravidlá sú korektné:

$$\gamma^* \quad \frac{\mathbf{T} \forall x_1 \dots \forall x_n A}{\mathbf{T} A\{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}} \quad \frac{\mathbf{F} \exists x_1 \dots \exists x_n A}{\mathbf{F} A\{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}}$$

$$\delta^* \quad \frac{\mathbf{F} \forall x_1 \dots \forall x_n A}{\mathbf{F} A\{x_1 \mapsto y_1, \dots, x_n \mapsto y_n\}} \quad \frac{\mathbf{T} \exists x_1 \dots \exists x_n A}{\mathbf{T} A\{x_1 \mapsto y_1, \dots, x_n \mapsto y_n\}}$$

kde A je formula, x_1, \dots, x_n sú premenné, t_1, \dots, t_n sú termy, y_1, \dots, y_n sú navzájom rôzne premenné, ktoré sa nevyskytujú voľné vo vetve, v liste ktorej je pravidlo použité, pričom pre každé $i \in \{1, \dots, n\}$ je term t_i substituovateľný za x_i v A a premenná y_i je substituovateľná za x_i v A .

Pravidlá pre ekvivalenciu

Tvrdenie 13.11. Nasledujúce pravidlá sú korektné:

$$ESTT \quad \frac{\mathbf{T}(A_1 \leftrightarrow A_2) \quad \mathbf{T} A_i}{\mathbf{T} A_{3-i}} \quad ESTF \quad \frac{\mathbf{T}(A_1 \leftrightarrow A_2) \quad \mathbf{F} A_i}{\mathbf{F} A_{3-i}}$$

$$ESFT \quad \frac{\mathbf{F}(A_1 \leftrightarrow A_2) \quad \mathbf{T} A_i}{\mathbf{F} A_{3-i}} \quad ESFF \quad \frac{\mathbf{F}(A_1 \leftrightarrow A_2) \quad \mathbf{F} A_i}{\mathbf{T} A_{3-i}}$$

kde A_1 a A_2 sú formuly, $i \in \{1, 2\}$.

Všimnite si: $3 - 1 = 2$ a $3 - 2 = 1$.

14 Explicitné definície

Pojmy

V mnohých doménach sú zaujímavé komplikovanejšie kombinácie základných vlastností alebo vzťahov:

- x má spoločného rodiča s y , ale x je rôzne od y
 $\exists z(\text{rodič}(z, x) \wedge \text{rodič}(z, y)) \wedge \neg x \doteq y$

- x je živočích, ktorý konzumuje iba rastliny
 $\text{živočích}(x) \wedge \forall y(\text{konzumuje}(x, y) \rightarrow \text{rastlina}(y))$

Často sa vyskytujúce kombinácie vzťahov a vlastností je výhodné:

- *pomenovať*
- a jasne *vyjadriť význam* nového mena pomocou doteraz známych vlastností a vzťahov,

teda *zadefinovať pojem*.

Definície pojmov

Definícia je tvrdenie, ktoré vyjadruje význam pojmu.

Explicitná definícia (najčastejší druh definície) je *ekvivalencia* medzi pojmom a opisom jeho významu, v ktorom sa definovaný pojem sám nevyskytuje.

Príklad 14.1. • *Objekt x je súrodencom objektu y práve vtedy, keď x nie je y a x má spoločného rodiča s y .*

$$\forall x \forall y (\text{súrodeneec}(x, y) \leftrightarrow (x \neq y \wedge \exists z (\text{rodič}(z, x) \wedge \text{rodič}(z, y))))$$

- *Objekt x je bylinožravec vtedy a len vtedy, keď x je živočích, ktorý konzumuje iba rastliny.*

$$\forall x (\text{bylinožravec}(x) \leftrightarrow (\text{živočích}(x) \wedge \forall y (\text{konzumuje}(x, y) \rightarrow \text{rastlina}(y))))$$

Explicitná def. a nutná a postačujúca podmienka

Všimnite si:

Definícia pojmu *súrodeneec* vyjadruje *nutnú aj postačujúcu* podmienku toho, aby medzi dvoma objektmi bol súrodenecký vzťah.

- Pre každú dvojicu objektov x a y , ktoré označíme za súrodencov, *musí* existovať ich spoločný rodič a musia byť navzájom rôzne.

- ← Každé dva navzájom rôzne objekty x a y , ktoré majú spoločného rodiča, *musia* byť súrodenci.

Podobne pre iné definície.

Použitie pojmov

Využitím definovaného pojmu

- skracujeme tvrdenia: *Škrečky sú bylinožravce*.
 $\forall x(\text{škrekok}(x) \rightarrow \text{bylinožravec}(x))$
- jednoduchšie definujeme ďalšie pojmy:
Objekt x je sestrou objektu y práve vtedy, keď x je žena, ktorá je súrodencom y .
 $\forall x \forall y(\text{sestra}(x, y) \leftrightarrow (\text{žena}(x) \wedge \text{súrodenec}(x, y)))$
- potenciálne skrácujeme dôkazy (napr. nájdeme spor vyjadrený novým pojmom a nemusíme analyzovať celý podstrom zodpovedajúci rozvinutiu pojmu cez jeho definíciu)

Vyskúšajte si 14.1

Zadefinujte pojem *teta* (chápaný ako vzťah dvoch ľudí) neformálne (v slovenčine) aj formálne (formulou logiky prvého rádu).

Riešenie. Objekt x je *tetou* y vtedy a len vtedy, keď x je sestrou rodiča y .
 $\forall x \forall y(\text{teta}(x, y) \leftrightarrow \exists z(\text{sestra}(x, z) \wedge \text{rodič}(z, y)))$

Podmienené definície

Niekedy má pojem význam iba pre niektoré druhy objektov, alebo má ten istý pojem rôzne významy pre rôzne druhy objektov.

Vtedy môžeme definície *podmieniť* druhmi:

- *Študent absolvuje predmet vtt je z, neho hodnotený inou známkou ako Fx .*
 $\forall x \forall y(\text{študent}(x) \wedge \text{predmet}(y) \rightarrow$
 $(\text{absolvuje}(x, y) \leftrightarrow$
 $\exists z(\text{hodnotený}(x, y) \doteq z \wedge \text{známka}(z) \wedge z \neq Fx)))$

- Študent absolvuje študijný program vtt absolvuje každý jeho povinný predmet.

$$\forall x \forall y (\text{študent}(x) \wedge \text{št_prog}(y) \rightarrow$$

$$(\text{absolvuje}(x, y) \leftrightarrow$$

$$\forall z (\text{pov_predmet_prog}(z, y) \rightarrow \text{absolvuje}(x, z))))$$

Explicitná definícia presne

Definícia 14.2. Nech \mathcal{L} a \mathcal{L}_1 sú jazyky logiky prvého rádu. Jazyk \mathcal{L}_1 je rozšírením jazyka \mathcal{L} vtt $\mathcal{V}_{\mathcal{L}_1} = \mathcal{V}_{\mathcal{L}}$, $\mathcal{C}_{\mathcal{L}} \subseteq \mathcal{C}_{\mathcal{L}_1}$, $\mathcal{P}_{\mathcal{L}} \subseteq \mathcal{P}_{\mathcal{L}_1}$, $\mathcal{F}_{\mathcal{L}} \subseteq \mathcal{F}_{\mathcal{L}_1}$.

Definícia 14.3. Nech \mathcal{L} je jazyk logiky prvého rádu, T je teória v jazyku \mathcal{L} , a \mathcal{L}_P je rozšírenie jazyka o predikátový symbol P je s aritou n , ktorý sa nevyskytuje v \mathcal{L} . Teóriu v jazyku \mathcal{L}_P

$$T \cup \{\forall x_1 \dots \forall x_n (P(x_1, \dots, x_n) \leftrightarrow A)\},$$

kde A je formula, v ktorej sa nevyskytuje P , nazývame *rozšírením teórie T explicitnou definíciou* $\forall x_1 \dots \forall x_n (P(x_1, \dots, x_n) \leftrightarrow A)$ predikátového symbolu P .

Jednoznačnosť interpretácie definovaného predikátu

Význam explicitne definovaného predikátu je jednoznačne určený.

Príklad 14.4. Majme nejakú teóriu T v jazyku \mathcal{L} s $\mathcal{P}_{\mathcal{L}} = \{\text{rodič}^2\}$. Rozšírime T o $X = \forall x \forall y (\text{súrodenec}(x, y) \leftrightarrow (x \neq y \wedge \exists z (\text{rodič}(z, x) \wedge \text{rodič}(z, y))))$.

Nech $\mathcal{M} = (\{\mathbf{i}_I, \mathbf{i}_J, \mathbf{i}_K, \mathbf{i}_L, \mathbf{i}_M, \mathbf{i}_N, \mathbf{i}_O\}, i)$ je model T , kde

$$i(\text{rodič}) = \{(\mathbf{i}_I, \mathbf{i}_M), (\mathbf{i}_L, \mathbf{i}_M), (\mathbf{i}_I, \mathbf{i}_N), (\mathbf{i}_O, \mathbf{i}_N), (\mathbf{i}_M, \mathbf{i}_K), (\mathbf{i}_M, \mathbf{i}_J)\}$$

Potom sa \mathcal{M} dá *jednoznačne* rozšíriť na model $T \cup \{X\}$:

$$\mathcal{M}_1 = (\{\mathbf{i}_I, \mathbf{i}_J, \mathbf{i}_K, \mathbf{i}_L, \mathbf{i}_M, \mathbf{i}_N, \mathbf{i}_O\}, i_1, i_1(\text{rodič}) = i(\text{rodič}),$$

$$i_1(\text{súrodenec}) = \{(\mathbf{i}_M, \mathbf{i}_N), (\mathbf{i}_N, \mathbf{i}_M), (\mathbf{i}_K, \mathbf{i}_J), (\mathbf{i}_J, \mathbf{i}_K)\}$$

Definícia ako dopyt

Explicitne definovaný predikát sa správa ako *dopyt* alebo *pohľad* nad ostatnými predikátmi.

Príklad 14.5.

rodič			súrodenec	
r	d		d1	d2
		CREATE VIEW súrodenec AS		
		SELECT r1.d AS d1, r2.d AS d2		
		FROM rodič AS r1		
		JOIN rodič AS r2 ON r1.r = r2.r		
		WHERE r1.d <> r2.d		
		$\forall x \forall y$		
		(súrodenec(x, y) \leftrightarrow		
		(x \neq y \wedge		
		$\exists z(\text{rodič}(z, x) \wedge \text{rodič}(z, y)))$)		

Jednoznačnosť definičného rozšírenia

Definícia 14.6. Nech \mathcal{L}_2 je rozšírenie jazyka \mathcal{L}_1 . Nech $\mathcal{M}_1 = (D_1, i_1)$ je štruktúra pre \mathcal{L}_1 a $\mathcal{M}_2 = (D_2, i_2)$ je štruktúra pre \mathcal{L}_2 . Potom \mathcal{M}_2 je rozšírením (expanziou) \mathcal{M}_1 vtt $D_2 = D_1$ a $i_2(s) = i_1(s)$ pre každý mimologický symbol s jazyka \mathcal{L}_1 .

Tvrdenie 14.7. Nech

- T je teória v jazyku \mathcal{L} ,
- T' je rozšírenie T explicitnou definíciou predikátového symbolu.

Potom

- pre každý model teórie T existuje práve jedno jeho rozšírenie, ktoré je modelom teórie T' ,
- každý model teórie T' je rozšírením práve jedného modelu teórie T .

Konzervatívnosť definičného rozšírenia

Konzervatívnosť spočíva v tom, že pridávaním nepokazíme význam existujúcich vecí.

Tvrdenie 14.8. Nech T je teória v jazyku \mathcal{L} a T' je rozšírenie T explicitnou definíciou nejakého predikátového symbolu. Nech X je uzavretá formula jazyka \mathcal{L} . Potom $T \models X$ vtt $T' \models X$.

Kontextová definícia funkčného symbolu

Nech $A(x, y)$ je formula s voľnými premennými x, y . Táto formula popisuje akýsi vzťah medzi x a y (a mohli by sme pridať predikát, ktorým tento vzťah pomenujeme). Ak tento vzťah je funkcia, t. j.

$$T \models \forall x \exists y (A(x, y) \wedge \forall y_2 (A(x, y_2) \rightarrow y_2 \doteq y)),$$

môžeme jazyk rozšíriť o nový funkčný symbol f^1 a teóriu T o *kontextovú definíciu funkcie f* :

$$\forall x \forall y (f(x) \doteq y \leftrightarrow A(x, y))$$

Príklady:

- Do jazyka teórie grúp pridáme unárny funkčný symbol $^{-1}$ označujúci inverzný prvok (v grupe existuje práve jeden).
- Do teórie popisujúcej rodinné vzťahy pridáme funkčný symbol na označenie matky (z teórie však musí vyplývať, že matka je len jedna).

Kontextová definícia individuovej konštanty

Nech $A(y)$ je formula s voľnou premennou y . Táto formula popisuje akúsi vlastnosť prvku domény (a mohli by sme pridať predikát, ktorým túto vlastnosť pomenujeme). Ak je takýto prvok jediný, t. j.

$$T \models \exists y (A(y) \wedge \forall y_2 (A(y_2) \rightarrow y_2 \doteq y)),$$

môžeme jazyk rozšíriť o novú individuovú konštantu a a teóriu T o *kontextovú definíciu konštanty a*

$$\forall x (a \doteq x \leftrightarrow A(x))$$

Napr. pre jazyk popisujúci (matematické) polia pridáme symboly 0 a 1; do jazyka teórie množín pridáme konštantu pre prázdnu množinu.

Pridávanie mimologických symbolov

Rozširovanie existujúceho jazyka (resp. teórie) o nové predikáty, konštanty a funkčné symboly explicitnými/kontextovými definíciami *nezvyšuje* vyjadrovaciu silu jazyka:

- nové symboly možno vnímať ako pohodlné skratky,
- nevieme však dokázať nič viac, ako bez nich.

Toto zachytáva tvrdenie 14.8; podobné tvrdenia možno sformulovať aj pre pridané konštanty a funkčné symboly.

Niekedy môže byť výhodnejšie ako definičné axiómy funkcií a konštant použiť:

$$\begin{aligned} &\forall x A(x, f(x)) \\ &A(a) \end{aligned}$$

Dokazovanie s explicitnými definíciami a rovnosťou

Využime nové pravidlá na dôkaz vyplývania z teórie s definíciou:

Príklad 14.9. Dokážme tablom, že $T \models X$ pre

$$\begin{aligned} T = &\{ \forall x \forall y (\text{študent}(x) \wedge \text{predmet}(y) \rightarrow \\ &(\text{absolvuje}(x, y) \leftrightarrow \\ &\exists z (\text{známka}(z) \wedge \text{hodnotený}(x, y) \doteq z \wedge z \neq Fx))), \\ &\forall x \forall y (\text{študent}(x) \wedge \text{št_prog}(y) \rightarrow \\ &(\text{absolvuje}(x, y) \leftrightarrow \\ &\forall z (\text{pov_predmet_prog}(z, y) \rightarrow \text{absolvuje}(x, z))))), \\ &\forall x (\text{št_prog}(x) \rightarrow \exists y \text{pov_predmet_prog}(z, x)), \\ &\forall x (\exists y \text{pov_predmet_prog}(x, y) \rightarrow \text{predmet}(x)) \} \\ X = &\forall x \forall y (\text{študent}(x) \wedge \text{št_prog}(y) \wedge \text{absolvuje}(x, y) \rightarrow \\ &\exists y \text{hodnotený}(x, y) \neq Fx) \end{aligned}$$

1. $\mathbf{T} \forall x \forall y (\text{student}(x) \wedge \text{predmet}(y) \rightarrow (\text{absolvuje}(x, y) \leftrightarrow \exists z (\text{hodnotený}(x, y) \doteq z \wedge \text{známka}(z) \wedge z \neq Fx)))$	S^+
2. $\mathbf{T} \forall x \forall y (\text{student}(x) \wedge \text{št_prog}(y) \rightarrow (\text{absolvuje}(x, y) \leftrightarrow \forall z (\text{pov_predmet_prog}(z, y) \rightarrow \text{absolvuje}(x, z))))$	S^+
3. $\mathbf{T} \forall x (\text{št_prog}(x) \rightarrow \exists y \text{ pov_predmet_prog}(z, x))$	S^+
4. $\mathbf{T} \forall x (\exists y \text{ pov_predmet_prog}(x, y) \rightarrow \text{predmet}(x))$	S^+
5. $\mathbf{F} \forall y \forall y (\text{student}(x) \wedge \text{št_prog}(y) \wedge \text{absolvuje}(x, y) \rightarrow \exists y \text{ hodnotený}(x, y) \neq Fx)$	S^+
6. $\mathbf{F} \text{student}(u) \wedge \text{št_prog}(v) \wedge \text{absolvuje}(u, v) \rightarrow \exists y \text{ hodnotený}(u, y) \neq Fx$	$\delta^*5\{x \mapsto u, y \mapsto v\}$
7. $\mathbf{T} \text{student}(u) \wedge \text{št_prog}(v) \wedge \text{absolvuje}(u, v)$	$\alpha 6$
8. $\mathbf{F} \exists y \text{ hodnotený}(u, y) \neq Fx$	$\alpha 6$
9. $\mathbf{T} \text{student}(u) \wedge \text{št_prog}(v)$	$\alpha 7$
10. $\mathbf{T} \text{absolvuje}(u, v)$	$\alpha 7$
11. $\mathbf{T} \text{student}(u) \wedge \text{št_prog}(v) \rightarrow (\text{absolvuje}(u, v) \leftrightarrow \forall z (\text{pov_predmet_prog}(z, v) \rightarrow \text{absolvuje}(u, z)))$	$\gamma^*2\{x \mapsto u, y \mapsto v\}$
12. $\mathbf{T} \text{absolvuje}(u, v) \leftrightarrow \forall z (\text{pov_predmet_prog}(z, v) \rightarrow \text{absolvuje}(u, z))$	$MP11, 9$
13. $\mathbf{T} \forall z (\text{pov_predmet_prog}(z, v) \rightarrow \text{absolvuje}(u, z))$	$ESTT12, 10$
14. $\mathbf{T} \text{št_prog}(v) \rightarrow \exists y \text{ pov_predmet_prog}(y, v)$	$\gamma 3\{x \mapsto v\}$
15. $\mathbf{T} \text{št_prog}(v)$	$\alpha 9$
16. $\mathbf{T} \exists y \text{ pov_predmet_prog}(y, v)$	$MP14, 15$
17. $\mathbf{T} \text{pov_predmet_prog}(w, v)$	$\delta^*16\{y \mapsto w\}$
18. $\mathbf{T} \text{pov_predmet_prog}(w, v) \rightarrow \text{absolvuje}(u, w)$	$\gamma 13\{z \mapsto w\}$
19. $\mathbf{T} \text{absolvuje}(u, w)$	$MP19, 17$
20. $\mathbf{T} \exists y \text{ pov_predmet_prog}(w, y) \rightarrow \text{predmet}(w)$	$\gamma 4\{x \mapsto w\}$

21. $\mathbf{F} \exists y \text{ pov_predmet_prog}(w, y)$ $\beta 20$

22. $\mathbf{F} \text{pov_predmet_prog}(w, v)$ $\gamma 21\{y \mapsto v\}$
* 17, 22

23. $\mathbf{T} \text{predmet}(w)$ $\beta 20$

24. $\mathbf{T} \text{student}(u) \wedge \text{predmet}(w) \rightarrow (\text{absolvuje}(u, w) \leftrightarrow \exists z (\text{hodnotený}(u, w) \doteq z \wedge \text{známka}(z) \wedge z \neq Fx))$ $\gamma^*1\{x \mapsto u, y \mapsto w\}$

25. $\mathbf{F} \text{student}(u) \wedge \text{predmet}(w)$ $\beta 24$

26. $\mathbf{F} \text{student}(u)$ $NCS25, 23$

27. $\mathbf{T} \text{student}(u)$ $\alpha 9$
* 26, 27

28. $\mathbf{T} \text{absolvuje}(u, w) \leftrightarrow \exists z (\text{hodnotený}(u, w) \doteq z \wedge \text{známka}(z) \wedge z \neq Fx)$ $\beta 24$

29. $\mathbf{T} \exists z (\text{hodnotený}(u, w) \doteq z \wedge \text{známka}(z) \wedge z \neq Fx)$ $ESTT28, 19$

30. $\mathbf{T} \text{hodnotený}(u, w) \doteq z \wedge \text{známka}(z) \wedge z \neq Fx$ $\delta 29\{z \mapsto z\}$

31. $\mathbf{T} \text{hodnotený}(u, w) \doteq z \wedge \text{známka}(z)$ $\alpha 30$

32. $\mathbf{T} \text{hodnotený}(u, w) \doteq z$ $\alpha 31$

33. $\mathbf{T} z \neq Fx$ $\alpha 30$

34. $\mathbf{T} \text{hodnotený}(u, w) \doteq \text{hodnotený}(u, w)$ Refl

35. $\mathbf{T} z \doteq \text{hodnotený}(u, w)$ $\text{Leib}32, 34$

36. $\mathbf{T} \text{hodnotený}(u, w) \neq Fx$ $\text{Leib}35, 33$

37. $\mathbf{F} \text{hodnotený}(u, w) \neq Fx$ $\gamma^*8\{y \mapsto w\}$
* 36, 37

15 Unifikácia termov

Dosádzanie termov za premenné

Pri kvantifikovaných formulách s funkčnými symbolmi môže byť ťažké povedať, aké termy dosádzať za všeobecne kvantifikované premenné.

Čo možno usúdiť z nasledujúcich dvoch formúl?

$$\begin{aligned}\forall y \quad & P(f(y), y) \\ \forall x(\neg P(x, d) \vee R(x))\end{aligned}$$

Ak by sme vhodným dosadením termov dosiahli totožnosť $f(y)$ s x a y s d , možno usúdiť $R(x)$.

Dosádzanie termov za premenné

$$\begin{aligned}\forall y \quad & P(f(y), y) \\ \forall x(\neg P(x, d) \vee R(x))\end{aligned}$$

Dosadenie popisujeme pomocou substitúcie. V našom prípade zjavne za y musíme substituovať d a za x ...

$$\sigma = \{x \mapsto f(d), y \mapsto d\}$$

Po substitúcii σ majú komplementárne literály rovnaké argumenty predikátu (preto σ nazývame *unifikátor*):

$$\begin{aligned}P(f(y), y)\sigma &= P(f(d), d) \\ \neg P(x, d)\sigma &= \neg P(f(d), d)\end{aligned}$$

Jedným z dôsledkov uvedených dvoch formúl je teda $R(f(d))$. (Aké iné dôsledky z uvedených formúl vyplývajú?)

Unifikátory

Definícia 15.1. Nech A, B sú postupnosti symbolov, σ je substitúcia. Substitúcia σ je *unifikátorom* A a B vtt $A\sigma = B\sigma$.

Príklad 15.2.

- $A_1 = R(\text{filantrop}, y), B_1 = R(x, d),$
 $\sigma_1 = \{x \mapsto \text{filantrop}, y \mapsto d\}$
- $A_2 = R(\text{nk}(y), y), B_2 = R(x, d),$
 $\sigma_2 = \{x \mapsto \text{nk}(d), y \mapsto d\}$
- $A_3 = R(\text{nk}(y), y), B_3 = R(e, x), \quad \sigma_3 = ??? \text{ neexistuje!}$
- $A_4 = R(\text{nk}(y), y), B_4 = R(x, x), \quad \sigma_4 = ??? \text{ neexistuje!}$
- $A_5 = R(f(y)), B_5 = R(x),$
 $\sigma_5 = \{x \mapsto f(d), y \mapsto d\} / \{x \mapsto f(f(d)), y \mapsto f(d)\} / \dots$

Skladanie substitúcií

Definícia 15.3. Nech $\sigma = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$ a $\theta = \{y_1 \mapsto s_1, \dots, y_m \mapsto s_m\}$ sú substitúcie. *Zložením (kompozíciou) substitúcií σ a θ* je substitúcia $\sigma\theta = \{x_1 \mapsto t_1\theta, \dots, x_n \mapsto t_n\theta, y_{i_1} \mapsto s_{i_1}, \dots, y_{i_k} \mapsto s_{i_k}\}$, kde $\{y_{i_1}, \dots, y_{i_k}\} = \{y_1, \dots, y_m\} \setminus \{x_1, \dots, x_n\}$.

Príklad 15.4.

$$\begin{aligned}\sigma &= \{x \mapsto \text{nk}(y), z \mapsto y\} \\ \theta &= \{y \mapsto d\} \\ \sigma\theta &= \{x \mapsto \text{nk}(d), \\ &\quad z \mapsto d, y \mapsto d\}\end{aligned}$$

Je pravda, že pre ľubovoľné substitúcie α, β, γ platí $(\alpha\beta)\gamma = \alpha(\beta\gamma)$?

Unifikátory

Definícia 15.5. Nech A, B sú postupnosti symbolov, σ a θ sú substitúcie.

σ je *všeobecnejšia* ako θ vtt existuje subst. γ taká, že $\theta = \sigma\gamma$.

σ je *najvšeobecnejším unifikátorom* A a B vtt

- σ je unifikátorom A a B a zároveň
- pre každý unifikátor θ A a B je σ všeobecnejšia ako θ .

Príklad 15.6. $A = R(\text{nk}(x), y), B = R(u, v)$

- $\sigma_1 = \{u \mapsto nk(d), v \mapsto y, x \mapsto d\}$ $\theta_1 = \{u \mapsto nk(d), v \mapsto Biba, x \mapsto d, y \mapsto Biba\}$ $\gamma_1 = \{y \mapsto Biba\}$
- $\sigma_2 = \{u \mapsto nk(x), v \mapsto y\}$ $\theta_2 = \{u \mapsto nk(d), v \mapsto y, x \mapsto d\}$ $\gamma_2 = \{x \mapsto d\}$

Unifikácia

Unifikácia má mnohoraké využitie:

- rezolvenca v prvorádovej logike
- inferencia typov kompilátormi (typy sú vlastne termy)
- niektoré druhy parserov (o. i. pattern matching)
- spracovanie prirodzeného jazyka (Prolog)
- deduktívne databázy
- expertné systémy, automatizované usudzovanie

Ukážeme si základný algoritmus na hľadanie najvšeobecnejšieho unifikátora (z r. 1965). <http://web.stanford.edu/class/linguist289/robinson65.pdf> <https://eli.thegreenplace.net/2018/unification/> <https://github.com/eliben/code-for-blog/blob/master/2018/unif/unifer.py>

Unifikácia: typy

```
class Term:
    pass

class Const(Term):
    """Constant"""
    def __init__(self, name):
        self.name = name

class Var(Term):
    """Variable"""
    def __init__(self, name):
        self.name = name

class App(Term):
```



```

"""Application of a function symbol"""
def __init__(self, fname, args=()):
    self.fname = fname
    self.args = args

```

```
Subst = dict[Var, Term]
```

```
# __eq__ and __hash__ omitted
```

Unifikácia: unify

```

def unify(s: Term, t: Term, sigma: Subst | None) -> Subst | None:
    """Unifies terms s and t, given an initial substitution.
    'None' means that a substitution does not exist."""
    if sigma is None:
        return None
    elif s == t:
        return sigma
    elif isinstance(s, Var):
        return unify_variable(s, t, sigma)
    elif isinstance(t, Var):
        return unify_variable(t, s, sigma)
    elif isinstance(s, App) and isinstance(t, App):
        if s.fname != t.fname:
            return None
        else:
            for i in range(len(s.args)):
                sigma = unify(s.args[i], t.args[i], sigma)
            return sigma
    else:
        # includes the case where s, t are different constants
        return None

```

Unifikácia: unify_variable

```

def unify_variable(x: Var, t: Term, sigma: Subst) -> Subst | None:
    """Unifies variable x with term t, using sigma.

    Returns updated sigma or None if impossible.
    """
    if x in sigma:
        return unify(sigma[x], t, sigma)

```

```

elif isinstance(t, Var) and t in sigma:
    return unify(x, sigma[t], sigma)
elif occurs_check(x, t, sigma):
    return None
else:
    # x is not yet in sigma and can't simplify t. Extend sigma.
    return {**sigma, x: t}

```

Unifikácia: occurs_check

```

def occurs_check(v: Var, t: Term, sigma: Subst) -> bool:
    """Does the variable v occur anywhere inside t?

    Variables in t are looked up in sigma and the check is applied
    recursively.
    """
    if v == t:
        return True
    elif isinstance(t, Var) and t in sigma:
        return occurs_check(v, sigma[t], sigma)
    elif isinstance(t, App):
        return any(occurs_check(v, arg, sigma) for arg in t.args)
    else:
        return False

```

Unifikácia

Korektný algoritmus: skončí a dá správny výsledok.

- Vďaka occurs_check algoritmus nikdy za premennú nedosadí term, ktorý ju obsahuje.
- Ak sme raz za premennú niečo dosadili, nedosadíme za ňu nič iné, a pri jej unifikovaní vždy použijeme existujúce dosadenie.
- unify_variable znižuje počet premenných. (Môžeme si predstaviť, že všetky termy pri každom dosadení prepíšeme už bez premennej, za ktorú sme dosadzovali.)
- unify zjednodušuje termy (postupne ubúdajú funkčné symboly).
- Algoritmus je preto konečný a nájde nejaký unifikátor (ak existuje).

Unifikácia

Nájdený unifikátor je najvšeobecnejší kvôli tomu, že algoritmus rozširuje substitúciu len vtedy, keď musí, a najvšeobecnejšie, ako sa dá (nepridáva zbytočné funkčné symboly).

(Toto by si zaslúžilo podrobný dôkaz, o. i. pretože najvšeobecnejší unifikátor nie je celkom jednoznačný — hoci ak ich existuje viac, líšiť sa môžu len označením premenných. Na tomto predmete ho však robiť nebudeme.)

Zaujímavosť: v r. 1991 bola objavená chyba v 7 rôznych seriózných knihách prezentujúcich tento algoritmus¹.

Unifikácia

Názorná predstava, ako unifikácia prebieha: máme sústavu rovností termov, ktorú upravujeme a postupne rozširujeme substitúciu o dosadenia za nové premenné. Povolené úpravy:

- Miesto rovnice, ktorá porovnáva dva rovnaké funkčné symboly s aritou k , zapíš k rovníc pre rovnosť jednotlivých argumentov.
- Ak je na niektorej strane rovnice osamotená premenná, dosad' za ňu term predpísaný rovnicou a prepíš všetky výskyty tejto premennej.
- Zmaž triviálne splnenú rovnicu.

Každá z operácií niečo znižuje (počet funkčných symbolov na ľavej strane, počet premenných, počet rovníc).

Unifikácia

¹<http://norvig.com/unify-bug.pdf>

Príklad 15.7 (Úspešný beh unifikačného algoritmu).

$$\begin{array}{c}
 f(X, h(X), Y, g(Y)) = f(g(Z), W, Z, X) \\
 \hline
 X = g(Z) \quad \{X \mapsto g(Z)\} \\
 h(X) = W \\
 Y = Z \\
 g(Y) = X \\
 \hline
 h(g(Z)) = W \quad \{W \mapsto h(g(Z))\} \\
 Y = Z \\
 g(Y) = g(Z) \\
 \hline
 Y = Z \quad \{Y \mapsto Z\} \\
 g(Y) = g(Z) \\
 \hline
 g(Z) = g(Z)
 \end{array}$$

Unifikácia

Uvedený algoritmus nie je veľmi efektívny (napr. ho `occurs_check` spomaľuje natoľko, že sa v niektorých implementáciách vynecháva²). Existujú teoreticky lepšie algoritmy (zhruba lineárne), ale tie zase na mnohých praktických vstupoch bežia prídlho, preto sa veľmi nepoužívajú.

Poznámka: Pre účely tohto predmetu je najdôležitejšie, aby ste plne rozumeli, o čo pri unifikácii ide, a vedeli nájsť najvšeobecnejší unifikátor v konkrétnom prípade. Úplná znalosť všeobecného algoritmu či zdôvodnenie jeho vlastností sú menej podstatné.

²https://en.wikipedia.org/wiki/Occurs_check

11. prednáška

Rezolvencia

16 Rezolvencia

Automatické dokazovanie v logike prvého rádu

Vyplývanie vo výrokovej logike je rozhodnuteľné.

SAT solver vždy skončí a rozhodne splniteľnosť, v najhoršom prípade v čase $O(2^n \cdot f)$ pre n atómov a formulu dĺžky f .

Logika prvého rádu *nie je rozhodnuteľná* (ak by bola, vedeli by sme riešiť problém zastavenia — viac nabadúce).

Vďaka tomu, že je úplná, však ku každému pravdivému tvrdeniu (vyplývaniu formuly z teórie) existuje dôkaz. Možno preto postupne enumerovať všetky dôkazy, až kým nenájdeme vyhovujúci. Problém vyplývania v prvorádovej logike je teda *čiastočne rozhodnuteľný*.

Dokazovací systém má podstatný vplyv na to, ako dlho v praxi potrvá nájdenie dôkazu (a či nám vystačí dostupná pamäť).

Ako fungujú automatické dokazovače v logike prvého rádu

Prvé automatické dokazovače využívali prvorádovú verziu DPLL.

Niektoré automatické dokazovače využívajú modifikované tablá.

Väčšina automatických dokazovačov (napr. Prover9 a Vampire) je ale založená na *rezolvencii*:

- špeciálne pravidlo na klauzulách,
- kombinuje výrokové a kvantifikátorové odvodzovanie.

Rezolvenčný dôkaz je lineárny, nevetví sa.

16.1 Rezolvencia vo výrokovej logike

Tranzitivita implikácie

Vráťme sa k neoznačeným formulám.

Je nasledujúce pravidlo korektné?

$$\frac{(A \rightarrow B) \quad (B \rightarrow C)}{(A \rightarrow C)}$$

Nahrad'me implikácie disjunkciami:

$$\frac{(\neg A \vee B) \quad (\neg B \vee C)}{(\neg A \vee C)}$$

Rezolvenca

Predchádzajúce pravidlo sa dá zovšeobecniť na ľub. dvojicu klauzúl:

Definícia 16.1. *Rezolvenčný princíp (rezolvenca, angl. resolution principle)* je pravidlo

$$\frac{(K_1 \vee \dots \vee A \vee \dots \vee K_m) \quad (L_1 \vee \dots \vee \neg A \vee \dots \vee L_n)}{(K_1 \vee \dots \vee K_m \vee L_1 \vee \dots \vee L_n)}$$

pre ľubovoľný atóm A a ľub. literály $K_1, \dots, K_m, L_1, \dots, L_n$.

Klauzulu $(K_1 \vee \dots \vee K_m \vee L_1 \vee \dots \vee L_n)$ nazývame *rezolventou* klauzúl $(K_1 \vee \dots \vee A \vee \dots \vee K_m)$ a $(L_1 \vee \dots \vee \neg A \vee \dots \vee L_n)$.

Tvrdenie 16.2. *Rezolvenca je korektné pravidlo. (Rezolventa je pravdivá v každom ohodnotení, v ktorom sú pravdivé pôvodné klauzuly.)*

Špeciálne prípady rezolvenacie

Viacero pravidiel sa dá chápať ako špeciálne prípady rezolvenacie:

$$\frac{(\neg A \vee B) \quad (\neg B \vee C)}{(\neg A \vee C)} \qquad \frac{(A \rightarrow B) \quad (B \rightarrow C)}{(A \rightarrow C)} \qquad \text{(HS)}$$

$$\frac{(\neg A \vee B) \quad A}{B} \qquad \frac{(A \rightarrow B) \quad A}{B} \qquad \text{(MP)}$$

$$\frac{(\neg A \vee B) \quad \neg B}{\neg A} \qquad \frac{(A \rightarrow B) \quad \neg B}{\neg A} \qquad \text{(MT)}$$

Pozorovania o rezolvencii

- Rezolvenca s *jednotkovou* klauzulou skrúti druhú klauzulu:

$$\frac{\neg B \quad (A \vee B \vee \neg C)}{(A \vee \neg C)}$$

- Rezolvenca môže odvodiť *prázdnu klauzulu*:

$$\frac{\neg A \quad A}{\square},$$

vtedy premisy *nie sú súčasne splniteľné*

- Nie každý logický dôsledok sa dá odvodiť rezolvenciou: $\{A, B\} \models (A \vee B)$

Častá chyba pri rezolvencii

Niektoré dvojice klauzúl možno rezolvovať na viacerých literáloch:

$$\frac{(\neg p \vee q) \quad (p \vee \neg q)}{(q \vee \neg q)} \quad \checkmark \qquad \frac{(\neg p \vee q) \quad (p \vee \neg q)}{(\neg p \vee p)} \quad \checkmark$$

ale je *chyba urobiť to naraz*:

~~$$\frac{(\neg p \vee q) \quad (p \vee \neg q)}{\square} \quad \times$$~~

Toto *nie je* inštancia rezolvenacie ani korektný úsudok.

Prečo?

Lebo $\{(\neg p \vee q), (p \vee \neg q)\}$ je ekvivalentná $(p \leftrightarrow q)$ a je splniteľná ($v_1 = \{p \mapsto t, q \mapsto t\}, v_2 = \{p \mapsto f, q \mapsto f\}$), ale \square je nesplniteľná.

Rezolvenčné odvodenie a problém

Opakovaním rezolvenacie môžeme odvodzovať ďalšie dôsledky:

Príklad 16.3. Z množiny $S = \{(A \vee B), (\neg A \vee C), (\neg B \vee A), (\neg A \vee \neg C)\}$ odvodíme:

- (1) $(A \vee B)$ predpoklad z S
 - (2) $(\neg A \vee C)$ predpoklad z S
 - (3) $(\neg B \vee A)$ predpoklad z S
 - (4) $(\neg A \vee \neg C)$ predpoklad z S
 - (5) $(A \vee A)$ rezolventa (3) a (1)
 - (6) $(B \vee C)$ rezolventa (1) a (2)
 - (7) $(B \vee \neg C)$ rezolventa (1) a (4)
 - (8) $(B \vee B)$ rezolventa (6) a (7)
- ⋮

Problematické prípady

Odvodeniami v príklade dostaneme iba existujúce alebo nové dvojprvkové klauzuly ($(A \vee A)$, $(B \vee C)$, $(B \vee B)$, ...) ale žiadnu jednotkovú, lebo rezolventa má $m + n - 2$ literálov.

$S = \{(A \vee B), (\neg A \vee C), (\neg B \vee A), (\neg A \vee \neg C)\}$ je ale nespĺniteľná, mali by sme nejako odvodiť prázdnu klauzulu.

To sa nedá bez odvodu nejakej jednotkovej klauzuly (napr. A).

Klauzula $(A \vee A)$ je evidentne ekvivalentná s A ; A sa ale z množiny S iba rezolveniou odvodíť nedá.

Potrebujeme ešte *pravidlo idempotencie*:

$$\frac{(K_1 \vee \dots \vee L \vee \dots \vee L \vee \dots \vee K_n)}{(K_1 \vee L \vee \dots \vee K_n)}$$

Rezolvenčné odvodenie a zamietnutie

Definícia 16.4. *Výrokovologické rezolvenčné odvodenie* z množiny klauzúl S je každá (aj nekonečná) postupnosť klauzúl $C_1, C_2, \dots, C_n, \dots$, ktorej každý člen C_i je:

- prvkom S alebo

- rezolventou dvoch predchádzajúcich klauzúl C_j a C_k pre $j < i$ a $k < i$, alebo
- záverom pravidla idempotencie pre nejakú predchádzajúcu klauzulu C_j , $j < i$.

Zamietnutím (angl. *refutation*) množiny klauzúl S je *konečné* rezolvenčné odvodenie, ktorého posledným prvkom je prázdna klauzula \square .

Rezolvenčné zamietnutie

Príklad 16.5. Nech $S = \{(A \vee B), (\neg A \vee C), (\neg B \vee A), (\neg A \vee \neg C)\}$.

Kombináciou rezolvenčie a idempotencie nájdeme zamietnutie S :

- (1) $(A \vee B)$ predpoklad z S
- (2) $(\neg A \vee C)$ predpoklad z S
- (3) $(\neg B \vee A)$ predpoklad z S
- (4) $(\neg A \vee \neg C)$ predpoklad z S
- (5) $(A \vee A)$ rezolventa (3) a (1)
- (6) A idempotencia (5)
- (7) C rezolvenčie (6) a (2)
- (8) $\neg C$ rezolvenčie (6) a (4)
- (9) \square rezolvenčie (7) a (8)

Rezolvenčné zamietnutie

Množine klauzúl budeme hovoriť aj *klauzálna teória*.

Tvrdenie 16.6. Ak pre klauzálnu teóriu S existuje zamietnutie, je nespĺniteľná.

(Ak by nejaké ohodnotenie bolo modelom S , bolo by vďaka korektnosti pravidla rezolvencie modelom každej odvodenej klauzuly, vrátane nesplniteľnej prázdnej.)

Vyskúšajte si 16.1

Dokážte nesplniteľnosť $S = \{(A \vee B \vee \neg C), (\neg A \vee \neg C), (A \vee \neg B), (\neg A \vee C), (A \vee B \vee C)\}$.

Rezolvenca a SAT

Možno pomocou rezolvencie znížiť počet atómov?

$$\frac{(\neg B \vee D) \quad (A \vee B \vee \neg C)}{(A \vee \neg C \vee D)}$$

Preskúmame nasledovný postup na hľadanie spĺňajúceho ohodnotenia:

- Ak v nejakej klauzule je A , v inej $\neg A$, spravíme na nich rezolvenciu. Ak odvodíme \square , vstupná formula je nesplniteľná.
- Ak už také dvojice nie sú, tak A alebo $\neg A$ je nezmiešaný literál, a preto vieme, ako A ohodnotiť. Takto sme sa úplne zbavili atómu A .
- Toto zopakujeme postupne s ďalšími atómami, až kým nenájdem spĺňajúce ohodnotenie.

Je tento postup polynomiálnym algoritmom pre SAT?

Rezolvenca a SAT

Ak uvedený postup vedie k zamietnutiu, ohodnotenie neexistuje. Ohodnotenie nájdené po eliminácii atómu popísaným spôsobom však nemusí vyhovovať pôvodným klauzulám!

$$\frac{(A \vee B) \quad (\neg A \vee C) \quad (\neg A \vee D) \quad \neg B \quad C}{(B \vee C) \quad (\neg A \vee D) \quad \neg B \quad C}$$

Spodné klauzuly sú pravdivé pri ohodnotení $\{A \mapsto f, B \mapsto f, C \mapsto t\}$, kým vrchné nie.

Postup sa však dá upraviť, aby fungoval. Miesto rezolvencie jednej dvojice klauzúl použijeme rezolvenciu *súčasne pre všetky možné dvojice* obsahujúce komplementárne literály s atómom A .

Rezolvenca a SAT

Nahradíme klauzuly S_1 obsahujúce A klauzulami S_2 (X_i, Y_j sú disjunkcie literálov neobsahujúcich A):

$$S_1 = \left\{ \begin{array}{ll} A \vee X_1 & \neg A \vee Y_1 \\ A \vee X_2 & \neg A \vee Y_2 \\ \vdots & \vdots \\ A \vee X_n & \neg A \vee Y_m \end{array} \right\} \quad S_2 = \left\{ \begin{array}{lll} X_1 \vee Y_1 & \dots & X_n \vee Y_1 \\ \vdots & & \vdots \\ X_1 \vee Y_m & \dots & X_n \vee Y_m \end{array} \right\}$$

Nech T je množina klauzúl, ktoré neobsahujú A . Predpokladajme, že pre nejaké ohodnotenie v_2 platí $v_2 \models_p S_2 \cup T$. Nájdeme v_1 také, že $v_1 \models_p S_1 \cup T$:

- Ak $v_2 \not\models_p X_i$ pre nejaké i , tak z $v_2 \models_p X_i \vee Y_j$ vyplýva $v_2 \models_p Y_j$ pre každé j . Vtedy stačí zvoliť $v_1 = v_2 \cup \{A \mapsto t\}$.
- Ak pre každé i platí $v_2 \models_p X_i$, zvolíme $v_1 = v_2 \cup \{A \mapsto f\}$.

Týmto nepokazíme splnenie klauzúl v T , lebo neobsahujú A .

Rezolvenca a SAT

Naopak, ak ohodnotenie v_1 je modelom $S_1 \cup T$, tak $v_1 \models_p S_2 \cup T$: Ak $v_1(A) = t$, tak $v_1 \models_p Y_j$ pre všetky j , preto $v_1 \models_p S_2$. Podobne pre $v_1(A) = f$.

Takto sme naozaj znížili počet atómov; podobné postupy sa využívajú pri predspracovaní vstupu pre SAT. (Čo sa stane s veľkosťou klauzúl?)

Počet pridaných klauzúl však môže narásť exponenciálne, preto sme polynomiálny algoritmus pre SAT nezískali.

Úplnosť rezolvenca

Využitím uvedeného postupu vieme dokázať úplnosť rezolvenca.

Tvrdenie 16.7 (Úplnosť rezolvenca). *Ak je klauzálna teória S nesplniteľná, existuje jej zamietnutie.*

Dôkaz. Uvažujme nesplniteľnú klauzálnu teóriu a rozdeľme jej klauzuly na dve množiny: v S_1 budú tie, čo obsahujú atóm A , v T ostatné. Každú klauzulu z S_2 vieme odvodiť z S_1 pomocou pravidla pre rezolvenca. Ako

sme ukázali, množina $T \cup S_1$ je nesplniteľná vtt $T \cup S_2$ je nesplniteľná. Zároveň $T \cup S_2$ má o jeden atóm menej. Opakovaním postupu nájdeme nesplniteľnú množinu klauzúl, ktorá má už len nezmiešané literály. Preto v nej musí byť aj \square . \square

(Kde v dôkaze využívame idempotenciu?)

Rezolvenca vo výrokovovej logike

Pomocou rezolvenca vieme rozhodovať splniteľnosť.

Veta 16.8 (Korektnosť a úplnosť rezolvenca). *Nech S je klauzálna teória. S je výrokovologicky nesplniteľná vtt existuje zamietnutie S .*

Pomocou rezolvenca možno rozhodovať aj výrokovologické vyplývanie formuly X z teórie T : vieme, že $T \models X$ vtt $T \cup \{\neg X\}$ je nesplniteľná. Aby sme mohli použiť rezolvenca, ostáva previesť všetky formuly z T aj $\neg X$ do CNF (čo sa vždy dá).

16.2 Prevod do klauzálnej teórie a skolemizácia

Rezolvenca vs. prvorádové teórie

Výrokovologická rezolvenca pracuje s klauzálnymi teóriami.

Výrokovologickú teóriu ľahko upravíme na klauzálnu – ekvivalentnými úpravami do CNF.

Ale čo s formulami v logike prvého rádu, kde sú spojky zložito skombinované s kvantifikátormi?

Prvorádové klauzuly a klauzálne teórie

Ujasnime si najprv, aký tvar chceme dosiahnuť.

Definícia 16.9. Nech \mathcal{L} je jazyk logiky prvého rádu.

Literál je atomická formula $P(t_1, \dots, t_m)$ jazyka \mathcal{L} alebo jej negácia $\neg P(t_1, \dots, t_m)$.

Klauzula je všeobecný uzáver disjunkcie literálov, teda uzavretá formula jazyka \mathcal{L} v tvare $\forall x_1 \cdots \forall x_k (L_1 \vee \cdots \vee L_n)$ kde L_1, \dots, L_n sú literály a x_1, \dots, x_k sú všetky voľné premenné formuly $L_1 \vee \cdots \vee L_n$. Klauzula môže byť aj *jednotková* ($\forall \vec{x} L_1$) alebo *prázdna* (\square).

Klauzálna teória je množina klauzúl $\{C_1, \dots, C_n\}$. Môže byť tvorená aj jedinou klauzulou alebo byť prázdna.

Prvorádová ekvivalencia

Postupovať budeme podobne ako vo výrokovologickej príhode: Postupne odstránime z teórie implikácie, negácie zložených formúl, *existenčné kvantifikátory*, disjunkcie konjunkcií, vnorené všeobecné kvantifikátory.

Podľa možnosti budeme používať ekvivalentné úpravy v prvorádovom zmysle:

Definícia 16.10 (Prvorádová ekvivalencia). Množiny formúl S a T sú (*prvorádovo*) ekvivalentné ($S \Leftrightarrow T$) vtt pre každú štruktúru \mathcal{M} a každé ohodnotenie e platí $\mathcal{M} \models S[e]$ vtt $\mathcal{M} \models T[e]$.

Tvrdenie 16.11 (Ekvivalentná úprava). *Nech X, A, B sú formuly a nech $\text{free}(A) = \text{free}(B)$. Ak $A \Leftrightarrow B$, tak $X \Leftrightarrow X[A \mid B]$.*

Nahradenie implikácií

Rovnako ako vo výrokovej logike môžeme každú formulu $(A \rightarrow B)$ ekvivalentne nahradiť formulou $(\neg A \vee B)$.

Príklad 16.12.

$$\begin{aligned} & \forall x(\text{dobré}(x) \wedge \text{dieťa}(x) \rightarrow \exists y(\text{dostane}(x, y) \wedge \text{darček}(y))) \\ \Leftrightarrow & \forall x(\neg(\text{dobré}(x) \wedge \text{dieťa}(x)) \vee \exists y(\text{dostane}(x, y) \wedge \text{darček}(y))) \\ & \forall x(\neg \text{dobré}(x) \rightarrow \neg \exists y \text{ dostane}(x, y)) \\ \Leftrightarrow & \forall x(\neg \neg \text{dobré}(x) \vee \neg \exists y \text{ dostane}(x, y)) \end{aligned}$$

Konverzia do negačného normálneho tvaru (NNF)

Definícia 16.13. Formula X je v *negačnom normálnom tvare* (NNF) vtt neobsahuje implikáciu a pre každú jej podformulu $\neg A$ platí, že A je atomická formula.

Formulu bez implikácií do NNF upravíme pomocou

- de Morganových zákonov pre spojky:

$$\neg(A \wedge B) \Leftrightarrow \neg A \vee \neg B \qquad \neg(A \vee B) \Leftrightarrow \neg A \wedge \neg B$$

- pravidla dvojitej negácie:

$$\neg\neg A \Leftrightarrow A$$

- zovšeobecnení de Morganových zákonov pre kvantifikátory:

$$\neg \exists x A \Leftrightarrow \forall x \neg A \qquad \neg \forall x A \Leftrightarrow \exists x \neg A$$

Konverzia do NNF

Tvrdenie 16.14. Pre každú formulu X existuje formula Y v NNF taká, že $X \Leftrightarrow Y$.

Príklad 16.15.

$$\begin{aligned} & \forall x(\neg(\text{dobré}(x) \wedge \text{dieťa}(x)) \vee \exists y(\text{dostane}(x, y) \wedge \text{darček}(y))) \\ \Leftrightarrow & \forall x((\neg \text{dobré}(x) \vee \neg \text{dieťa}(x)) \vee \exists y(\text{dostane}(x, y) \wedge \text{darček}(y))) \\ & \forall x(\neg \neg \text{dobré}(x) \vee \neg \exists y \text{ dostane}(x, y)) \\ \Leftrightarrow & \forall x(\text{dobré}(x) \vee \forall y \neg \text{dostane}(x, y)) \end{aligned}$$

Skolemizácia

Skolemizácia (podľa nórskeho logika Thoralfu Skolema) je úprava formuly X v NNF, ktorou nahradíme existenčné kvantifikátory novými konštantami alebo funkčnými symbolmi.

Podobá sa pravidlu δ v tabľách, ale aplikuje sa naraz na všetky existenčné kvantifikátory.

Výsledná formula je v novom, rozšírenom jazyku.

Nie je ekvivalentná s pôvodnou, ale je *ekvisplnitelná*.

Definícia 16.16 (Prvorádová ekvisplnitelnosť). Množiny formúl S a T sú (prvorádovo) rovnako splniteľné (ekvisplnitelné, equisatisfiable) vtt S má model práve vtedy, keď T má model.

Skolemizácia — skolemovská konštanta

Lahký prípad (v podstate pravidlo δ):

Vo formule X sa vyskytuje $\exists y A$ mimo všetkých oblastí platnosti všeobecných kvantifikátorov.

1. Pridáme do jazyka novú *skolemovskú konštantu* c (symbol c doteraz nebol v jazyku v žiadnej úlohe).
2. Každý výskyt podformuly $\exists y A$ v X mimo všetkých oblastí platnosti všeobecných kvantifikátorov nahradíme formulou

$$A\{y \mapsto c\}$$

Konštanta c pomenúva objekt, ktorý existuje podľa $\exists y A$.

Príklad 16.17.

$$\exists x (\text{dobré}(x) \wedge \text{dieťa}(x))$$

$$\rightsquigarrow \text{dobré}(\text{nejaké_dobré_dieťa}) \wedge \text{dieťa}(\text{nejaké_dobré_dieťa})$$

Skolemizácia — skolemovská funkcia

Vo formule X sa vyskytuje $\exists y A$ v oblasti platnosti všeobecných kvantifikátorov premenných x_1, \dots, x_n :

$$X = \dots \forall x_1 (\dots \forall x_2 (\dots \forall x_n (\dots \exists y A \dots) \dots) \dots) \dots$$

1. Pridáme do jazyka nový funkčný symbol, *skolemovskú funkciu* f .
2. Každý výskyt $\exists y A$ v X v oblasti platnosti kvantifikátorov $\forall x_1, \dots, \forall x_n$ nahradíme formulou

$$A\{y \mapsto f(x_1, x_2, \dots, x_n)\}$$

Funkcia f pomenúva priradenie objektu y objektom x_1, \dots, x_n .

Príklad 16.18.

$$\forall x (\neg \text{dobré}(x) \vee \neg \text{dieťa}(x) \vee \exists y (\text{dostane}(x, y) \wedge \text{darček}(y)))$$

$$\rightsquigarrow \forall x (\neg \text{dobré}(x) \vee \neg \text{dieťa}(x) \vee (\text{dostane}(x, \text{darček_pre}(x)) \wedge \text{darček}(\text{darček_pre}(x))))$$

Skolemizácia

Tvrdenie 16.19. Pre každú uzavretú formulu X v jazyku \mathcal{L} existuje formula Y vo vhodnom rozšírení \mathcal{L}' jazyka \mathcal{L} taká, že Y neobsahuje existenčné kvantifikátory a X a Y sú ekvivalentné.

Príklad 16.20.

$$\begin{aligned} & \exists z \left(R(z, z) \wedge \forall x (\neg R(x, z) \vee \exists u (R(x, u) \wedge R(u, z)) \right. \\ & \quad \vee \forall y \exists v (\neg R(y, v) \wedge R(x, v)) \\ & \quad \left. \vee \exists v \forall w (R(x, v) \wedge R(v, w)) \right) \\ \rightsquigarrow & R(c, c) \wedge \forall x (\neg R(x, c) \vee (R(x, f_1(x)) \wedge R(f_1(x), c)) \\ & \quad \vee \forall y (\neg R(y, f_2(x, y)) \wedge R(x, f_2(x, y))) \\ & \quad \vee \forall w (R(x, f_3(x)) \wedge R(f_3(x), w))) \end{aligned}$$

Konverzia do PNF

Definícia 16.21. Formula X je v *prenexnom normálnom tvare* (PNF) vtt má tvar $Q_1 x_1 Q_2 x_2 \cdots Q_n x_n A$, kde $Q_i \in \{\forall, \exists\}$, x_i je premenná a A je formula bez kvantifikátorov (*matica* formuly X).

Skolemizovanú formulu v NNF upravíme do PNF opakovanou aplikáciou nasledujúcich transformácií:

- ak x nemá voľný výskyt v B ,

$$\begin{aligned} (\forall x A \wedge B) &\Leftrightarrow \forall x (A \wedge B) & (B \wedge \forall x A) &\Leftrightarrow \forall x (B \wedge A) \\ (\forall x A \vee B) &\Leftrightarrow \forall x (A \vee B) & (B \vee \forall x A) &\Leftrightarrow \forall x (B \vee A) \end{aligned}$$

- ak x má voľný výskyt v B a y je nová premenná,

$$\begin{aligned} (\forall x A \wedge B) &\Leftrightarrow (\forall y A\{x \mapsto y\} \wedge B) & (B \wedge \forall x A) &\Leftrightarrow (B \wedge \forall y A\{x \mapsto y\}) \\ (\forall x A \vee B) &\Leftrightarrow (\forall y A\{x \mapsto y\} \vee B) & (B \vee \forall x A) &\Leftrightarrow (B \vee \forall y A\{x \mapsto y\}) \end{aligned}$$

Konverzia do PNF

Tvrdenie 16.22. Pre každú formulu X v NNF bez existenčných kvantifikátorov existuje ekvivalentná formula Y v PNF a NNF.

Príklad 16.23.

$$\begin{aligned} & \forall x(\text{dobré}(x) \vee \forall y \neg \text{dostane}(x, y)) \\ \Leftrightarrow & \forall x \forall y (\text{dobré}(x) \vee \neg \text{dostane}(x, y)) \end{aligned}$$

Pozor! Pre ekvivalentnosť prenexovania je nutné, aby boli premenné viazané rôznymi kvantifikátormi rôzne:

$$\begin{aligned} & (\forall x A(x) \vee \forall x B(x)) \not\equiv \forall x (A(x) \vee B(x)) \quad \text{✗} \\ & (\forall x A(x) \vee \forall x B(x)) \Leftrightarrow \forall x (A(x) \vee \forall x B(x)) \Leftrightarrow \\ & \forall x (A(x) \vee \forall y B(y)) \Leftrightarrow \forall x \forall y (A(x) \vee B(y)) \quad \text{✓} \end{aligned}$$

Prenexujte *po jednom* alebo *premenujte* premenné (ešte pred skolemizáciou)

Konverzia do CNF

Maticu (najväčšiu podformulu bez kvantifikátorov) formuly v PNF upravíme do CNF pomocou distributívnosti a komutatívnosti disjunktie:

$$\begin{aligned} (A \vee (X \wedge Y)) & \Leftrightarrow ((A \vee X) \wedge (A \vee Y)) \\ ((X \wedge Y) \vee A) & \Leftrightarrow ((X \vee A) \wedge (Y \vee A)) \end{aligned}$$

Príklad 16.24.

$$\begin{aligned} & \forall x(\neg \text{dobré}(x) \vee \neg \text{dieťa}(x) \vee \\ & \quad (\text{dostane}(x, \text{darček_pre}(x)) \wedge \text{darček}(\text{darček_pre}(x)))) \\ \Leftrightarrow & \forall x((\neg \text{dobré}(x) \vee \neg \text{dieťa}(x) \vee \text{dostane}(x, \text{darček_pre}(x))) \wedge \\ & \quad (\neg \text{dobré}(x) \vee \neg \text{dieťa}(x) \vee \text{darček}(\text{darček_pre}(x)))) \end{aligned}$$

Konverzia do klauzálnej teórie

Formula, ktorej matica je v CNF, je ekvivalentná s konjunkciou klauzúl:

$$\forall x(A \wedge B) \Leftrightarrow (\forall x A \wedge \forall x B)$$

a konjunkcia klauzúl je ekvivalentná s ich množinou:

$$\{(\forall x A \wedge \forall x B)\} \Leftrightarrow \{\forall x A, \forall x B\}$$

Príklad 16.25.

$$\begin{aligned} & \{ \forall x((\neg \text{dobré}(x) \vee \neg \text{dieťa}(x) \vee \text{dostane}(x, \text{darček_pre}(x))) \wedge \\ & \quad (\neg \text{dobré}(x) \vee \neg \text{dieťa}(x) \vee \text{darček}(\text{darček_pre}(x)))) \} \\ \Leftrightarrow & \{ (\forall x(\neg \text{dobré}(x) \vee \neg \text{dieťa}(x) \vee \text{dostane}(x, \text{darček_pre}(x))) \wedge \\ & \quad \forall x(\neg \text{dobré}(x) \vee \neg \text{dieťa}(x) \vee \text{darček}(\text{darček_pre}(x)))) \} \\ \Leftrightarrow & \{ \forall x(\neg \text{dobré}(x) \vee \neg \text{dieťa}(x) \vee \text{dostane}(x, \text{darček_pre}(x))), \\ & \quad \forall x(\neg \text{dobré}(x) \vee \neg \text{dieťa}(x) \vee \text{darček}(\text{darček_pre}(x))) \} \end{aligned}$$

Konverzia do klauzálnej teórie

Veta 16.26. *Ku každej teórii T v jazyku logiky prvého rádu \mathcal{L} existuje ekvivalentná klauzálna teória v nejakom rozšírení \mathcal{L}' jazyka \mathcal{L} o skolemovské konštanty a funkcie.*

Príklad 16.27.

$$\left\{ \begin{array}{l} \forall x(\text{dobré}(x) \wedge \text{dieťa}(x) \rightarrow \exists y(\text{dostane}(x, y) \wedge \text{darček}(y))), \\ \exists x(\text{dobré}(x) \wedge \text{dieťa}(x)), \\ \forall x(\neg \text{dobré}(x) \rightarrow \neg \exists y \text{dostane}(x, y)) \end{array} \right\} \rightsquigarrow \left\{ \begin{array}{l} \forall x_1(\neg \text{dobré}(x_1) \vee \neg \text{dieťa}(x_1) \vee \text{dostane}(x_1, \text{darček_pre}(x_1))), \\ \forall x_2(\neg \text{dobré}(x_2) \vee \neg \text{dieťa}(x_2) \vee \text{darček}(\text{darček_pre}(x_2))), \\ \text{dobré}(\text{nejaké_dobré_dieťa}), \text{dieťa}(\text{nejaké_dobré_dieťa}), \\ \forall x_3 \forall y(\text{dobré}(x_3) \vee \neg \text{dostane}(x_3, y)) \end{array} \right\}$$

Konverzia do prvorádovej CNF

Dôkaz/algoritmus

T_I : Implikácie nahradíme disjunkciami.

T_N : *Negačný normálny tvar* (NNF): Presunieme negácie k atómom.

T_V : Premenujeme premenné tak, aby *každý kvantifikátor viazal inú premennú* ako ostatné kvantifikátory.

T_S : *Skolemizácia*: Existenčné kvantifikátory nahradíme substitúciou nimi viazaných premenných za skolemovské konštanty/aplikácie skolemovských funkcií na príslušné všeobecne kvantifikované premenné.

T_P : *Prenexný normálny tvar* (PNF): presunieme všeobecné kvantifikátory na začiatok formuly.

T_C : *Konjunktívny normálny tvar* (CNF): distribuujeme disjunkcie do konjunktív.

T_K : Odstránime konjunktie rozdelením konjunktov do samostatne kvantifikovaných klauzúl.

Skolemizácia vytvorí ekvivalentnú teóriu, ostatné úpravy sú ekvivalentné.

16.3 Rezolvenca v logike prvého rádu

Rezolvenca a skrátenie zápisu

Prvorádovou rezolvencom budeme odvodzovať dôsledky klauzálnych teórií.

Dohoda 16.28. Všeobecné kvantifikátory v zápise klauzúl budeme zanedbávať.

Teda namiesto $\forall x_1 \cdots \forall x_n (L_1 \vee \cdots \vee L_m)$ píšeme iba $L_1 \vee \cdots \vee L_m$.

Pozor: konštanty a premenné treba naďalej striktné rozlišovať, *za konštanty nie je možné dosádzať iné termy!*

Úsudky s klauzulami

Príklad 16.29. Každého má niekto rád — jeho najlepší kamarát/najlepšia kamarátka (NK):

$$\forall y r(\text{nk}(y), y)$$

Kto má rád Dadu, toho Edo nemá rád:

$$\forall x (\neg r(x, D) \vee \neg r(E, x)),$$

Teda aj Dadu má niekto rád:

$$r(\text{nk}(D), D)$$

Ak Dadin NK má rád Dadu, tak ho Edo nemá rád:

$$\neg r(\text{nk}(D), D) \vee \neg r(E, \text{nk}(D)).$$

Preto (výrokovou rezolenciou):

$$\frac{r(\text{nk}(D), D) \quad (\neg r(\text{nk}(D), D) \vee \neg r(E, \text{nk}(D)))}{\neg r(E, \text{nk}(D))}$$

Úsudky s klauzulami

Celý úsudok z príkladu aj s dosadeniami:

$$\frac{\forall y r(\text{nk}(y), y) \quad \forall x (\neg r(x, D) \vee \neg r(E, x))}{\neg r(E, \text{nk}(D))}$$

Aby sme klauzuly mohli rezolvovať, použili sme *unifikátor*:

$$\sigma = \{x \mapsto \text{nk}(D), y \mapsto D\}$$

Po substitúcii σ majú komplementárne literály rovnaké argumenty predikátu r :

$$\begin{aligned} r(\text{nk}(y), y)\sigma &= r(\text{nk}(D), D) \\ \neg r(x, D)\sigma &= \neg r(\text{nk}(D), D) \end{aligned}$$

Ak chceme čo najvšeobecnejší úsudok, hľadáme *najvšeobecnejší unifikátor*.

Úsudky s klauzulami

Uvedený úsudok predstavuje jeden krok rezolvenencie.

Príklad 16.30.

$$\frac{r(\text{nk}(y), y) \sigma \quad (\neg r(x, D) \vee \neg r(E, x)) \sigma}{\neg r(E, x) \sigma}$$

$$\sigma = \{x \mapsto \text{nk}(D), y \mapsto D\}$$

$$\frac{r(\text{nk}(D), D) \quad \neg r(\text{nk}(D), D) \vee \neg r(E, \text{nk}(D))}{\neg r(E, \text{nk}(D))}$$

Premenovanie premenných

Príklad 16.31. Rovnaké premenné v klauzulách môžu zabrániť unifikácii literálov:

$$r(\text{nk}(x), x) \quad \neg r(x, D) \vee \neg r(E, x)$$

Klauzuly sú však všeobecne kvantifikované *nezávisle* od seba. Premenovanie premenných v jednej z nich nezmení jej význam, ale umožní unifikáciu (viď predchádzajúci príklad).

$$r(\text{nk}(y), y) \quad \neg r(x, D) \vee \neg r(E, x)$$

Definícia 16.32. *Premenovaním premenných* je každá substitúcia $\sigma = \{x_1 \mapsto y_1, \dots, x_n \mapsto y_n\}$, kde y_1, \dots, y_n sú premenné.

Prvorádová rezolvenca — pravidlá

Definícia 16.33. Nech C a D sú prvorádové klauzuly, nech A a B sú atómy, nech L a K sú literály.

Rezolvenca (angl. resolution) je odvodzovacie pravidlo

$$\frac{A \vee C \quad \neg B \vee D}{(C\theta \vee D)\sigma} \quad \begin{array}{l} \sigma \text{ je unifikátor } A\theta \text{ a } B, \\ \theta \text{ je premenovanie premenných.} \end{array}$$

Faktorizácia (angl. factoring) je odvodzovacie pravidlo

$$\frac{L \vee K \vee C}{(L \vee C)\sigma} \quad \sigma \text{ je unifikátor } L \text{ a } K.$$

Faktorizácia je zovšeobecnenie idempotencie pri výrokovovej rezolvencii.

Rezolvenca postupne

Rezolvenciu

$$\frac{\neg P(x) \vee \neg Q(y, x) \vee R(f(x, y), y) \quad \neg R(x, c)}{\neg P(x) \vee \neg Q(c, x)}$$

si môžeme predstaviť ako postupný proces:

	$\neg R(x, c)$	
	$\downarrow \{x \mapsto z\}$	premenovanie
$\neg P(x) \vee \neg Q(y, x) \vee R(f(x, y), y)$	$\neg R(z, c)$	
\downarrow	$\{y \mapsto c, z \mapsto f(x, c)\}$	unifikácia
$\neg P(x) \vee \neg Q(c, x) \vee R(f(x, c), c)$	$\neg R(f(x, c), c)$	
$\neg P(x) \vee \neg Q(c, x)$		výrokovolog. rezolvenca

Rezolvenčné odvodenie a zamietnutie

Definícia 16.34. Nech T je klauzálna teória.

Rezolvenčným odvođením z T je každá (aj nekonečná) postupnosť klauzúl

$\mathcal{Z} = (C_1, C_2, \dots, C_n, \dots)$, kde každá klauzula C_i , $1 \leq i \leq n$, je:

- prvkom T , alebo
- odvođená pravidlom rezolvenacie z klauzúl C_j a C_k , ktoré sa v \mathcal{Z} nachádzajú pred C_i (teda $j, k < i$), alebo
- odvođená pravidlom faktorizácie z klauzuly C_j , ktorá sa v \mathcal{Z} nachádza pred C_i (teda $j < i$).

Zamietnutím T (angl. refutation) je každé konečné rezolvenčné odvodenie

$\mathcal{Z} = (C_1, C_2, \dots, C_n)$, kde $C_n = \square$.

Refutačná korektnosť a úplnosť rezolvenzie

Pri klasickom poňatí dôkazu ako postupnosti formúl, ktoré sú odvodené z predošlých formúl pomocou fixnej sady pravidiel, pod *úplnosťou* rozumieme schopnosť odvodiť z teórie hociktorú formulu, ktorá je jej logickým dôsledkom. Rezolvenzia je v tomto zmysle neúplná (napr. z A nevieme odvodiť $A \vee B$ či $A \vee \neg A$).

Vieme však rezolvenziou z ľubovoľnej nesplniteľnej teórie odvodiť nesplniteľnú prázdnu klauzulu \square . Tejto vlastnosti hovoríme *refutačná úplnosť*.

Veta 16.35 (Refutačná korektnosť a úplnosť rezolvenzie). *Nech T je klauzálna teória. Potom existuje zamietnutie T vtt T je nesplniteľná.*

Refutačná korektnosť a úplnosť rezolvenzie

Pretože každú teóriu môžeme transformovať na ekvisplniteľnú klauzálnu teóriu, dostávame:

Dôsledok 16.36 (Úplnosť rezolvenzie). *Nech T je teória, nech X je uzavretá formula. Nech $T'_X = \{C_1, \dots, C_n\}$ je klauzálna teória ekvisplniteľná s $T \cup \{\neg X\}$. Potom z T vyplýva X vtt existuje zamietnutie T'_X .*

Príklad 16.37. Dokážme nesplniteľnosť:

$$\left\{ \begin{array}{l} \forall x r(nk(x), x), \\ \forall x \forall y r(x, nk(y)), \\ \forall x (\neg r(x, D) \vee \neg r(E, x)) \end{array} \right\}$$

12. prednáška

Nerozhodnuteľnosť, neúplnosť a logika 2. rádu

Cieľom tejto prednášky je jemne doplniť filozofický, matematický, algoritmický a historický kontext týkajúci sa logiky.

Nie je nutné, aby ste uvedené veci podrobne ovládali — ide o veľké idey, nezaujímate sa o technické detaily.

Presná formulácia a dôkazy tvrdení z tejto prednášky:

- predmet Matematická logika (2-INF-114)

17 Logika 1. a 2. rádu

Zhrnutie: logika 1. rádu

Uvažujme jazyk logiky 1. rádu a v ňom teóriu T a formulu F .

Veta 17.1 (Korektnosť logiky 1. rádu). *Ak $T \vdash F$, tak $T \models F$.*

Veta 17.2 (Úplnosť logiky 1. rádu). *Ak $T \models F$, tak $T \vdash F$.*

Veta 17.3 (Kompaktnosť logiky 1. rádu). *Každá nespĺniteľná teória má konečnú podmnožinu, ktorá je nespĺniteľná.*

Ako vyjadriť tranzitívny uzáver

Majme binárnu reláciu reprezentovanú predikátom R . Pre každé $n \geq 1$ označme $R_n(a, b)$ formulu

$$\exists x_1 \exists x_2 \cdots \exists x_n (R(a, x_1) \wedge R(x_1, x_2) \wedge \cdots \wedge R(x_n, b))$$

(jednotlivé formuly R_n sú konečné, ale je ich nekonečne veľa — pre každé n by sme mohli náš jazyk rozšíriť o predikát, ktorý R_n definuje).

Ďalej nech $R_0 = R$ a nech

$$R^*(a, b) \text{ vtt existuje } n \text{ také, že platí } R_n(a, b).$$

Náše R^* vyjadruje tranzitívny uzáver relácie R .

Dá sa však R^* vyjadriť prvorádovou formulou?

Ako vyjadriť tranzitívny uzáver

Predpokladajme, že R^* sa dá vyjadriť prvorádovou formulou.

Uvažujme nekonečnú prvorádovú teóriu

$$T = \{\neg R_0(a, b), \neg R_1(a, b), \dots, \neg R_n(a, b), \dots, R^*(a, b)\}.$$

1. T nemá model: ak $R^*(a, b)$, tak pre nejaké n platí $R_n(a, b)$.
2. Každá konečná podmnožina T má model: ak n je najväčší z indexov R_n obsiahnutých v podmnožine, vyhovujúci model je napr. $D = \mathbb{N}$, $i(R) = \{(0, 1), (1, 2), \dots, (n + 1, n + 2)\}$, $i(a) = 0$, $i(b) = n + 2$.

Tieto dve pozorovania sú však v spore s kompaktnosťou logiky 1. rádu.

Formulou logiky 1. rádu tranzitívny uzáver nevieme vyjadriť.

Ako vyjadriť tranzitívny uzáver

V logike 2. rádu by to už šlo:

$$R^*(a, b) \leftrightarrow$$

$$\forall P \left(\left(\forall x (R(a, x) \rightarrow P(x)) \wedge \forall x \forall y (P(x) \wedge R(x, y) \rightarrow P(y)) \right) \rightarrow P(b) \right)$$

Tento popis je založený na tom, že vlastnosť „ b je R -potomkom a “ interpretujeme ako „ b dedí od a každú vlastnosť, ktorú majú všetci priami R -potomkovia a a zároveň sa zachováva cez R “.

Syntax a sémantiku logiky 1. rádu možno rozšíriť tak, aby umožňovala kvantifikovanie cez predikáty; dostaneme tak logiku 2. rádu.

Použitie formúl 2. rádu zvyšuje vyjadrovaciu silu jazyka, ale zároveň stratíme kompaktnosť a úplnosť.

Ako vyjadriť indukciu

Predstavme si, že chceme pridať axiómu umožňujúcu využívať v dôkazoch matematickú indukciu. Nech P^1 je predikát a S^1 funkčný symbol pre nasledovníka. Chceli by sme

$$\forall P \left(\left(P(0) \wedge \forall x (P(x) \rightarrow P(S(x))) \right) \rightarrow \forall x P(x) \right).$$

Toto je však formula logiky 2. rádu – v logike 1. rádu nič ako $\forall P$ pre predikát P nemáme. Jediné, čo nám ostáva, je pridať axiómu

$$(P(0) \wedge \forall x(P(x) \rightarrow P(S(x)))) \rightarrow \forall x P(x),$$

kde P je predikát **vyjadriteľný formulou** v našom konkrétnom prvorádovom jazyku \mathcal{L} . To má nevýhody:

- dôkaz využívajúci túto axiómu bude platiť len pre jeden konkrétny predikát;
- dôkazy sa budú týkať len predikátov vyjadriteľných v \mathcal{L} .

Peanova aritmetika

Jednou z možností, ako formalizovať aritmetiku, je *Peanova aritmetika* (PA). Jazyk vychádza z prvorádovej logiky s rovnosťou, individuová konštanta je 0, funkčné symboly sú S^1 , $+$, \cdot (nasledovník, sčítanie, násobenie). Prirodzené čísla reprezentujeme ako *termy* bez premenných, napr. 3 je $S(S(S(0)))$. Axiómy:

$$\begin{aligned} \forall x \ 0 &\neq S(x) \\ \forall x \forall y \ (S(x) \doteq S(y) &\rightarrow x \doteq y) \\ \forall x \ x + 0 &\doteq x \\ \forall x \forall y \ x + S(y) &\doteq S(x + y) \\ \forall x \ x \cdot 0 &\doteq 0 \\ \forall x \forall y \ x \cdot S(y) &\doteq (x \cdot y) + x \end{aligned}$$

Táto množina axióm je nezávislá (žiadna nevyplýva z iných).

Peanova aritmetika

K uvedeným axiómam treba pridať axiómu pre indukciu. Máme dve možnosti:

- Formuly 1. rádu. Nevýhoda: neštandardné modely (aj s nespočítateľnou doménou); ich existencia vyplýva o.i. z vety o kompaktnosti.
- Formula 2. rádu. Výhoda: všetky modely sú izomorfné s prirodzenými číslami.

Peanova aritmetika

Prečo práve tri funkčné symboly?

Ak vypustíme násobenie (hoci definovateľné pomocou sčítania), nevieme vyjadrovať vlastnosti sčítania (napr. vzťah pre násobok by zahŕňal tranzitívny uzáver).

Naopak, pridať funkčný symbol pre umocňovanie (či inú funkciu na prirodzených číslach) netreba, lebo prvorádová PA (dokonca aj bez indukcie) zachytáva každú vypočítateľnú funkciu.

Peanova aritmetika

Presnejšie, jazyk PA umožňuje formulou vyjadriť každú *primitívne rekurzívnu funkciu* (t. j. takú, čo sa dá počítať programom využívajúcim len cykly s vopred daným počtom opakovaní), a následne vie PA dokazovaním rozhodnúť o platnosti takýchto formúl.

Označme \bar{x} reprezentáciu prirodzeného čísla x v PA (ako termu vznikajúceho opakovanou aplikáciou symbolu S , napr. $\bar{2} = S(S(0))$). Funkciu f^1 možno reprezentovať akousi formulou F takou, že $F(\bar{x}, \bar{y})$ je pravda vtt $f(x) = y$.

Ak $f(x) = y$, tak $PA \vdash F(\bar{x}, \bar{y})$, a ak $f(x) \neq y$, tak $PA \vdash \neg F(\bar{x}, \bar{y})$.

18 Nerozhodnuteľnosť platnosti formuly

Nerozhodnuteľnosť platnosti formuly vo FOL

Veta 18.1. *Platnosť formuly v prvorádovej logike je nerozhodnuteľná.*

Dôkaz: redukciou na problém zastavenia.

Ukážeme, ako by sme vedeli rozhodnúť, či daný Turingov stroj M zastaví, ak by sme vedeli rozhodovať platnosť prvorádových formúl.

Nerozhodnuteľnosť platnosti formuly vo FOL

Jazyk našej logiky bude obsahovať

- individuová konštanta ε pre prázdny reťazec;
- unárny funkčný symbol a^1 pre každé písmeno a v abecede;

- binárny predikát f_q pre každý stav q TS M .

Naša interpretácia tohto jazyka:

- $a(w)$ označuje reťazec aw ;
- $f_q(x, y)$ indikuje, že M dosiahne na danom vstupe stav q , pričom na páske je reťazec $\bar{x}y$ (x v opačnom poradí) a hlava M je na prvom znaku y .

Nerozhodnuteľnosť platnosti formuly vo FOL

Krok výpočtu zachytáva formulu

$$\forall x \forall y \quad f_q(x, a(y)) \rightarrow f_{q'}(b(x), y)$$

M prečíta z pásky a , zapíše b , prejde zo stavu q do stavu q' a posunie hlavu doprava. (Takúto formulu pridáme do teórie popisujúcej činnosť M pre každú dvojicu a, b .)

Pre pohyb hlavy doľava máme formulu

$$\forall x \forall y \quad f_q(c(x), a(y)) \rightarrow f_{q'}(x, c(b(y))).$$

Nerozhodnuteľnosť platnosti formuly vo FOL

Pre polohu hlavy na ľavom okraji pásky pridáme

$$\forall y \quad f_q(\varepsilon, a(y)) \rightarrow f_{q'}(\varepsilon, b(y))$$

(hlava sa nehýbe, len prepisuje znak na páske).

Podobne doriešime aj polohu na pravom okraji, keď sa hlava posunie na časť pásky, kam sa ešte nezapisovalo.

Konkrétne detaily závisia od uvažovaného variantu Turingovho stroja (páska môže byť obojstranne nekonečná apod.).

Nerozhodnuteľnosť platnosti formuly vo FOL

Začiatok výpočtu z počiatočného stavu q_0 na slove w popisuje formulu $f_{q_0}(\varepsilon, w)$ a zastavenie stroja v (jedinom) akceptačnom stave q_{acc} vyjadruje formulu

$$F_M = f_{q_0}(\varepsilon, w) \wedge T \rightarrow \exists x \exists y \quad f_{q_{acc}}(x, y),$$

kde T je konjunkcia implikácií popisujúcich povolené prechody TS M .

Každý akceptačný výpočet M vieme prerobiť na dôkaz F_M (konečné slovo na páske a poloha hlavy popisujú hodnoty premenných x a y v konzekvente implikácie F_M ; jednotlivé kroky výpočtu naznačujú, ako inšancovať všeob. kvantifikátory formúl z T). Keďže prvorádová logika je korektná, tak ak M zastaví, F_M je platná formula.

Naopak, ak F_M je platná, tak je pravdivá v každej interpretácii (štruktúre), čiže aj v tej našej týkajúcej sa TS M . Keďže premisy F_M sú splnené, musí byť splnený aj záver $\exists x \exists y f_{q_{acc}}(x, y)$, takže M zastaví.

Čiastočná rozhodnuteľnosť platnosti formuly

Veta 18.2. *Platnosť formuly v prvorádovej logike (so spočítateľným jazykom) je čiastočne rozhodnuteľná.*

Dôkaz: stačí enumerovať všetky dôkazy v danom jazyku.

Formula je platná vtt jej negácia je nesplniteľná. Dôsledky:

Veta 18.3. (1) *Nesplniteľnosť formuly v prvorádovej logike je nerozhodnuteľná a čiastočne rozhodnuteľná.* (2) *Splniteľnosť v prvorádovej logike nie je ani čiastočne rozhodnuteľná.*

Dôkaz (2): ak by sme splniteľnosť vedeli rozhodovať čiastočne, môžeme paralelne spustiť testovanie splniteľnosti aj nesplniteľnosti, a jeden z týchto výpočtov by musel skôr či neskôr skončiť, čím by sme vedeli rozhodovať nesplniteľnosť formuly, a to je spor.

Monadic FOL

Monadická prvorádová logika (MFOL) je prvorádová logika, v ktorej nemáme funkčné symboly a predikáty majú len jeden argument. (V takomto jazyku možno vyjadriť bežné sylogizmy a počas väčšiny 19. storočia sa verilo, že postačuje na formalizáciu uvažovania.)

Veta 18.4 (1915). *Pravdivosť formuly v MFOL je rozhodnuteľná.*

Ak pridáme čo len jeden predikát arity 2, stratíme rozhodnuteľnosť.

Tento príklad ilustruje, že zvoliť jazyk s veľkou vyjadrovacou silou (napr. pre databázový systém) neraz prináša algoritmické problémy, preto to nie je samozrejماً voľba.

19 Deskriptívna zložitosť

Deskriptívna zložitosť

Pri klasickom pohľade na zložitosť algoritmov skúmame, koľko krokov spraví Turingov stroj v závislosti od veľkosti vstupu, resp. či Turingov stroj s dodatočnými obmedzeniami vôbec vie problém riešiť.

Alternatívny pohľad: akú zložitú logickú formulu potrebujeme na popis daného problému (jazyka akceptovaného Turingovým strojom)?

Príklady rôzne zložitých formúl:

- prvorádová logika (FOL)
- druhorádová logika (SOL) – kvantifikujeme aj predikáty/množiny objektov
- existencial SOL: $\exists X_1 \exists X_2 \dots \exists X_n F$, F je formula FOL
- universal SOL: $\forall X_1 \forall X_2 \dots \forall X_n F$, F je formula FOL

Deskriptívna zložitosť

Existencia trojuholníka v grafe sa dá vyjadriť vo FOL:

$$\exists x \exists y \exists z (V(x) \wedge V(y) \wedge V(z) \wedge \\ \wedge x \neq y \wedge y \neq z \wedge z \neq x \wedge E(x, y) \wedge E(y, z) \wedge E(z, x))$$

Pri testovaní, či konkrétny graf spĺňa túto formulu, nerozhodujeme o platnosti formuly vo všeobecnosti (to je nerozhodnuteľný problém), ale len vyhodnocujeme jej splnenie v konkrétnej interpretácii: predikát V popisuje vrcholy, E hrany.

Pri vyhodnotení kvantifikátora stačí preskúmať všetky vrcholy, čiže pre k kvantifikátorov vo formule preveríme $O(|V|^k)$ možností. Preto tento problém patrí do triedy P (zjemnenie úvah umožňuje dokázať príslušnosť do LOGSPACE).

Deskriptívna zložitosť

Existencia 3-farbenia grafu: NP-complete; nevyjadriteľné vo FOL, vyjadriteľné v existential SOL:

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

(R, G, B sú predikáty vyjadrujúce jednotlivé farby). Pri vyhodnocovaní formuly pre konkrétny graf za doménu zoberieme vrcholy grafu. Existencia predikátu zodpovedá existencii podmnožiny všetkých vrcholov; túto podmnožinu vieme nedeterministicky uhádnuť a vyhodnotenie prvorádovej časti formuly už pridá len polynomiálny faktor (exponent závisí od počtu vnorených kvantifikátorov), preto tento problém patrí do triedy NP.

Deskriptívna zložitosť

- FOL: trieda zložitosti AC^0 (vlastná podmnožina LOGSPACE)
- FOL + tranzitívny uzáver: non-deterministic LOGSPACE
- FOL + least fixed point operator: P (súvisí s databázami: dotazy sú podmnožinou prvorádových formúl a navyše je povolená rekurzia počítaná seminaivnou evaluáciou, čiže ako least fixed point)
- existential SOL: NP
- universal SOL: co-NP
- SOL: PH (obsahuje NP aj coNP, ale vlastná podmnožina PSPACE)
- SOL + least fixed point operator: EXPTIME

20 Neúplnosť

Efektívna axiomatizovateľnosť

Ako formalizovať, že niečo „možno vypočítať“?

Vieme vymýšľať rôzne konkrétne modely na počítanie a porovnávať, čo dokážu. Ukazuje sa, že všeobecné modely počítania s prirodzenými číslami (nie naschvál oslabené dodatočnými reštrikciami) vedú k tomu istému pojmu vypočítateľnosti.

- Turingove stroje
- frázové gramatiky
- Minského registrové automaty
- lambda kalkul
- čiastočné rekurzívne funkcie

Church-Turing Thesis

Ak sa funkcia z \mathbb{N} do \mathbb{N} dá vypočítať, dá sa vypočítať na Turingovom stroji.

Efektívna axiomatizovateľnosť

Z hľadiska logiky nás zaujímajú formálne systémy, s ktorými možno algoritmicke pracovať:

- konečná množina symbolov
- algoritmy na prácu s termami
- algoritmy na prácu s formulami
- jednoznačné interpretovanie formuly v danej štruktúre
- rozhodnuteľnosť, či niečo je axióma (resp. enumerácia axióm/formúl v teórii)
- rozhodnuteľnosť, či niečo je dôkaz (resp. enumerácia dôkazov)

Teórii s týmito vlastnosťami (resp. formálnemu systému, ktorého je súčasťou) hovoríme *efektívne axiomatizovateľná* (v angličtine *effectively*, nie *efficiently*).

Sémantická a negačná úplnosť

Doterajší pojem úplnosti, tzv. *sémantická úplnosť* formálneho systému, hovoril o tom, že pravdivé veci možno dokázať: Pre každú teóriu T a uzavretú formulu F , ak $T \models F$, tak $T \vdash F$.

Zaujímavá je však aj *negačná úplnosť* teórie T : pre každú uzavretú formulu F (v jazyku tejto teórie) platí $T \vdash F$ alebo $T \vdash \neg F$.

Sémantická a negačná úplnosť

Uvažujme výrokovú logiku s atómami A, B, C a teóriu $T = \{A \wedge B\}$. Sémantická úplnosť vyplýva z vlastností výrokovej logiky.

Avšak napr. formula C je nezávislá od teórie T , neplatí $T \models C$ ani $T \models \neg C$. Preto $T \not\models C$ a $T \not\models \neg C$, čiže T nie je negačne úplná.

Sémantická a negačná úplnosť

Je ľahké vytvárať teórie, ktoré sú negačne neúplné: stačí vynechať zopár podstatných axióm/formúl.

Ak sa však snažíme nájsť teóriu T , ktorá čo najúplnejšie popisuje nejakú časť sveta vyjadriteľnú v jazyku \mathcal{L} , chceli by sme, aby pre každú uzavretú formulu jazyka \mathcal{L} platilo buď $T \models F$ alebo $T \models \neg F$. Preto by sme chceli vedieť efektívne rozhodnúť, či $T \vdash F$ alebo $T \vdash \neg F$. Hľadáme teda negačne úplné teórie.

Prvá Gödelova veta o neúplnosti

Veta 20.1. *Nech T je prvorádová teória v jazyku \mathcal{L} s týmito vlastnosťami:*

- je konzistentná (nevyplýva z nej nepravda);
- je efektívne axiomatizovateľná;
- obsahuje aritmetiku.

Potom T nie je negačne úplná (t. j. existuje tvrdenie v \mathcal{L} , ktoré nemožno z T dokázať ani vyvrátiť).

Prvá Gödelova veta o neúplnosti

Idea dôkazu – Gödelova veta G : „toto je veta, ktorá sa nedá dokázať“.

Aritmetiku potrebujeme na to, aby sme vedeli jednoznačne očíslovať všetky možné formuly a tiež všetky možné dôkazy (efektívna axiomatizovateľnosť zaručuje, že formuly aj dôkazy sa dajú enumerovať, je ich teda spočítateľne veľa). Potom vieme kľúčovú vetu G zapísať ako prvorádovú formulu.

Ak by sa G dala dokázať, je nepravdivá, čo je spor s konzistentnosťou. Ak sa G dokázať nedá, je pravdivá. Ale potom $\neg G$ je nepravdivá, a teda sa nemôže dať v konzistentnom systéme dokázať. Teória T teda nie je negačne úplná.

Prvá Gödelova veta o neúplnosti

Skonstruovaná Gödelova veta G_T je síce dlhá, ale má jednoduchú štruktúru, nejde o nejakú absurdne zložitú formulu.

A nielen to. Ak do T doplníme ako axiómu jej G_T alebo $\neg G_T$, naďalej budú splnené predpoklady prvej Gödelovej vety, a výsledná teória bude stále negatívne neúplná.

Toto prekvapivé tvrdenie z r. 1931 znamenalo krach Hilbertovho programu z r. 1900 (snahy o kompletnú axiomatizáciu matematiky v štýle Euklidových základov).

Tarského veta

Kľúčovým krokom v dôkaze prvej Gödelovej vety je konštrukcia formuly $F_1(k)$ vyjadrujúcej, že formula s kódom k je dokázateľná v rámci daného logického systému.

Tarského veta: v rámci daného logického systému nie je možné zostrojiť formulu $F_2(k)$, ktorá by vyjadrovala, že formula s kódom k je platná.

Pozorovanie:[2mm]

Veľká časť logiky ako vedného odboru je zameraná na nahradenie práce s *pravdivosťou* prácou s *dokázateľnosťou*, ktorá je na rozdiel od pravdivosti algoritmicke uchopiteľná.

Pojem výpočtu je s logikou veľmi úzko spätý: kvôli aplikáciám v logike vlastne vznikli prvé formalizácie pojmu výpočet či algoritmus (Gödel cca 1930 — primitívne rekurzívne funkcie).

Druhá Gödelova veta o neúplnosti

Veta 20.2. *Ak S je bezosporný efektívne axiomatizovateľný formálny systém obsahujúci aritmetiku (napr. PA), jeho bezospornosť nemožno dokázať v rámci S .*

Čiže dôkaz treba spraviť neformálnou metaúvahou alebo v rámci iného „silnejšieho“ formálneho systému. Ako však dokážeme bezospornosť tohto druhého systému? ...

V tomto zmysle nevieme dokázať ani len bezospornosť PA, nieto zložitejších formalizácií matematiky.

21 Dôkazy ako programy

Prirodzená dedukcia / natural deduction

Prirodzená dedukcia: gentzenovský systém, konštruktívne dôkazy (nie ako tablá, kde odvodzujeme spor).

Dôkaz pre $(A \wedge C) \rightarrow (B \rightarrow A)$:

1. Predpokladajme $A \wedge C$ (hypotéza H_1).
2. Predpokladajme B (hypotéza H_2).
3. Z $A \wedge C$ odvodíme A – t.j. $\{H_1, H_2\} \vdash A$ (pravidlo $\wedge E_L$ použité na 1).
4. Z predpokladu B sme odvodili A , preto $\{H_1\} \vdash B \rightarrow A$ (pravidlo $\rightarrow I$ použité na 3, discharging H_2).
5. Z predpokladu $A \wedge C$ sme odvodili $B \rightarrow A$, preto $\vdash (A \wedge C) \rightarrow (B \rightarrow A)$ (pravidlo $\rightarrow I$ použité na 4, discharging H_1).

Lambda kalkul

Lambda kalkul:

- Formálny systém na zapisovanie výpočtov.
- *Výpočty* sú termy vznikajúce kombinovaním dvoch jednoduchých operácií:
 - function abstraction (viazanie voľných premenných),
 - function application (substitúcia / skladanie funkcií).
- Termy pozostávajú z funkcií (aj konštanty vnímame ako konštantné funkcie).

Lambda kalkul: viazanie premenných

Viazanie premennej:

- K danému termu M možno vytvoriť nový term $\lambda x.M$.
- Premenná x , ktorá reprezentuje vstup do funkcie, je tým viazaná v rámci termu M (všetky jej výskyty).

- Term M môže obsahovať voľné premenné.
- Príklad: v $\lambda x.(x+y)$ je x je viazaná, ale y voľná premenná (jej hodnota prichádza kdesi zvonku).

Lambda kalkul: skladanie funkcií

Aplikovanie funkcie / skladanie funkcií / substitúcia / β -redukcia:

- Z termov M, N môžeme vytvoriť term (MN) .
- Príklad: $((\lambda x.M)N)$ – funkciu $\lambda x.M$ aplikujeme na argument N , čiže všetky viazané výskyty x v rámci M sú nahradené N .
- Príklad: $((\lambda x.(x + 2)) 3)$.
 - Tu je M výraz $(x + 2)$, x je viazaná premenná a N je 3.
 - Substitúcia: nahraď x vo výraze $(x + 2)$ číslom 3.
 - Výsledok: $(3 + 2)$, čo sa vyhodnotí na 5.

Lambda kalkul: typy

Lambda kalkul je turingovsky úplný, čiže pripúšťa aj nekonečné výpočty (napr. je možné funkciu aplikovať na seba). Ak sa nám to nepáči, môžeme pridať mechanizmus na prácu s typmi, ktorý zabezpečí konečnosť (napr. Simply Typed Lambda Calculus). Ukážka:

- atomické typy: void (príp. bool, int...)
- $T_1 \rightarrow T_2$: funkcia s argumentom typu T_1 vráti hodnotu typu T_2 , napr. $\lambda x : T_1.e$, kde e je telo funkcie vracajúce T_2
- product type $T_1 \times T_2$, usporiadaná dvojica objektov daných typov

Typy pridávajú obmedzenia pre skladanie funkcií, vďaka čomu máme kontrolu aj nad algoritmickými aspektmi (termy v lambda kalkule popisujú výpočty).

Lambda kalk: typy a logické spojky

K danej výrokovologickej formule vieme zostaviť term v lambda kalkule.

- Formule $A \wedge C$ zodpovedá typ $A \times C$. Na prácu s objektmi tohto typu sa hodí funkcia proj_i , ktorá vráti i -tú zložku danej dvojice.
- Formule

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

zodpovedá term

$$\lambda p : (A \times C).(\lambda q : B.\text{proj}_1(p)).$$

Táto funkcia prijíma dôkaz p výroku $A \wedge C$ a vracia „vnútornú“ funkciu. Táto vnútorná funkcia prijíma dôkaz q výroku B a vracia dôkaz A (získaný z p).

Curryho-Howardova korešpondencia

Medzi dôkazmi a výpočtami existuje „izomorfizmus“.

Logika (Prirodzená dedukcia)	Výpočty (Lambda kalk)
Výrok (A, B, C) $P \wedge Q$ $P \rightarrow Q$	Typ (A, B, C) product type $P \times Q$ functional type $P \rightarrow Q$
Dôkaz / pravidlá $\wedge E_L$ (napr. z $p : A \wedge C$ dostaň A) $\rightarrow I$ (predpokladaj X , odvod' Y)	Term (program) / operácie Projekcia ($\text{proj}_1(p)$) Lambda abstrakcia ($\lambda x : X.y$)
Formula: $(A \wedge C) \rightarrow (B \rightarrow A)$	Typ: $(A \times C) \rightarrow (B \rightarrow A)$
Náš dôkaz (kroky dedukcie)	Náš program (lambda term): $\lambda p.(\lambda q.\text{proj}_1(p))$

Formula je dokázateľná vtt existuje term zodpovedajúceho typu.

Lean: jazyk a interaktívny dokazovací systém

Lean je dôkazový asistent založený na teórii typov (**Calculus of Constructions**).

- Implementuje Curryho-Howardovu korešpondenciu: výroky sú typy, dôkazy sú termy.

- Na dokázanie formuly treba skonštruovať term zodpovedajúceho typu.
- Lean vie verifikovať existujúce dôkazy (termy).
- Lean tiež ponúka tzv. „taktiky“, ktoré pomáhajú automatizovať konštrukciu dôkazu (napr. mu možno dať pokyn „sprav rozbor prípadov“, „zovšeobecni“, „použi lemu“).
- `mathlib` je rastúca knižnica formálne verifikovaných matematických tvrdení
- Ľudia s Fieldsovou medailou v tom robia formalizované dôkazy

Lean: dôkaz pomocou taktík

```
-- Declare A, B, C as propositional atoms
variable (A B C : Prop)

theorem my_tactic : (A ∧ C) → (B → A) := by
  -- Assume the hypothesis (A ∧ C) and call it h_ac
  -- The goal becomes: B → A
  intro h_ac
  -- Deconstruct h_ac : A ∧ C into its components.
  -- This introduces h_a : A and h_c : C into the context.
  -- The goal remains: B → A
  obtain ⟨h_a, h_c⟩ := h_ac
  -- Assume the hypothesis B (the antecedent of the implication) and
  -- call it h_b
  -- The goal becomes: A
  intro h_b
  -- The goal is now A, and we have h_a : A in our context.
  -- Use h_a to exactly prove the goal.
  exact h_a
```

Lean: dôkaz priamou konštrukciou termu

```
-- Declare A, B, C as propositions
variable (A B C : Prop)

-- Direct proof term
theorem my_term : (A ∧ C) → (B → A) :=
  fun h_ac : A ∧ C => -- Assume h_ac is a proof of A ∧ C
```

```

fun h_b : B =>      -- Assume h_b is a proof of B
  -- Return the left component of h_ac, which is a proof of A (
  like proj_1(p))
  h_ac.left

```

Lean: kvantifikátory a použitie vety

```

-- Import necessary definitions (Nat, <, succ) and theorems
import Mathlib.Data.Nat.Order.Basic

-- Theorem from Mathlib: 0 is less than the successor of any n.
theorem zero_lt_succ (n : Nat) : 0 < Nat.succ n

-- Prove that there exists a natural number greater than 0.
example : ∃ x : Nat, x > 0 :=
  -- The type corresponding to an existential quantifier
  -- is a pair (witness, proof for the witness).
  -- We provide the witness '47' (which is 'Nat.succ 46')
  -- and use 'zero_lt_succ' to prove '47 > 0'.
  ⟨47, zero_lt_succ 46⟩

```

Lean: AlphaProof

AlphaProof (od Google DeepMind) má za cieľ objavovať a overovať matematické dôkazy pomocou umelej inteligencie. Za základ si vybrali Lean:

- Lean ponúka spoľahlivú formálnu verifikáciu.
- Táto verifikácia je veľmi rýchla, používa ju napr. Amazon ako základnú metódu na verifikáciu softvéru.
- Hoci problém splniteľnosti formuly je nerozhodnutelný, Lean má dobre navrhnutý systém obmedzení, ktorý garantuje konečnosť verifikácie (napr. umožňuje použiť len rekurziu, ktorá dokázateľne skončí; ak dôkaz nevie nájsť Lean, žiada ho od užívateľa).

Lean: AlphaProof

- Jazyk, ktorý používa Lean, zahŕňa nielen FOL, ale aj logiku vyšších rádov (napr. je možné kvantifikovať cez funkcie, typy a predikáty).

Vyjadrovacia sila formalizmu, na ktorom je Lean postavený (Calculus of Inductive Constructions), stačí viac-menej na všetko (okrem „naschvál skonštruovaných extrémov“).

- Systém taktík umožňuje automatizáciu dôkazov. AI môže generovať stratégie dôkazu na vysokej úrovni alebo konkrétne volania taktík, ktoré Lean následne vykoná a overí.
- `mathlib` — obrovská aktívne vyvíjaná knižnica formalizovanej matematiky; možno na nej trénovať AI.

22 Nadväzujúce predmety

Nadväzujúce predmety

Presná formulácia a dôkazy tvrdení z tejto prednášky:

- Matematická logika (2-INF-114)

Viac o vypočítateľnosti:

- Teória vypočítateľnosti (2-INF-121)

Aplikácie logiky:

- Špecifikácia a verifikácia programov (1-AIN-470)
- Výpočtová logika (2-AIN-108)
- Reprezentácia znalostí a inferencia (2-AIN-144)
- Výpočtová fuzzy logika, modelovanie a systémy (2-AIN-113)
- Ontológie a znalostné inžinierstvo (2-AIN-286)

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_course2.pdf, https://www.inf.ed.ac.uk/teaching/courses/propm/papers/Zhang/sat_course3.pdf.