I008 Výpočtová logika

nikoliv *teorie modelů* (algebraická logika), ale *teorie důkazů*

Vstup: množina formulí
Výstup: logický důsledek vstupní množiny

mechanické prostředky pro dokazování formulí

Proces dokazování $=$*v jistém smyslu totéž jako* proces výpočtu programu

1 Úvod. Logický kalkul, syntaxe, sémantika

1.1 Úvod

$$\exists F\{\ F(a)\ =\ b\ \wedge\ \forall\ x\ [\ p(x)\ \rightarrow\ F(x)\ =\ G(x, f(x))\ ]\ \}$$

(syntakticky) dobře utvořená formule (well formed formula, wff)

Sémantika

- standardní $\wedge$ $\vee$ $\neg$ $\rightarrow$ $\leftrightarrow$ $\forall$ $\exists$
- interpretací
  - $a, b \in D$
  - $p : D \rightarrow \{true, false\}$
  - $f : D \rightarrow D, g : D\mathrm{x}D \rightarrow D$

Příklad 1.1 $\exists F\{\ F(a)\ =\ b\ \wedge\ \forall\ x\ [\ p(x)\ \rightarrow\ F(x)\ =\ G(x, f(x))\ ]\ \}$

Možné interpretace

1. D=N, a=0, b=1, f(x)=x-1, g(x,y)=x*y, p(x)= x>0
2. D=Σ^*, $\Sigma = \{\alpha, \beta\}$, a=b=nil, f(x)= x bez 1. znaku zřetězeno s 1. znakem z x
3. D=N, a=0, b=1, f(x)=x, g(x,y)=y+1, p(x)= x>0

1.2 Syntaxe

1.2.1 Abeceda

pravdivostní symboly T,F

výrokové spojky $\wedge$ $\vee$ $\neg$ $\rightarrow$ $\leftrightarrow$

operátory $=$

kvantifikátory $\forall$ $\exists$

konstanty
- funkční f_i^n (n-ární), n=0 individuové
- predikátové p_i^n, n=0 výrokové

proměnné
- funkční F_i^n, n=0 x,y,z,... individuové
- predikátové P_i^n, n=0 výrokové

1.2.2 Dobře utvořená formule

Definice 1.2 *Term*

1. *Individuové konstanty a proměnné jsou termy;*
2. *jsou-li* $t_1, t_2, ..., t_n$ *termy, jsou* $f_i^n(t_1, ..., t_n), F_i^n(t_1, ..., t_n)$ *termy.*

Definice 1.3 *Atomická formule (atf)*

1. T *a* F *jsou atf;*
2. *jsou-li* $t_1, t_2, ..., t_n$ *termy, jsou* $p_i^n(t_1, ..., t_n), P_i^n(t_1, ..., t_n)$ *atf;*
3. *jsou-li* t_1, t_2 *termy, je* $t_1 = t_2$ *atf.*

Definice 1.4 *Dobře utvořená formule (wff)*

1. *Každá atf je wff;*
2. *jsou-li* A,B *wff, jsou i* $\neg A,\ A \wedge B,\ A \vee B,\ A \rightarrow B,\ A \leftrightarrow B$ *wff;*
3. *je-li* x *proměnná a* A *wff, jsou* $(\forall x\ A),\ (\exists x\ A)$ *wff.*

Definice 1.5 *Proměnná je vázaná, je-li kvantifikovaná. Je-li mimo dosah kvantifikátoru, je volná.*

Příklad 1.6 $\forall P\ [\ P(a) \wedge \exists x\ (\ x \neq a \wedge P(f(x))\ )\ \rightarrow P(x)\ ]$

1.3 Klasifikace logických kalkulů

1. pouze výrokové konstanty (p^0), žádné proměnné ($\rightarrow$ žádné kvantifikátory):
   výrokový počet(kalkul)
   $(p \wedge q) \rightarrow (\neg q \vee r)$

2. pouze individuové konstanty a funkční individuové proměnné:
   kalkul rovnosti
   $\forall x \forall y \forall z \; [\; (x = y \wedge y = z) \rightarrow x = z \;]$

3. všechny konstanty, ale pouze funkční individuové proměnné:
   predikátový počet 1. řádu
   $(x \neq a) \rightarrow \forall y \; [\; \exists z \; p(x, f(y, z)) \rightarrow q(y) \;]$
   s funkcemi a rovnostmi

4. **predikátový počet 2.řádu**

1.4 Sémantika

Definice 1.7 *Interpretace I dobře utvořené formule A je trojice* (D, I_C, I_V),
kde $D \neq \emptyset$ *je obor interpretace*
I_C *je interpretace konstant*
I_V *je interpretace proměnných.*

Pozn: Dosazujeme jen volné proměnné.

Příklad 1.8 *A:* $\exists x\ \forall y\ p(x, y)$
I_1: $D=N$, $p(x,y) = x \leq y$; $<A, I_1>$?
I_2: $D=N$, $p(x,y) = x \geq y$; $<A, I_2>$?
I_3: $D=N$, $p(x,y) = x=1$; $<A, I_3>$?

B: $\forall x\ \exists y\ p(x, y)$
$<B, I_1>$?
$<B, I_2>$?
$<B, I_3>$?

Příklad 1.9 $[\ \forall x\ p(x)\ \vee\ \forall x\ q(x)\ ] \rightarrow \forall x\ (\ p(x) \vee q(x)\ )$

Příklad 1.10 $\forall x\ (\ p(x) \vee q(x)\ ) \rightarrow [\ \forall x\ p(x)\ \vee\ \forall x\ q(x)\ ]$

Definice 1.11 *Interpretace dané formule, která vede k pravdivému výroku, se nazývá model formule. Jinými slovy,tato interpretace splňuje formuli.*

Definice 1.12 *Formule je splnitelná, existuje-li alespoň jeden její model. V opačném případě je nesplnitelná.*

Definice 1.13 *Je-li formule pravdivá v každé interpretaci, nazývá se logicky pravdivá (validní). (Ve výrokovém počtu též tautologie).*

Definice 1.14 *Dvě formule A,B nazýváme ekvivalentní, jestliže každá interpretace, která je modelem jedné z nich, je modelem i druhé (tj. právě když $A \leftrightarrow B$ je logicky pravdivá formule). Fromule B je logickým důsledkem formule A, jestliže každá interpretace, která je modelem A , je i modelem B (tj. právě když $A \rightarrow B$ je logicky pravdivá.*

Příklad 1.15 $\forall x\ p(x) \rightarrow \exists x\ p(x)$

$\forall x\ q(x) \rightarrow q(x)$

$\exists x\ p(x) \rightarrow \forall x\ p(x)$

$p(a)\ \wedge\ \neg p(a)$

1 Intro

2 Výroková logika

2.1 Uspořádání a strom

Definice 2.1 *Částečné uspořádání je množina S spolu s binární relací $<$ na S, která je tranzitivní, tj. $(x < y) \wedge (y < z) \rightarrow x < z$, a ireflexivní, tj. $x < x$ neplatí pro žádné x.*

Definice 2.2 *Částečné uspořádání $<$ je lineární uspořádání (uspořádání), jestliže splňuje podmínku $(x < y) \vee (x = y) \vee (y < x)$ pro $\forall x, y \in S$.*

Příklad 2.3

Definice 2.4 *Lineární uspořádání je dobře založené (nebo dobré) uspořádání, když neobsahuje žádný nekonečný sestupný řetězec* $x_0, x_1, \ldots \in S$ *takový, že*
$\ldots\ x_2 < x_1 < x_0$.

Definice 2.5 *Strom je množina* T *, jejíž prvky nazýváme uzly, částečně uspořádaná relací* $<_T$ *s jediným nejmenším prvkem, zvaným kořen, kde předchůdci každého uzlu jsou dobře uspořádány relací* $<_T$ *(např. neexistuje tam cyklus).*

Cesta P *ve stromu* T *je maximální lineárně uspořádaná podmnožina* T *.*

Příklad 2.6 1. *Vytvářecí strom*

2. *Rozhodovací strom*

Definice 2.7 *Úroveň ve stromu T je definována indukcí takto:*

1. *Nultá úroveň obsahuje kořen T .*
2. *$(k + 1)$-tá úroveň T obsahuje všechny bezprostřední následníky všech uzlů k-té úrovně.*

Hloubka stromu T je maximální n takové, že strom T obsahuje uzel úrovně n. Existuje-li uzel úrovně n pro $\forall n \in N$, pak strom T je nekonečný.

Jestliže každý uzel stromu T má nejvýše n následovníků, pak strom nazýváme n-ární. Pokud mají všechny uzly konečně mnoho bezprostředních následovníků, strom nazýváme strom s konečným větvením.

Příklad 2.8

Věta 2.9 *Königova věta*
Jestliže strom s konečným větvením je nekonečný (má nekonečnou hloubku), pak obsahuje nekonečnou cestu.

Důsledek: Žádný strom T nemá současně následující 3 vlastnosti:

1. Každý uzel v T má konečně mnoho přímých následníků.
2. Kařdá cesta v T je konečně dlouhá.
3. Strom T má nekonečně mnoho uzlů.

Definice 2.10 *Ohodnocený strom T je strom T spolu s funkcí f, která přiřazuje každému uzlu v T nějaký objekt, který nazýváme ohodnocení nebo hodnota uzlu.*

Příklad 2.11

2.2 Výroky, spojky a pravdivostní tabulky

Definice 2.12 *Formule je řetězec nad abecedou výrokových symbolů a logických spojek*

1. *výrokový symbol je formule;*
2. *jsou-li* $\alpha, beta$ *formule, jsou i* $(\alpha \wedge beta)$, $(\alpha \vee beta)$, $(\alpha \rightarrow beta)$, $(\neg\alpha)$ *formule;*
3. *řetězec symbolů je formule, právě když ji lze získat aplikací (1) a (2).*

Lemma 2.13 *Každá formule má jednoznačný vytvářecí strom.*

Definice 2.14 *Množina pravdivostně funkcionálních spojek je postačující, jestliže pro libovolnou pravdivostní tabulku existuje výrok složený z těchto spojek, který má stejnou pravdivostní tabulku.*

Věta 2.15 *Množina* $\{\wedge, \vee, \neg\}$ *je postačující.*

Příklad 2.16 *(idea důkazu věty 2.15)*

A	B	C	?
T	T	T	T
T	T	F	F
T	F	T	F
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	T

Z pohledu na tuto tabulku vidíme, kdy hledaná funkce nabývá hodnot T (pravda). V našem případě js ou to 1., 5. a 8. řádek. Tedy výsledná funkce je logickým součtem podmínek, které je nutno splnit , aby hledaná funkce nabyla hodnoty pravda. Každá podmínka je konjunkcí požadavků na pravdivost všech výrokových symbolů A; B; C. Tedy hledaná funkce je:

Důsledek:

1. Množina $\{\neg, \vee\}$ je postačující.
2. Shefferova spojka je postačující.

A	B	A|B
T	T	F
T	F	T
F	T	T
F	F	T

2.3 Přiřazeni pravdivostních hodnot. Valuace

Definice 2.17 *Pravdivostní přiřazení $\mathcal{A}$ je funkce, která přiřazuje každému výrokovému symbolu A jedinou pravdivostní hodnotu $\mathcal{A}(\mathcal{A}) \in \{\mathcal{T}, \mathcal{F}\}$.*

Definice 2.18 *Pravdivostní ohodnocení (valuace, interpretace) $\mathcal{V}$ je funkce přiřazující každému výroku α jedinou pravdivostní hodnotu $\mathcal{V}(\alpha)$ podle pravdivostních tabulek pro logické spojky.*

Věta 2.19 *Pro danné pravdivostní přiřazení $\mathcal{A}$ existuje právě jedno pravdivostní ohodnocení $\mathcal{V}$ takové, že $V(\alpha) = A(\alpha)$ pro každý výrokový symbol α.*

Důsledek: Jestliže dvě valuace $\mathcal{V}_1, \mathcal{V}_2$ souhlasí na všech výrokových proměnných formule α, pak $\mathcal{V}_1(\alpha) = \mathcal{V}_2(\alpha)$ pro lib. fomuli α.

Platná formule $\mathcal{V}(\alpha) = \mathcal{T}$

α je (logický) důsledek množiny formulí Σ ($\Sigma \models \alpha$)

Tautologie

$\mathcal{V}$ je modelem množiny formulí Σ

1 Intro

2 Výroková logika

2.1 Výroky, spojky a pravdivostní tabulky

2.2 Přiřazeni pravdivostních hodnot. Valuace

Tabulkové du*kazy, o nichž si zde budeme povídat, jsou stromy s označenými formulemi v uzlech. Probrat všechny možnosti pro du*kaz formule není v podstatě možný (neuvažujeme matematickou proveditelnost, ale proveditelnost z hlediska automatizovaného dokazování formulí; zde uvedenou nemožnost chápeme jako extrémně vysokou složitost). Proto rozvíjíme strom s označenými formulemi tak, abychom dostali kontradikci.

2.3 Tabulkové důkazy ve výrokové logice

Definice 2.1 *Označená formule je TA nebo FA pro libovolnou formuli A.*

Definice 2.2 *Konečná tabulka je binární strom, kde v uzlech jsou označené formule, definovaný takto:*

1. *Každá atomická tabulka je konečná tabulka;*
2. *Je-li τ konečná tabulka, P cesta v tabulce τ , E je uzel na P a τ' je tabulka, která vznikne z tabulky P přidáním atomické tabulky s kořenem E na konec cesty P, pak τ' je též konečná tabulka.*

Definice 2.3

1. *Uzel E na cestě P je redukovaný, jestliže se na cestě P vyskytuje jako kořen atomické tabulky;*
2. *Cesta P je kontradiktorická, jestliže se na ní vyskytuje dvojice uzlů TA a FA pro nějaké A;*
3. *Tabulka τ se nazývá ukončená, jsou-li v ní na každé nekontradiktorické cestě všechny uzly redukované. Jinak je neukončená.*

Tabulka τ se nazývá kontradiktorická, je-li v ní každá cesta kontradiktorická.

Definice 2.4 *Tabulkový důkaz formule A je kontradiktorická tabulka s kořenem FA.*
Tabulkové vyvrácení formule A je kontradiktorická tabulka s kořenem TA.

1 Intro

2 Výroková logika

2.1 Výroky, spojky a pravdivostní tabulky

2.2 Přiřazeni pravdivostních hodnot. Valuace

2.3 Tabulkové důkazy ve výrokové logice

2.4 Rezoluce

Definice 2.1 Literál *l je výrokový symbol P nebo jeho negace $\neg P$.*
Literál s negací je negativní, bez ní je pozitivní.

Klauzule *C je konečná množina literálů (ve smyslu jejich dizjunkce).*
Klauzule je pravdivá, jestliže alespoň jeden z jejích prvků je pravdivý.
Prázdná klauzule $\square$ je vždy nepravdivá.

Formule *je (potenciálně nekonečná) množina klauzulí.*
Prázdná formule je vždy pravdivá.

2.4.1 Prolog, alternativní notace

Definice 2.2 Rezoluční pravidlo. *Buďte* $C_1 = \{p\} \sqcup C_1^{'}, C_2 = \{\neg p\} \sqcup C_2^{'}$ *klauzule. Jejich* **rezolventu** *definujeme jako klauzuli* $C = C_1^{'} \cup C_2^{'}$ *.*

C_1, C_2 ... rodiče, C ... dítě

Příklad 2.3 $\{p, q, \neg r, s\}$, $\{\neg p, q, r, t\}$

Definice 2.4 Rezoluční důkaz *klauzule* C *z množiny klauzulí* S *je konečná posloupnost klauzulí* $C_1, C_2, ..., C_n = C$ *taková, že každé* C_i *je buď prvkem* S *nebo je rezolventou klauzulí* C_j, C_k *pro* $j, k < i$.

Existuje-li rezoluční důkaz klauzule C *z množiny klauzulí* S*, říkáme, že* C *je* **(rezolučně) dokazatelná** *z* S *a píšeme* $S \vdash_R C$.

Odvození prázdné klauzule $\square$ *z* S *se nazývá* **vyvrácením** *množiny klauzulí* S.

Definice 2.5 Strom rezolučního důkazu *klauzule* C *z množiny* S *je strom* T *následujících vlastností:*

1. C *je kořenem stromu* T*;*
2. *Listy* T *jsou prvky množiny* S*;*
3. *Je-li* C_2 *uzel, který není listem, a jeho následníky jsou uzly* C_1, C_2*, pak* C_2 *je rezolventou* C_1, C_2*.*

Příklad 2.6 $\{\{p, r\}, \{q, \neg r\}, \{\neg q\}, \{\neg p, t\}, \{s, \neg t\}\}$

Věta 2.7 Korektnost a úplnost rezoluce.
Existuje-li rezoluční vyvrácění množiny klauzulí S*, pak* S *je nesplnitelná.*
Je-li množina S *nesplnitelná, pak existuje rezoluční vyvrácení* S*.*

Důsledek: Existuje-li rezoluční strom s listy z množiny S a kořenem $\square$, pak S je nesplnitelná.

2.4.2 Zjemnění rezoluce

- Vyloučení všech klauzulí, které obsahují literál, který se v S vyskytuje jen v jedné paritě;
- Odstranění tautologií, tj. klauzulí C takových, že existuje literál p, který se vyskytuje v klauzuli jako pozitivní i jako negativní.
  Příklad: $\{m, a, l, \neg a\}$;
- Sémantická rezoluce. Volba libovolného přiřazení pravdivostních hodnot A a odmítnutí rodičovských klauzulí, které jsou v něm splněny.

Každé zjemnění korektní metody je opět korektní metodou.

Uvedená zjemnění nemění nic na úplnosti zjemněné metody.

2.5 Lineární rezoluce, Hornovské klauzule, a Prolog

Definice 2.8 Lineární (rezoluční) dedukce *(důkaz)*
klauzule C z S, kde S je množina klauzulí, je posloupnost dvojic $(C_0; B_0)$, ... , $(C_n; B_n)$ *taková, že* $C_{n+1} = C$ *a*

1. C_0 *a všechna* B_i *jsou z množiny* S *nebo nějaké* C_j *takové, že* $j < i$*;*
2. *Každá* $C_{i+1}, i \leq n$ *je rezolventou* C_i *a* B_i*.*

Klauzule C *je* **lineárně odvoditelná***(dokazatelná) z* S*, jestliže existuje lineární dedukce klauzule* C *z* S*. Množinu* S *nazveme množina* **vstupních klauzulí***, klauzule Ci nazveme* **střední** *(průběžné; angl. center) klauzule a klauzule* B_j *nazveme* **boční** *(side) klauzule.*

Příklad 2.9

Definice 2.10 Prologovský program.

Hornova klauzule *je klauzule s nejvýše jedním pozitivním literálem.*
Programová klauzule *je klauzule s právě jedním pozitivním literálem.*
Pravidlo *je programová klauzule s negativními literály.*
Fakt *je programová klauzule bez negativních literálů.*
Cíl *je Hornova klauzule bez pozitivních literálů.*
Prologovský program *je množina klauzulí obsahující jen programové klauzule (pravidla nebo fakta).*

Lemma 2.1 *Je-li S množina Hornových klauzulí nesplnitelná, pak S obsahuje alespoň jeden fakt a alespoň jeden cíl.*

Důkaz: Uvažujme ohodnocení, které přiřazuje všem výrokovým symbolům TRUE. Potom je splněna každá programová klauzule (fakt nebo pravidlo). Přiřazení, které přiřazuje všem výrokovým symbolům FALSE, splňuje cílovou klauzuli a každé pravidlo. Tedy lib. nesplnitelná množina Hornových klauzulí musí obsahovat jak fakt, tak cílovou klauzuli.

Věta 2.2 Úplnost lineární rezoluce pro Hornovy klauzule.
Je-li S nesplnitelná množina Hornových klauzulí, pak existuje lineární (rezoluční) odvození $\square$ z S.

Definice 2.3 *Nechť P je množina programových klauzulí a G cílová klauzule.* **Lineární vstupní rezoluce** *(LI-rezoluce) množiny $P \cup \{G\}$ je lineární vyvrácení S začínající v G, kde všechny boční klauzule jsou z P.*

Příklad 2.4 Neúplnost LI-rezoluce.

$$S = \{\{p, q\}, \{p, \neg q\}, \{\neg p, q\}, \{\neg p, \neg q\}\}$$

Věta 2.5 *Buď P množina Hornových klauzulí a G cíl. Jestliže $S = P \cup \{G\}$ je nesplnitelná, pak existuje lineární vstupní vyvrácení S.*

Idea důkazu: cílovou klauzuli můžeme rezolvovat jen s programovou klauzulí ($\neg p$ z cílové s p z programové). Výsledkem rezoluce je opět cílová klauzule. Tj. pro lib. lineární důkaz $\Box$ z P, který začína v G platí, že všechny děti jsou cílové klauzule a všechny boční klauzule musí být programové. Stači tedy dokázat, že existuje lineární důkaz $\Box$ z P začínající v G (indukcí přes počet literálů v S).

Definice 2.6 Uspořádané klauzule *(definite clauses) jsou konečné posloupnosti literálů.*

Definice 2.7 *Je-li $P \cup \{G\}$ množina uspořádaných klauzulí, pak* **LD-rezoluční vyvrácení** $P \cup \{G\}$ *je posloupnost* $< G_0, C_0 >, \ \ldots \ , < G_n, C_n >$ *uspořádaných klauzulí* $< G_i, C_i >$ *taková, že* $G_0 = G$, $G_{n+1} = \Box$ *a*

1. *každá* G_i, $i \leq n$ *je uspořádaná cílová klauzule* $\{\neg A_{i,0}, \ \ldots \ , \neg A_{i,n(i)}\}$
2. *každá* $C_i = \{B_i, \neg B_{i,0}, \ \ldots \ , \neg B_{i,m(i)}\}$ *je programová klauzule délky* $m(i) + 2$ *z* P. *(délka* $C_i = \{B_i\}$ *je 1).*
3. *pro každé* $i < m$ *existuje rezoluce usporadanych klauzuli* G_i *a* C_i *s rezolventou jako uspořádanou klauzulí*
   $G_{i+1} = \{\neg A_{i,0}, \ \ldots \ , \neg A_{i,k-1}, \neg B_{i,0}, \ \ldots \ , \neg B_{i,m(i)}, \neg A_{i,k+1}, \ \ldots \ , \neg A_{i,n(i)}\}$
   (rezoluce pro $B_i = A_{i,k}$*)*

Lemma 2.8 *Je-li* $S = P \cup \{G\}$ *nesplnitelná, pak existuje LD-rezoluční vyvrácení* S, *které začíná klauzulí* G.

Důkaz: indukcí podle délky LI-rezolučního vyvrácení $P \cup \{G\}$

Selekční pravidlo R je funkce, která vybírá literál z uspořádané cílové klauzule.

SLD-rezoluce - lineární vstupní rezoluce se selekčním pravidlem

Definice 2.9 SLD-rezoluční vyvrácení $P \cup \{G\}$ pomocí selekčního pravidla R *je LD-rezoluční vyvrácení* $< G_0, C_0 >$, ... $< G_n, C_n >$, $G_0 = G, G_{n+1} = \square$, *kde* $R(G_i)$ *je literál z* G_n *rezolvovaný v i+1-ním kroku.*

Poznámka: Není-li R explicitně uvedeno, předpokládá se výběr nejlevějšího literálu.

Příklad 2.10

Věta 2.11 Úplnost SLD-rezoluce pro Prolog.
Je-li $P \cup \{G\}$ *nesplnitelná a* R *libovolné selekční pravidlo, potom existuje SLD-rezoluční vyvrácení* $P \cup \{G\}$ *pomocí* R*.*