AD4M33AU
Automatické uvažování

Rezoluční kalkulus pro výrokovou logiku

Petr Pudlák

Výroková logika

- Výhody
  - Jednoduchý jazyk.
  - Rozhodnutelnost dokazatelnosti i nedokazatelnosti.
  - Rychlejší algoritmy.
- Nevýhody
  - Relativně malé vyjadřovací schopnost jazyka – mnohé úlohy nelze efektivně nebo vůbec zapsat.

Jazyk výrokové logiky

v syntaxi TPTP:

- *výroková_proměnná ::= 'a' | 'b' | 'c' |... | 'aa' | 'ab' | ...*
- *výroková_formule ::=*
  výroková_proměnná
  | '(' '~' výroková_formule ')'
  | '(' výroková_formule '&' výroková_formule ')'
  | '(' výroková_formule '|' výroková_formule ')'
  | '(' výroková_formule '=>' výroková_formule ')'
  | '(' výroková_formule '<=' výroková_formule ')'
  | '(' výroková_formule '<=>' výroková_formule ')'
  | '(' výroková_formule '<~>' výroková_formule ')'

Spojky „&“ a „|“ jsou asociativní (lze psát „a & b & c“ apod.), ostatní spojky je třeba závorkovat (např. „a => b => c“ není jednoznačné). Spojka „&“ se váže silněji než „|“, lze psát např. „a & b | c & d“.

Interpretace a model

- **Interpretace** *M* je zobrazení, které přiřazuje každé výrokové proměnné význam – nepravda nebo pravda, nebo jako hodnoty 0 nebo 1.
- V rámci dané interpretace pak můžeme dosadit za proměnné dané formule φ a vyčíslit její pravdivost podle známých pravidel.
- Jestliže je formule *φ* pravdivá v dané interpretaci *M*, říkáme, že **M je model φ**, značíme: **M |= φ**
- Říkáme též, že **M splňuje φ.**

Sémantický důsledek

- Jestliže ψ platí ve všech interpretacích, ve kterých platí φ, říkáme, že
  ψ je sémantický důsledek φ.
- Značení: **φ ⊨ ψ**
- Pro množinu formulí **A** také píšeme **A ⊨ φ**, pokud φ platí ve všech interpretacích, ve kterých jsou splněny všechny formule z **A**.

Poznámky

- Pozorování: Relace „$\models$“ na formulích je reflexivní a tranzitivní.
- Je-li současně $\varphi \models \psi$ a $\psi \models \varphi$, říkáme, že tyto formule jsou sémanticky ekvivalentní nebo také ekvi-splnitelné, někdy značeno $\varphi \models\!\!\models \varphi$.
- Někdy se rozlišuje značením, že M je model formule ψ a že ψ je sémantickým důsledkem φ:
  - $M \models \psi$
  - $\varphi \models \psi$

Sémantický důsledek – příklady

- a ⊨ (a | b)
- (a & b) ⊨ a
- (a & b) ⊨ (a | b)
- b ⊨ (a ⇒ b)
- (a <=> b) ⊨ (a ⇒ b)
- (a <~> b) ⊨ (a | b)
- (a & b) ⊨ (a <=> b)

Tautologie

- Definice: Jestliže formule φ platí ve všech interpretacích, nazýváme jí **tautologie**.
- Značení: $\models \varphi$
- Příklad: $\models (a \mid \neg a)$
- Někdy se též zavádí speciální symbol (nulární logická spojka) pro tautologii „$\top$“.

Tautologie - vlastnosti

- Každá interpretace je modelem (libovolné) tautologie.
- Každá tautologie je sémantickým důsledkem libovolné formule, například
  $b \models (a \mid \neg a)$
- Důsledek: všechny tautologie jsou sématicky ekvivalentní.

Kontradikce (spor)

- Definice:
  - Jestliže formule ψ není splněná v žádné interpretaci, nazveme jí **kontradikce**, nebo **sporná formule**. Příklad: (a & ¬a)
  - Množinu formulí, která není splněná v žádné interpretaci, nazveme **spornou množinou**.
- Někdy se zavádí pro kontradikci speciální symbol (nulární logická spojka) „⊥“.

Kontradikce (spor) – vlastnosti

- Sporná formule (množina) nemá žádný model.
- Libovolná formule je sémantickým důsledkem sporné formule (množiny), například:
  $(a \,\&\, \neg a) \models b$
- Důsledek: Všechny kontradikce jsou sémanticky ekvivalentní.
- Formule je sporná právě když její negace je tautologie a naopak.

Poznámka: ⊥ a |, ⊤ a &

- Jelikož je spojka „|“ asociativní, lze se na ní dívat jako na spojku libobolné arity, kterou aplikujeme na *n* formulí (*n* může být 0). Disjunkce formulí $\varphi_1, \ldots, \varphi_n$ bude

  $\bot \mid (\varphi_1 \mid (\varphi_2 \mid (\ldots \mid \varphi_n)\ldots))$.
- Obdobně pro konjunkci: Konjunkce formulí $\varphi_1, \ldots, \varphi_n$ bude

  $\top \,\&\, (\varphi_1 \,\&\, (\varphi_2 \,\&\, (\ldots \,\&\, \varphi_n)\ldots))$;

Nevýhody sémantického zjišťování pravdivosti

- Příklad: je následující formule tautologie?

  $a \Rightarrow (a \mid a_1 \mid a_2 \mid ... \mid a_n)$
- Sémantickým zjišťováním pravdivosti (dosazením) bychom museli prozkoumat 2^{n+1} možností.
- Z „tvaru“ formule je přitom zřejmé, že se jedná o tautologii pro libovolné *n*.

Logické deduktivní kalkuly

- Mechanismy, které umožňují zjistit (odvodit) pravdivost formule **syntaktickými** prostředky, tedy pouze prací se symboly, kterými jsou formule zapsány.

Jazyk, axiomy a odvozovací pravidla logického kalkulu

Každý logický kalkulus se skládá z:

- **jazyka**, ve kterém se zapisují jeho formule;
- **axiomů**, což jsou formule, jejichž platnost v daném kalulu implicitně předpokládáme;
- **odvozovacích pravidel**, která říkají, jaké formule můžeme odvodit z axiomů, nebo z jiných již odvozených formulí.

Důkaz

Definice: Důkaz v daném logickém kalkulu je taková konečná posloupnost formulí, kde každá formule je buďto:

- jeden z axiomů, nebo
- odvozená pomocí některého logického pravidla kalkulu z předcházejících formulí v posloupnosti.

Dokazatelnost

- Definice: Řekneme, že formule ψ je dokazatelná z množiny formulí **A**, pokud existuje důkaz ψ z **A**.
- Značení: $\mathbf{A} \vdash \psi$.
- Místo $\{ \varphi_1, ..., \varphi_n \} \vdash \psi$

  píšeme jen $\varphi_1, ..., \varphi_n \vdash \psi$.
- Pozorování: Každý axiom je triviálně dokazatelný důkazem délky 1.

Příklad – Hilbertův implikativní kalkulus pro výrokovou logiku

- Schémata axiomů: Pro libovolné formule φ, ψ, ρ jsou následující formule axiomy:
  - (φ ⇒ (ψ ⇒ ρ)) ⇒ ((φ ⇒ ψ) ⇒ (φ ⇒ ρ))
  - φ ⇒ (ψ ⇒ φ)
  - (¬ψ ⇒ ¬φ) ⇒ (φ ⇒ ψ)
- Odvozovací pravidlo *modus ponens:*

$$\frac{\varphi \quad \varphi \Rightarrow \psi}{\psi}$$

- Též zapisujeme φ, φ ⇒ ψ ⊢ ψ.

Korektnost logického kalkulu

- Definice: Logický kalkulus je **korektní**, jestliže platí: pokud **A** $\vdash \psi$ pak **A** $\models \psi$.
- Tedy jestliže platí:
  - všechny axiomy jsou tautologie, a
  - pokud z formulí $\varphi_1,...,\varphi_n$ odvodíme ψ, musí platit $\varphi_1,...,\varphi_n \models \psi$.
- Čili, vše, co je dokazatelné, je pravda.
- Příklad: Hilbertův kalkulus je korektní, protože všechny jeho axiomy jsou tautologie, a pravidlo *modus* ponens je korektní:
  $\varphi, \varphi \Rightarrow \psi \models \psi$ (nakreslit tabulku).

Úplnost logického kalkulu

- Definice: Logický kalkulus je **úplný**, jestliže platí: pokud $\mathbf{A} \models \psi$ pak $\mathbf{A} \vdash \psi$.
- Čili, vše, co je pravda, je dokazatelné.
- Příklad: Hilbertův kalkulus pro výrokovou logiku je úplný.

Rezoluční kalkulus

- Rezoluční kalkulus umožňuje dokázat, že je daná množina tzv. **klauzulí** sporná, tedy zda v žádné interpretaci nelze splnit všechny dané formule.
- Úlohu je tedy nejprve nutné převést na hledání důkazu sporu z nějaké množiny klauzulí.

Jazyk rezolučního kalkulu pro výrokovou logiku

Definice:

- **Literál** je výroková proměnná nebo její negace.
- **Klauzule** je disjunkce libovolného počtu literálů.
- Příklad:

  a | b | ~b | c
- **Prázdná klauzule** je klauzule, která neobsahuje žádný literál. Tato je ekvivalentní (každé) kontradikci. Značení □, někdy též ⊥.

Literály a klauzule pro výrokovou logiku

- Klauzule se obvykle uvažují jako množiny symbolů, nikoliv jako posloupnosti.
- Následující klauzule se tedy považují za totožné:
  - a | a | b | ~c | ~c
  - b | a | ~c
  - ~c | b | b | a
- Z toho je odvozeno časté značení pro klauzule: {a, b, ~c}
- Prázdná klauzule □ se proto také někdy zapisuje jako {}.

Axiomy, odvozovací pravidlo výrokové rezoluce

- Výroková rezoluce nemá žádné axiomy (stejně tak predikátová).
- Výroková rezoluce má pouze jedno odvozovací pravidlo:

$$\frac{D \mid a \qquad \neg a \mid G}{D \mid G}$$

kde D a G jsou disjunkce libovolného počtu literálů (klauzule).

- Výsledná klauzule „D | G“ se nazývá **rezolventa**.

Více resolvent ze dvou klauzulí

- Je zřejmé, že mohou být dvě takové výrokové klauzule, na které nelze použít rezoluční pravidlo.
- Úloha:

1. Najděte příklad dvou klauzulí, na které lze použít rezoluční pravidlo dvěma různými způsoby.
2. Ukažte, že v každém takovém případě (ve výrokové logice!) je výsledná rezolventa tautologií
   (a tedy pro další dokazování zbytečná).

Korektnost rezolučního kalkulu

- $\dfrac{D \mid a \qquad \neg a \mid G}{D \mid G}$
- Buď M interpretace taková, že $M \models (D \mid a)$ a $M \models (\sim a \mid G)$.
- Pokud $M \models a$, pak musí být $M \models G$ a tedy $M \models (D \mid G)$
- Pokud $M \models \neg a$, pak musí být $M \models D$ a tedy $M \models (D \mid G)$.
- Rezoluční pravidlo je tedy korektní.

Úplnost rezolučního kalkulu

- **Pozor!** Rezoluční kalkulus není úplný v tom smyslu, že lze odvodit každou pravdivou klauzuli!
- Příklad: z jedné klauzule {*a*} nelze odvodit žádnou další klauzuli, tedy ani klauzuli {*a, b*}, přestože a $\models a \mid b$.

Úplnost rezolučního kalkulu

- Věta (o úplnosti rezolučního kalkulu):
  Buď **A** sporná množina klauzulí. Pak lze rezolučním kalkulem odvodit prázdnou klauzuli (spor).
- V angličtině se toto označuje pojmem **„refutationally complete“**.

- Úloha: Ukažte, že je-li **A** splnitelná množina klauzulí, pak:
  - výrokovým rezolučním kalkulem lze provést jen konečně mnoho odvození,
  - a žádná resolventa nebude prázdná klauzule.

Použití rezolučního kalkulu

1. Převedeme úlohu na hledání důkazu spornosti množiny formulí.
2. Formule převedeme do CNF (na klauzule).
3. Aplikujeme rezoluční kalkulus.

Metoda důkazu sporem

- Věta: **A** $\models \varphi$ právě když $A \cup \{\neg\varphi\} \models \square$ (jinak řečeno, **A**$\cup\{\neg\varphi\}$ je sporná množina formulí). (obrázek).

- Takže chceme-li zkoumat, zda **A** $\models \varphi$, převedeme problém na ekvivalentní problém zkomající, zda **A**$\cup\{\neg\varphi\}$ je sporná.

Převedení formulí na CNF

- Všechny logické spojky přepíšeme pomocí konjunkce, disjunkce a negace.
- Pomocí DeMorganových pravidel přesuneme všechny negace co nejhlouběji, až k výrokovým proměnným.
- Průběžně eliminujeme dvojité negace.
- Distributivním pravidlem roznásobíme konjunkce a disjunkce tak, aby všechny disjunkce byly uvnitř konjunkcí.

Převedení formulí na CNF

- Příklad: (na tabuli).
- Příklad: použití `eprover –cnf ....`

Strategie aplikace rezolučního pravidla

- Při použití rezoluce (i jiných důkazových mechanismů) musíme (my, nebo automatický dokazovač) volit, na které dvě klauzule použít rezoluční pravidlo.
- Snahou je vybírat takové klauzule, které nejspíš povedou ke krátkému důkazu.

Strategie aplikace rezolučního pravidla – způsob prohledávání

- **Do šířky** – nejčastější, velké nároky na paměť, ale možno dobře filtrovat odvozené klauzule (subsumpce, viz. dále); výsledný důkaz je nejkratší možný.
- **Do hloubky –** menší nároky na pamět ale větší nároky na čas (a v případě rezoluce v logice 1. řádu neúplné).
- **Unit resolution** – preferujeme klauzule, které mají jen jeden literál.

Základní optimalizace rezoluce

1. Zkrácení výsledných klauzulí při konverzi na CNF.
2. Subsumpce.
3. Omezit rezoluci jen na některé klauzule.

Optimalizace konverze na CNF

- Problém: Roznásobování & a | vede v některých případech k exponenciálnímu nárůstu formule.
- Příklad:
  $(a_1 \,\&\, b_1) \mid (a_2 \,\&\, b_2) \mid \ldots \mid (a_n \,\&\, b_n)$
  Formule pro roznásobení bude mít 2^n členů, dostaneme tedy 2^n klauzulí.

Optimalizace konverze na CNF

- $(a_1 \,\&\, b_1) \mid (a_2 \,\&\, b_2) \mid \ldots \mid (a_n \,\&\, b_n)$
- Zavedeme <u>nové</u> výrokové proměnné reprezentující jednotlivé konjunce původní formule: $z_i \Rightarrow (a_i \,\&\, b_i)$,

  zapsáno klauzulemi $\neg z_i \mid a_i$, $\neg z_i \mid b_i$.
- Takto dostaneme jen lineárně mnoho nových formulí (klauzulí).
- Původní formuli nahradíme klauzulí

  $z_1 \mid z_2 \mid \ldots \mid z_n$
- Z těchto klauzulí dovedeme odvodit všechny ty, které bychom získali roznásobením.

Subsumpce

- Definice: Jestliže $\varphi \subseteq \psi$ ve smyslu množin literálů, říkáme, že formule φ **subsumuje** formuli ψ.
  Značme $\varphi \sqsubseteq \psi$.
  - Příklad: a | ~c $\sqsubseteq$ a | b | ~c | ~d
- Tvrzení: Jestliže $\varphi \sqsubseteq \psi$ pak $\varphi \models \psi$.
  Naopak to neplatí (příklad).
- Pro množiny klauzulí **A** a **B** řekneme, že množina **A** subsumuje množinu **B** (zn. $\mathbf{A} \sqsubseteq \mathbf{B}$), pokud každou klauzuli z **B** subsumuje nějaká klauzule z **A**.
  - Příklad: { a, ~b | ~c } $\sqsubseteq$ { a | d, ~b | ~c | e, a | ~e }

Subsumpce v rezoluci

- Věta: Pokud existuje rezoluční důkaz sporu z **B** a platí-li **A**$\sqsubseteq$**B**, pak:
  - existuje rezoluční důkaz sporu z **A**;
  - ten lze mechanicky sestrojit, a
  - není delší než důkaz z **B**.
- Struktura důkazu bude stejná, jen nám „vypadnou“ některé literály a následně ta rezoluční pravidla, ve kterých jsme se těchto literálů předtím „zbavovali“.
- Použití: Pokud jsme odvodili dvě formule φ a ψ, a pokud $\varphi \sqsubseteq \psi$, ponecháme si pouze φ a ψ můžeme zahodit.

Subsumpce v rezoluci – příklad

- Mějme fragment rezolučního důkazu:

$$\dfrac{\dfrac{\dfrac{a \mid b \qquad \neg a \mid c}{b \mid c} \qquad \neg c}{b} \qquad \neg b}{\square}$$

Subsumpce v rezoluci – příklad (pokračování)

- Pokud jsme na začátku tohoto důkazu měli již odvozenu mezitím klauzuli *a*, můžeme klauzuli *a* | *b* nahradit *a* (protože platí *a* ⊑ *a* | *b*), a pak mechanicky upravit důkaz, který bude v tomto případě i o jeden krok kratší – nebude třeba provádět rezoluci s ~*b*:

a ~a | c
---
c ~c

□

Zpětná a dopředná subsumpce

- **Dopředná (forward) subsumpce:** Pokud odvodíme klauzuli ψ, a ψ je subsumována nějakou z již dříve odvozených klauzulí φ (tedy $\varphi \sqsubseteq \psi$), klauzuli ψ zahodíme. (Cokoliv bychom dokázali z ψ můžeme dokázat i z φ.)
- **Zpětná (backward) subsumpce:** Pokud odvodíme klauzuli φ, klauzulí φ nahradíme všechny doposud odvozené klauzule ψ_i, které jsou subsumovány φ (tedy $\varphi \sqsubseteq \psi_i$).

Zpětná a dopředná subsumpce – shrnutí

- Vždy **ponecháváme subsumující** klauzuli (tu s méně literály, vlevo od $\sqsubseteq$) a
- **zahazujeme subsumovanou** klauzuli (tu s více literály, vpravo od $\sqsubseteq$).

Omezení rezoluce jen na některé klauzule.

- Vhodné omezení na to, jaké klauzule „smíme“ použít pro rezoluci, omezí významně zmenší prohledávací prostor, ale zachová úplnost kalkulu.

Uspořádaná rezoluce

- Myšlenka: Abychom dospěli k prázdné klauzuli, musíme z nějaké klauzule odstranit postupně všechny literály.
- Nezávisí na pořadí, v jakém literály odstraňujeme.
- Tím, že nějaké pořadí zvolíme, omezíme množství odvozených klauzulí.

Uspořádaná rezoluce

- Zvolíme kvaziuspořádání $\preccurlyeq$ na symbolech výrokových proměnných takové, že pro dvě různé proměnné a a b neplatí současně $a \preccurlyeq b$ a $b \preccurlyeq a$.
- Jinak je volba libovolná, metoda funguje pro jakékoliv takové kvaziuspořádání.
- V případě výrokové rezoluce obvykle volíme lineární uspořádání.
- Definice: Rezoluční pravidlo

  $$\Gamma \mid A, \quad \Delta \mid \neg A \vdash \quad \Gamma \mid \Delta$$

  lze v uspořádané rezoluci použít pouze pokud pro všechny výrokové proměnné z Γ i Δ platí, že nejsou menší než A.

Lineární uspořádání, kvaziuspořádání

- Definice: **Kvaziuspořádání** $\preccurlyeq$ je reflexivní tranzitivní relace.
- Definice: **Lineární uspořádání** $\preccurlyeq$ je totální kvaziuspořádání, tedy takové, kde platí, že pro libovolné *A* a *B* platí $A \preccurlyeq B$ nebo $B \preccurlyeq A$ (obecně může platit i obojí najednou).

- Poznámky:
  - V lineárním uspořádání jsou každé dva prvky porovnatelné.

Úplnost uspořádané rezoluce

- Věta: Z každé sporné množiny klauzulí ***S*** lze uspořádanou rezolucí odvodit spor.

- Důsledek: Jestliže uspořádanou rezolucí z množiny klauzulí ***S*** odvodíme tzv. **saturovanou množinu klauzulí**, tedy množinu, ze které nelze odvodit žádné nové rezolventy, je množina ***S*** splnitelná (a dovedeme nalézt splňující interpretaci).

Úplnost uspořádané rezoluce

Důkaz: Provedeme indukcí podle počtu výrokových proměnných v ***S***.

- Neobsahuje-li S žádnou výrokovou proměnnou, je
  - sporná právě když obsahuje prázdnou klauzuli, a
  - splnitelná právě když je prázdná (libovolná interpretace je její model).
- Buď ***V*** množina výrokových proměnných vyskytující se ***S***, $|\boldsymbol{V}| > 0$. Buď $Q \in \boldsymbol{V}$ taková výroková proměnná, která není větší než žádná jiná z ***V***.

Úplnost uspořádané rezoluce

- Rozdělme $\boldsymbol{S}$ na tři disjunktní množiny:
  - $\boldsymbol{S}_0 = \{ \Gamma \in \boldsymbol{S} \mid \Gamma$ neobsahuje ani Q ani ¬Q $\}$
  - $\boldsymbol{S}_Q = \{ \Gamma \in \boldsymbol{S} \mid \Gamma$ obsahuje Q $\}$
  - $\boldsymbol{S}_{\neg Q} = \{ \Gamma \in \boldsymbol{S} \mid \Gamma$ obsahuje ¬Q $\}$
- Buď $\boldsymbol{S'} = \boldsymbol{S}_0 \cup \{$ všechny rezolventy z $\boldsymbol{S}_Q$ a $\boldsymbol{S}_{\neg Q}$ přes Q $\}$
- Jelikož *Q* není menší než žádná jiná výroková proměnná z $\boldsymbol{S}$, lze všechny rezolventy získat uspořádanou rezolucí.

Úplnost uspořádané rezoluce

- Tvrdíme: Je-li ***S*** sporná, je i ***S′*** sporná.
- Neboli, je-li ***S′*** splnitelná, je i ***S*** splnitelná.
- Když toto dokážeme, podle indukčního předpokladu existuje důkaz sporu z ***S′***, a ten rozšíříme na důkaz sporu z ***S*** přidáním odpovídajících rezolucí přes *Q*.
  Tím bude věta dokázána.

Úplnost uspořádané rezoluce

- Buď $I \models \boldsymbol{S'}$. Označme I_Q a $I_{\neg Q}$ interpretace vzniklé přidáním $Q=1$ resp. $Q=0$ k I.
- Jelikož $\boldsymbol{S_0} \subseteq \boldsymbol{S'}$, platí, že $I \models \boldsymbol{S_0}$.
- Pokud by současně $I_Q \not\models \boldsymbol{S}$ i $I_{\neg Q} \not\models \boldsymbol{S}$, musely by existovat dvě klauzule $\Gamma, \Delta \in \boldsymbol{S}$ takové, že $I_Q \not\models \Gamma$ a $I_{\neg Q} \not\models \Delta$. Ty nemohou být ze $\boldsymbol{S_0}$, a musí tedy být $\Gamma \equiv \neg Q \mid \Gamma'$ a $\Delta \equiv Q \mid \Delta'$.
- Pak ale $I \not\models \Gamma'$ a $I \not\models \Delta'$, a tedy $I \not\models \Gamma' \mid \Delta'$.
- To je spor, jelikož $\Gamma' \mid \Delta' \in \boldsymbol{S'}$.
- $\boldsymbol{S}$ je tedy splnitelná alespoň jednou z interpretací I_Q nebo $I_{\neg Q}$.

Úplnost klasické rezoluce

- Zvolíme-li degradované kvaziuspořádání, kde každá výroková proměnná je porovnatelná jen sama se sebou, není na rezoluci žádné omezení uspořádaná rezoluce s tímto kvaziuspořádáním je identická klasické rezoluci.
- Dokázali jsme tedy i úplnost klasické rezoluce.

Uspořádaná rezoluce v Prover9

- V tomto dokazovači lze nastavit uspořádanou rezoluci pomocí příznaků:

```
set(raw).
set(binary_resolution).
set(ordered_res).
```

- A zobrazit všechny operace, které provádí:

```
set(print_kept).
set(print_given).
set(print_gen).
```