6. AUTOMATSKO ZAKLJUČIVANJE

Zaključak (novoizvedeno znanje) zapravo nije ništa drugo nego **logička posljedica** premisa (postojećeg znanja)

Problem s dokazivanjem semantičke posljedice

Netraktabilnost: moramo provjeriti 2^n interpretacija

Neodlučivost: u predikatnoj logici broj interpretacija je beskonačan, pa nemamo šanse sve ih ispitati **||** Zapravo, FOL je poluodlučiva (engl. semi-decidable): možemo dokazati valjanost onih formula koje jesu valjane, ali za formule koje nisu valjane ne možemo to uvijek dokazati

---

Prednosti teorije dokaza naspram dokazivanja logičke posljedice

Učinkovitost: umjesto da iscrpno pretražujemo sve moguće interpretacije, deduktivnu posljedicu možemo brže dokazati (pogotovo ako koristimo pametnu strategiju dokazivanja). Primijetite, međutim, da ne možemo izbjeći neodlučivost FOL-a

Interpretabilnost: možemo objasniti zašto nešto slijedi iz premisa (pozivajući se na pravila zaključivanja) ⇒ dobivamo dokaz

Deduktivna posljedica

Formula G je **dedukcija** ili **deduktivna posljedicaž** formula $F_1, F_2, \ldots, F_n$ akko je G moguće izvesti iz premisa $F_1, F_2, \ldots, F_n$ pravilima zaključivanja.

Pravila zaključivanja

Ispravnost i potpunost pravila

Ispravnost => Pravilo zaključivanja je ispravno ako, primijenjeno na skup premisa, izvodi formulu koja je logička posljedica tih premisa.

ako $F_1, \ldots, F_n \vdash_r G$ onda $F_1, \ldots, F_n \vDash G$

Potpunost => Skup pravila R je potpun ako i samo ako je njime moguće izvesti sve logičke posljedice

ako $F_1, \ldots, F_n \vDash G$ onda $F_1, \ldots, F_n \vdash_R G$

Automatsko zaključivanje

Metoda rezolucije

Rezolucijsko pravilo

$$\frac{A \vee F \qquad \neg A \vee G}{F \vee G} \qquad \text{ili} \qquad A \vee F,\ \neg A \vee G \vdash F \vee G$$

Klauzula

Rezolucijsko pravilo može se primijeniti samo na disjunkcije

Ako ˇželimo primjenjivati isključivo rezolucijsko pravilo, premise trebaju biti u obliku disjunkcije. Takav oblik nazivamo **klauzula.**

Literal je atom ili njegova negacija. **Klauzula** je disjunkcija konačnog broja literala G_i:

$$G_1 \vee G_2 \vee \cdots \vee G_n,\ n \geq 0$$

Klauzula koja sadrži samo jedan literal naziva se **jedinična klauzula.**

Rezolucijsko pravilo nad PL klauzulama

$$\frac{F_1 \vee \cdots \vee F_i \vee \cdots \vee F_n \qquad\qquad G_1 \vee \cdots \vee G_j \vee \cdots \vee G_m}{F_1 \vee \cdots \vee F_{i-1} \vee F_{i+1} \vee \cdots \vee F_n \vee G_1 \vee \cdots \vee G_{j-1} \vee G_{j+1} \vee \cdots \vee G_m}$$

gdje su F_i i G_j **komplementarni literali** (jedan je negacija drugoga).

Premise nazivamo **roditeljske klauzule**, a dedukciju nazivamo **rezolventa**.

Konjunktivna normalna forma

Konjunktivna normalna forma (engl. *conjuctive normal form*, CNF)

Formula F je u **konjunktivnoj normalnoj formi** akko je F u obliku

$$F_1 \land F_2 \land \cdots \land F_n$$

pri čemu je F_i oblika

$$G_{i1} \lor G_{i2} \lor \cdots \lor G_{im}$$

gdje su G_{ij} literali (atomi ili njihove negacije).

Pretvorba formule u CNF

(1) Uklanjanje ekvivalencije: $F \leftrightarrow G \equiv (\neg F \lor G) \land (\neg G \lor F)$

(2) Uklanjanje implikacije: $F \rightarrow G \equiv \neg F \lor G$

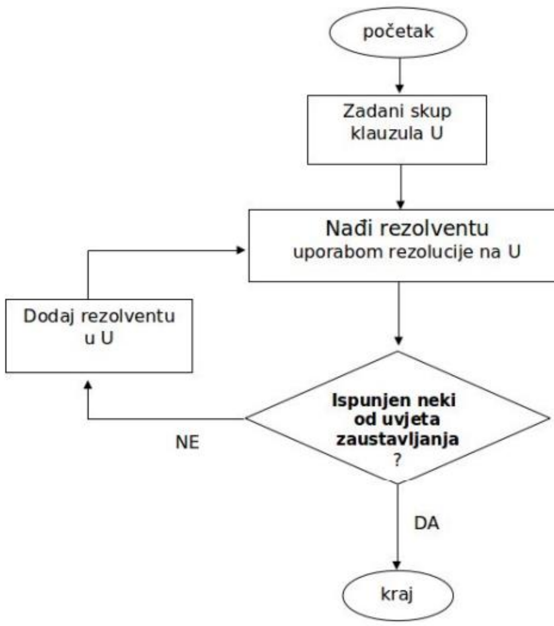
(3) Potiskivanje negacije do atoma: $\neg(F \lor G) \equiv \neg F \land \neg G$
$\neg(F \land G) \equiv \neg F \lor \neg G$

(4) Primjena distributivnosti: $F \lor (G \land H) \equiv (F \lor G) \land (F \lor H)$

Svaki se korak ponavlja sve dok je primjenjiv.
U svim koracima, kad god je to moguće, primijenjuje se ekvivalencija za involuciju $\neg\neg F \equiv F$.

Klauzalni oblik => skup skupova literala

Rezolucija



Budući da vrijedi

$$\mathbf{F \models F \lor G},$$

a da rezolucijskim pravilom ne možemo deduktivno izvesti

$$\mathbf{F \vdash F \lor G},$$

zaključujemo da rezolucijskim pravilom ne možemo dokazati sve logičke posljedice, pa zaključujemo da **rezolucijsko pravilo nije potpuno.**

Izravna rezolucija vs. rezolucija opovrgavanjem

Izravna rezolucija je nepotpuna, međutim **rezolucija opovrgavanjem je potpuna**

Rezolucija opovrgavanjem

Umjesto da dokazujemo $F_1, \ldots, F_n \vdash G$, nastojimo dokazati da je $F_1 \land \cdots \land F_n \land \neg G$ proturječna formula

NIL označava praznu klauzulu čija je semantička vrijednost ⊥

$$\frac{A \qquad \neg A}{\text{NIL}}$$

Dokazano je da, uvijek kada je skup klauzula proturječan, rezolucijom možemo izvesti klauzulu NIL

=> uvijek možemo dokazati nekonzistentnost skupa klauzula
=> možemo dokazati svaku logičku posljedicu

Znači da je rezolucija opovrgavanjem **potpuna**, zato što njome možemo dokazati bilo koju logičku posljedicu.

Rezolucija opovrgavanjem je **ispravna i potpuna**!

Faktorizacija

Faktorizacija je primjena ekvivalencije $G \lor G \equiv G$ kojom se višekratno pojavljivanje istog literala zamjenuje jednim literalom

Kako bismo zadržali **potpunost**, treba primjenjivati faktorizaciju kad god je to moguće

Algoritam rezolucije opovrgavanjem (za propozicijsku logiku)

function $\mathrm{plResolution}(F, G)$
  $clauses \leftarrow \mathrm{cnfConvert}(F \wedge \neg G)$
  $new \leftarrow \emptyset$
  loop do
    for each (c_1, c_2) **in** $\mathrm{selectClauses}(clauses)$ **do**
      $resolvents \leftarrow \mathrm{plResolve}(c_1, c_2)$
      if $\mathrm{NIL} \in resolvents$ **then return** $true$
      $new \leftarrow new \cup resolvents$
    if $new \subseteq clauses$ **then return** $false$
    $clauses \leftarrow clauses \cup new$

Broj mogućih različitih klauzula je **konačan** (ako se provodi **faktorizacija**), pa algoritam sigurno završava u **konačnom broju** koraka

Rezolucijske strategije

Strategija pojednostavljenja - strategija brisanja

Uklanjanje redundantnih klauzula => klauzula koja je pokrivena drugom klauzulom može se obrisati prema ekvivalenciji apsorpcije: $\mathbf{F \wedge (F \vee G) \equiv F}$

Uklanjanje nevažnih klauzula => klauzula koja je valjana (tautologija) je nevažna

klauzula je valjana akko sadrži komplementaran par literala $\mathbf{F_i}$ **i** $\mathbf{\neg F_i}$

Upravljačke rezolucijske strategije

Strategija zasićenja po razinama

Rezolvente izvodimo razinu po razinu (kao kod pretraživanja u ˇširinu): razrješavamo sve moguće parove klauzula na prvoj razini (početni skup klauzula), zatim na drugoj razini, itd.

Ovo je potpuna strategija, ali je vrlo neučinkovita *(problem kombinatorne eksplozije)*

Strategija skupa potpore, SoS

Temelji se na pretpostavci da je **skup ulaznih premisa *konzistentan***

Skup potpore (SoS): klauzule dobivene negacijom cilja i sve novo izvedene klauzule

<u>Barem jedna</u> roditeljska klauzula uvijek dolazi iz SoS-a