Logică computațională
Curs 10

Lector dr. Mihiș Andreea-Diana

Sistem formal (axiomatic) asociat Rezoluției predicative

- $\mathrm{Res}^{\mathrm{Pr}} = (\Sigma_{\mathrm{Res}}^{\mathrm{Pr}}, F_{\mathrm{Res}}^{\mathrm{Pr}}, A_{\mathrm{Res}}^{\mathrm{Pr}}, R_{\mathrm{Res}}^{\mathrm{Pr}})$
  - $\Sigma_{\mathrm{Res}}^{\mathrm{Pr}} = \Sigma_{\mathrm{Pr}} \setminus \{ \forall, \exists, \wedge, \rightarrow, \leftrightarrow \}$ – **alfabetul**
  - $F_{\mathrm{Res}}^{\mathrm{Pr}} \cup \{\square\}$ – **mulţimea formulelor bine-formate**
    - $F_{\mathrm{Res}}^{\mathrm{Pr}}$ mulţimea tuturor clauzelor ce se pot forma folosind alfabetul $\Sigma_{\mathrm{Res}}^{\mathrm{Pr}}$
    - $\square$ **- clauza vidă** care nu conţine nici un literal, simbolizează inconsistenţa
  - $A_{\mathrm{Res}}^{\mathrm{Pr}} = \varnothing$ – **mulțimea axiomelor**
  - $R_{\mathrm{Res}}^{\mathrm{Pr}} = \{res^{\mathrm{Pr}}, fact\}$ **mulţimea regulilor de inferenţă** care conţine:

Reguli de inferență predicative

- **regula rezoluției predicative**:

$A \vee l_1, B \vee \neg l_2 \vdash_{res}^{\Pr} \theta(A) \vee \theta(B)$,

unde $\theta = mgu(l_1, l_2)$ și $A, B \in F_{\text{Res}}^{\Pr}$

  - $C_1 = A \vee l_1$, $C_2 = B \vee \neg l_2$ clauzele care rezolvă,

    dacă literalii l_1 și l_2 sunt unificabili

  - Rezolventul binar $C_3 = \text{Res}_{\theta}^{\Pr}(C_1, C_2) = \theta(A) \vee \theta(B)$

- **regula factorizării**:

$C \vdash_{fact} C'$, C' – factor al lui C

unde $C = l_1 \vee l_2 \vee \ldots \vee l_k \vee l_{k+1} \vee \ldots \vee l_n$,

$\lambda = mgu(l_1, l_2, \ldots, l_k)$

$C' = \lambda(l_k) \vee \lambda(l_{k+1}) \vee \ldots \vee \lambda(l_n)$

Teoremă

Fie $U_1, U_2, \ldots, U_n$ şi V formule predicative.

- $\vdash V$ dacă şi numai dacă $(\neg V)^C \vdash_{res}^{Pr} \square$
- $U_1, U_2, \ldots, U_n \vdash V$ dacă şi numai dacă

$$\{U_1^C, U_2^C, \ldots, U_n^C, (\neg V)^C\} \vdash_{res} \square$$

Observație: Variabilele din clauze distincte se recomandă să fie distincte.

Algoritmul rezoluției predicative:

Date de intrare: $U_1, U_2, \ldots, U_n$, V – formule predicative

Date de ieșire: ”are loc $U_1, U_2, \ldots, U_n \vdash V$” sau ”nu are loc $U_1, U_2, \ldots, U_n \vdash V$”

Se construiește $S = \{ U_1^C, U_2^C, \ldots, U_n^{\,C}, (\neg V)^C \}$

Repetă

Se selectează literalii l_1, l_2 și clauzele C_1, C_2 astfel încât sunt clauze sau factori ai unor clauze din S

Fie $l_1 \in C_1$ și $\neg l_2 \in C_2$

Dacă l_1 și l_2 sunt unificabili cu $\theta = mgu(l_1, l_2)$

Atunci

$C = Res_{\theta}^{Pr}(C_1, C_2)$

Algoritmul rezoluției predicative - continuare

Dacă C= □

Atunci Scrie "are loc $U_1, U_2, \ldots, U_n \vdash V$"; **STOP**

Altfel $S = S \cup \{C\}$

Sfârșit_dacă

Sfârșit_dacă

Până când nu se mai pot deriva noi rezolvenți **sau** un număr fixat de iterații au fost executate

Dacă nu se mai pot deriva noi rezolvenți

Atunci Scrie "nu are loc $U_1, U_2, \ldots, U_n \vdash V$"

Altfel Scrie "nu se poate decide dacă are loc sau nu $U_1, U_2, \ldots, U_n \vdash V$"

Sfârșit_dacă

Sfârșit algoritm

Strategii și rafinări ale rezoluției predicative

- Strategii:
  - Strategia eliminării !unificarea, factorizarea
  - Strategia saturării pe nivele
  - Strategia mulțimii suport

- Rafinări:
  - Rezoluția blocării
  - Rezoluția liniară
    - input
    - unit

Completitudinea și corectitudinea

- Toate rafinările şi strategiile rezolutive păstrează completitudinea şi corectitudinea.
- Combinarea lor poate impune prea multe restricţii și deși mulțimea inițială de clauze este inconsistentă, s-ar putea să nu se poată deriva clauza vidă.
- sunt complete:
  - rezoluţia generală + strategia eliminării
  - rezoluţia generală + strategia mulţimii suport
  - rezoluţia generală + strategia mulţimii suport + strategia eliminării
  - rezoluția liniară + strategia eliminării
  - rezoluţia blocării + strategia mulţimii suport
- nu sunt complete:
  - rezoluția blocării + strategia eliminării
  - rezoluția blocării + strategia mulţimii suport
  - rezoluția blocării + rezoluția liniară
  - rezoluţia unitară
  - rezoluția de intrare

Completitudinea rezoluției de intrare

Definiții:

- O *clauză* se numește *pozitivă* dacă aceasta conține literali pozitivi.
- O *clauză* se numește *negativă* dacă aceasta conține doar literali negativi.
- O clauză se numește *clauză Horn* dacă aceasta conține un singur literal pozitiv, ceilalți fiind negativi.

Teoremă:

- Rezoluția de intrare este completă pe o mulțime de clauze Horn, cu o clauză negativă ca și clauză de vârf (PROLOG).

  - H_i: $U_1 \wedge U_2 \wedge \ldots \wedge U_n \rightarrow V$, $i \in \{1, \ldots, k\}$
  - C: $Z_1 \wedge Z_2 \wedge \ldots \wedge Z_m$?