LOGICĂ COMPUTAŢIONALĂ

Lect.univ.dr. Doru Constantin, Asist.univ.dr. Alina Florentina Ştefan

Cuprins

Capitolul 1

Introducere în limbajul propoziţiilor logice

1.1 Sintaxa limbajului

Vocabularul limbajului este $V \cup L \cup S$, unde

V este mulţimea propoziţiilor elementare; $V \neq \emptyset$. Convenim să notăm simbolurile din mulţimea V prin literele alfabetului latin.

$L = \{\wedge, \vee, \rightarrow, \leftrightarrow, \neg\}$ este mulţimea conectivelor logice: conjuncţie, disjuncţie, implicaţie, echivalenţă şi negaţie;

$S = \{(,)\}$ este mulţimea simbolurilor de punctuaţie.

Presupunem îndeplinite condiţiile $V \cap L = \emptyset$, $V \cap S = \emptyset$.

Convenim să numim asamblaje elementele mulţimii $A = (V \cup L \cup S)^*$. Pentru $\alpha \in A$ şi $x \in V \cup L \cup S$ indicăm prin $\alpha \langle x \rangle$ faptul că simbolul x apare cel puţin o dată printre simbolurile asamblajului α respectiv prin $\alpha \rangle x \langle$ situaţia contrară.

Structurile simbolice de interes în calculul cu propoziţii sunt formulele logice; mulţimea formulelor notată $FORM$.

Exemplul 1.1.1 *Fie mulţimea de şiruri binare:*

$$\left\{ \begin{array}{cccccc} (000000), & (100000), & (110000), & (111000), & (111100), & (111110), \\ (111111), & (011111), & (001111), & (000111), & (000011), & (000001), \end{array} \right\}$$

Explicaţi modul în care este posibil să se scrie această mulţime sub forma unei formule şi găsiţi o reprezentare a sa.

Fie a_i propoziţie elementară având bitul de pe poziţia "i" valoarea 1, $i = \overline{0..5}$. Evident, $\neg a_i$ are bitul de pe poziţia "i" egal cu 0. O scriere compactă a mulţimii date se poate reprezenta sub forma:

$$\bigvee_{k=0}^{5} \left(\left(\bigwedge_{i=0}^{k} \neg a_i \wedge \bigwedge_{i=k+1}^{5} a_i \right) \vee \left(\bigwedge_{i=0}^{k} a_i \wedge \bigwedge_{i=k+1}^{5} \neg a_i \right) \right).$$

Considerăm mulţimea $\{T, F\}$, elementele ei având semnificaţiile $T =$ „*adevărat*”, respectiv $F =$ „*fals*”. Pe mulţimea $\{T, F\}$ definim operaţiile logice :

$$\neg : \{T, F\} \to \{T, F\} \text{ (negaţie)}$$
$$\wedge : \{T, F\}^2 \to \{T, F\} \text{ (conjuncţie)}$$
$$\vee : \{T, F\}^2 \to \{T, F\} \text{ (disjuncţie)}$$
$$\to : \{T, F\}^2 \to \{T, F\} \text{ (implicaţie)}$$
$$\leftrightarrow : \{T, F\}^2 \to \{T, F\} \text{ (echivalenţă)}$$

prin tabelele:

$\neg$	T	F
	F	T

$\wedge$	T	F
T	T	F
F	F	F

$\vee$	T	F
T	T	T
F	T	F

$\to$	T	F
T	T	F
F	T	T

$\leftrightarrow$	T	F
T	T	F
F	F	T

Stabilirea valorii de adevăr pentru o formulă se realizează evaluând pe baza tabelelor corespunzătoare conectivelor logice, valorile de adevăr pentru subexpresiile componente în funcţie de valorile de adevăr considerate pentru propoziţiile ce intervin în formula respectivă.

Observaţia 1.1.1 *În cazul în care intervin n propoziţii în componenţa unei formule α, tabela prin care se calculează valoarea de adevăr pentru α va avea 2^n linii.*

Definiţia 1.1.1 *Spunem că formula α este **tautologie** dacă oricare ar fi valorile de adevăr corespunzătoare propoziţiilor componente ale lui α, valoarea de adevăr a formulei este adevărată.*

Definiţia 1.1.2 *Spunem că formula α este **contradicţie** dacă oricare ar fi valorile de adevăr corespunzătoare propoziţiilor componente ale lui α, valoarea de adevăr a formulei este falsă.*

Exemplul 1.1.2 *Se poate verifica cu uşurinţă că următoarele forumle sunt tautologii: $a \vee (\neg a)$, $a \to (\neg(\neg a))$, şi $(a \wedge b) \to a$, $a \to (a \vee b)$.*

Definiţia 1.1.3 *Fie α, β forumle. Dacă $a \leftrightarrow b$ este tautologie spunem că formulele α, β sunt logic echivalente.*

Exemplul 1.1.3 *Oricare ar fi α şi β formule, urmă toarele perechi de formule sunt logic echivalente:*

1) α şi $\neg(\neg\alpha)$;

2) $\alpha \to \beta$ şi $(\neg\alpha) \vee \beta$;

3) $\neg(\alpha \to \beta)$ şi $\alpha \wedge (\neg\beta)$.

Definiţia 1.1.4 *Formula α se spune că este **logic falsă** sau **contradicţie** dacă oricare ar fi valorile de adevăr corespunzătoare propoziţiilor componente, valoarea de adevăr rezultată pentru α este F.*

Observaţia 1.1.2 *Dacă α este formulă, atunci α este tautologie dacă şi numai dacă $\neg\alpha$ este logic falsă.*

O modalitate de descrierea a regulilor de bună formare pentru structurile simbolice din mulţimea $FORM$ este bazată pe noţiunea de SGF (**S**ecvenţă **G**enerativă **F**ormule)

Definiţia 1.1.5 *Secvenţa de asamblaje $\alpha_1, \alpha_2, \ldots, \alpha_n$ este* SGF*, dacă pentru orice i, $1 \leq i \leq n$, este îndeplinită una dintre condiţiile:*

(i) $\alpha_i \in V$,

(ii) există $j, 1 \leq j < i$ astfel încât $\alpha_i = (\neg\alpha_j)$,

(iii) există j, k, $1 \leq j, k < i$ şi există $\rho \in L \setminus \{\neg\}$ astfel încât $\alpha_i = (\alpha_j \rho \alpha_k)$.

Observaţia 1.1.3 Fie α asamblaj. Atunci, există $n \geq 1$ şi $\alpha_1, \alpha_2, \ldots, \alpha_n$ – SGF astfel încât $\alpha_n = \alpha$ dacă şi numai dacă $\alpha \in FORM$.

Exemplul 1.1.4 Pentru formula

$$\alpha = ((\neg\,((\neg a) \vee b)) \leftrightarrow (a \wedge (\neg b)))\,,$$

putem construi următorul SGF,

$$a, b, (\neg a)\,, (\neg b)\,, (a \wedge (\neg b))\,, ((\neg a) \vee b)\,, (\neg\,((\neg a) \vee b))\,, ((\neg\,((\neg a) \vee b)) \leftrightarrow (a \wedge (\neg b))) = \alpha.$$

Exemplul 1.1.5 Formulei

$$\beta = (\neg\,(a \vee (\neg b)) \rightarrow (a \leftrightarrow (\neg c))) \rightarrow (a \vee (\neg b))$$

îi putem construi următorul SGF,
$a, b, c, \neg b, \neg c, (a \vee (\neg b))\,, (\neg\,(a \vee (\neg b)))\,, (a \leftrightarrow (\neg c))\,, (\neg\,(a \vee (\neg b)) \rightarrow (a \leftrightarrow (\neg c)))\,, \beta.$

Observaţia 1.1.4

1. Dacă $\alpha \in V$ atunci secvenţa α este SGF, deci $V \subset FORM$;
2. Dacă $\alpha_1, \alpha_2, \ldots, \alpha_n$ este SGF, atunci pentru orice i, $1 \leq i \leq n$, $\alpha_1, \alpha_2, \ldots, \alpha_i$ este SGF. Cu alte cuvinte, toate componentele unui SGF sunt formule;
3. Dacă $\alpha_1, \ldots, \alpha_n$ şi $\beta_1, \ldots, \beta_m$ sunt SGF atunci $\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_m$ este de asemenea SGF;
4. Parantezarea rezultată pentru structurile simbolice din $FORM \setminus V$, permite identificarea conectivei principale corespunzătoare fiecărei formule care nu este propoziţie elementară. Intr-adevăr, orice $\alpha \in FORM \setminus V$ este de unul şi numai de unul din tipurile: $\alpha = (\neg\beta)$ sau $\alpha = (\beta\rho\gamma)$ cu $\rho \in L \setminus \{\neg\}$. Dacă $\alpha = (\neg\beta)$ atunci conectiva principală a formulei α este "$\neg$" şi respectiv dacă $\alpha = (\beta\rho\gamma)$ atunci α are conectiva principală "ρ".

Reprezentarea neparantezată a structurilor simbolice din mulţimea $FORM$ poate fi obţinută pe baza arborilor de structură.

Definiţia 1.1.6 *Arborele T, binar, direcţionat, cu rădăcină având vârfurile etichetate cu simboluri din mulţimea $V \cup L$ este* arbore de structură *dacă etichetele vârfurilor sunt astfel încât, pentru orice vârf $n \in V(T)$,* *(i) dacă* $od(n) = 0$ *atunci* $\varphi(n) \in V$*;*

(ii) dacă $od(n) = 1$ *atunci* $\varphi(n) = \neg$*;*

(iii) dacă $od(n) = 2$ *atunci* $\varphi(n) \in L \setminus \{\neg\}$ *unde* $\varphi(n)$ *este eticheta vârfului* n*.*

Spunem că T *este arbore de structură pentru* $\varphi(r)$ *unde* r *este vârful rădăcină.*

Fiecare formulă α poate fi reprezentată printr-un arbore de structură $T(\alpha)$. Construcţia arborelui $T(\alpha)$ poate fi realizată recursiv utilizând observaţia 1.1.4 4 astfel,

(i) dacă $\alpha \in V$ atunci

$$T(\alpha) : r$$

şi $\varphi(r) = \alpha$;

(ii) dacă $\alpha = (\neg\beta)$ atunci

$$T(\alpha) : \begin{array}{c} r \\ \downarrow \\ T(\beta) \end{array},$$

unde $\varphi(r) = \neg$ şi $T(\beta)$ este arborele de structură al formulei β;

(iii) dacă $\alpha = (\beta\rho\gamma)$ cu $\rho \in L \setminus \{\neg\}$ atunci

$$T(\alpha) : \begin{array}{ccc} & r & \\ & \swarrow \searrow & \\ T(\beta) & & T(\gamma) \end{array},$$

unde $\varphi(r) = \rho$ şi $T(\beta), T(\gamma)$ sunt arborii de structură corespunzători formulelor β respectiv γ.

Exemplul 1.1.6 Fie $\alpha = ((\neg((\neg a) \vee b)) \leftrightarrow (a \wedge (\neg b)))$.

Deoarece $\alpha = (\beta\rho\gamma)$ cu $\rho = \leftrightarrow, \beta = (\neg((\neg a) \vee b))$ şi $\gamma = (a \wedge (\neg b))$ obţinem

$$T(\alpha) : \begin{array}{ccc} & r & \\ & \swarrow \searrow & \\ T(\beta) & & T(\gamma) \end{array},$$

unde $\varphi(r) = \leftrightarrow$.

Deoarece $\beta = (\neg\beta_1)$, unde $\beta_1 = ((\neg a) \vee b)$, obţinem,

$$T(\beta) : \begin{array}{c} n_1 \\ \downarrow \\ T(\beta_1) \end{array},$$

$\varphi(n_1) = \neg$.

Repetând aceleaşi argumente obţinem,

$$T(\beta_1) : \begin{array}{ccc} & n_2 & \\ & \swarrow \searrow & \\ T(\beta_2) & & T(\beta_3) \end{array},$$

$\varphi(n_2) = \vee,\ \beta_2 = (\neg a),\ \beta_3 = b.$

$$T(\beta_2):\ \begin{array}{c} n_3 \\ \downarrow \\ T(\beta_4) \end{array},$$

$\varphi(n_3) = \neg,\ T(\beta_4) : n_4,\ \varphi(n_4) = a.$

$$T(\beta_3):\ n_5,$$

$\varphi(n_5) = b.$

$$T(\gamma):\ \begin{array}{ccc} & n_6 & \\ & \swarrow\ \searrow & \\ T(\gamma_1) & & T(\gamma_2) \end{array},$$

$\varphi(n_6) = \wedge, \gamma_1 = a, \gamma_2 = (\neg b).$

$$T(\gamma_1):\ n_7,$$

$\varphi(n_7) = a.$

$$T(\gamma_2):\ \begin{array}{c} n_8 \\ \downarrow \\ T(\gamma_3) \end{array},$$

$\varphi(n_8) = \neg.$

$$T(\gamma_3):\ n_9,$$

$\varphi(n_9) = b.$

Obţinem în final,

$$T(\alpha):\ \begin{array}{ccccc} & & r(\leftrightarrow) & & \\ & & \swarrow\ \searrow & & \\ & n_1(\neg) & & n_6(\wedge) & \\ & \downarrow & & \swarrow\ \searrow & \\ & n_2(\vee) & n_7(a) & & n_8(\neg) \\ & \swarrow\ \searrow & & & \downarrow \\ n_3(\neg) & & n_5(b) & & n_9(b) \\ \downarrow & & & & \\ n_4(a) & & & & \end{array}.$$

Exemplul 1.1.7 Arborele de structură specific formulei

$$\alpha = (a \vee (\neg b)) \to ((b \vee c) \to (\neg a \to c))$$

$$T(\alpha):\ \begin{array}{ccccc} & & r(\to) & & \\ & & \swarrow\ \searrow & & \\ & n_1(\vee) & & n_5(\to) & \\ & \swarrow\ \searrow & & \swarrow\ \searrow & \\ n_2(a) & n_3(\neg) & & n_6(\vee) & n_9(\to) \\ & \downarrow & & \swarrow\ \searrow & \swarrow\ \searrow \\ & n_7(b) & n_8(c) & n_{10}(\neg) & n_{11}(c) \\ & & & \downarrow & \\ & & & n_{11}(a) & \end{array}.$$

Pentru o structură simbolică α din sortul FORM, reprezentarea prin arborele de structură $T(\alpha)$ este unic determinată şi permite obţinerea unor reprezentări neambigui şi neparantezate pentru formulele limbajului calculului cu propoziţii şi anume reprezentarea *poloneză prefixată* şi reprezentarea *poloneză prefixată* introduse de către logicianul de origine poloneză J.Lukasiewicz. Reprezentarea conform regulilor de bună formare definite prin intermediul SGF este numită *reprezentare infixată.*

Reprezentarea *poloneză prefixată* a unei formule $\alpha \in FORM$ este secvenţa de etichete corespunzătoare vârfurilor rezultate prin traversarea r-s-d (rădăcină-subarbore stâng-subarbore drept), respectiv reprezentarea *poloneză postfixată* rezultă prin traversarea s-d-r (subarbore stâng-subarbore drept-rădăcină) a arborelui de structură $T(\alpha)$.

Exemplul 1.1.8 Prin traversarea r-s-d a arborelui de structură al formulei,

$$\alpha = ((\neg((\neg a) \vee b)) \leftrightarrow (a \wedge (\neg b))),$$

rezultă secvenţa de vârfuri :

$$r, n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8, n_9,$$

secvenţa etichetelor corespunzătoare lor fiind

$$\leftrightarrow \neg \vee \neg ab \wedge a \neg b\alpha,$$

deci reprezetarea α_{prefix}. Prin traversarea s-d-r a arborelui de structură rezultă secvenţa de vârfuri

$$n_4, n_3, n_5, n_2, n_1, n_7, n_9, n_8, n_6, r,$$

secvenţa etichetelor corespunzătoare lor fiind

$$a \neg b \vee \neg ab \neg \wedge \leftrightarrow,$$

deci reprezentarea $\alpha_{postfix}$.

Complexitatea structurală a formulelor este exprimată prin intermediul valorilor funcţiei adâncime.

Definiţia 1.1.7 *Funcţia adâncime* $\hbar$*: FORM*$\longrightarrow N$ *este definită prin:* $\alpha \in FORM$,

$$\hbar(\alpha) = \begin{cases} 0, & dacă\ \alpha \in V, \\ 1 + \hbar(\beta), & dacă\ \alpha = (\neg\beta), \\ 1 + \max\{\hbar(\beta), \hbar(\gamma)\}, & dacă\ \alpha = (\beta\rho\gamma), \rho \in L \setminus \{\neg\}. \end{cases}$$

Exemplul 1.1.9 Conform cu Definiţia **??**, pentru

$$\alpha = ((\neg((\neg a) \vee b)) \leftrightarrow (a \wedge (\neg b))),$$

obţinem,

$$\hbar(\alpha) = 1 + \max\{\hbar(\beta), \hbar(\gamma)\},$$

unde $\beta = (\neg((\neg a) \vee b)), \gamma = (a \wedge (\neg b))$

$$\begin{aligned} \hbar(\beta) &= 1 + \hbar(((\neg a) \vee b)) = 2 + \max\{\hbar((\neg a)), \hbar(b)\} = \\ &= 2 + \max\{1 + \hbar(a), 0\} = 2 + \max\{1, 0\} = 3 \end{aligned}$$

$$\hbar(\gamma) = 1 + \max\{\hbar(a), \hbar((\neg b))\} = 1 + \max\{0, 1 + \hbar(b)\} = 2$$

Rezultă $\hbar(\alpha) = 4$.

Observaţia 1.1.5 Pentru orice $\alpha \in FORM$, valoarea $\hbar(\alpha)$ exprimă numărul conectivelor logice (simbolurile din mulţimea L) care apar în structura simbolică α. De asemenea, $\hbar(\alpha)$ este înălţimea arborelui de structură corespunzător formulei α.

1.2 Abordarea axiomatică în formalizarea raţionamentelor

Modelarea raţionamentelor în contextul limbajului calculului cu propoziţii presupune alegerea unui sistem de axiome şi definirea unui sistem de demonstraţie în termenii unei mulţimi de reguli de inferenţă.

Fie a, b, c propoziţii elementare. Mulţimea axiomelor teoriei, notată $AXIOM$ este, $AXIOM = \{\overline{\alpha}_1, \overline{\alpha}_2, \ldots, \overline{\alpha}_2\}$ unde,

$$\begin{aligned}
&\overline{\alpha}_1 = (a \to (b \to a)) \\
&\overline{\alpha}_2 = ((a \to (a \to b)) \to (a \to b))\,, \\
&\overline{\alpha}_3 = ((a \to b) \to ((b \to c) \to (a \to c)))\,, \\
&\overline{\alpha}_4 = ((a \leftrightarrow b) \to (a \to b))\,, \\
&\overline{\alpha}_5 = ((a \leftrightarrow b) \to (b \to a))\,, \\
&\overline{\alpha}_6 = ((a \to b) \to ((b \to a) \to (a \leftrightarrow b)))\,, \\
&\overline{\alpha}_7 = (((\neg a) \to (\neg b)) \to (b \to a))\,, \\
&\overline{\alpha}_8 = ((a \vee b) \leftrightarrow ((\neg a) \to b))\,, \\
&\overline{\alpha}_9 = ((a \wedge b) \leftrightarrow (\neg ((\neg a) \vee (\neg b))))\,.
\end{aligned}$$

Evident, $AXIOM \subset FORM$

Definiţia 1.2.1 *Se numeşte* substituţie *o mulţime finită de perechi*

$$\sigma = \{\gamma_1 \mid a_1, \ldots, \gamma_n \mid a_n\}\,,$$

unde $\gamma_1, \ldots, \gamma_n \in FORM$*,* $a_1, \ldots, a_n \in V$ *astfel încât pentru orice* $i \neq j$*,* $a_i \neq a_j$*. Formulele* $\gamma_1, \ldots, \gamma_n$ *se numesc* formule substituente*, respectiv* $a_1, \ldots, a_n$ *se numesc propoziţii substituite. Notăm cu* ε *substituţia vidă,* $\varepsilon = \emptyset$*. Notăm cu* $SUBST$ *mulţimea substituţiilor.*

Definiţia 1.2.2 *Fie* $\alpha \in FORM$*,* $\sigma \in SUBST$*. Se numeşte* rezultat *al aplicării substituţiei* σ *formulei* α*, structura simbolică* $\alpha\sigma$ *calculată astfel:*

$\alpha\sigma =$
  if $\sigma = \varepsilon$ **then** α
    else if $\sigma = \{\gamma_1 \mid a_1, \ldots, \gamma_n \mid a_n\}$ **then**
      if $\alpha \in V \setminus \{a_1, \ldots, a_n\}$ **then** α
        else if $\alpha = a_i$ **then** γ_i
          else if $\alpha = (\neg\beta)$ **then** $(\neg\beta\sigma)$
            else if $\alpha = (\beta\rho\delta)$ **pentru anume** $\rho \in L \setminus \{\neg\}$ **then** $(\beta\sigma\rho\delta\sigma)$

Observaţia 1.2.1 Pentru orice $\alpha \in FORM$, $\sigma \in SUBST$, $\alpha\sigma \in FORM$. Dacă $\alpha \in AXIOM$ atunci $\alpha\sigma$ se numeşte exemplu de axiomă (instanţiere a axiomei α).

Exemplul 1.2.1 Fie $\alpha = ((a \wedge b) \to c)$ şi $\sigma = \{((\neg a) \vee p) \mid a, c \mid b, (a \vee x) \mid p, b \mid q\}$. Rezultă,

$$\alpha\sigma = ((a \wedge b)\,\sigma \to c\sigma) = ((a\sigma \wedge b\sigma) \to c\sigma) = ((((\neg a) \vee p) \wedge c) \to c)\,,$$

deoarece $a\sigma = ((\neg a) \vee p)$, $b\sigma = c$, $c\sigma = c$.

Definiţia 1.2.3 *Numim* regulă de inferenţă *orice relaţie* $\Re \subset FORM^p \times FORM^q$ *unde* p, q *sunt numere naturale; perechea* (p, q) *defineşte tipul relaţiei.*

Dacă $(\{\alpha_1,\ldots,\alpha_p\},\{\beta_1,\ldots,\beta_q\}) \in \Re$, spunem că $\{\beta_1,\ldots,\beta_q\}$ se deduce din $\{\alpha_1,\ldots,\alpha_p\}$ pe baza regulei de inferenţa $\Re$. Pentru simplificarea notaţiei, convenim sa reprezentăm $(\{\alpha_1,\ldots,\alpha_p\},\{\beta_1,\ldots,\beta_q\}) \in \Re$ prin $\frac{\alpha_1,\ldots,\alpha_p}{\beta_1,\ldots,\beta_q}\Re$. Mulţimile de formule $\{\alpha_1,\ldots,\alpha_p\}$ şi $\{\beta_1,\ldots,\beta_q\}$ sunt referite ca *premisă* şi respectiv *concluzie* ale regulei $\Re$.

Modelarea proceselor de raţionament în contextul limbajului calculului cu propoziţii este realizată pe baza a două reguli de inferenţă şi anume, regula substituţiei SUB de tip $(1,1)$ şi regula modus ponens MP de tip $(2,1)$, definite prin,

$$\begin{aligned} SUB &= \{(\alpha,\beta) \mid \alpha \in FORM, \exists\sigma \in SUBST, \beta = \alpha\sigma\}, \\ MP &= \{(\{\alpha,(\alpha \to \beta)\},\beta) \mid \alpha,\beta \in FORM\}. \end{aligned}$$

Dacă $\alpha \in AXIOM$ şi $\sigma \in SUBST$, convenim să numim $\alpha\sigma$ exemplu de axiomă α sau instanţiere a axiomei α.

Definiţia 1.2.4 *Secvenţa de formule $\alpha_1,\ldots,\alpha_n$ este o demonstraţie formală dacă pentru orice i, $1 \le i \le n$ este îndeplinită una din condiţiile,*

(i) α_i este instanţiere a unei axiome;

(ii) $\exists j,\exists k,\ 1 \le j,\ k < i$ astfel încât $(\{\alpha_j,\alpha_k\},\alpha_i) \in MP$.

Definiţia 1.2.5 *Spunem că $\alpha \in FORM$ este* formulă demonstrabilă (teoremă) *şi notăm $\vdash \alpha$, dacă există $n \ge 1$ şi $\alpha_1,\ldots,\alpha_n$ demonstraţie formală astfel încât $\alpha_n = \alpha$. Notăm cu T_h mulţimea teoremelor;*

$$T_h = \{\alpha \mid \alpha \in FORM, \vdash \alpha\}.$$

Observaţia 1.2.2 Dacă $\alpha_1,\ldots,\alpha_n$ şi $\beta_1,\ldots,\beta_m$ sunt demonstraţii formale, atunci

$$\alpha_1,\ldots,\alpha_n,\beta_1,\ldots,\beta_m$$

este o demonstraţie formală. În consecinţă, putem include direct teoreme ca etape într-o demonstraţie formală.

Exemplul 1.2.2 Pentru orice $\alpha \in FORM, \vdash (\alpha \to \alpha)$.
Într-adevăr, următoarea secvenţă de formule este o demonstraţie formală pentru $(\alpha \to \alpha)$,

$$\begin{aligned} &\beta_1 = (\alpha \to (\alpha \to \alpha)) = \overline{\alpha}_1\{\alpha \mid a,\ \alpha \mid b\}, \\ &\beta_2 = ((\alpha \to (\alpha \to \alpha)) \to (\alpha \to \alpha)) = \overline{\alpha}_2\{\alpha \mid a,\ \alpha \mid b\}, \\ &\beta_3 = (\alpha \to \alpha),\ \frac{\beta_1,\beta_2}{\beta_3}MP. \end{aligned}$$

Exemplul 1.2.3 Pentru orice $\alpha \in FORM, \vdash (\alpha \vee (\neg\alpha))$.

Considerăm demonstraţia formală,

$$\begin{aligned} &\beta_1 = ((\alpha \vee (\neg\alpha)) \leftrightarrow ((\neg\alpha) \to (\neg\alpha))) = \overline{\alpha}_8\{\alpha \mid a,\ (\neg\alpha) \mid b\}, \\ &\beta_2 = (((\alpha \vee (\neg\alpha)) \leftrightarrow ((\neg\alpha) \to (\neg\alpha))) \to (((\neg\alpha) \to (\neg\alpha)) \to (\alpha \vee (\neg\alpha)))) = \\ &\quad = \overline{\alpha}_5\{(\alpha \vee (\neg\alpha)) \mid a,\ ((\neg\alpha) \to (\neg\alpha)) \mid b\}, \\ &\beta_3 = (((\neg\alpha) \to (\neg\alpha)) \to (\alpha \vee (\neg\alpha))),\ \frac{\beta_1,\beta_2}{\beta_3}MP. \\ &\beta_4 = ((\neg\alpha) \to (\neg\alpha)),\ \vdash \beta_4, \\ &\beta_5 = (\alpha \vee (\neg\alpha)),\ \frac{\beta_4,\beta_3}{\beta_5}MP. \end{aligned}$$

Observaţia 1.2.3 Mulţimea teoremelor este nevidă.

Definiţia 1.2.6 *Formula α este logic falsă dacă* $(\neg\alpha) \in T_h$.

1.3 Deductibilitatea sub o familie de ipoteze

Frecvent, în cazul raţionamentelor sunt implicate cunoştinţe suplimentare care sunt reprezentate prin formule care nu sunt în mod necesar demonstrabile. Formulele reprezentând cunoştinţe suplimentare sunt referite ca ipoteze, modelarea raţionamentelor în care ipotezele sunt acceptate ca etape ale procesului deductiv se realizează pe baza conceptelor de deductibilitate simplă şi respectiv globală sub o familie de ipoteze.

Definiţia 1.3.1 *Fie $H \subset FORM$. Secvenţa de formule $\alpha_1, \ldots, \alpha_n$ este o* H-secvenţă deductivă *dacă pentru orice i, $1 \leq i \leq n$ este indeplinită una din condiţiile:*

(i) $\alpha_i \in H \cup T_h$

(ii) $\exists j,\ \exists k,\ 1 \leq j,\ k < i$ *astfel încât* $\frac{\alpha_j, \alpha_k}{\alpha_i} MP$.

Spunem că formula α este *deductibilă* sub ipotezele H, notat $H \vdash \alpha$, dacă există $n \geq 1$ şi există o H-secvenţă deductivă $\alpha_1, \ldots, \alpha_n$ astfel încât $\alpha_n = \alpha$.

Definiţia 1.3.2 *Mulţimea* $T(H) = \{\alpha \mid \alpha \in FORM, H \vdash \alpha\}$ *este* mulţimea de deductibilitate *corespunzătoare mulţimii de ipoteze H.*

Definiţia 1.3.3 *Mulţimea de formule H este compatibilă dacă $T(H) \neq FORM$. Dacă $T(H) = FORM$ atunci H este incompatibilă.*

Observaţia 1.3.1

1. Pentru orice $H \subset FORM$, $H \cup T_h \subset T(H)$;
2. Dacă $H_1, H_2 \subset FORM$ şi $H_1 \subset H_2$ atunci $T(H_1) \subset T(H_2)$;
3. $T(\emptyset) = T(T_h) = T_h$;
4. Pentru orice $H \subset FORM, T(T(H)) = T(H)$;
5. Dacă $\alpha_1, \ldots, \alpha_n$ şi $\beta_1, \ldots, \beta_m$ sunt H-secvenţe deductive, atunci

$$\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_m$$

   este de asemenea o H-secvenţă deductivă. În consecinţă, pentru simplificare putem conveni să includem direct formule din $T(H)$ ca etape într-o H-secvenţă deductivă.

Din observaţiile precedente, notând cu $\wp(FORM)$ mulţimea părţilor mulţimii $FORM$, rezultă că

$$\Im : \wp(FORM) \to \wp(FORM)$$

este un operator de închidere şi cum pentru orice $H \subset FORM$, evident $T_h \subset T(H)$, rezultă că T_h este cel mai mic punct fix al operatorului $\Im$ (în raport cu relaţia de incluziune).

Lema 1.3.1 *Fie $H \subset FORM$. Pentru orice $\alpha \in FORM$, $\alpha \in T(H)$ dacă şi numai dacă există $H_0(\alpha) \subset H$, $H_0(\alpha)$ mulţime finită astfel încât $\alpha \in T(H_0(\alpha))$*

Demonstraţie. Evident, dacă $H_0(\alpha) \subset H$, $H_0(\alpha)$ mulţime finită astfel încât $\alpha \in T(H_0(\alpha))$, cum $T(H_0(\alpha)) \subset T(H)$, obţinem $\alpha \in T(H)$.

Dacă $\alpha \in T(H)$ atunci există o H-secvenţă deductivă pentru α, fie aceasta

$$\alpha_1, \ldots, \alpha_n = \alpha.$$

Evident $\alpha_1 \in T_h \cup H$, deci

$$\{\alpha_1, \ldots, \alpha_n\} \cap (T_h \cup H) \neq \emptyset.$$

Notând $H_0(\alpha) = H \cap \{\alpha_1, \ldots, \alpha_n\}$ rezultă imediat că $\alpha_1, \ldots, \alpha_n$ este o $H_0(\alpha)$-secvenţă deductivă deci $\alpha \in T(H_0(\alpha))$. ■

Observaţia 1.3.2 Concluzia stabilită de lema 1.3.1 evidenţiază caracterul finit al proprietăţii de a fi deductibilită sub o familie de ipoteze.

Lema 1.3.2 *Fie $H = \{\alpha\}$. Atunci, $T(H) = \{\beta \mid \beta \in FORM, \vdash (\alpha \to \beta)\}$.*

Demonstraţie. Pentru $\beta \in FORM$, notăm $\gamma = (\alpha \to \beta)$. Evident, dacă $\beta = \alpha$, atunci secvenţa $\gamma,\ \alpha,\ \beta$ este o H-secvenţă deductivă, deci $\beta \in T(H)$.

Demonstrăm prin inducţie asupra lungimii H-secvenţei deductive că dacă $\beta \in T(H)$, atunci $\vdash \gamma$. Notăm cu $l(\beta)$ lungimea minimă a unei H-secvenţe deductive pentru β.

Dacă $l(\beta) = 1$ atunci β este H-secvenţă deductivă, deci $\beta = \alpha$ sau $\vdash \beta$.

Dacă $\beta = \alpha$, rezultă evident $\vdash (\alpha \to \beta)$ (exemplul 1).

Dacă $\vdash \beta$, atunci secvenţa

$$\beta,\ (\beta \to (\alpha \to \beta)) = \overline{\alpha}_1 \{\beta \mid a,\ \alpha \mid b\},\ (\alpha \to \beta)$$

este demonstraţie formală, deci $\vdash (\alpha \to \beta)$.

Presupunem că pentru orice $\beta \in FORM$ astfel încât $l(\beta) \leq n$ rezultă $\vdash (\alpha \to \beta)$. Fie $\beta \in FORM$ astfel încât $l(\beta) \leq n + 1$ şi $\beta_1, \ldots, \beta_n, \beta_{n+1} = \beta$ o H-secvenţă deductivă. Deoarece pentru orice $p \leq n$, $\beta_1, \ldots, \beta_p$ este H-secvenţă deductivă, rezultă $\vdash (\alpha \to \beta_p), 1 \leq p \leq n$.

Dacă $\beta_{n+1} \in T_h \cup H$, evident rezultă $\vdash (\alpha \to \beta_{n+1})$, adică $\vdash (\alpha \to \beta)$.

Rămâne de analizat cazul în care β_{n+1} rezultă prin regula *modus ponens* aplicată unei perechi de formule de ranguri mai mici sau cel mult egale cu n. Fie i, j astfel încât $1 \leq i, j \leq n$ şi $\beta_{j=}(\beta_i \to \beta_{n+1})$. Aplicând ipoteza inductivă, rezultă $\vdash (\alpha \to \beta_i)$ şi $\vdash (\alpha \to (\beta_i \to \beta_{n+1}))$.

Rezultă următoarea demonstraţie formală:

$\gamma_1 = (\alpha \to \beta_i)\,,$
$\gamma_2 = (\alpha \to (\beta_i \to \beta_{n+1}))\,,$
$\gamma_3 = ((\alpha \to \beta_i) \to ((\beta_i \to \beta_{n+1}) \to (\alpha \to \beta_{n+1}))) = \overline{\alpha}_3\,\{\alpha \mid a,\ \beta_i \mid b,\ \beta_{n+1} \mid c\}\,,$
$\gamma_4 = ((\beta_i \to \beta_{n+1}) \to (\alpha \to \beta_{n+1}))\,,\ \dfrac{\gamma_1,\gamma_3}{\gamma_4}MP,$
$\gamma_5 = ((\alpha \to (\beta_i \to \beta_{n+1})) \to (((\beta_i \to \beta_{n+1}) \to (\alpha \to \beta_{n+1})) \to (\alpha \to (\alpha \to \beta_{n+1}))))$
$\quad = \overline{\alpha}_3\,\{\alpha \mid a,\ (\beta_i \to \beta_{n+1}) \mid b,\ (\alpha \to \beta_{n+1}) \mid c\}\,,$
$\gamma_6 = (((\beta_i \to \beta_{n+1}) \to (\alpha \to \beta_{n+1})) \to (\alpha \to (\alpha \to \beta_{n+1})))\,,\ \dfrac{\gamma_2,\gamma_5}{\gamma_6}MP,$
$\gamma_7 = (\alpha \to (\alpha \to \beta_{n+1}))\,,\ \dfrac{\gamma_4,\gamma_6}{\gamma_7}MP,$
$\gamma_8 = ((\alpha \to (\alpha \to \beta_{n+1})) \to (\alpha \to \beta_{n+1})) = \overline{\alpha}_2\,\{\alpha \mid a,\ \beta_{n+1} \mid b\}\,,$
$\gamma_9 = (\alpha \to \beta_{n+1})\,,\ \dfrac{\gamma_7,\gamma_8}{\gamma_9}MP,$

deci $\vdash (\alpha \to \beta_{n+1})$ ■

Unul dintre cele mai importante rezultate în deductibilitatea sub o familie de ipoteze este *teorema deducţiei* (Herbrand).

Teorema 1.3.1 (Teorema deducţiei) *Fie $H \subset FORM$, $\alpha, \beta \in FORM$. Atunci, $H \cup \{\alpha\} \vdash \beta$ dacă şi numai dacă $H \vdash (\alpha \to \beta)$.*

Demonstraţie. Presupunem că $H \vdash (\alpha \to \beta)$ deci $H \cup \{\alpha\} \vdash (\alpha \to \beta)$. Evident $\alpha, (\alpha \to \beta)$, β este o $H \cup \{\alpha\}$-secvenţă deductivă deci $H \cup \{\alpha\} \vdash \beta$.

Demonstrăm prin inducţie asupra lungimii $H \cup \{\alpha\}$-secvenţei deductive că, dacă β admite o $H \cup \{\alpha\}$-secvenţă deductivă, atunci $H \vdash (\alpha \to \beta)$.

Dacă β admite o $H \cup \{\alpha\}$-secvenţă deductivă de lungime $l(\beta) = 1$, atunci $\beta \in T_h \cup H \cup \{\alpha\}$.

1. Dacă $\beta = \alpha$, cum $\vdash (\alpha \to \alpha)$ rezultă
$$H \vdash (\alpha \to \alpha)\,.$$
2. Dacă $\beta \in T_h$ atunci
$$\beta, (\beta \to (\alpha \to \beta))\,,$$
$(\alpha \to \beta)$ este o demonstraţie formală, deci
$$\vdash (\alpha \to \beta)$$
ceea ce implică în particular
$$H \vdash (\alpha \to \beta)\,.$$
3. Dacă $\beta \in H$ atunci
$$\beta, (\beta \to (\alpha \to \beta))\,,\ (\alpha \to \beta)$$
este o H-secvenţă deductivă deci
$$H \vdash (\alpha \to \beta)\,.$$

Presupunem că proprietatea este adevărată pentru toate formulele ce admit $H \cup \{\alpha\}$-secvenţe deductive de lungimi cel mult egale cu n. Fie $\beta_1, \ldots, \beta_n, \beta_{n+1} = \beta$ o $H \cup \{\alpha\}$-secvenţă deductivă. Evident pe baza ipotezei inductive rezultă $H \vdash (\alpha \to \beta_i)$, $1 \le i \le n$.

Dacă $\beta_{n+1} \in T_h \cup H \cup \{\alpha\}$, atunci pe baza argumentului precedent rezultă

$$H \vdash (\alpha \to \beta_{n+1})\,.$$

Dacă β_{n+1} rezultă prin aplicarea regulei *modus ponens* formulelor $\beta_i, \beta j$, cu $1 \le i, j \le n$, atunci $\beta_j = (\beta_i \to \beta_{n+1})$. În acest caz, se observă cu uşurinţă că secvenţa $\gamma_1 - \gamma_9$ din demonstraţia lemei precedente este o H-secvenţă deductivă, deci $H \vdash (\alpha \to \beta_{n+1})$. ■

În cazul particular al mulţimilor finite de ipoteze, poate fi stabilit următorul corolar al teoremei deducţiei.

Corolarul 1.3.1 *Fie $H = \{\alpha_1, \ldots, \alpha_n\}$ o mulţime finită de ipoteze. Pentru orice $\beta \in FORM$, $H \vdash \beta$ dacă şi numai dacă*

$$\vdash (\alpha_1 \to (\alpha_2 \to (\cdots \to (\alpha_n \to \beta))))\,.$$

Demonstraţie. Fie $\gamma_n = (\alpha_n \to \beta)\,, \gamma_k = (\alpha_k \to \gamma_{k+1})\,; k = n-1, \ldots, 1$ deci

$$\gamma_1 = (\alpha_1 \to (\alpha_2 \to (\cdots \to (\alpha_n \to \beta))))\,.$$

Presupunem că $H \vdash \beta$ deci $\{\alpha_1, \ldots, \alpha_{n-1}\} \cup \{\alpha_n\} \vdash \beta$.

Utilizând teorema deducţiei obţinem $\{\alpha_1, \ldots, \alpha_{n-1}\} \vdash (\alpha_n \to \beta) = \gamma_n$.

Iterând acelaşi argument obţinem $\{\alpha_1, \ldots, \alpha_k\} \vdash (\alpha_{k+1} \to \gamma_{k+2}) = \gamma_{k+1}; k = n-1, \ldots, 1$. Pentru $k = 1$ rezultă $\{\alpha_1\} \vdash (\alpha_2 \to \gamma_3) = \gamma_2$ deci pe baza concluziei stabilite de Lema 1.3.2 obţinem $\vdash (\alpha_1 \to \gamma_2) = \gamma_1$

Reciproc, dacă $\vdash \gamma_1 = (\alpha_1 \to \gamma_2)$ atunci din Lema 1.3.2 rezultă $\{\alpha_1\} \vdash \gamma_2 = (\alpha_2 \to \gamma_3)$. Din teorema deducţiei rezultă $\{\alpha_1, \alpha_2\} \vdash \gamma_3 = (\alpha_3 \to \gamma_4)$.

Iterând acelaşi argument obţinem în continuare $\{\alpha_1, \ldots, \alpha_k\} \vdash \gamma_{k+1} = (\alpha_{k+1} \to \gamma_{k+2})\,,$ $k = 1, \ldots, n-1$.

Pentru $k = n-1$ rezultă

$$\{\alpha_1, \ldots, \alpha_{n-1}\} \vdash \gamma_n = (\alpha_n \to \beta)$$

din care pe baza teoremei deducţiei obţinem în final $\{\alpha_1, \ldots, \alpha_n\} \vdash \beta$. ■

Corolarul teoremei deducţiei oferă o modalitate de a stabili că anumite formule sunt demonstrabile, şi anume permite reducerea problemei determinării unei demonstraţii formale la determinarea unei secvenţe deductive. În particular, deoarece o regulă de inferentă $\Re$ de tipul $(p, 1)$, $p \ge 1$ exprimă în fapt ideea că formula din partea de concluzie a regulei este „direct” deductibilă sub mulţimea de ipoteze reprezentată de mulţimea premisă, rezultă că $((\alpha_1, \ldots, \alpha_p)\,, \beta) \in \Re$ poate fi reprezentată $\{\alpha_1, \ldots, \alpha_p\} \vdash \beta$, sau echivalent, pe baza corolarului teoremei deducţiei,

$$\vdash (\alpha_1 \to (\alpha_2 \to (\cdots \to (\alpha_n \to \beta))))\,.$$

În particular deoarece pentru orice $\alpha, \beta \in FORM$, $(\{\alpha, (\alpha \to \beta)\}\,, \beta) \in MP$ rezultă $\{\alpha, (\alpha \to \beta)\} \vdash \beta$ deci, pe baza corolarului teoremei deducţiei obţinem următoarele concluzii:

$$\begin{aligned}
&\vdash (\alpha \to ((\alpha \to \beta) \to \beta))\,,\\
&\vdash ((\alpha \to \beta) \to (\alpha \to \beta))\,,\\
&\{\alpha\} \vdash ((\alpha \to \beta) \to \beta)\,,\\
&\{(\alpha \to \beta)\} \vdash (\alpha \to \beta)\,.
\end{aligned}$$

În continuare vor fi stabilite câteva reguli (scheme) de inferenţă auxiliare a căror utilizare facilitează construirea demonstraţiilor formale şi a secvenţelor deductive.

Corolarul 1.3.2 *Dacă* $\alpha, \beta \in FORM$. *atunci* $\{\alpha\} \vdash \beta$ *dacă şi numai dacă* $(\alpha \to \beta) \in T_h$.

Demonstraţie. Evident $T_h = T(\emptyset)$ deci pentru $H = \emptyset$ din teorema deducţiei obţinem, $\emptyset \cup \{\alpha\} \vdash \beta$ dacă şi numai dacă $\emptyset \vdash (\alpha \to \beta)$ adică $\{\alpha\} \vdash \beta$ dacă şi numai dacă $(\alpha \to \beta) \in T_h$. ■

1.4 Aplicaţii la raţionamentul axiomatic

1.4.1 Schema silogismului (RS)

Pentru orice $\alpha, \beta, \gamma \in FORM, \{(\alpha \to \beta), (\beta \to \gamma)\} \vdash (\alpha \to \gamma)$.
Deoarece

$$\vdash \overline{\alpha}_3 \{\alpha \mid a,\ \beta \mid b,\ \gamma \mid c\} = ((\alpha \to \beta) \to ((\beta \to \gamma) \to (\beta \to \gamma)))$$

utilizând corolarul teoremei deducţiei obţinem

$$\{(\alpha \to \beta), (\beta \to \gamma)\} \vdash (\alpha \to \gamma).$$

Schema (regula) silogismului este reprezentată convenţional $\frac{(\alpha \to \beta), (\beta \to \gamma)}{(\alpha \to \gamma)} RS$.

1.4.2 Schema "trecerii" de la implicaţie la echivalenţă (IE)

Pentru orice $\alpha, \beta, \gamma \in FORM$,

$$\{(\alpha \to \beta), (\beta \to \alpha)\} \vdash (\alpha \leftrightarrow \beta).$$

Deoarece

$$\vdash \overline{\alpha}_6 \{\alpha \mid a,\ \beta \mid b\} = ((\alpha \to \beta) \to ((\beta \to \alpha) \to (\alpha \leftrightarrow \beta)))$$

utilizând corolarul teoremei deducţiei obţinem

$$\{(\alpha \to \beta), (\beta \to \alpha)\} \vdash (\alpha \leftrightarrow \beta).$$

Schema trecerii de la implicaţie la echivalenţă este reprezentată $\frac{(\alpha \to \beta), (\beta \to \alpha)}{(\alpha \leftrightarrow \beta)} IE$.

1.4.3 Schema permutării premiselor (PP)

Pentru orice $\alpha, \beta, \gamma \in FORM$,

$$\vdash ((\alpha \to (\beta \to \gamma)) \to (\beta \to (\alpha \to \gamma))).$$

Fie $H = \{\alpha, \beta, (\alpha \to (\beta \to \gamma))\}$. Evident, secvenţa $\alpha, (\alpha \to (\beta \to \gamma)), (\beta \to \gamma), \beta, \gamma$ este o H-secvenţă deductivă, deci $\{\alpha, \beta, (\alpha \to (\beta \to \gamma))\} \vdash \gamma$.

Aplicând teorema deducţiei rezultă

$$\begin{aligned}
&\{\beta, (\alpha \to (\beta \to \gamma))\} \vdash (\alpha \to \gamma), \\
&\{(\alpha \to (\beta \to \gamma))\} \vdash (\beta \to (\alpha \to \gamma)), \\
&\vdash ((\alpha \to (\beta \to \gamma)) \to (\beta \to (\alpha \to \gamma))).
\end{aligned}$$

Schema permutării premiselor este reprezentată $\frac{(\alpha \to (\beta \to \gamma))}{(\beta \to (\alpha \to \gamma))} PP$.

1.4.4 Schemele "trecerii" de la echivalenţă la implicaţie (EI)

Deoarece pentru orice $\alpha, \beta \in FORM$,

$$\overline{\alpha}_4 \{\alpha \mid a,\ \beta \mid b\} = ((\alpha \leftrightarrow \beta) \to (\alpha \to \beta))$$

şi

$$\overline{\alpha}_5 \{\alpha \mid a,\ \beta \mid b\} = ((\alpha \leftrightarrow \beta) \to (\beta \to \alpha))$$

rezultă

$$\{(\alpha \leftrightarrow \beta)\} \vdash (\alpha \to \beta) \text{ şi } \{(\alpha \leftrightarrow \beta)\} \vdash (\beta \to \alpha).$$

Schemele „trecerii" de la echivalenţă la implicaţie (EI) sunt reprezentate prin $\frac{(\alpha\leftrightarrow\beta)}{(\alpha\to\beta)}EI$ respectiv $\frac{(\alpha\leftrightarrow\beta)}{(\beta\to\alpha)}EI$.

1.4.5 Aplicaţie 5

Pentru orice $\alpha, \beta \in FORM$,

$$\vdash ((\neg\alpha) \to (\alpha \to \beta)).$$

Secvenţa

$$\begin{aligned}
&\gamma_1 = (((\neg\beta) \to (\neg\alpha)) \to (\alpha \to \beta)) = \overline{\alpha}_7 \{\beta \mid a,\ \alpha \mid b\},\\
&\gamma_2 = ((\neg\alpha) \to ((\neg\beta) \to (\neg\alpha))) = \overline{\alpha}_1 \{(\neg\alpha) \mid a,\ (\neg\beta) \mid b\},\\
&\gamma_3 = ((\neg\alpha) \to (\alpha \to \beta)), \frac{\gamma_2, \gamma_1}{\gamma_3}RS,
\end{aligned}$$

este o demonstraţie formală, deci $\vdash ((\neg\alpha) \to (\alpha \to \beta))$.

În particular rezultă

$$\{(\neg\alpha), \alpha\} \vdash \beta$$

pentru orice $\beta \in FORM$, deci pentru orice $\alpha \in FORM$,

$$T(\{(\neg\alpha), \alpha\}) = FORM.$$

1.4.6 Schemele dublei negaţii (DN)

Pentru orice $\alpha \in FORM$, $\vdash (\alpha \leftrightarrow (\neg(\neg\alpha)))$.

Pentru $H = \{(\neg(\neg\alpha))\}$ şi $\beta \in T_h$ (de exemplu $\beta = (\alpha \to \alpha)$) fie H-secvenţa deductivă,

$$\begin{aligned}
&\gamma_1 = ((\neg(\neg\alpha)) \to ((\neg\alpha) \to (\neg\beta))) \in T_h(\text{aplicaţia } 1.4.5),\\
&\gamma_2 = (\neg(\neg\alpha)) \in H,\\
&\gamma_3 = ((\neg\alpha) \to (\neg\beta)),\ \frac{\gamma_2, \gamma_1}{\gamma_3}MP,\\
&\gamma_4 = (((\neg\alpha) \to (\neg\beta)) \to (\beta \to \alpha)) = \overline{\alpha}_7 \{\alpha \mid a,\ \beta \mid b\},\\
&\gamma_5 = (\beta \to \alpha), \frac{\gamma_3, \gamma_4}{\gamma_5}MP,\\
&\gamma_6 = \beta \in T_h,\\
&\gamma_7 = \alpha, \frac{\gamma_6, \gamma_5}{\gamma_7}MP.
\end{aligned}$$

Rezultă

$$\{(\neg (\neg \alpha))\} \vdash \alpha,$$

deci

$$\vdash ((\neg (\neg \alpha)) \rightarrow \alpha)$$

pentru orice formulă α, ceea ce în particular implică,

$$\vdash ((\neg (\neg (\neg \alpha))) \rightarrow (\neg \alpha)) .$$

Considerăm demonstraţia formală

$$\begin{aligned}
\delta_1 &= ((\neg (\neg (\neg \alpha))) \rightarrow (\neg \alpha)) \in T_h, \\
\delta_2 &= (((\neg (\neg (\neg \alpha))) \rightarrow (\neg \alpha)) \rightarrow (\alpha \rightarrow (\neg (\neg \alpha)))) = \\
&= \overline{\alpha}_7 \{(\neg (\neg \alpha)) \mid a,\ \alpha \mid b\} , \\
\delta_3 &= (\alpha \rightarrow (\neg (\neg \alpha))) ,\ \frac{\delta_1, \delta_2}{\delta_3} MP.
\end{aligned}$$

deci

$$\vdash (\alpha \rightarrow (\neg (\neg \alpha))) ,$$

adică

$$\{\alpha\} \vdash (\neg (\neg \alpha)) .$$

Fie demonstraţia formală,

$$\begin{aligned}
\eta_1 &= ((\neg (\neg \alpha)) \rightarrow \alpha) \in T_h, \\
\eta_2 &= (\alpha \rightarrow (\neg (\neg \alpha))) \in T_h, \\
\eta_3 &= (\alpha \leftrightarrow (\neg (\neg \alpha))) , \frac{\eta_1, \eta_2}{\eta_3} IE.
\end{aligned}$$

Din rezultatele stabilite obţinem schemele de inferenţă (DN) reprezentate prin

$$\frac{\alpha}{(\neg (\neg \alpha))} DN$$

şi respectiv

$$\frac{(\neg (\neg \alpha))}{\alpha} DN.$$

Observaţia 1.4.1 Mulţimea formulelor false este nevidă. Într-adevăr, deoarece $T_h \neq \emptyset$, fie $\alpha \in T_h$. Evident, rezultă $(\neg (\neg \alpha)) \in T_h$ deci $(\neg \alpha)$ este formulă falsă. Convenim să notăm prin $\top$ o formulă demonstrabilă, respectiv prin $\perp$ o formulă falsă oarecare.

1.4.7 Schema negaţiei (NN)

Pentru orice $\alpha, \beta \in FORM$,

$$\vdash ((\alpha \rightarrow \beta) \leftrightarrow ((\neg \beta) \rightarrow (\neg \alpha))) .$$

Fie demonstraţia formala,

$$\begin{aligned}
\delta_1 &= \overline{\alpha}_7 \{\beta \mid a,\ \alpha \mid b\} = (((\neg \beta) \rightarrow (\neg \alpha)) \leftrightarrow (\alpha \rightarrow \beta)) , \\
\delta_2 &= (((\neg \beta) \rightarrow (\neg \alpha)) \rightarrow (\alpha \rightarrow \beta)) ,\ \frac{\delta_1}{\delta_2} EI,
\end{aligned}$$

deci

$$\{((\neg\beta) \to (\neg\alpha))\} \vdash (\alpha \to \beta) .$$

Obţinem astfel prima schemă de negaţie $\frac{((\neg\beta)\to(\neg\alpha))}{(\alpha\to\beta)} NN$.

Considerăm mulţimea de ipoteze $H = \{(\alpha \to \beta)\}$ şi H-secvenţă deductivă,

$$\begin{aligned}
&\eta_1 = (\alpha \to \beta) \in H,\\
&\eta_2 = ((\neg(\neg\alpha)) \to \alpha) \in T_h,\\
&\eta_3 = ((\neg(\neg\alpha)) \to \beta) , \ \frac{\eta_2, \eta_1}{\eta_3} RS,\\
&\eta_4 = (\beta \to (\neg(\neg\beta))) \in T_h,\\
&\eta_5 = ((\neg(\neg\alpha)) \to (\neg(\neg\beta))) , \ \frac{\eta_3, \eta_4}{\eta_5} RS,\\
&\eta_6 = ((\neg\beta) \to (\neg\alpha)) , \frac{\eta_5}{\eta_6} NN.
\end{aligned}$$

Rezultă

$$\{(\alpha \to \beta)\} \vdash ((\neg\beta) \to (\neg\alpha))$$

deci

$$\vdash ((\alpha \to \beta) \to ((\neg\beta) \to (\neg\alpha)))$$

din care obţinem cea de a doua schemă de negaţie, $\frac{(\alpha\to\beta)}{(\neg\beta)\to(\neg\alpha)} NN$.

Aplicând corolarul teoremei deducţiei rezultă

$$\vdash ((\alpha \to \beta) \to ((\neg\beta) \to (\neg\alpha))) .$$

În final, putem considera demonstraţia formală,

$$\begin{aligned}
&\zeta_1 = (((\neg\beta) \to (\neg\alpha)) \to (\alpha \to \beta)) ,\\
&\zeta_2 = ((\alpha \to \beta) \to ((\neg\beta) \to (\neg\alpha))) ,\\
&\zeta_3 = (((\neg\beta) \to (\neg\alpha)) \leftrightarrow (\alpha \to \beta)) , \ \frac{\zeta_1, \zeta_2}{\zeta_3} IE.
\end{aligned}$$

1.4.8 Aplicaţie 8

Pentru orice $\alpha, \beta \in FORM$,

$$\vdash ((\alpha \to (\neg\beta)) \leftrightarrow (\beta \to (\neg\alpha))) .$$

Fie $H = \{(\alpha \to (\neg\beta))\}$.

Considerăm H-secvenţa deductivă,

$$\begin{aligned}
&\zeta_1 = (\alpha \to (\neg\beta)) \in H,\\
&\zeta_2 = ((\neg(\neg\alpha)) \to \alpha) \in T_h,\\
&\zeta_3 = ((\neg(\neg\alpha)) \to (\neg\beta)) , \ \frac{\zeta_2, \zeta_1}{\zeta_3} RS,\\
&\zeta_4 = (((\neg(\neg\alpha)) \to (\neg\beta)) \to (\beta \to (\neg\alpha))) \in T_h,\\
&\zeta_5 = (\beta \to (\neg\alpha)) , \ \frac{\zeta_3, \zeta_4}{\zeta_5} MP,
\end{aligned}$$

deci

$$\{(\alpha \to (\neg\beta))\} \vdash (\beta \to (\neg\alpha)),$$

adică

$$\vdash ((\alpha \to (\neg\beta)) \to (\beta \to (\neg\alpha))).$$

Deoarece proprietatea a fost stabilită pentru orice formule α, β rezultă,

$$\vdash ((\beta \to (\neg\alpha)) \to (\alpha \to (\neg\beta))),$$

deci utilizând schema trecerii de la implicaţie la echivalenţă rezultă,

$$\vdash ((\beta \to (\neg\alpha)) \leftrightarrow (\alpha \to (\neg\beta))).$$

1.4.9 Schema rezoluţiei (REZ)

Pentru orice $\alpha, \beta, \gamma \in FORM$,

$$\vdash ((\alpha \to \beta) \to (((\neg\alpha) \to \gamma) \to (\beta \vee \gamma))).$$

Fie $H = \{(\alpha \to \beta), ((\neg\alpha) \to \gamma)\}$ şi H-secvenţa deductivă,

$$\begin{aligned}
&\delta_1 = ((\beta \vee \gamma) \leftrightarrow ((\neg\beta) \to \gamma)) = \overline{\alpha}_8 \{\beta \mid a,\ \gamma \mid b\},\\
&\delta_2 = (((\neg\beta) \to \gamma) \to (\beta \vee \gamma)),\ \frac{\delta_1}{\delta_2} EI,\\
&\delta_3 = (\alpha \to \beta) \in H,\\
&\delta_4 = ((\alpha \to \beta) \to ((\neg\beta) \to (\neg\alpha))) \in T_h,\\
&\delta_5 = ((\neg\beta) \to (\neg\alpha)),\ \frac{\delta_3, \delta_4}{\delta_5} RS,\\
&\delta_6 = ((\neg\alpha) \to \gamma) \in H,\\
&\delta_7 = ((\neg\beta) \to \gamma),\ \frac{\delta_5, \delta_6}{\delta_7} RS,\\
&\delta_8 = (\beta \vee \gamma),\ \frac{\delta_7, \delta_2}{\delta_8} RS.
\end{aligned}$$

Obţinem astfel,

$$\{(\alpha \to \beta), ((\neg\alpha) \to \gamma)\} \vdash (\beta \vee \gamma),$$

deci aplicând corolarul teoremei deducţiei, rezultă,

$$\vdash ((\alpha \to \beta) \to (((\neg\alpha) \to \gamma) \to (\beta \vee \gamma)))$$

şi

$$\vdash (((\neg\alpha) \to \gamma) \to ((\alpha \to \beta) \to (\beta \vee \gamma))).$$

Schema rezoluţiei este reprezentată prin, $\frac{(\alpha \to \beta), ((\neg\alpha) \to \gamma)}{(\beta \vee \gamma)} REZ$.

1.4.10 Aplicaţie 10

Pentru orice $\alpha, \beta \in FORM, \vdash (\alpha \to (\alpha \vee \beta)), \vdash (\beta \to (\alpha \vee \beta))$

Pentru $H = \{\alpha\}$, considerăm secvenţa deductivă,

$$\begin{aligned}
&\delta_1 = ((\alpha \vee \beta) \leftrightarrow ((\neg\alpha) \to \beta)) = \overline{\alpha}_8 \{\alpha \mid a,\ \beta \mid b\},\\
&\delta_2 = (((\neg\alpha) \to \beta) \to (\alpha \vee \beta)),\ \frac{\delta_1}{\delta_2} EI,\\
&\delta_3 = ((\neg\alpha) \to (\alpha \to \beta)) \in T_h,\ (aplicactia\ 1.4.5),\\
&\delta_4 = (\alpha \to ((\neg\alpha) \to \beta)),\ \frac{\delta_3}{\delta_4} PP,\\
&\delta_5 = \alpha \in H,\\
&\delta_6 = ((\neg\alpha) \to \beta),\ \frac{\delta_5, \delta_4}{\delta_6} MP,\\
&\delta_7 = ((\alpha \vee \beta) \leftrightarrow ((\neg\alpha) \to \beta)) = \overline{\alpha}_8 \{\alpha \mid a,\ \beta \mid b\},\\
&\delta_8 = (((\neg\alpha) \to \beta) \to (\alpha \vee \beta)),\ \frac{\delta_7}{\delta_8} EI,\\
&\delta_9 = (\alpha \vee \beta),\ \frac{\delta_6, \delta_8}{\delta_9} MP,
\end{aligned}$$

deci

$$\{\alpha\} \vdash (\alpha \vee \beta),$$

din care rezultă

$$\vdash (\alpha \to (\alpha \vee \beta)).$$

Fie demonstraţia formală,

$$\begin{aligned}
&\eta_1 = ((\alpha \vee \beta) \leftrightarrow ((\neg\alpha) \to \beta)) = \overline{\alpha}_8 \{\alpha \mid a,\ \beta \mid b\},\\
&\eta_2 = (((\neg\alpha) \to \beta) \to (\alpha \vee \beta)),\ \frac{\eta_1}{\eta_2} EI,\\
&\eta_3 = (\beta \to ((\neg\alpha) \to \beta)) = \overline{\alpha}_1 \{\beta \mid a, (\neg\alpha) \mid b\},\\
&\eta_4 = (\beta \to (\alpha \vee \beta)),\ \frac{\eta_3, \eta_2}{\eta_4} RS,
\end{aligned}$$

deci

$$\vdash (\beta \to (\alpha \vee \beta)).$$

1.4.11 Aplicaţie 11

Pentru orice $\alpha, \beta \in FORM, \vdash ((\alpha \vee \beta) \leftrightarrow (\beta \vee \alpha))$.

Fie $H = \{(\beta \vee \alpha)\}$. Considerăm H-secvenţa deductivă,

$$\begin{aligned}
\delta_1 &= ((\alpha \vee \beta) \leftrightarrow ((\neg\alpha) \to \beta)) = \overline{\alpha}_8 \{\alpha \mid a,\ \beta \mid b\}, \\
\delta_2 &= (((\neg\alpha) \to \beta) \to (\alpha \vee \beta)),\ \frac{\delta_1}{\delta_2} EI, \\
\delta_3 &= ((\beta \vee \alpha) \leftrightarrow ((\neg\beta) \to \alpha)) = \overline{\alpha}_8 \{\beta \mid a,\ \alpha \mid b\}, \\
\delta_4 &= ((\beta \vee \alpha) \to ((\neg\beta) \to \alpha)),\ \frac{\delta_3}{\delta_4} EI, \\
\delta_5 &= (\beta \vee \alpha) \in H, \\
\delta_6 &= ((\neg\beta) \to \alpha),\ \frac{\delta_5, \delta_4}{\delta_6} MP, \\
\delta_7 &= (\alpha \to (\neg(\neg\alpha))) \in T_h, \\
\delta_8 &= ((\neg\beta) \to (\neg(\neg\alpha))),\ \frac{\delta_6, \delta_7}{\delta_8} RS, \\
\delta_9 &= (((\neg\beta) \to (\neg(\neg\alpha))) \to ((\neg\alpha) \to \beta)) = \overline{\alpha}_7 \{\beta \mid a,\ (\neg\alpha) \mid b\}, \\
\delta_{10} &= ((\neg\alpha) \to \beta),\ \frac{\delta_8, \delta_9}{\delta_{10}} MP, \\
\delta_{11} &= (\alpha \vee \beta),\ \frac{\delta_{10}, \delta_2}{\delta_{11}} MP.
\end{aligned}$$

Rezultă

$$\{(\beta \vee \alpha)\} \vdash (\alpha \vee \beta),$$

deci pentru orice $\alpha, \beta \in FORM$,

$$\vdash ((\beta \vee \alpha) \to (\alpha \vee \beta)).$$

Deoarece

$$\vdash ((\beta \vee \alpha) \to (\alpha \vee \beta)) \text{ şi } \vdash ((\alpha \vee \beta) \to (\beta \vee \alpha)),$$

utilizând schema (IE) obţinem

$$\vdash ((\beta \vee \alpha) \leftrightarrow (\alpha \vee \beta)).$$

1.5 Aplicaţii

Exemplul 1.5.1 *Să se verifice dacă următoarele expresii logice sunt tautologii folosind tebele de adevăr: a)* $\alpha_1 = ((p \to q) \wedge p) \to q$

Soluţie

p	q	$p \to q$	$(p \to q) \wedge p$	$((p \to q) \wedge p) \to q$
1	*1*	*1*	*1*	*1*
1	*0*	*0*	*0*	*1*
0	*1*	*1*	*0*	*1*
0	*0*	*1*	*0*	*1*

b) $\alpha_2 = ((p \to q) \wedge (\neg q)) \to (\neg p)$

p	q	$\neg p$	$\neg q$	$p \to q$	$(p \to q) \wedge (\neg q)$	$((p \to q) \wedge (\neg q)) \to (\neg p)$
1	1	0	0	1	0	1
1	0	0	1	0	0	1
0	1	1	0	1	0	1
0	0	1	1	1	1	1

c) $\alpha_3 = ((p \vee q) \wedge (\neg p)) \to q$

p	q	$\neg p$	$p \vee q$	$(p \vee q) \wedge (\neg p)$	$((p \vee q) \wedge (\neg p)) \to q$
1	1	0	1	0	1
1	0	0	1	0	1
0	1	1	1	1	1
0	0	1	0	0	1

d) $\alpha_4 = \neg(p \vee (q \wedge t)) \leftrightarrow (\neg p \wedge (\neg q \vee \neg t))$

p	q	t	$\neg p$	$\neg q$	$\neg t$	$q \wedge t$	$p \vee (q \wedge t)$	$\neg(p \vee (q \wedge t))$	$\neg q \vee (\neg t)$	$\neg p \wedge (\neg q \vee \neg t)$	α
1	1	1	0	0	0	1	1	0	0	0	1
1	1	0	0	0	1	0	1	0	1	0	1
1	0	1	0	1	0	0	1	0	1	0	1
1	0	0	0	1	1	0	1	0	1	0	1
0	1	1	1	0	0	1	1	0	0	0	1
0	1	0	1	0	1	0	0	1	1	1	1
0	0	1	1	1	0	0	0	1	1	1	1
0	0	0	1	1	1	0	0	1	1	1	1

e) $\alpha_5 = \neg(p \to q) \wedge (\neg(q \to p))$

p	q	$p \to q$	$q \to p$	$\neg(p \to q)$	$\neg(q \to p)$	$\neg(p \to q) \wedge (\neg(q \to p))$
1	1	1	1	0	0	0
1	0	0	1	1	0	0
0	1	1	0	0	1	0
0	0	1	1	0	0	0

Se observă că primele 4 expresii sunt tautologii, iar ultima formulă este o contradicţie.

Exemplul 1.5.2 *Folosind tabela de adevăr determinaţi dacă formula* $(p \to q) \vee (p \to \neg q)$ *este validă.*

Soluţie

p	q	$p \to q$	$\neg q$	$p \to \neg q$	$(p \to q) \vee (p \to \neg q)$
T	T	T	F	F	T
T	F	F	T	T	T
F	T	T	F	T	T
F	F	T	T	T	T

Formula este validă deoarece valoarea de adevăr este T indiferent de valoarea de adevăr a propoziţiilor componente.

Exemplul 1.5.3 *Folosind tabela de adevăr demonstraţi că formule* $p \to (q \wedge \neg q)$ *şi* $\neg p$ *sunt echivalente*

Soluţie

p	q	$p \wedge \neg q$	$p \rightarrow (q \wedge \neg q)$	$\neg p$
T	T	F	F	F
T	F	F	F	F
F	T	F	T	T
F	F	F	T	T

Cele două formule sunt echivalente deoarece au aceeaşi valoare de adevăr.

Exemplul 1.5.4 *Se consideră formulele*
a) $\alpha_1 = (\neg b \vee (a \wedge \neg c)) \leftrightarrow (b \vee (\neg a \rightarrow (c \rightarrow \neg b)))$
b) $\alpha_2 = (\neg a \vee b) \rightarrow (\neg(a \rightarrow b) \rightarrow (\neg b \wedge a))$.

Să se determine: secvenţa generativă formule (SGF), arborele de structură pentru formulele date şi să se calculeze funcţia de adâncime $h(\alpha_i)$, $i \in \{1, 2\}$ *pentru expresiile date.*

Soluţie

Formulei $\alpha_1 = (\neg b \vee (a \wedge \neg c)) \leftrightarrow (b \vee (\neg a \rightarrow (c \rightarrow \neg b)))$ *îi putem construi următorul SGF:* $a, b, c, \neg a, \neg b, \neg c, a \wedge \neg c), (\neg b \vee (a \wedge \neg c)), (c \rightarrow \neg b), (\neg a \rightarrow (c \rightarrow \neg b)), (b \vee (\neg a \rightarrow (c \rightarrow \neg b))), \alpha_1$.

Arborele de structură specific formulei de mai sus este:

$$
\begin{array}{cccccccccc}
 & & & & r\,(\leftrightarrow) & & & & & \\
 & & & & \swarrow\searrow & & & & & \\
 & & n_1(\vee) & & & & n_2(\vee) & & & \\
 & & \swarrow\searrow & & & & \swarrow\searrow & & & \\
 & n_3(\neg) & & n_4(\wedge) & & n_5(b) & & n_6(\rightarrow) & & \\
T\,(\alpha_1): & \downarrow & & \swarrow\searrow & & & & \swarrow\searrow & & \\
 & n_7(a) & n_8(a) & & n_9(\neg) & & & n_{10}(\neg) & n_{11}(\rightarrow) & \\
 & & & & \downarrow & & & \downarrow & \swarrow\searrow & \\
 & & & & n_{12}(c) & & & n_{13}(a) & n_{14}(c) & n_{15}(\neg) \\
 & & & & & & & & & \downarrow \\
 & & & & & & & & & n_{16}(b)
\end{array}.
$$

Pentru formula

$$\alpha_1 = (\neg b \vee (a \wedge \neg c)) \leftrightarrow (b \vee (\neg a \rightarrow (c \rightarrow \neg b)))$$

obţinem următoarea funcţie de adâncime

$$\hbar\,(\alpha_1) = 1 + \max\{\hbar\,(\beta)\,, \hbar\,(\gamma)\}\,,$$

unde $\beta = (\neg b \vee (a \wedge \neg c)$, $\gamma = b \vee (\neg a \rightarrow (c \rightarrow \neg b))$

$$
\begin{aligned}
\hbar\,(\beta) &= 1 + \max\{\hbar\,(\neg b)\,, \hbar(a \wedge \neg c)\} = 1 + \max\{1 + \hbar(b); 1 + \max\{\hbar(a), \hbar(\neg c)\}\} = \\
&= 2 + \max\{0; \max\{0, 1 + \hbar(c)\}\} = 2 + \max\{0, 1\} = 3
\end{aligned}
$$

$$
\begin{aligned}
\hbar\,(\gamma) &= 1 + \max\{\hbar\,(b)\,, \hbar\,(\neg a \rightarrow (c \rightarrow \neg b))\} = \\
&= 1 + \max\{0, 1 + \max\{\hbar(\neg a), \hbar(c \rightarrow \neg b)\}\} = \\
&= 2 + \max\{1 + \hbar(a); 1 + \max\{\hbar(c), \hbar(\neg b)\}\} = 3 + \max\{0; 1 + \hbar(b)\} = 4
\end{aligned}
$$

Rezultă $\hbar\,(\alpha_1) = 5$.

Pentru formula $\alpha_2 = (\neg a \vee b) \rightarrow (\neg(a \rightarrow b) \rightarrow (\neg b \wedge a))$ *arborele de structură este următorul*

$$
\begin{array}{lcccccccc}
 & & & & r\,(\rightarrow) & & & & \\
 & & & & \swarrow\searrow & & & & \\
 & & n_1(\vee) & & & & n_2(\rightarrow) & & \\
 & & \swarrow\searrow & & & & \swarrow\searrow & & \\
T\,(\alpha_2): & n_3(\neg) & & n_4(b) & & n_5(\neg) & & n_6(\wedge) & \\
 & \downarrow & & & & \downarrow & & \swarrow\searrow & \\
 & n_7(a) & & & & n_8(\rightarrow) & n_9(\neg) & & n_{10}(a) \\
 & & & & & \swarrow\searrow & \downarrow & & \\
 & & & & n_{11}(a) & n_{12}(b) & n_{13}(b) & &
\end{array}
$$

Funcţia de adâncime specifică formulei $\alpha_2 = (\neg a \vee b) \rightarrow (\neg(a \rightarrow b) \rightarrow (\neg \wedge a))$ *este următoarea:*

$$\hbar(\alpha_2) = 1 + \max\{\hbar(\beta), \hbar(\gamma)\}$$

unde $\beta = \neg a \vee b$ *şi* $\gamma = \neg(a \rightarrow b) \rightarrow (\neg b \wedge a)$

$$\hbar(\beta) = 1 + \max\{\hbar(\neg a), \hbar(b)\} = 1 + \max\{1 + \hbar(a), 0\} = 2$$

$$
\begin{aligned}
\hbar(\gamma) &= 1 + \max\{\neg(a \rightarrow b), \hbar(\neg b \wedge a)\} = 2 + \max\{\hbar(a \rightarrow b); \max\{\hbar(\neg b), \hbar(a)\}\} = \\
&= 2 + \max\{\hbar(a \rightarrow b); \max\{1 + \hbar(b), 0\}\} = 2 + \max\{1 + \max\{\hbar(a), \hbar(b)\}, 1\} = 3.
\end{aligned}
$$

În acest caz, funcţia de adâncime este $\hbar(\alpha_2) = 4$.

Exemplul 1.5.5 *Fie formulele*
a) $\alpha_1 = (\neg\beta \rightarrow \neg(\gamma \vee \theta)) \rightarrow ((\neg\gamma \rightarrow \theta) \rightarrow \beta)$
b) $\alpha_2 = (\neg\gamma \vee \beta) \rightarrow ((\neg\beta \rightarrow (\delta \vee \alpha)) \rightarrow (\gamma \rightarrow \beta))$
c) $\alpha_3 = \neg((\alpha \vee \beta) \rightarrow \neg(\gamma \rightarrow \alpha)) \rightarrow ((\neg\alpha \rightarrow \beta) \wedge (\neg\gamma \vee \alpha))$
d) $\alpha_4 = (\neg\gamma \rightarrow (\beta \vee \alpha)) \rightarrow (\gamma \vee (\neg\beta \rightarrow \alpha))$
e) $\alpha_5 = (\neg\beta \rightarrow (\neg(\gamma \vee \theta) \rightarrow \neg\beta)) \rightarrow \beta \vee (\neg(\neg\gamma \wedge \neg\theta) \vee \neg\beta)$
şi substituţiile

$$\sigma = \{(\neg a \vee b)|\beta,\ (\neg b \rightarrow c)|\gamma,\ (\neg\gamma \rightarrow \delta)|\omega,\ (\neg\omega \rightarrow \delta)|\pi\}.$$

Să se calculeze rezultatele aplicării substituţiei σ *asupra formulelor date.*
Soluţie
În urma aplicării substituţiei σ *formulelor* α_i, $i \in \{1, 2, ..., 5\}$ *se obţin structurile simbolice* $\alpha_i\sigma$, $i \in \{1, 2, ..., 5\}$ *astfel:*
a) $\alpha_1\sigma = (\neg(\neg a \vee b) \rightarrow \neg((\neg b \rightarrow c) \vee \theta)) \rightarrow ((\neg(\neg b \rightarrow c) \rightarrow \theta) \rightarrow (\neg a \vee b))$
b) $\alpha_2\sigma = (\neg(\neg b \rightarrow c) \vee (\neg a \vee b)) \rightarrow ((\neg(\neg a \vee b) \rightarrow (\delta \vee \alpha)) \rightarrow ((\neg b \rightarrow c) \rightarrow (\neg a \vee b)))$
c) $\alpha_3\sigma = \neg((\alpha \vee (\neg a \vee b)) \rightarrow \neg((\neg b \rightarrow c) \rightarrow \alpha)) \rightarrow ((\neg\alpha \rightarrow (\neg a \vee b)) \wedge (\neg(\neg b \rightarrow c) \vee \alpha))$
d) $\alpha_4\sigma = (\neg(\neg b \rightarrow c) \rightarrow ((\neg a \vee b) \vee \alpha)) \rightarrow ((\neg b \rightarrow c) \vee (\neg(\neg a \vee b) \rightarrow \alpha))$
e) $\alpha_5\sigma = (\neg(\neg a \vee b) \rightarrow (\neg((\neg b \rightarrow c) \vee \theta) \rightarrow \neg(\neg a \vee b))) \rightarrow (\neg a \vee b) \vee (\neg(\neg(\neg b \rightarrow c) \wedge \neg\theta) \vee \neg(\neg a \vee b))$.

Exemplul 1.5.6 *Să se identifice demonstraţii formale pentru* $(((\alpha \wedge \theta) \rightarrow \beta) \wedge (\beta \rightarrow \gamma)) \rightarrow ((\alpha \wedge \theta) \rightarrow \gamma)$, *unde* α, β, $\gamma \in FORM$.
Soluţie
Folosind

$$\vdash \overline{\alpha}_3\,\{(\alpha \wedge \theta) \mid a,\ \beta \mid b,\ \gamma \mid c\} = (((\alpha \wedge \theta) \rightarrow \beta) \rightarrow ((\beta \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma)))$$

şi regula silogismului (RS) obţinem

$$\frac{((\alpha \wedge \theta) \to \beta)\,, (\beta \to \gamma)}{((\alpha \wedge \theta) \to \gamma)} RS.$$

Exemplul 1.5.7 *Folosind axiomelele să se identifice demonstraţii formale pentru:*
a) $(\neg\alpha \to \beta) \to ((\neg\gamma \vee \theta) \to (\alpha \vee \beta))$
b) $(\neg(\beta \vee \gamma) \to \neg(\theta \vee \alpha)) \to ((\neg\theta \to \alpha) \to (\beta \vee \gamma))$
c) $\neg((\beta \vee \neg\alpha) \vee (\gamma \wedge \neg\theta)) \leftrightarrow ((\neg\beta \wedge \alpha) \wedge (\neg\gamma \vee \theta))$ *unde* α, β, γ, $\theta \in FORM$.

Soluţie

Folosind $\overline{\alpha}_1 = a \to (b \to a)$ *şi substituţia* $\sigma = \{(\neg\alpha \vee \beta)|a,\, (\neg\gamma \vee \theta)|b\}$ *obţinem* $\overline{\alpha}_1\sigma = (\neg\alpha \to \beta) \to ((\neg\gamma \vee \theta) \to (\alpha \vee \beta))$, *relaţia a), unde* a, b, c *propoziţii elementare.*

Pentru demonstrarea relaţiei b) vom folosi axioma $\overline{\alpha}_7 = (((\neg a) \to (\neg b)) \to (b \to a))$, *substituţia* $\sigma = \{(\beta \vee \gamma)|a,\, (\neg\theta \to \alpha)|b\}$ *şi următoarea axiomă* $(\neg\alpha \to \beta) \leftrightarrow (\alpha \vee \beta)$.

Plecând de la relaţiile $\neg(a \vee b) \leftrightarrow (\neg a \wedge \neg b)$ *şi* $\neg(a \wedge b) \leftrightarrow (\neg a \vee \neg b)$, *cu* $a, b \in V$ *şi aplicând substituţia* $\sigma = \{(\beta \vee \neg\alpha)|a,\, (\gamma \wedge \neg\theta)|b\}$ *formulelor prezentate obţinem cu uşurinţă c).*

Exemplul 1.5.8 *Să se identifice axiomele care se pot aplica pentru demonstrarea formulelor:*
a) $f_1 : \vdash \neg(\gamma \vee \theta) \to ((\neg\gamma \to \theta) \to (\omega \vee \delta))$
b) $f_2 : \vdash (\neg(\omega \vee (\theta \to \delta)) \vee (\neg\theta \to \delta)) \leftrightarrow ((\neg\omega \to (\neg\theta \vee \delta)) \to (\theta \vee \delta))$
c) $f_3 : \vdash (\neg\delta \wedge \neg\gamma) \to ((\neg\delta \to \gamma) \to (\neg\theta \vee (\delta \to \neg\omega)))$
d) $f_4 : \vdash \theta \to (\theta \vee (\gamma \to \omega))$.

Soluţie

În cazul forumlei f_1 *s-a folosit aplicaţia 1.4.5,* $\neg\alpha \to (\alpha \to \beta)$ *cu* $\alpha, \beta \in FORM$ *căreia i s-a aplicat substituţia* $\sigma = \{(\gamma \vee \theta)|\alpha,\, (\omega \vee \delta)|\beta\}$.

Pentru cazul b) s-a utilizat axioma $\alpha = (\neg a \to b) \leftrightarrow (a \vee b)$ *împreună cu substituţia* $\sigma = \{(\neg\omega \to (\neg\theta \vee \delta))|a,\, (\theta \vee \delta)|b\}$ *rezultând* $\alpha\sigma$, *adică relaţia dată.*

În subpunctul c) s-a plecat de la aplicaţia 1.4.5, $\neg\alpha \to (\alpha \to \beta)$ *cu* $\alpha, \beta \in FORM$, *căreia i s-a aplicat substituţia* $\sigma = \{(\beta \vee \gamma)|\alpha,\, (\neg\theta \vee (\delta \to \neg\omega))|\beta\}$. *De asemenea s-au folosit şi legile lui De Morgan* $\neg(a \vee b) = \neg a \wedge \neg b$, *dar şi axioma* $\alpha = (\neg a \to b) \leftrightarrow (a \vee b)$.

În ultimul caz, aplicaţiei 1.4.10, $\delta = \alpha \to (\alpha \vee \beta)$ *cu* $\alpha, \beta \in FORM$ *i s-a aplicat substituţia* $\sigma = \{\theta|\alpha,\, (\gamma \to \omega)|\beta\}$ *obţinându-se* $\delta\sigma$, *adică relaţia dorită.*

Exemplul 1.5.9 *Verificaţi dacă formulele* ***A1*** *şi* ***A2*** *sunt tautologii folosind metoda tabelelor de adevăr:*
A1: $a \to (b \to a)$ ***A2***: $(a \to (b \to c)) \to ((a \to b) \to (a \to c))$
A3: $((\neg b) \to (\neg a)) \to ((\neg b) \to a) \to b)$

Soluţie

a	b	$b \to a$	***A1***
1	*1*	*1*	*1*
1	*0*	*1*	*1*
0	*1*	*0*	*1*
0	*0*	*1*	*1*

a	b	c	$b \to c$	$a \to (b \to c)$	$a \to b$	$a \to c$	$(a \to b) \to (a \to c)$	***A2***
1	1	1	1	1	1	1	1	1
1	1	0	0	0	1	0	0	1
1	0	1	1	1	0	1	1	1
1	0	0	1	1	0	0	1	1
0	1	1	1	1	1	1	1	1
0	1	0	0	1	1	1	1	1
0	0	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1

a	b	$\neg a$	$\neg b$	$\neg b \to \neg a$	$\neg b \to a$	$(\neg b \to a) \to b$	***A3***
1	1	0	0	1	1	1	1
1	0	0	1	0	1	0	1
0	1	1	0	1	1	1	1
0	0	1	1	1	0	1	1

Exemplul 1.5.10 *a) Să se arate că* $\vdash a \to a$.

Soluţie

1. $(a \to ((a \to a) \to a)) \to ((a \to (a \to a)) \to (a \to a))$	*A2*
2. $a \to ((a \to a) \to a)$	*A1*
3. $(a \to (a \to a)) \to (a \to a)$	*1, 2 MP*
4. $a \to (a \to a)$	*A1*
5. $a \to a$	*3, 4 MP*

b) Să se arate că $a \to b, b \to c \vdash a \to c$.

1. $(b \to c) \to (a \to (b \to c))$	*A1*
2. $b \to c$	*ipoteză*
3. $a \to (b \to c)$	*1, 2 MP*
4. $(a \to (b \to c)) \to ((a \to b) \to (a \to c))$	*A2*
5. $(a \to b) \to (a \to c)$	*4, 3 MP*
6. $a \to b$	*ipoteză*
7. $a \to c$	*5, 6 MP*

c) Să se arate că $a, \neg a \vdash b$.

1. $\neg a$	*ipoteză*
2. $(\neg a) \to ((\neg b) \to (\neg a))$	*A1*
3. $(\neg b) \to (\neg a)$	*1, 2 MP*
4. a	*ipoteză*
5. $a \to ((\neg b) \to a)$	*A1*
6. $(\neg b) \to a$	*4, 5 MP*
7. $((\neg b) \to (\neg a)) \to (((\neg b) \to a) \to b)$	*A3*
8. $((\neg b) \to a) \to b$	*3, 7 MP*
9. b	*6, 8 MP*

1.6 Exerciţii

Exerciţiul 1.6.1 *Să se determine care dintre următoarele expresii sunt tautologii, care sunt contradicţii şi respectiv care nu sunt nici tautologii şi nici contradicţii.*

1) $a \leftrightarrow (a \vee a)$;

2) $((a \to b) \wedge b) \to a$;

3) $(\neg a) \to (a \wedge b)$;

4) $(a \to b) \to ((b \to c) \to (a \to c))$;

5) $a \wedge (\neg(a \vee b))$;

6) $(a \to b) \leftrightarrow ((\neg a) \vee b)$;

7) $(a \to b) \leftrightarrow (\neg(a \wedge (\neg b)))$;

8) $(b \leftrightarrow (b \to a)) \to a$.

Exerciţiul 1.6.2 *Sa se demonstreze că dacă formulele α şi $\alpha \to \beta$ sunt tautologii, atunci şi β este tautologie.*

Exerciţiul 1.6.3 *Să se demonstreze că următoarele perechi de formule sunt logic echivalente.*

1) $a \to (b \to c)$ *şi* $(a \wedge b) \to c$;

2) $a \wedge (b \vee c)$ *şi* $(a \wedge b) \vee (a \wedge c)$;

3) $a \vee (b \wedge c)$ *şi* $(a \vee b) \wedge (a \vee c)$;

4) $(a \wedge b) \vee (\neg b)$ *şi* $a \vee (\neg b)$;

5) $(a \vee b) \wedge (\neg b)$ *şi* $a \wedge (\neg b)$;

6) $a \to b$ *şi* $(\neg b) \to (\neg a)$;

7) $a \leftrightarrow b$ *şi* $(a \wedge b) \vee ((\neg a) \wedge (\neg b))$

8) $\neg(a \vee b)$ *şi* $(\neg a) \wedge (\neg b)$

9) $\neg(a \wedge b)$ *şi* $(\neg a) \vee (\neg b)$

10) $\neg(a \leftrightarrow b)$ *şi* $a \leftrightarrow (\neg b)$.

Exerciţiul 1.6.4 *Se definesc următoarele conective "$\triangle$" şi "$\otimes$" având următoarele tabele de adevăr*

p	q	$p \triangle q$
T	T	F
F	T	F
T	F	F
F	F	T

p	q	$p \otimes q$
T	T	F
F	T	T
T	F	T
F	F	T

Să se demonstreze că următoarele formule sunt tautologii
a) $(\neg a) \leftrightarrow (a \triangle a)$
b) $(a \wedge b) \leftrightarrow ((a \triangle a) \triangle (b \triangle b))$
c) $(a \vee b) \leftrightarrow ((a \otimes a) \otimes (b \otimes b))$.

Exerciţiul 1.6.5 *a) Să se verifice expresiile tautologice:* $\overline{\alpha}_1, \overline{\alpha}_2, ..., \overline{\alpha}_9$.

b) Fie substituţiile $\sigma = \{(\alpha \vee \beta)|a,\ (\neg\beta \wedge \neg\alpha)|b,\ (\alpha \vee \neg\beta)|c, (\neg\alpha \wedge \beta)|d\}$ *şi* $\theta = \{(\gamma \vee \delta)|a,\ (\neg\gamma \wedge \delta)|b,\ (\neg\gamma \vee \omega)|d\}$. *Să se calculeze* $\overline{\alpha}_i\sigma$ *şi* $\overline{\alpha}_i\theta$, $i = \overline{1..9}$ *şi să se verifice expresiile tautologice.*

Exerciţiul 1.6.6 *Să se determine secvenţa generativă formule (SGF) pentru formulele de mai jos:*

a) $\alpha_1 = (a \vee (\neg b)) \rightarrow ((b \vee c) \rightarrow (\neg a \rightarrow c))$;

b) $\alpha_2 = (a \wedge (b \vee c)) \leftrightarrow ((a \wedge b) \vee (a \wedge c))$;

c) $\alpha_3 = (a \vee (\neg b)) \rightarrow ((b \vee (c \wedge (\neg d))) \rightarrow (\neg a \rightarrow (c \wedge (\neg d))))$;

d) $\alpha_4 = ((\neg a \vee b) \wedge (b \rightarrow c)) \rightarrow (c \vee (\neg a))$;

e) $\alpha_5 = ((\neg a \wedge b) \wedge (\neg c)) \leftrightarrow ((\neg b \rightarrow (a \rightarrow (\neg c))) \vee c)$;

f) $\alpha_6 = (c \rightarrow (a \wedge b)) \wedge (\neg d \vee (b \wedge a)) \rightarrow ((c \vee d) \rightarrow d)$.

Exerciţiul 1.6.7 *Pentru formulele de mai jos, determinaţi arborii de structură:*

a) $\alpha_1 = (\neg a \vee (b \wedge (\neg c))) \leftrightarrow (a \vee (\neg b \rightarrow (c \rightarrow (\neg a))))$;

b) $\alpha_2 = (\neg p \vee q) \wedge (q \rightarrow ((\neg r \wedge (\neg p)) \wedge (p \vee r)))$;

c) $\alpha_3 = ((\neg b \wedge a) \rightarrow (c \wedge d)) \rightarrow (\neg\, (\neg a \vee b) \rightarrow (\neg\, (c \wedge d)))$;

d) $\alpha_4 = (a \rightarrow b) \rightarrow ((a \vee b) \rightarrow (\neg a \vee b))$;

e) $\alpha_5 = ((\neg a \vee b) \rightarrow (a \rightarrow b)) \rightarrow (\neg\, (\neg a \vee b) \rightarrow \neg\, (a \rightarrow b))$;

f) $\alpha_6 = (\neg a \rightarrow \neg\, (b \vee c)) \rightarrow ((\neg b \rightarrow c) \rightarrow a)$;

g) $\alpha_7 = ((\neg a \rightarrow \neg b) \rightarrow (\neg b \rightarrow \neg c)) \rightarrow (a \vee \neg c)$;

h) $\alpha_8 = \neg\, (a \vee (b \wedge c)) \leftrightarrow (\neg a \wedge (\neg b \neg t))$;

i) $\alpha_9 = (a \rightarrow ((b \vee c) \rightarrow d)) \rightarrow (\neg a \vee (\neg\, (\neg b \rightarrow c) \vee d))$.

Exerciţiul 1.6.8 *Pentru expresiile de mai jos să se obţină reprezentările în scriere poloneză prefixată, respectiv reprezentarea poloneză postfixată:*

a) $\alpha_1 = (\neg b \vee a) \rightarrow ((\neg a \wedge b) \vee b)$;

b) $\alpha_2 = (a \vee ((\neg c \wedge b) \rightarrow a))$;

c) $\alpha_3 = (\neg b \rightarrow \neg(c \vee a)) \rightarrow ((\neg c \rightarrow a) \rightarrow b))$;

d) $\alpha_4 = (\neg a \rightarrow (b \vee c)) \rightarrow (a \vee (\neg b \rightarrow c))$;

e) $\alpha_5 = (\neg b \rightarrow (\neg(c \vee a) \rightarrow b)) \rightarrow (b \vee (\neg(\neg c \wedge \neg a) \vee b))$;

Exerciţiul 1.6.9 *Să se detremine dacă următoarele secvenţe sunt reprezentări în forma poloneză prefixată a unor formule din calculul propoziţional. În caz afirmativ, să se obţină reprezentările în forma infixată corespunzătoare.*

i) $\neg \rightarrow abc \vee ab\neg c$;

ii) $\rightarrow\rightarrow ab \rightarrow\rightarrow bc \rightarrow \neg ac$.

Exerciţiul 1.6.10 *Determinaţi funcţia de adâncime pentru următoarele formule:*

a) $\alpha_1 = ((\neg\,((\neg(a \wedge b)) \vee b)) \leftrightarrow ((a \wedge b) \wedge (\neg b)))$;

b) $\alpha_2 = ((p \rightarrow (\neg q)) \wedge ((\neg p \vee q) \rightarrow (\neg q))) \rightarrow (\neg q \wedge p)$;

c) $\alpha_3 = (\neg p \vee q) \wedge (q \rightarrow ((\neg\,(p \vee q) \wedge (\neg p)) \wedge (p \vee (p \vee q))))$;

d) $\alpha_4 = (b \rightarrow a) \rightarrow ((a \vee b) \rightarrow (\neg\,(\neg a \wedge b)))$;

e) $\alpha_5 = \neg\,(a \vee \neg\,(a \rightarrow b)) \leftrightarrow (\neg a \wedge (\neg a \vee b))$.

Exerciţiul 1.6.11 *Fie formulele*

a) $\alpha_1 = (\neg b \rightarrow (\neg\,(c \vee a))) \rightarrow ((\neg c \rightarrow a) \rightarrow b)$;

b) $\alpha_2 = (\neg a \vee b) \rightarrow ((\neg b \rightarrow (c \vee b)) \rightarrow (a \rightarrow b))$;

c) $\alpha_3 = \neg\,((a \vee b) \rightarrow (\neg\,(c \rightarrow a))) \rightarrow ((\neg a \rightarrow b) \wedge (\neg c \vee a))$;

d) $\alpha_4 = (\neg a \rightarrow (b \vee c)) \rightarrow (a \vee (\neg b \rightarrow c))$;

e) $\alpha_5 = (\neg b \rightarrow (\neg\,(a \vee c) \rightarrow (\neg b))) \rightarrow (b \vee (\neg\,(\neg a \wedge (\neg c)) \wedge (\neg a)) \vee (\neg b))$.

i) Să se construiască arborii de structură pentru formulele date.

ii) Să se calculeze funcţia de adâncime pentru formulele considerate anterior.

Exerciţiul 1.6.12 *Se consideră următoarele expresii logice:*

a) $((a \rightarrow b) \wedge (\neg a \rightarrow c) \wedge (a \vee \neg a)) \wedge (b \vee c)$;

b) $(a \rightarrow (b \wedge c)) \rightarrow ((a \vee (b \wedge c)) \rightarrow (\neg(a \wedge \neg(\neg b \wedge \neg))))$;

c) $(\neg a \rightarrow (\beta \vee \neg c)) \leftrightarrow (a \vee (c \rightarrow b))$;

d) $\neg(\neg a \wedge b) \rightarrow (\neg b \vee a)$;

e) $(\neg b \rightarrow (b \vee a)) \rightarrow (b \vee (\neg a \rightarrow b))$.

Cerinţe:
1) Să se verifice dacă expresiile logice date sunt tautologii.
2) Să se construiască arborele de structură pentru expresiile considerate anterior.
3) Să se calculeze funcţia de adâncime $h(x)$ pentru expresiile date.

Exerciţiul 1.6.13 *Se consideră conectivele logice notate " $\parallel$ " şi " $\perp$ " definite prin:*

$$(a \parallel b) \overset{def}{\leftrightarrow} (\neg a \vee \neg b) \text{ şi } (a \perp b) \overset{def}{\leftrightarrow} (\neg a \wedge \neg b).$$

Cerinţe:
1) Să se construiască tabele de adevăr pentru " $\parallel$ şi $\perp$.
2) Să se construiască tabele de adevăr pentru:

a) $(a \parallel b) \to (a \perp b)$;

b) $(a \perp b) \to (a \parallel b)$;

c) $(a \parallel b) \to ((a \parallel b) \vee \neg(a \perp b))$;

d) $(a \perp b) \to (\neg(a \parallel b) \to \neg(a \perp b))$.

3) Pentru conectivele date " $\parallel$ " şi " $\perp$ ", să se verifice proprietăţile de comutativitate şi asociativitate:

$$a \parallel b \iff b \parallel a \text{ şi } (a \parallel (b \parallel c)) \iff ((a \parallel b) \parallel c);$$

$$a \perp b \iff b \perp a \text{ şi } (a \perp (b \perp c)) \iff ((a \perp b) \perp c).$$

Exerciţiul 1.6.14 *Să se demonstreze că dacă formula α este tautologie şi $a_1, a_2, ..., a_n$ sunt simboluri propoziţionale ce apar în expresia lui α, atunci pentru orice $b_1, b_2, ..., b_n$ formule, formula obţinută prin substituirea simbolurilor $a_1, a_2, ..., a_n$ respectiv $b_1, b_2, ..., b_n$ este de asemenea tautologie.*

Exerciţiul 1.6.15 *Se consideră formula $\alpha = ((a \vee b) \to (\neg a \vee (c \wedge d)) \to (\neg b \to (c \vee d)))$ şi substituţia $\sigma = \{(p \vee \neg q)\,|\alpha, (\neg a \vee t)\,|c, (c \wedge d)\,|x, \neg b|t\}$. Să se determine rezultatul aplicării substituţiei σ pentru formula α.*

Exerciţiul 1.6.16 *Se consideră formula $\alpha = (a \vee \neg\,(\neg b \vee c)) \leftrightarrow (\neg a \vee (b \vee (c \to a)))$ şi substituţia $\sigma = \{(a \vee \neg c)\,|\alpha, (d \vee c)\,|a, (c \vee \neg b)\,|d, \neg d|b, \neg a|c\}$. Să se determine rezultatul aplicării substituţiei σ pentru formula α şi arborele de structură asociat lui $\alpha\sigma$.*

Exerciţiul 1.6.17 *Fie formulele:*
a) $(\neg b \to \neg(c \vee a)) \to ((\neg c \to a) \to b)$
b) $(\neg a \vee b) \to ((\neg b \to (c \vee d)) \to (a \to b))$
c) $\neg((a \vee b) \to (c \to a)) \to ((\neg a \to b) \wedge (\neg c \vee a))$
d) $(\neg a \to (b \vee c)) \to (a \vee (\neg b \to c))$
e) $(\neg b \to (\neg(b \vee c) \to \neg a)) \to a \vee (\neg(\neg b \wedge \neg c) \vee \neg b)$ *şi substituţiile*

$$\sigma = \{(\neg\alpha \vee \beta)|b,\ (\neg\beta \to \gamma)|a,\ (\neg a \vee b)|\omega,\ (\neg\beta \wedge \alpha)|d\}.$$

Cerinţe:
1) Să se calculeze rezultatele aplicării substituţiei σ asupra formulelor date ($\alpha\sigma$).
2) Să se verifice dacă formulele iniţiale sunt tautologii.
3) Să se verifice dacă rezultatele aplicării substituţiei σ asupra formulelor date sunt tautologii.
4) Să se construiască arborii de structură pentru funcţiile iniţiale, precum şi pentru formulele rezultate în urma aplicării subxtituţiei σ asupra formulelor iniţiale.
5) Să se calculeze funcţia de adâncime $h(\alpha)$ pentru formulele iniţiale, precum şi pentru formulele rezultate în urma aplicării substituţiei σ asupra formulelor iniţiale.

Exerciţiul 1.6.18 *Se consideră formula $\alpha = ((\neg a \vee (b \wedge \neg c)) \leftrightarrow (a \vee (\neg b \to (c \to \neg a))))$ şi substituţia $\sigma = \{(p \vee \neg q)|\alpha,\ (p \wedge \neg q)|a,\ (r \vee p)|p,\ b|p\}$. Să se determine: secvenţa generativă formule (SGF) pentru formula α; tabelul de adevăr pentru formula α; arborele de structură pentru formula α; $\alpha\sigma$- rezultatul aplicării substituţiei σ pentru formula α şi arborele de structură asociat lui $\alpha\sigma$.*

Exercițiul 1.6.19 *Folosind regula silogismului, să se verifice relația:*

$$((\alpha \to \beta) \wedge (\beta \to \gamma)) \to (\alpha \to \gamma).$$

Exercițiul 1.6.20 *Folosind axiomelele, să se identifice demonstrații formale pentru:*
a) $(\neg(\alpha \wedge \neg\beta) \to \beta) \to ((\neg\gamma \vee \theta) \to ((\alpha \wedge \neg\beta) \vee \beta))$;
b) $(\neg((\beta \wedge \alpha) \vee \gamma) \to \neg(\theta \vee \alpha)) \to ((\neg\theta \to \alpha) \to ((\beta \wedge \alpha) \vee \gamma))$;
c) $\neg((\beta \vee \neg\alpha) \vee ((\gamma \vee \neg\theta) \wedge \neg\theta)) \leftrightarrow ((\neg\beta \wedge \alpha) \wedge (\neg(\gamma \vee \neg\theta) \vee \theta))$.

Exercițiul 1.6.21 *Să se demonstreze că pentru orice* $a, b \in FORM$, *următoarele formule sunt teoreme.*

1) $(a \wedge b) \to a$;

2) $(a \wedge b) \to b$;

3) $(a \wedge b) \leftrightarrow (b \wedge a)$;

4) $((a \to b) \to a) \to a$;

5) $(a \wedge b) \leftrightarrow (\neg\,(a \to \neg b))$;

6) $\neg\,(a \to b) \leftrightarrow (a \wedge (\neg b))$.

Exercițiul 1.6.22 *Să se stabilească demonstrații formale pentru următoarele formule:*

1) $\alpha_1 = (a \to (b \to c)) \to (b \to (a \to c))$;

2) $\alpha_2 = (\neg b \wedge a) \to ((\neg a \vee b) \to c)$;

3) $\alpha_3 = (a \to b) \to (\neg b \to (\neg a))$;

4) $\alpha_4 = (a \to (\neg a)) \to (\neg a)$;

5) $\alpha_5 = (\neg a \to a) \to a$;

6) $\alpha_6 = \neg\,(\neg a \vee b) \to ((a \to (\neg b)) \to (\neg b \wedge a))$.

Capitolul 2

Sistemul deducţiei logice Gentzen

2.1 Introducere

Deşi dezvoltat independent, sistemul deducţiei logice naturale (Gentzen) poate fi considerat un pas spre o metodă mai intuitivă în demonstrarea validităţii unui argument. În logica propoziţiilor validitatea unui argument poate fi testată prin tabelele de adevăr. Însă metoda tabelelor de adevăr devine practic greoaie atunci când numărul variabilelor propoziţionale este mare (pentru două variabile propoziţionale avem patru distribuiri sau combinaţii ale valorilor de adevăr, pentru 3 avem 8, pentru 2 avem 16 etc.).

Din acest motiv, în locul tabelelor de adevăr, putem folosi metoda deducţiei naturale, un mod de testare a validităţii argumentelor mai familiar, mai asemănător şi mai apropiat atât de formele de argumentare folosite în limbajul cotidian, cât şi de modalităţile de demonstraţie din matematică.

2.2 Regulile de inferenţă Gentzen

Definiţia 2.2.1 *Se numeşte* secvent *o pereche de mulţimi* (H, Γ) *de formule în care apar numai conective din mulţimea* $L \setminus \{\leftrightarrow\}$ *. Prima componenta a perechii se numeşte antecedent, respectiv cea de a doua componentă se numeşte succedent. Convenţional secventul* (H, Γ) *este reprezentat* $H \Longrightarrow \Gamma$*.*

Definiţia 2.2.2 *Secventul* $H \Longrightarrow \Gamma$ *este secvent axiomă dacă* $H \cap \Gamma \neq \emptyset$*.*

Definiţia 2.2.3 *Secventul* $H \Longrightarrow \Gamma$ *este secvent încheiat dacă mulţimile componente conţin numai propoziţii elementare;* $H \cup \Gamma \subset V$*.*

Sistemul deducţiei naturale este construit pe baza mulţimii regulilor de inferenţă Gentzen.

Definiţia 2.2.4 *Regulile de inferenţă Gentzen sunt:*

G1. $\frac{H \Longrightarrow \Gamma \cup \{\alpha\}}{H \cup \{(\neg\alpha)\} \Longrightarrow \Gamma}$, *(*regula negaţie stânga*);*

G2. $\frac{H \cup \{\alpha, \beta\} \Longrightarrow \Gamma}{H \cup \{(\alpha \wedge \beta)\} \Longrightarrow \Gamma}$, *(*regula conjuncţiei stânga*);*

G3. $\frac{H \cup \{\alpha\} \Longrightarrow \Gamma,\ H \cup \{\beta\} \Longrightarrow \Gamma}{H \cup \{(\alpha \vee \beta)\} \Longrightarrow \Gamma}$,*(*regula disjuncţie stânga*);*

G4. $\frac{H\cup\{\beta\}\Longrightarrow\Gamma,\ H\Longrightarrow\Gamma\cup\{\alpha\}}{H\cup\{(\alpha\to\beta)\}\Longrightarrow\Gamma}$, *(*regula implicaţiei stânga*);*

G5. $\frac{H\cup\{\alpha\}\Longrightarrow\Gamma}{H\Longrightarrow\Gamma\cup\{(\neg\alpha)\}}$, *(*regula negaţiei dreapta*);*

G6. $\frac{H\Longrightarrow\Gamma\cup\{\alpha\},\ H\Longrightarrow\Gamma\cup\{\beta\}}{H\Longrightarrow\Gamma\cup\{(\alpha\wedge\beta)\}}$, *(*regula conjuncţie dreapta*);*

G7. $\frac{H\Longrightarrow\Gamma\cup\{\alpha,\beta\}}{H\Longrightarrow\Gamma\cup\{(\alpha\vee\beta)\}}$, *(*regula disjuncţie dreapta*);*

G8. $\frac{H\cup\{\alpha\}\Longrightarrow\Gamma\cup\{\beta\}}{H\Longrightarrow\Gamma\cup\{(\alpha\to\beta)\}}$, *(*regula implicaţie dreapta*).*

unde H,Γ sunt mulţimi arbitrare de formule; $\alpha,\beta\in FORM$.

Observaţia 2.2.1 Regulile Gentzen definesc modalităţile de „descompunere în formule de adâncime mai mică, a formulelor selectate din mulţimile antecedent/succedent. Se observă ca pentru o formulă selectată, regula Gentzen aplicabilă există şi este unică şi anume este definită de conectiva principală corespunzătoare formulei şi de provenienţa ei (antecedent respectiv succedent). În continuare, regulile $G1, G2$, $G5, G7, G8$ vor fi referite ca reguli de tipul 1, respectiv regulile $G3$, $G4, G6$ sunt reguli de tipul 2.

Exemplul 2.2.1 Presupunem că formula selectată este

$$\theta=(((\alpha\to\beta)\vee((\neg\alpha)\to\gamma))\to(\beta\vee\gamma))$$

şi $S=H\cup\{\theta\}\Longrightarrow\Gamma$. Atunci regula Gentzen aplicabilă este $G4$ şi rezultă $\frac{S_1,S_2}{S}$ unde

$$S_1=H\cup\{(\beta\vee\gamma)\}\Longrightarrow\Gamma,\ S_2=H\Longrightarrow\Gamma\cup\{((\alpha\to\beta)\vee((\neg\alpha)\to\gamma))\}.$$

Pentru

$$S=H\Longrightarrow\Gamma\cup\{\theta\},$$

dacă θ este formula selectată, atunci regula Gentzen aplicabilă este $G8$ şi rezultă

$$\frac{S_1}{S}$$

unde

$$S_1=H\cup\{((\alpha\to\beta)\vee((\neg\alpha)\to\gamma))\}\Longrightarrow\Gamma\cup\{(\beta\vee\gamma)\}$$

Definiţia 2.2.5 *Arborele $T=(V(T),E(T))$ binar, cu rădăcină şi vârfurile etichetate cu secvenţi este un arbore de deducţie, dacă pentru orice $n\in V(T)$,notând cu $\varphi(n)$ secventul etichetă corespunzător vârfului n,*

(i) dacă $od(n)=1$ şi n_1 este unicul succesor al lui n atunci $\frac{\varphi(n_1)}{\varphi(n)}$ este regulă Gentzen;

(ii) dacă $od(n)=2$ şi n_1,n_2 sunt succesorii vârfului n atunci $\frac{\varphi(n_1),\ \varphi(n_2)}{\varphi(n)}$ este regulă Gentzen.

Definiţia 2.2.6 *Arborele de deducţie T este un arbore de demonstractie pentru secventul etichetă a vârfului rădăcină dacă orice vârf terminal are ca etichetă un secvent axiomă.*

Definiţia 2.2.7 *Arborele de deducţie T este un arbore încheiat dacă toate vârfurile terminale au etichete secvenţi încheiaţi.*

Definiţia 2.2.8 *rborele de deducţie T este un arbore contraexemplu dacă există $n \in V(T)$ astfel încât $\varphi(n)$ este secvent încheiat şi nu este secvent axiomă.*

Definiţia 2.2.9 *Secventul S este demonstrabil, notat $\vdash S$, dacă există T arbore de demonstractie cu S etichetă a rădăcinii.*

Exemplul 2.2.2 Fie secventul $S = \{(\alpha \to \beta), ((\neg\alpha) \to \gamma)\} \Longrightarrow \{(\beta \vee \gamma)\}$.

Arborele T reprezentat în Figura 2.1 este un arbore de deducţie cu S etichetă a radăcinii. Deoarece vârfurile terminale sunt n_5, n_7, n_{10}, n_{11} şi au etichetele secvenţi axiomă rezultă $\vdash S$.

$$\begin{tikzcd}
 & & n_7 & & \\
n_5 & & n_6 \arrow[u] & n_{10} & n_{11} \\
n_3 \arrow[u] & & n_4 \arrow[u] & n_8 \arrow[u] & n_9 \arrow[u] \\
 & n_1 \arrow[ul] \arrow[ur] & & n_2 \arrow[u] \arrow[ur] & \\
 & & r \arrow[ul] \arrow[ur] & &
\end{tikzcd}$$

Figura 2.1:

Secvenţii etichete ale vârfurilor arborelui T sunt :

$$\begin{aligned}
&\varphi(r) = S,\\
&\varphi(n_1) = \{\beta, ((\neg\alpha) \to \gamma)\} \Longrightarrow \{(\beta \vee \gamma)\},\\
&\varphi(n_2) = \{((\neg\alpha) \to \gamma)\} \Longrightarrow \{\alpha, (\beta \vee \gamma)\},\\
&\varphi(n_3) = \{\beta, \gamma\} \Longrightarrow \{(\beta \vee \gamma)\},\\
&\varphi(n_4) = \{\beta\} \Longrightarrow \{(\neg\alpha), (\beta \vee \gamma)\},\\
&\varphi(n_5) = \{\beta, \gamma\} \Longrightarrow \{\beta, \gamma\},\\
&\varphi(n_6) = \{\alpha, \beta\} \Longrightarrow \{(\beta \vee \gamma)\},\\
&\varphi(n_7) = \{\alpha, \beta\} \Longrightarrow \{\beta, \gamma\},\\
&\varphi(n_8) = \{\gamma\} \Longrightarrow \{\alpha, (\beta \vee \gamma)\},\\
&\varphi(n_9) = \emptyset \Longrightarrow \{(\neg\alpha), \alpha, (\beta \vee \gamma)\},\\
&\varphi(n_{10}) = \{\gamma\} \Longrightarrow \{\alpha, \beta, \gamma\},\\
&\varphi(n_{11}) = \{\alpha\} \Longrightarrow \{\alpha, (\beta \vee \gamma)\}.
\end{aligned}$$

Tentativa de construcţie a unui arbore de demonstraţie pentru un secvent dat S poate fi realizată pe baza unei tehnici de expandare a arborelui curent generat. Fie $T = (V(T), E(T))$ arborele curent. Iniţial, $V(T) = \{r\}$, $\varphi(r) = S$, $E(T) = \emptyset$.

Pasul de expandare constă în efectuarea următoarelor operaţii:

P1 Selectează un vârf n aflat pe frontiera arborelui curent astfel încât $\varphi(n)$ nu este secvent încheiat şi $\varphi(n)$ nu este secvent axiomă

P2. Selectează o formulă $\alpha \notin V$ din antecedentul/succedentul secventului $\varphi(n)$

P3. Identifică regula Gentzen aplicabilă formulei α

(i) Dacă regula este de tipul 1, fie aceasta $\frac{S_1}{\varphi(n)}$ atunci extinde T;

$$T \leftarrow (V(T) \cup \{n_1\}, E(T) \cup \{nn_1\})$$

unde

$$n_1 \notin V(T), \varphi(n_1) = S_1$$

(ii) Dacă regula este de tipul 2, fie aceasta $\frac{S_1, S_2}{\varphi(n)}$ atunci extinde T;

$$T \leftarrow (V(T) \cup \{n_1, n_2\}, E(T) \cup \{nn_1, nn_2\})$$

unde

$$n_1, n_2 \notin V(T), \varphi(n_1) = S_1, \varphi(n_2) = S_2$$

Condiţia de oprire a procesului de expandare: pentru efectuarea pasului P1 este necesară verificarea etichetelor vârfurilor aflate pe frontiera arborelui curent generat. Dacă există cel puţin un secvent încheiat şi care nu este axiomă atunci se decide terminarea procesului de expandare cu decizia „S nu este demonstrabil". Dacă toate vârfurile aflate pe frontiera arborelui curent generat au ca etichete secvenţi axiomă, atunci arborele curent generat este un arbore de demonstraţie pentru S şi în acest caz procesul de expandare se încheie cu decizia „S este secvent demonstrabil".

Pentru ilustrarea metodei descrise vom relua exemplul 2.2.2; notăm cu ∂T frontiera arborelui T

Iniţial $T = (\{r\}, \emptyset)$, $\varphi(r) = \{(\alpha \to \beta), ((\neg\alpha) \to \gamma)\} \Longrightarrow \{(\beta \vee \gamma)\}$;

1. $\partial T = \{r\}, n = r$; formula selectată $(\alpha \to \beta)$; regula G4;

$$T: \quad \begin{array}{ccc} n_1 & & n_2 \\ & \nwarrow \nearrow & \\ & r & \end{array}$$

$\varphi(n_1) = \{\beta, ((\neg\alpha) \to \gamma)\} \Longrightarrow \{(\beta \vee \gamma)\}$, $\varphi(n_2) = \{((\neg\alpha) \to \gamma)\} \Longrightarrow \{\alpha, (\beta \vee \gamma)\}$.

2. $\partial T = \{n_1, n_2\}; n = n_1$; formula selectată $((\neg\alpha) \to \gamma)$; regula G4;

$$T: \quad \begin{array}{ccccc} n_3 & & n_4 & & \\ & \nwarrow \nearrow & & & \\ & n_1 & & & n_2 \\ & & & \nwarrow \nearrow & \\ & & & r & \end{array}$$

$\varphi(n_3) = \{\beta, \gamma\} \Longrightarrow \{(\beta \vee \gamma)\}$, $\varphi(n_4) = \{\beta\} \Longrightarrow \{(\neg\alpha), (\beta \vee \gamma)\}$.

3. $\partial T = \{n_3, n_4, n_2\}$; $n = n_3$; formula selectată $(\beta \vee \gamma)$; regula G7;

$$T: \quad \begin{array}{ccccc} n_5 & & & & \\ \uparrow & & & & \\ n_3 & & n_4 & & \\ & \nwarrow \nearrow & & & \\ & n_1 & & & n_2 \\ & & & \nwarrow \nearrow & \\ & & & r & \end{array}$$

$\varphi(n_5) = \{\beta, \gamma\} \Longrightarrow \{\beta, \gamma\}$ este secvent axiomă.

4. $\partial T = \{n_5, n_4, n_2\}$; $n = n_4$; formula selectată $(\neg\alpha)$; regula G5;

$$T:\ \begin{array}{ccccc} n_5 & & n_6 & & \\ \uparrow & & \uparrow & & \\ n_3 & & n_4 & & \\ & \nwarrow\!\nearrow & & & \\ & n_1 & & & n_2 \\ & & & \nwarrow\!\nearrow & \\ & & r & & \end{array}$$

$\varphi(n_6) = \{\alpha, \beta\} \Longrightarrow \{(\beta \vee \gamma)\}$.

5. $\partial T = \{n_5, n_6, n_2\}$; $n = n_6$; formula selectată $(\beta \vee \gamma)$; regula G7;

$$T:\ \begin{array}{ccccc} & & n_7 & & \\ & & \uparrow & & \\ n_5 & & n_6 & & \\ \uparrow & & \uparrow & & \\ n_3 & & n_4 & & \\ & \nwarrow\!\nearrow & & & \\ & n_1 & & & n_2 \\ & & & \nwarrow\!\nearrow & \\ & & r & & \end{array}$$

$\varphi(n_7) = \{\alpha, \beta\} \Longrightarrow \{\beta, \gamma\}$ este secvent axiomă.

6. $\partial T = \{n_5, n_7, n_2\}$; $n = n_2$; formula selectată $((\neg\alpha) \to \gamma)$; regula G4;

$$T:\ \begin{array}{ccccc} & & n_7 & & \\ & & \uparrow & & \\ n_5 & & n_6 & & \\ \uparrow & & \uparrow & & \\ n_3 & & n_4 & n_8 & n_9 \\ & \nwarrow\!\nearrow & & \nwarrow\!\nearrow & \\ & n_1 & & n_2 & \\ & & \nwarrow\!\nearrow & & \\ & & r & & \end{array}$$

$\varphi(n_8) = \{\gamma\} \Longrightarrow \{\alpha, (\beta \vee \gamma)\}$, $\varphi(n_9) = \emptyset \Longrightarrow \{(\neg\alpha), \alpha, (\beta \vee \gamma)\}$.

7. $\partial T = \{n_5, n_7, n_8, n_9\}$; $n = n_8$; formula selectată $(\beta \vee \gamma)$; regula G7;

$$T:\ \begin{array}{ccccc} & & n_7 & & \\ & & \uparrow & & \\ n_5 & & n_6 & n_{10} & \\ \uparrow & & \uparrow & \uparrow & \\ n_3 & & n_4 & n_8 & n_9 \\ & \nwarrow\!\nearrow & & \nwarrow\!\nearrow & \\ & n_1 & & n_2 & \\ & & \nwarrow\!\nearrow & & \\ & & r & & \end{array}$$

$\varphi(n_{10}) = \{\gamma\} \Longrightarrow \{\alpha, \beta, \gamma\}$ este secvent axiomă.

8. $\partial T = \{n_5, n_7, n_{10}, n_9\}$; $n = n_9$; formula selectată $(\neg\alpha)$; regula G5;

$$
\begin{array}{ccccccccc}
 & & & & n_7 & & & & \\
 & & & & \uparrow & & & & \\
 & n_5 & & & n_6 & & n_{10} & & n_{11} \\
 & \uparrow & & & \uparrow & & \uparrow & & \uparrow \\
T: & n_3 & & & n_4 & & n_8 & & n_9 \\
 & & \nwarrow & \nearrow & & & \nwarrow\nearrow & & \\
 & & & n_1 & & & n_2 & & \\
 & & & & \nwarrow\nearrow & & & & \\
 & & & & r & & & &
\end{array}
$$

$\varphi(n_{11}) = \{\alpha\} \Longrightarrow \{\alpha, (\beta \vee \gamma)\}$ este secvent axiomă.

9. $\partial T = \{n_5, n_7, n_{10}, n_{11}\}$; toate vârfurile aflate pe frontiera arborelui curent generat au ca etichete secvenţi axiomă,deci se procesul de expandare se încheie cu decizia „S este secvent demonstrabil".

2.3 Aplicaţii

Exemplul 2.3.1 *Să se verifice dacă următorii secvenţi sunt demonstrabili:*

a) $S = \{(\neg a \wedge \neg b)\} \Longrightarrow \{\neg(a \vee b)\}$
Soluţie
Folosind regulile de inferenţă Gentzen descrise în Definiţia 2.2.4 obţinem

$$\varphi(n_1) = \{(\neg a, \neg b)\} \Longrightarrow \{\neg(a \vee b)\} \textit{ (G2)}$$

$$\varphi(n_2) = \emptyset \Longrightarrow \{\neg(a \vee b), a, b\} \textit{ (G1)}$$

$$\varphi(n_3) = \{(a \vee b)\} \Longrightarrow \{a, b\} \textit{ (G5)}$$

$$\varphi(n_4) = \{a\} \Longrightarrow \{a, b\} \textit{ (G3) (secvent axiomă)}$$

$$\varphi(n_5) = \{b\} \Longrightarrow \{a, b\} \textit{ (G3) (secvent axiomă).}$$

Secventul considerat este demonstrabil, deoarece etichetele nodurilor terminale sunt de tipul secvent axiomă.

b) $S = \{(\neg r \vee (\neg(\neg a \wedge \neg b))), (\neg r \to \neg b)\} \Longrightarrow \{\neg(a \vee b) \to (\neg b)\}$

Plecând de la relaţia dată

$$S = \{(\neg r \vee (\neg(\neg a \wedge \neg b))), (\neg r \to \neg b)\} \Longrightarrow \{\neg(a \vee b) \to (\neg b)\}$$

şi aplicând regula implicaţie la dreapta (G8) rezultă

$$\varphi(n_1) = \{(\neg r \vee (\neg(\neg a \wedge \neg b))), (\neg r \to \neg b), \neg(a \vee b)\} \Longrightarrow \{(\neg b)\}.$$

Folosind în continuare regulile enunţate în Definiţia 2.2.4 se obţin succesiv relaţiile:

$$\varphi(n_2) = \{(\neg r \vee (\neg(\neg a \wedge \neg b))), (\neg r \to \neg b), \neg(a \vee b), b\} \Longrightarrow \emptyset \textit{ (G5)}$$

$$\varphi(n_3) = \{(\neg r \vee (\neg(\neg a \wedge \neg b))), (\neg r \to \neg b), b\} \implies \{(a \vee b)\} \text{ (G4)}$$
$$\varphi(n_4) = \{(\neg r \vee (\neg(\neg a \wedge \neg b))), (\neg r \to \neg b), b\} \implies \{(a, b)\} \text{ (G7)}$$
$$\varphi(n_5) = \{(\neg r \vee (\neg(\neg a \wedge \neg b))), (\neg b), b\} \implies \{a, b\} \text{ (G4)}$$

şi

$$\varphi(n_6) = \{(\neg r \vee (\neg(\neg a \wedge \neg b))), (\neg b), b\} \implies \{a, b, \neg r\} \text{ (G4)}$$

În continuare selectăm formula $\varphi(n_5) = \{(\neg r \vee (\neg(\neg a \wedge \neg b))), (\neg b), b\} \implies \{a, b\}$.

$$\varphi(n_7) = \{(\neg r \vee (\neg(\neg a \wedge \neg b))), b\} \implies \{a, b, b\} \text{ (G1)}$$
$$\varphi(n_8) = \{(\neg r), b\} \implies \{a, b\} \text{ (G3)}$$
$$\varphi(n_9) = \{(\neg(\neg a \wedge \neg b)), b\} \implies \{a, b\} \text{ (G3)}$$

Din $\varphi(n_8)$ *obţinem*

$$\varphi(n_{10}) = \{b\} \implies \{a, b, r\} \text{ (G1) (secvent axiomă).}$$

Folosind $\varphi(n_9)$ *şi (G1) rezultă*

$$\varphi(n_{11}) = \{b\} \implies \{a, b, (\neg a \wedge \neg b))\}$$
$$\varphi(n_{12}) = \{b\} \implies \{a, b, (\neg a))\} \text{ (G6)}$$
$$\varphi(n_{13}) = \{b\} \implies \{a, b, (\neg b))\} \text{ (G6)}$$

Relaţiile obţinute din $\varphi(n_{12})$, $\varphi(n_{13})$ *cărora li s-a aplicat (G5) sunt secvenţi axiomă:*

$$\varphi(n_{14}) = \{b, a\} \implies \{a, b\}$$
$$\varphi(n_{15}) = \{b, b\} \implies \{a, b\}$$

Revenim la $\varphi(n_6) = \{(\neg r \vee (\neg(\neg a \wedge \neg b))), (\neg b), b\} \implies \{a, b, \neg r\}$ *şi procedăm asemănător*

$$\varphi(n_{16}) = \{(\neg r \vee (\neg(\neg a \wedge \neg b))), (\neg b), b, r\} \implies \{a, b\} \text{ (G5)}$$
$$\varphi(n_{17}) = \{(\neg r \vee (\neg(\neg a \wedge \neg b))), b, r\} \implies \{a, b, b\} \text{ (G1)}$$
$$\varphi(n_{18}) = \{(\neg r), b, r\} \implies \{a, b\} \text{ (G3)}$$
$$\varphi(n_{19}) = \{(\neg(\neg a \wedge \neg b)), b, r\} \implies \{a, b\} \text{ (G3)}$$

Din $\varphi(n_{18})$ *obţinem*

$$\varphi(n_{20}) = \{b, r\} \implies \{a, b, r\} \text{ (G1) (secvent axiomă).}$$

Folosind $\varphi(n_{19})$ *şi (G1) rezultă*

$$\varphi(n_{21}) = \{b, r\} \implies \{a, b, (\neg a \wedge \neg b))\}$$
$$\varphi(n_{22}) = \{b, r\} \implies \{a, b, (\neg a))\} \text{ (G6)}$$
$$\varphi(n_{23}) = \{b, r\} \implies \{a, b, (\neg b))\} \text{ (G6)}$$

În ultimele două relaţii aplicăm regula negaţiei la dreapta şi obţinem secvenţi axiomă:

$$\varphi(n_{22}) = \{b, r, a\} \implies \{a, b\} \text{ (G5)}$$
$$\varphi(n_{23}) = \{b, r, b\} \implies \{a, b\} \text{ (G5).}$$

Secventul este demonstrabil deoarece orice vârf terminal are ca etichetă un secvent axiomă.

2.4 Exerciţii

Exerciţiul 2.4.1 *Să se verifice dacă următorii secvenţi sunt demonstrabili:*

a) $S = \{((\neg a \vee \neg c) \vee \neg b)\} \implies \{\neg(a \wedge c) \wedge b)\}$

b) $S = \{(\neg c \vee (\neg(\neg a \wedge \neg b))),\ (\neg c \to \neg b)\} \implies \{\neg(a \vee b) \to (\neg b)\}$

c) $S = \{(\neg a \vee b),\ (a \vee (c \wedge d))\} \implies \{\neg b \to (c \wedge d)\}$

d) $S = \{a \wedge (\neg(\neg b \vee c)), (\neg a \to \neg b)\} \implies \{(\neg a \wedge \neg b) \to (\neg b)\}$

e) $S = \{\neg b \wedge (\neg(a \vee \neg b)), (\neg c \to (\neg a \vee \neg b))\} \implies \{\neg(a \vee \neg b) \to c\}$

f) $S = \{\neg a \vee (a \vee \neg b)\} \implies \{(a \wedge \neg(a \vee \neg b)),\ (\neg a \vee (\neg b \vee \neg a))\}.$

Capitolul 3

Semantica limbajului calculului cu propoziţii

3.1 Introducere

Definiţia 3.1.1 *Numim* funcţie de adevăr *orice funcţie* $h : V \to \{T, F\}$.

Lema 3.1.1 *Fie h o funcţie de adevăr. Atunci există şi este unică*

$$I(h) : FORM \to \{T, F\}$$

astfel încât următoarele cerinţe să fie îndeplinite:

1. *pentru orice $\alpha \in V$, $I(h)(\alpha) = h(\alpha)$;*
2. *pentru orice $\alpha, \beta \in FORM$,*
   (a) $I(h)((\neg\alpha)) = \neg I(h)(\alpha)$;
   (b) $I(h)((\alpha \wedge \beta)) = I(h)(\alpha) \wedge I(h)(\beta)$;
   (c) $I(h)((\alpha \vee \beta)) = I(h)(\alpha) \vee I(h)(\beta)$;
   (d) $I(h)((\alpha \to \beta)) = I(h)(\alpha) \to I(h)(\beta)$;
   (e) $I(h)((\alpha \leftrightarrow \beta)) = I(h)(\alpha) \leftrightarrow I(h)(\beta)$.

Demonstraţie. Definim $I(h) : FORM \to \{T, F\}$ prin, $\alpha \in FORM$,

$$I(h)(\alpha) = \begin{cases} h(\alpha), & \text{dacă } \alpha \in V, \\ \neg I(h)(\beta), & \text{dacă } \alpha = (\neg\beta), \\ I(h)(\beta) \wedge I(h)(\gamma), & \text{dacă } \alpha = (\beta \wedge \gamma), \\ I(h)(\beta) \vee I(h)(\gamma), & \text{dacă } \alpha = (\beta \vee \gamma), \\ I(h)(\beta) \to I(h)(\gamma), & \text{dacă } \alpha = (\beta \to \gamma), \\ I(h)(\beta) \leftrightarrow I(h)(\gamma), & \text{dacă } \alpha = (\beta \leftrightarrow \gamma). \end{cases}$$

Deoarece orice $\alpha \in FORM \setminus V$ este de unul şi numai de unul din tipurile: $\alpha = (\neg\beta)$ sau $\alpha = (\beta\rho\gamma)$ cu $\rho \in L \setminus \{\neg\}$, rezultă că funcţia $I(h)$ este bine definită şi verifică proprietăţile $1, 2.a - 2.e$.

Presupunem că funcţia $I : FORM \to \{T, F\}$ verifică proprietăţile din enunţ. Demonstrăm prin inducţie asupra nivelului de adâncime că $I = I(h)$. Evident, orice

$\alpha \in FORM$ cu $\hbar(\alpha) = 0$ rezultă $\alpha \in V$, deci $I(\alpha) = h(\alpha) = I(h)(\alpha)$. Presupunem că $I(\alpha) = I(h)(\alpha)$ pentru orice $\alpha \in FORM$ cu $\hbar(\alpha) \leq k$. Fie $\alpha \in FORM$ astfel încât $\hbar(\alpha) = k+1$. Dacă $\alpha = (\neg\beta)$atunci $\hbar(\beta) = k$ şi conform ipotezei inductive $I(\beta) = I(h)(\beta)$deci $I(\alpha) = I(h)(\alpha)$. Un argument similar conduce evident la concluzia $I(\alpha) = I(h)(\alpha)$ pentru fiecare din cazurile $\alpha = (\beta\rho\gamma)$ cu $\rho \in L \setminus \{\neg\}$. ■

Numim *interpretare* orice funcţie $I : FORM \to \{T, F\}$ care verifică proprietăţile din enunţul Lemei 3.1.1. Extensia $I(h)$ a funcţiei de adevăr h la mulţimea formulelor este *interpretarea indusă de h.* Dacă $I(\alpha) = T$ atunci spunem că interpretarea I validează formula α ,sau echivalent, I este model pentru α. Notăm cu $\aleph(\alpha)$ mulţimea modelelor formulei α. Dacă $I(\alpha) = F$ atunci spunem că I falsifică α. Notăm cu $\Im$ mulţimea tuturor interpretărilor.

Observaţia 3.1.1 Dacă $I : FORM \to \{T, F\}$ verifică cerinţele din enunţul Lemei 3.1.1, atunci există şi este unică funcţia de adevăr h astfel încât $I = I(h)$. Intr-adevăr, utilizând rezultatul stabilit de Lema 3.1.1, pentru restricţia $h = I\mid_{FORM}$, obţinem concluzia $I = I(h)$.

Definiţia 3.1.2 *Formula α este validabilă dacă $\aleph(\alpha) \neq \emptyset$. Dacă $\aleph(\alpha) = \emptyset$ atunci α este invalidabilă. Termenii sinonimi cu invalidabil, frecvent utilizaţi, sunt insatisfiabil respectiv logic fals.*

Definiţia 3.1.3 *Formula α este tautologie dacă $\aleph(\alpha) = \Im$.*

Definiţia 3.1.4 *Fie $\alpha \in FORM$. Dacă $(\alpha \leftrightarrow \beta)$ este o tautologie spunem că formulele α, β sunt logic echivalente.*

Definiţia 3.1.5 *Fie $H \subset FORM$ şi $I \in \Im$. Spunem că I este model pentru H dacă pentru orice $\alpha \in H$, $I(\alpha) = T$. Notăm cu $\aleph(H)$ mulţimea modelelor mulţimii H. Dacă $\aleph(H) \neq \emptyset$ atunci spunem că H este consistentă (validabilă), respectiv, dacă $\aleph(H) = \emptyset$ atunci H este inconsistentă (invalidabilă).*

Observaţia 3.1.2 Din Definiţia 3.1.5 rezultă imediat

$$\aleph(H) = \bigcap_{\alpha \in H} \aleph(\alpha).$$

De asemenea, dacă $H, \Gamma \subset FORM$, atunci

$$\aleph(H \cup \Gamma) = \aleph(H) \cap \aleph(\Gamma),\ \aleph(H \cap \Gamma) \supseteq \aleph(H) \cup \aleph(\Gamma).$$

Lema 3.1.2 *Fie $H \subset FORM$ finită, $H = \{\alpha_1, \ldots, \alpha_n\}$.Atunci, $\aleph(H) = \aleph(\delta_n)$, unde $\delta_1 = \alpha_1, \delta_k = (\delta_{k-1} \wedge \alpha_k), 2 \leq k \leq n$, adică*

$$\aleph(H) = \aleph\left(\bigwedge_{i=1}^{n} \alpha_i\right).$$

Demonstraţie. Demonstrăm proprietatea afirmată în enunţ prin inducţie asupra cardinalului mulţimii H. Evident proprietatea este verificată pentru mulţimi de cardinal egal cu 1. Fie $H = \{\alpha_1, \alpha_2\}$ deci

$$\delta_2 = (\delta_1 \wedge \alpha_2) = (\alpha_1 \wedge \alpha_2).$$

Demonstrăm că $\aleph(\{\alpha_1, \alpha_2\}) = \aleph(\{(\alpha_1 \wedge \alpha_2)\})$. Într-adevăr, pentru orice

$$I \in \Im, I((\alpha_1 \wedge \alpha_2)) = I(\alpha_1) \wedge I(\alpha_2),$$

deci

$$I((\alpha_1 \wedge \alpha_2)) = T \text{ dacă şi numai dacă } I(\alpha_1) = I(\alpha_2) = T$$

adică

$$\aleph(\{(\alpha_1 \wedge \alpha_2)\}) = \aleph(\alpha_1) \cap \aleph(\alpha_1) = \aleph(\{\alpha_1, \alpha_2\}).$$

Presupunem proprietatea adevărată pentru orice mulţime H de cardinal cel mult egal cu k. Fie $H = \{\alpha_1, \ldots, \alpha_k, \alpha_{k+1}\}$. Utilizând observaţia precedentă şi ipoteza inductivă obţinem,

$$\begin{aligned}\aleph(H) &= \aleph(\{\alpha_1, \ldots, \alpha_k\} \cup \{\alpha_{k+1}\}) = \aleph(\{\alpha_1, \ldots, \alpha_k\}) \cap \aleph(\{\alpha_{k+1}\}) \\ &= \aleph(\{\delta_k\}) \cap \aleph(\{\alpha_{k+1}\}) = \aleph(\{(\delta_k \wedge \alpha_{k+1})\}) = \aleph(\delta_{k+1}) \\ &= \aleph(\textstyle\bigwedge \alpha_i).\end{aligned}$$

■

Observaţia 3.1.3 *Deoarece pentru orice $\alpha \in FORM$, şi $I \in \Im$, şi $I((\neg\alpha)) = \neg I(\alpha)$ rezultă $I(\alpha) = T$ dacă şi numai dacă $I((\neg\alpha)) = F$ deci $\aleph((\neg\alpha)) = \Im \setminus \aleph(\alpha)$.*

Lema 3.1.3 *Dacă $\gamma_1 = \beta_1, \gamma_k = (\gamma_{k-1} \vee \beta_k), 2 \leq k \leq n$ unde $\{\beta_1, \ldots, \beta_n\} \subset FORM$, atunci $\aleph(\gamma_n) = \bigcup\limits_{k=1}^{n} \aleph(\beta_k)$.*

Demonstraţie. Demonstrăm proprietatea afirmată în enunţ prin inducţie asupra cardinalului mulţimii. Evident proprietatea este adevărată pentru $n = 1$. Pentru $n = 2$,

$$\gamma_2 = (\gamma_1 \vee \beta_2) = (\beta_1 \vee \beta_2).$$

Conform definiţiei, pentru orice $I \in \Im$,

$$I((\beta_1 \vee \beta_2)) = I(\beta_1) \vee I(\beta_2),$$

deci $I((\beta_1 \vee \beta_2)) = T$ dacă şi numai dacă $I(\beta_1) = T$ sau $I(\beta_2) = T$. Rezultă

$$\aleph(\gamma_2) = \bigcup_{k=1}^{2} \aleph(\beta_k).$$

Presupunem proprietatea adevărată pentru orice $\{\beta_1, \ldots, \beta_k\} \subset FORM$, $2 \leq k \leq n$. Fie

$$\{\beta_1, \ldots, \beta_{n+1}\} \subset FORM;\ \gamma_{n+1} = (\gamma_n \vee \beta_{n+1}).$$

Obţinem, utilizând ipoteza inductivă,

$$\aleph(\gamma_{n+1}) = \aleph(\gamma_n) \cup \aleph(\beta_{n+1}) = \bigcup_{k=1}^{n} \aleph(\beta_k) \cup \aleph(\beta_{n+1}) = \bigcup_{k=1}^{n+1} \aleph(\beta_k).$$

■

Definiţia 3.1.6 *Spunem că formulele α, β sunt semantic echivalente (notat $\alpha \equiv \beta$), dacă $\aleph(\alpha) = \aleph(\beta)$.*

Observaţia 3.1.4 Relaţia de echivalenţă semantică este o relaţie de echivalenţă pe mulţimea $FORM$.

Lema 3.1.4 *Pentru orice $\alpha, \beta, \gamma \in FORM$,*

1. $(\alpha \vee \alpha) \equiv \alpha,\ (\alpha \wedge \alpha) \equiv \alpha$;
2. $(\neg(\neg\alpha)) \equiv \alpha$;
3. $(\alpha \to \beta) \equiv ((\neg\alpha) \vee \beta)\,,\ (\alpha \leftrightarrow \beta) \equiv ((\alpha \to \beta) \wedge (\beta \to \alpha))\,;$
4. $(\neg(\alpha \wedge \beta)) \equiv ((\neg\alpha) \vee (\neg\beta))\,,\ (\neg(\alpha \vee \beta)) \equiv ((\neg\alpha) \wedge (\neg\beta))\,;$
5. $(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))\,,\ (\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))\,;$
6. $(\alpha \vee \beta) \equiv ((\neg\alpha) \to \beta)\,,\ (\alpha \wedge \beta) \equiv (\neg(\alpha \to (\neg\beta)))\,.$

Demonstraţie. Justificarea este imediată şi rezultă direct pe baza definiţiilor. ■

Observaţia 3.1.5 Din proprietăţile (3) şi (6) rezultă că mulţimea de conective $\{\to, \neg\}$ este completă din punct de vedere semantic, în sensul că pentru orice formulă α există formula α' astfel încât $\alpha \equiv \alpha'$ şi singurele simboluri de tip conectivă din structura simbolică α' sunt $\{\to, \neg\}$.

Lema 3.1.5 *Dacă $\alpha \equiv \beta$ şi $\gamma \equiv \delta$, atunci*

1. $(\neg\alpha) \equiv (\neg\beta)$;
2. $(\alpha\rho\gamma) \equiv (\beta\rho\delta)$ *pentru orice* $\rho \in L \setminus \{\rho\}$.

Demonstraţie. Justificarea este imediată şi rezultă direct pe baza definiţiilor. ■

Teorema 3.1.1 (Teorema de consistenţă a calculului cu propoziţii) *Orice teoremă este tautologie.*

Demonstraţie.
Justificarea afirmaţiei rezultă imediat prin inducţie asupra lungimii demonstraţiei formale, dacă demonstrăm că orice exemplu de axiomă este tautologie.
Fie $\alpha, \beta, \gamma \in FORM$ arbitrare,

$$\sigma = \{\alpha \mid a,\ \beta \mid b,\ \gamma \mid c\}\,.$$

Pentru $I \in \Im$ arbitrară, utilizând observaţiile precedente, obţinem
1. Axioma $\overline{\alpha}_1$:

$$\begin{aligned} I(\overline{\alpha}_1\sigma) &= I((\alpha \to (\beta \to \alpha))) = I(\alpha) \to (I(\beta) \to I(\alpha)) = \\ &= \neg I(\alpha) \vee \neg I(\beta) \vee I(\alpha) = T. \end{aligned}$$

2. Axioma $\overline{\alpha}_2$:

$$\begin{aligned} I(\overline{\alpha}_2\sigma) &= I(((\alpha \to (\alpha \to \beta)) \to (\alpha \to \beta))) = \\ &= \neg I((\alpha \to (\alpha \to \beta))) \vee I(\alpha \to \beta) = \\ &= \neg(\neg I(\alpha) \vee \neg I(\alpha) \vee I(\beta)) \vee (\neg I(\alpha) \vee I(\beta)) = \\ &= \neg(\neg I(\alpha) \vee I(\beta)) \vee (\neg I(\alpha) \vee I(\beta)) = \\ &= (I(\alpha) \wedge \neg I(\beta)) \vee (\neg I(\alpha) \vee I(\beta)) = \\ &= (I(\alpha) \vee \neg I(\alpha) \vee I(\beta)) \wedge (\neg I(\beta) \vee \neg I(\alpha) \vee I(\beta)) = T. \end{aligned}$$

3. Axioma $\overline{\alpha}_3$:

$$\begin{aligned}
I(\overline{\alpha}_3\sigma) &= I(((\alpha \to \beta) \to ((\beta \to \gamma) \to (\alpha \to \gamma)))) = \\
&= \neg I((\alpha \to \beta)) \vee (\neg I((\beta \to \gamma)) \vee I((\alpha \to \gamma))) = \\
&= \neg(\neg I(\alpha) \vee I(\beta)) \vee (\neg(\neg I(\beta) \vee I(\gamma)) \vee (\neg I(\alpha) \vee I(\gamma))) = \\
&= (I(\alpha) \wedge \neg I(\beta)) \vee ((I(\beta) \wedge \neg I(\gamma)) \vee (\neg I(\alpha) \vee I(\gamma))) = \\
&= (I(\alpha) \wedge \neg I(\beta)) \vee ((I(\beta) \vee \neg I(\alpha) \vee I(\gamma)) \wedge (\neg I(\gamma) \vee \neg I(\alpha) \vee I(\gamma))) = \\
&= (I(\alpha) \wedge \neg I(\beta)) \vee ((I(\beta) \vee \neg I(\alpha) \vee I(\gamma)) \wedge T) = \\
&= (I(\alpha) \wedge \neg I(\beta)) \vee (I(\beta) \vee \neg I(\alpha) \vee I(\gamma)) = \\
&= (I(\alpha) \vee I(\beta) \vee \neg I(\alpha) \vee I(\gamma)) \wedge (\neg I(\beta) \vee I(\beta) \vee \neg I(\alpha) \vee I(\gamma)) = \\
&= T \wedge T = T.
\end{aligned}$$

4. Axioma $\overline{\alpha}_4$:

$$\begin{aligned}
I(\overline{\alpha}_4\sigma) &= I(((\alpha \leftrightarrow \beta) \to (\alpha \to \beta))) = \\
&= \neg I((\alpha \leftrightarrow \beta)) \vee I((\alpha \to \beta)) = \neg(I((\alpha \to \beta)) \wedge I((\beta \to \alpha))) \vee I((\alpha \to \beta)) = \\
&= \neg I((\alpha \to \beta)) \vee \neg I((\beta \to \alpha)) \vee I((\alpha \to \beta)) = T.
\end{aligned}$$

5. Axioma $\overline{\alpha}_5$:

$$\begin{aligned}
I(\overline{\alpha}_5\sigma) &= I(((\alpha \leftrightarrow \beta) \to (\beta \to \alpha))) = \\
&= \neg I((\alpha \leftrightarrow \beta)) \vee I((\beta \to \alpha)) = \neg(I((\alpha \to \beta)) \wedge I((\beta \to \alpha))) \vee I((\beta \to \alpha)) = \\
&= \neg I((\alpha \to \beta)) \vee \neg I((\beta \to \alpha)) \vee I((\beta \to \alpha)) = T.
\end{aligned}$$

6. Axioma $\overline{\alpha}_6$:

$$\begin{aligned}
I(\overline{\alpha}_6\sigma) &= I(((\alpha \to \beta) \to ((\beta \to \alpha) \to (\alpha \leftrightarrow \beta)))) = \\
&= \neg I((\alpha \to \beta)) \vee (\neg I((\beta \to \alpha)) \vee I((\alpha \leftrightarrow \beta))) = \\
&= \neg I((\alpha \to \beta)) \vee (\neg I((\beta \to \alpha)) \vee (I((\alpha \to \beta)) \wedge I((\beta \to \alpha)))) = \\
&= \neg I((\alpha \to \beta)) \vee ((\neg I((\beta \to \alpha)) \vee I((\alpha \to \beta))) \wedge (\neg I((\beta \to \alpha)) \vee I((\beta \to \alpha)))) = \\
&= \neg I((\alpha \to \beta)) \vee ((\neg I((\beta \to \alpha)) \vee I((\alpha \to \beta))) \wedge T) = \\
&= \neg I((\alpha \to \beta)) \vee (\neg I((\beta \to \alpha)) \vee I((\alpha \to \beta))) = T.
\end{aligned}$$

7. Axioma $\overline{\alpha}_7$:

$$\begin{aligned}
I(\overline{\alpha}_7\sigma) &= I((((\neg\alpha) \to (\neg\beta)) \to (\beta \to \alpha))) = \\
&= \neg I(((\neg\alpha) \to (\neg\beta))) \vee I((\beta \to \alpha)) = \\
&= \neg(\neg I((\neg\alpha)) \vee I((\neg\beta))) \vee (\neg I(\beta) \vee I(\alpha)) = \\
&= \neg(I(\alpha) \vee \neg I(\beta)) \vee (\neg I(\beta) \vee I(\alpha)) = \\
&= (\neg I(\alpha) \wedge I(\beta)) \vee (\neg I(\beta) \vee I(\alpha)) = \\
&= (\neg I(\alpha) \vee \neg I(\beta) \vee I(\alpha)) \wedge (I(\beta) \vee \neg I(\beta) \vee I(\alpha)) = T \wedge T = T.
\end{aligned}$$

8. Axioma $\overline{\alpha}_8$:

$$\begin{aligned}
I(\overline{\alpha}_8\sigma) &= I(((\alpha \vee \beta) \leftrightarrow ((\neg\alpha) \to \beta))) = \\
&= I(((\alpha \vee \beta) \to ((\neg\alpha) \to \beta))) \wedge I((((\neg\alpha) \to \beta) \to (\alpha \vee \beta))) = \\
&= (\neg I((\alpha \vee \beta)) \vee I(((\neg\alpha) \to \beta))) \wedge (\neg I(((\neg\alpha) \to \beta)) \vee I((\alpha \vee \beta))) = \\
&= (\neg I((\alpha \vee \beta)) \vee (\neg I((\neg\alpha)) \vee I(\beta))) \wedge (\neg(\neg I((\neg\alpha)) \vee I(\beta)) \vee I((\alpha \vee \beta))) = \\
&= (\neg I((\alpha \vee \beta)) \vee (I(\alpha) \vee I(\beta))) \wedge (\neg(I(\alpha) \vee I(\beta)) \vee I((\alpha \vee \beta))) = \\
&= (\neg I((\alpha \vee \beta)) \vee I((\alpha \vee \beta))) \wedge (\neg I((\alpha \vee \beta)) \vee I((\alpha \vee \beta))) = T.
\end{aligned}$$

9. Axioma $\overline{\alpha}_9$:

$$\begin{aligned}
I(\overline{\alpha}_9\sigma) &= I(((\alpha\wedge\beta)\leftrightarrow(\neg((\neg\alpha)\vee(\neg\beta)))))\\
&= I(((\alpha\wedge\beta)\rightarrow(\neg((\neg\alpha)\vee(\neg\beta)))))\wedge I(((\neg((\neg\alpha)\vee(\neg\beta)))\rightarrow(\alpha\wedge\beta)))\\
&= (\neg I((\alpha\wedge\beta))\vee I((\neg((\neg\alpha)\vee(\neg\beta)))))\wedge(\neg I((\neg((\neg\alpha)\vee(\neg\beta))))\vee I((\alpha\wedge\beta)))\\
&= (\neg I((\alpha\wedge\beta))\vee\neg(I((\neg\alpha))\vee I((\neg\beta))))\wedge((I((\neg\alpha))\vee I((\neg\beta)))\vee I((\alpha\wedge\beta)))\\
&= (\neg I((\alpha\wedge\beta))\vee\neg(\neg I(\alpha)\vee\neg I(\beta)))\wedge((\neg I(\alpha)\vee\neg I(\beta))\vee I((\alpha\wedge\beta)))\\
&= (\neg I((\alpha\wedge\beta))\vee\neg\neg(I(\alpha)\wedge I(\beta)))\wedge(\neg(I(\alpha)\wedge I(\beta))\vee I((\alpha\wedge\beta)))\\
&= (\neg I((\alpha\wedge\beta))\vee I((\alpha\wedge\beta)))\wedge(\neg I((\alpha\wedge\beta))\vee I((\alpha\wedge\beta))) = T\wedge T = T.
\end{aligned}$$

În concluzie, orice exemplu de axiomă este tautologie.

Să observăm că pentru orice formule α,β,

$$\aleph(\alpha)\cap\aleph((\alpha\rightarrow\beta))\subset\aleph(\beta).$$

Într-adevăr, dacă

$$I\in\aleph(\alpha)\cap\aleph((\alpha\rightarrow\beta))$$

atunci

$$I(((\alpha\rightarrow\beta)))=I(\alpha)=T,$$

şi

$$I(((\alpha\rightarrow\beta)))=I(((\neg\alpha)\vee\beta))=\neg I(\alpha)\vee I(\beta)=F\vee I(\beta)=I(\beta)$$

deci

$$I\in\aleph(\beta)$$

Demonstrăm prin inducţie asupra lungimii demonstraţiei formale că orice $\alpha\in T_h$ este tautologie.

Dacă admite o demonstraţie formală de lungime 1, atunci α este exemplu de axiomă, deci α este tautologie. Presupunem că orice formulă demonstrabilă ce admite o demonstraţie formală de lungime mai mică sau egală cu n este tautologie. Fie

$$\alpha_1,\ldots,\alpha_n,\alpha_{n+1}=\alpha$$

demonstraţie formală. Evident, pentru orice $k, 1\leq k\leq n$, $\alpha_1,\ldots,\alpha_k$ este demonstraţie formală, deci conform ipotezei inductive, formulele α_k , $k=1,\ldots,n$ sunt tautologii.

Dacă α_{n+1} este exemplu de axiomă atunci α este tautologie. Dacă există $1\leq i,j\leq n$ astfel încât $\alpha_j=(\alpha_i\rightarrow\alpha_{n+1})$, atunci

$$\aleph(\alpha_i)\cap\aleph((\alpha_i\rightarrow\alpha_{n+1}))\subset\aleph(\alpha_{n+1}).$$

Deoarece α_i,α_j sunt tautologii,

$$\aleph(\alpha_i)=\aleph((\alpha_i\rightarrow\alpha_{n+1}))=\Im$$

ceea ce implică $\aleph(\alpha_{n+1})=\Im$ deci α este tautologie. ■

3.2 Aplicaţii

Exemplul 3.2.1 *Să se determine rezultatul aplicării funcţiei de interpretare I asupra următoarei formule:*

a) $\alpha_1 = (\neg a \vee b) \to (\neg(a \to b) \to (\neg b \wedge a))$

Soluţie

$$
\begin{aligned}
I(\alpha_1) &= I(\neg a \vee b) \to I(\neg(a \to b) \to (\neg b \wedge a)) \\
&= \neg(\neg I(a) \vee I(b)) \vee (\neg(I(a) \to I(b)) \to (\neg I(b) \wedge I(a))) \\
&= (I(a) \wedge \neg I(b)) \vee ((\neg I(a) \vee I(b)) \vee (\neg I(b) \wedge I(a))) \\
&= (I(a) \wedge \neg I(b)) \vee ((\neg I(a) \vee I(b) \vee \neg I(b)) \wedge (\neg I(a) \vee I(b) \vee I(a))) \\
&= (I(a) \wedge \neg I(b)) \vee (T \wedge T) \\
&= I(a) \wedge \neg I(b)) \vee T = T.
\end{aligned}
$$

b) $\alpha_2 = \neg a \to (\neg b \to \neg a)$

$$I(\alpha_2) = \neg I(a) \to (\neg I(b) \to \neg I(a)) = I(a) \vee (I(b) \vee \neg I(a)) = I(a) \vee I(b) \vee \neg I(a) = T.$$

c) $\alpha_3 = (\neg a \to \neg b) \to (b \to a)$

$$
\begin{aligned}
I(\alpha_3) \quad &= (\neg I(a) \to \neg I(b)) \to (I(b) \to I(a)) = \neg(I(a) \vee \neg I(b)) \vee (\neg I(b) \vee I(a)) = \\
&= (\neg I(a) \wedge I(b)) \vee (\neg I(b) \vee I(a)) = \\
&= (\neg I(b) \vee I(a) \vee \neg I(a)) \wedge (\neg I(b) \vee I(a) \vee I(b)) = T \wedge T = T
\end{aligned}
$$

d) $\alpha_4 = (\neg a \wedge \neg b) \leftrightarrow \neg(\neg a \to b)$

$$
\begin{aligned}
I(\alpha_4) &= (\neg I(a) \wedge \neg I(b)) \leftrightarrow \neg(\neg I(a) \to I(b)) = \\
&= ((\neg I(a) \wedge \neg I(b)) \to \neg(\neg I(a) \to I(b))) \wedge (\neg(\neg I(a) \to I(b)) \to (\neg I(a) \wedge \neg I(b)))= \\
&= ((I(a) \vee I(b)) \vee \neg(I(a) \vee I(b))) \wedge ((I(a) \vee I(b)) \vee (\neg I(a) \wedge \neg I(b))) = \\
&= T \wedge ((I(a) \vee I(b)) \vee \neg(I(a) \vee I(b))) = T \wedge T = T.
\end{aligned}
$$

Exemplul 3.2.2 *Arăţţi că $(p \wedge q) \to (p \vee q)$ este o tautologie.*

Soluţie

Aplicând funcţia de interpretare expresiei considerate obţinem

$$
\begin{aligned}
I((p \wedge q) \to (p \vee q)) &= \neg(I(p) \wedge I(q)) \vee (I(p) \vee I(q)) = (\neg I(p) \vee \neg I(q)) \vee (I(p) \vee I(q)) = \\
&= (\neg I(p) \vee I(p)) \vee (\neg I(q) \vee I(q)) = T \vee T = T
\end{aligned}
$$

Exemplul 3.2.3 *Arătaţi că $\neg(p \vee (\neg p \wedge q)) \equiv \neg p \wedge \neg q$.*

Soluţie

Relaţia dată se poate rescrie sub forma următoare:

$$
\begin{aligned}
\neg(p \vee (\neg p \wedge q)) \quad &\equiv \neg p \wedge \neg(\neg p \wedge q) \equiv \neg p \wedge (p \vee \neg q) \equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q) \\
&\equiv F \vee (\neg p \wedge \neg q) \equiv (\neg p \wedge \neg q).
\end{aligned}
$$

3.3 Exerciţii

Exerciţiul 3.3.1 *Să se aplice funcţia de interpretare următoarelor formule logice:*

a) $\neg(\alpha \vee \beta) \to (\neg\gamma \to \neg(\alpha \vee \beta))$
b) $(\neg\theta \to \neg\alpha) \to (\alpha \to \theta)$
c) $((\neg\theta \vee \gamma) \wedge ((\theta \to \gamma) \to (\omega \vee \beta))) \to (\neg\omega \to \beta)$
d) $\neg((\neg\gamma \vee \theta) \to \beta) \to (\neg(\gamma \wedge \neg\theta) \wedge \neg\beta)$

e) $(\neg\alpha \to (\beta \vee \neg\delta)) \leftrightarrow (\alpha \vee (\neg(\delta \to b)))$
f) $(\neg\beta \to (\beta \vee \gamma)) \to (\beta \vee (\neg\gamma \to \beta))$.

Exerciţiul 3.3.2 *Să se arate fără să se folosească tabele de adevăr că următoarele formule sunt tautologii:*
a) $\neg p \to (p \to q)$
b) $(p \wedge q) \to p$
c) $\neg p \to (p \vee q) \to q$
d) $(p \wedge (p \to q)) \to q$
e) $(\neg p \wedge (p \wedge q)) \to \neg q$.

Exerciţiul 3.3.3 *Verificaţi fără tabele de adevăr dacă următoarele formule sunt echivalente:*
a) $(p \wedge q) \to r$ *şi* $(p \to r) \wedge (q \to r)$
b) $\neg(p \leftrightarrow q)$ *şi* $p \leftrightarrow \neg q$
c) $\neg p \to (q \to r)$ *şi* $q \to (p \vee r)$
d) $(p \to q) \to r$ *şi* $p \to (q \to r)$.

Capitolul 4

Verificarea automată

Fie H, Γ mulţimi de formule. Procedura $\wp$ este o procedură de demonstrare automată dacă decide $H \vdash \Gamma$ cu alternativa $H \nvdash \Gamma$. Convenim să numim soluţie calculată de procedura de demonstrare automată $\wp$ secvenţa de etape intermediare ale evoluţiei determinate de procedură. În aplicaţii de interes practic, mulţimile H, Γ sunt finite, caz în care, problema verificării dacă $H \vdash \Gamma$ se reduce la verificarea validabilităţii/invalidabilităţii unei singure formule. Dacă $H = \{\alpha_1, \ldots, \alpha_n\}, \Gamma = \{\beta_1, \ldots, \beta_m\}$, atunci $H \vdash \Gamma$ dacă şi numai dacă

$$\left(\bigwedge_{i=1}^{n} \alpha_i \rightarrow \bigvee_{j=1}^{m} \beta_j \right)$$

este tautologie. De asemenea, o caracterizare echivalentă este $H \vdash \Gamma$ dacă şi numai dacă

$$\left(\bigwedge_{i=1}^{n} \alpha_i \right) \wedge \left(\bigwedge_{j=1}^{m} (\neg \beta_j) \right)$$

este invalidabilă. În consecinţă, procedurile de demonstrare automată pot fi gândite ca proceduri pentru verificarea validabilităţii/invalidabilităţii unei singure formule. În esenţă, o procedură de demonstrare automată este o metodă de căutare sistematică a unui model pentru o formulă dată. O tehnică de demonstrare automată $\wp$ care verifică $\{\alpha_1, \ldots, \alpha_n\} \vdash \{\beta_1, \ldots, \beta_m\}$ procedând la verificarea invalidabilităţii formulei

$$\left(\bigwedge_{i=1}^{n} \alpha_i \right) \wedge \left(\bigwedge_{j=1}^{m} (\neg \beta_j) \right)$$

se numeşte *metodă de respingere*.

O procedură de verificare a validabilităţii/invalidabilităţii unei formule date α este imediată şi anume, revine la generarea a 2^n funcţii de adevăr, unde n este numărul propoziţiilor elementare care apar în structura simbolică α. Dacă valoarea calculată pentru α este T pentru toate interpretările induse de funcţiile de adevăr astfel generate, atunci α este tautologie. Dacă cel puţin o astfel de interpretare calculează T atunci α este validabilă, respectiv dacă în toate cele 2^n interpretări valoarea calculată este F atunci α este invalidabilă.

Exemplul 4.0.1 Fie $a, b \in V$, $\alpha = (a \rightarrow (b \rightarrow (a \wedge b)))$. Deoarece $n = 2$ vor fi generate 2^2 funcţii de adevăr.

$\hbar(a)$	$\hbar(b)$	$I(\hbar)((a \wedge b))$	$I(\hbar)((b \to (a \wedge b)))$	$I(\hbar)((a \to (b \to (a \wedge b))))$
T	T	T	T	T
T	F	F	T	T
F	T	F	F	T
F	F	F	T	T

Rezultă $\alpha = (a \to (b \to (a \wedge b)))$ este tautologie, deci $\alpha \in T_h$.

4.1 Normalizarea CNF a formulelor logice

Definiţia 4.1.1 *Un literal este o propoziţie elementară sau negaţia unei propoziţii elementare. Mulţimea literalilor este $V \cup \neg V$. Dacă $p \in V$ atunci literalii p,$(\neg p)$se numesc literali complementari. Literalii din V se numesc literali pozitivi respectiv literalii din $\neg V$ se numesc* literali negativi.

Definiţia 4.1.2 *Se numeşte* clauză *o structură simbolică $k = \bigvee_{i=1}^{n} L_i$ de tip disjunctiv pentru o mulţime de literali $\{L_1, \ldots, L_n\}$. Clauza corespunzătoare mulţimii vide de literali se numeşte clauza vidă şi este notata $k = \Box$.*

Observaţia 4.1.1 Pentru $k \neq \Box$, mulţimea modelelor clauzei $k = \bigvee_{i=1}^{n} L_i; n \geq 1$ este evident $\aleph(k) = \bigcup_{i=1}^{n} \aleph(L_i)$. Conform convenţiilor Bourbaki, calculul mulţimii $\aleph(\Box)$ revine la reuniune după mulţimea vidă de indici, deci $\aleph(\Box) = \emptyset$. Cu alte cuvinte, clauza vidă este invalidabilă.

Definiţia 4.1.3 *Se numeşte* formulă normalizată CNF *(**C**onjunctive **N**ormal **F**orm) o structură simbolică $\alpha = \bigwedge_{i=1}^{n} k_i$ de tip conjunctiv pentru o mulţime de clauze $\{k_1, \ldots, k_n\}$. Mulţimea clauzelor $S(\alpha) = \{k_1, \ldots, k_n\}$ se numeşte* reprezentare clauzală *pentru formula α. Formula CNF corespunzătoare mulţimii vide de clauze se numeşte* formulă vidă *şi este notată $\alpha = \emptyset$.*

Observaţia 4.1.2 Dacă $\alpha = \bigwedge_{i=1}^{n} k_i; n \geq 1$, atunci $\aleph(\alpha) = \aleph(S(\alpha)) = \bigcap_{i=1}^{n} \aleph(k_i)$. Conform convenţiilor Bourbaki, calculul mulţimii $\aleph(\emptyset)$ revine la intersecţie după mulţime vidă de indici, deci $\aleph(\emptyset) = \Im$. Cu alte cuvinte, formula vidă este tautologie.

Rezultatul stabilit de următoarea teoremă exprimă faptul că din punct de vedere semantic, se poate presupune întotdeauna că se dispune de reprezentări normalizate CNF pentru formulele limbajului calculului cu propoziţii.

Definiţia 4.1.4 *Fie $\alpha, \beta \in FORM$. Spunem că β este o subformulă a formulei α dacă β este o componentă a unui SGF de lungime minimă pentru α.*

Exemplul 4.1.1 Fie $a, b \in V$, $\alpha = (a \to (b \to (a \wedge b)))$.

Evident, secvenţa

$$a, b, (a \wedge b), (b \to (a \wedge b)), (a \to (b \to (a \wedge b))) = \alpha$$

este un SGF pentru α.

Rezultă că mulţimea subformulelor lui α este

$$\{a, b, (a \wedge b), (b \rightarrow (a \wedge b)), (a \rightarrow (b \rightarrow (a \wedge b)))\}.$$

Observaţia 4.1.3 Orice subformulă a unei formule α este o formulă. Formula β este subformulă a formulei α dacă şi numai dacă β este o componentă a oricărui SGF pentru α.

Teorema 4.1.1 *Pentru orice $\alpha \in FORM$ există α' formulă normalizată CNF şi $\alpha \equiv \alpha'$.*

Demonstraţie. Aplicăm succesiv subformulelor formulei α următoarele transformări:

$T1$. Eliminarea conectivei "$\leftrightarrow$": Fiecare subformulă de tipul $(\beta \leftrightarrow \gamma)$ se substituie cu

$$((\beta \rightarrow \gamma) \wedge (\gamma \rightarrow \beta)),\ (\beta \leftrightarrow \gamma) \equiv ((\beta \rightarrow \gamma) \wedge (\gamma \rightarrow \beta)).$$

$T2$. Eliminarea conectivei "$\rightarrow$": Fiecare subformulă de tipul $(\beta \rightarrow \gamma)$ se substituie cu

$$((\neg\beta) \vee \gamma),\ (\beta \rightarrow \gamma) \equiv ((\neg\beta) \vee \gamma).$$

$T3$. Se aduc negaţiile în faţa literalilor:

i) Fiecare subformulă de tipul $(\neg(\beta \vee \gamma))$ se substituie cu

$$((\neg\beta) \wedge (\neg\gamma)),\ (\neg(\beta \vee \gamma)) \equiv ((\neg\beta) \wedge (\neg\gamma)).$$

ii) Fiecare subformulă de tipul $(\neg(\beta \wedge \gamma))$ se substituie cu

$$((\neg\beta) \vee (\neg\gamma)),\ (\neg(\beta \wedge \gamma)) \equiv ((\neg\beta) \vee (\neg\gamma)).$$

$T4$. Eliminarea negaţiilor multiple: Fiecare subformulă de tipul $(\neg(\neg\beta))$ se substituie cu

$$\beta,\ (\neg(\neg\beta)) \equiv \beta$$

$T5$. Obţinerea structurii CNF: Fiecare subformulă de tipul $(\beta \vee (\gamma \wedge \delta))$ se substituie prin

$$((\beta \vee \gamma) \wedge (\beta \vee \delta)),\ (\beta \vee (\gamma \wedge \delta)) \equiv ((\beta \vee \gamma) \wedge (\beta \vee \delta)).$$

respectiv $((\gamma \wedge \delta) \vee \beta)$ se substituie prin

$$((\gamma \vee \beta) \wedge (\delta \vee \beta)),\ ((\gamma \wedge \delta) \vee \beta) \equiv ((\gamma \vee \beta) \wedge (\delta \vee \beta)).$$

Deoarece aplicarea fiecărei transformări determină substituirea unei subformule cu o subformulă semantic echivalentă cu ea, la fiecare etapă se obţine o formulă semantic echivalentă cu formula iniţială. Rezultă în final o reprezentare normalizată CNF semantic echivalentă cu formula iniţială. ■

Exemplul 4.1.2 Fie $a, b \in V$, $\alpha = (((\neg b) \rightarrow (\neg a)) \longleftrightarrow (a \rightarrow b))$

Mulţimea subformulelor formulei α este

$$F_0 = \{a, b, (\neg a), (\neg b), (a \rightarrow b), (((\neg b) \rightarrow (\neg a)) \longleftrightarrow (a \rightarrow b))\}$$

Aplicarea transformării $T1$ determină substituirea subformulei

$$(((\neg b) \rightarrow (\neg a)) \longleftrightarrow (a \rightarrow b))$$

prin
$$((((\neg b) \to (\neg a)) \to (a \to b)) \wedge ((a \to b) \to ((\neg b) \to (\neg a))))$$

Rezultă

$$\alpha_1 = ((((\neg b) \to (\neg a)) \to (a \to b)) \wedge ((a \to b) \to ((\neg b) \to (\neg a))))$$

Aplicarea transformării $T2$ determină:
substituirea subformulei

$$(((\neg b) \to (\neg a)) \to (a \to b)) \text{ prin } ((\neg ((\neg b) \to (\neg a))) \vee (a \to b)).$$

substituirea subformulei

$$((a \to b) \to ((\neg b) \to (\neg a))) \text{ prin } ((\neg (a \to b)) \vee ((\neg b) \to (\neg a))).$$

substituirea subformulei
$$(a \to b) \text{ prin } ((\neg a) \vee b).$$

substituirea subformulei

$$((\neg b) \to (\neg a)) \text{ prin } ((\neg (\neg b)) \vee (\neg a)).$$

Rezultă $\alpha_2 = (((\neg ((\neg (\neg b)) \vee (\neg a))) \vee ((\neg a) \vee b)) \wedge ((\neg ((\neg a) \vee b)) \vee ((\neg (\neg b)) \vee (\neg a))))$

Aplicarea transformării $T3$ determină:
substituirea subformulei

$$(\neg ((\neg (\neg b)) \vee (\neg a))) \text{ prin } ((\neg (\neg (\neg b))) \wedge (\neg (\neg a))).$$

substituirea subformulei

$$(\neg ((\neg a) \vee b)) \text{ prin } ((\neg (\neg a)) \wedge (\neg b)).$$

Rezultă

$$\alpha_3 = ((((\neg (\neg (\neg b))) \wedge (\neg (\neg a))) \vee ((\neg a) \vee b)) \wedge (((\neg (\neg a)) \wedge (\neg b)) \vee ((\neg (\neg b)) \vee (\neg a)))).$$

Aplicarea transformării $T4$ determină:
substituirea subformulei
$$(\neg (\neg b)) \text{ prin } b.$$

substituirea subformulei
$$(\neg (\neg a)) \text{ prin } a.$$

Rezultă
$$\alpha_4 = ((((\neg b) \wedge a) \vee ((\neg a) \vee b)) \wedge ((a \wedge (\neg b)) \vee (b \vee (\neg a)))).$$

Aplicarea transformării $T5$ determină:
substituirea subformulei

$$(((\neg b) \wedge a) \vee ((\neg a) \vee b)) \text{ prin } (((\neg b) \vee ((\neg a) \vee b)) \wedge (a \vee ((\neg a) \vee b))).$$

substituirea subformulei

$$((a \wedge (\neg b)) \vee (b \vee (\neg a))) \text{ prin } ((a \vee (b \vee (\neg a))) \wedge ((\neg b) \vee (b \vee (\neg a)))).$$

Rezultă reprezentarea normalizată CNF pentru formula α,

$$\alpha' = ((((\neg b) \vee ((\neg a) \vee b)) \wedge (a \vee ((\neg a) \vee b))) \wedge ((a \vee (b \vee (\neg a))) \wedge ((\neg b) \vee (b \vee (\neg a)))))$$

şi $\alpha \equiv \alpha'$.

Reprezentarea clauzală este

$$S(\alpha) = \{k_1 = ((\neg b) \vee ((\neg a) \vee b)), k_2 = (a \vee ((\neg a) \vee b)), k_3 = (a \vee (b \vee (\neg a)))\},$$

$$k_4 = ((\neg b) \vee (b \vee (\neg a)))$$

4.2 Metoda Davis-Putnam

Demonstratorul de teoreme Davis-Putnam este un sistem de verificare a validabilităţii formulelor bazat pe tehnica de respingere. Prelucrarea iniţiată de metoda Davis-Putnam este efectuată asupra reprezentării clauzale corespunzătoare formulei testate.

Dacă α este o structură simbolică şi λ un simbol, convenim să notăm $\alpha\langle\lambda\rangle$ dacă λ apare printre simbolurile din α , respectiv $\alpha\rangle\lambda\langle$ cazul contrar.

Definiţia 4.2.1 *Fie k clauză care nu este tautologie şi $p \in V$. Clauza k este p-pozitivă dacă $k\langle p\rangle$, p-negativă dacă $k\langle(\neg p)\rangle$ respectiv este p-neutră dacă $k\rangle(\neg p)\langle$ şi $k\rangle p\langle$.*

Pentru k clauză λ-pozitivă, notăm $k \setminus \lambda$ clauza rezultată prin eliminarea literalului λ din α.

Observaţia 4.2.1 Fie $\lambda = (\neg p)$, $p \in V$. Convenim să acceptăm $(\neg\lambda)$ ca literal şi anume $(\neg\lambda)$ reprezintă literalul propoziţie elementară p. Justificarea acestei convenţii rezidă din proprietatea $(\neg(\neg p)) \equiv p$. Pentru k clauză şi λ literal, notăm $k \setminus \lambda$ clauza rezultată prin eliminarea literalului λ din α.

Definiţia 4.2.2 *Fie $S(\alpha)$ o reprezentare clauzală liberă de tautologii şi λ literal. Fie submulţimile de clauze,*

$$\begin{aligned}
\alpha_\lambda^+ &= \{k \mid k \in S(\alpha), k\langle\lambda\rangle\},\\
\alpha_\lambda^- &= \{k \mid k \in S(\alpha), k\langle(\neg\lambda)\rangle\},\\
\alpha_\lambda^0 &= \{k \mid k \in S(\alpha), k\rangle\lambda\langle,\ k\rangle(\neg\lambda)\langle\},\\
POS_\lambda(\alpha) &= \alpha_\lambda^0 \cup \left\{k \setminus \lambda \mid k \in \alpha_\lambda^+\right\},\\
NEG_\lambda(\alpha) &= \alpha_\lambda^0 \cup \left\{k \setminus (\neg\lambda) \mid k \in \alpha_\lambda^-\right\}.
\end{aligned}$$

Observaţia 4.2.2 Evident, $\alpha_\lambda^+ = \alpha_{(\neg\lambda)}^-, \alpha_\lambda^- = \alpha_{(\neg\lambda)}^+, \alpha_\lambda^0 = \alpha_{(\neg\lambda)}^0$. Deoarece $S(\alpha)$ este reprezentare clauzală liberă de tautologii, pentru orice literal λ , $\alpha_\lambda^+, \alpha_\lambda^-, \alpha_\lambda^0$ este o partiţie a mulţimii $S(\alpha)$.

De asemenea, $\alpha_\lambda^+ = \alpha_{(\neg\lambda)}^-, \alpha_\lambda^- = \alpha_{(\neg\lambda)}^+, \alpha_\lambda^0 = \alpha_{(\neg\lambda)}^0$.

Exemplul 4.2.1 Fie $S(\alpha) = \{k_1, k_2, k_3, k_4, k_5, k_6\}$ unde

$$\begin{aligned} k_1 &= (\neg p) \vee o, \\ k_2 &= (\neg p) \vee (\neg c), \\ k_3 &= (\neg m) \vee c \vee i, \\ k_4 &= m, \\ k_5 &= p, \\ k_6 &= (\neg i). \end{aligned}$$

Pentru $\lambda = (\neg p)$,

$$\begin{aligned} \alpha_\lambda^+ &= \{(\neg p) \vee o, (\neg p) \vee (\neg c)\}, \\ \alpha_\lambda^- &= \{p\}, \\ \alpha_\lambda^0 &= \{(\neg m) \vee c \vee i, m, (\neg i)\}, \\ POS_\lambda(\alpha) &= \{(\neg m) \vee c \vee i, m, (\neg i)\} \cup \{o, (\neg c)\}, \\ NEG_\lambda(\alpha) &= \{(\neg m) \vee c \vee i, m, (\neg i)\} \cup \{\Box\}. \end{aligned}$$

Observaţia 4.2.3 *Daca $S(\alpha)$ este o reprezentare clauzală liberă de tautologii atunci pentru orice literal λ , $POS_\lambda(\alpha), NEG_\lambda(\alpha)$ sunt de asemenea reprezentări clauzale libere de tautologii. Literalul λ nu apare în clauzele reprezentărilor clauzale $POS_\lambda(\alpha), NEG_\lambda(\alpha)$.*

Definiţia 4.2.3 *Fie $S(\alpha)$ o reprezentare clauzală liberă de tautologii şi λ literal. Spunem că λ este literal pur dacă $\alpha_\lambda^- = \emptyset$. Clauza $k = \bigvee_{i=1}^{n} L_i$ este clauză unitară dacă $n = 1$.*

Teorema 4.2.1 (Teorema de separare Davis-Putnam) . *Fie $S(\alpha)$ o reprezentare clauzală liberă de tautologii. Dacă $S(\alpha)$ este validabilă, atunci pentru orice literal λ, cel puţin una dintre mulţimile de clauze $POS_\lambda(\alpha)$, $NEG_\lambda(\alpha)$ este validabilă. Dacă există un literal λ astfel încât cel puţin una dintre mulţimile de clauze $POS_\lambda(\alpha)$, $NEG_\lambda(\alpha)$ este validabilă atunci $S(\alpha)$ este validabilă.*

Demonstraţie. Presupunem că $S(\alpha)$ este validabilă şi fie $I \in \aleph(S(\alpha))$ arbitrară. Demonstrăm că pentru orice λ literal, dacă $I(\lambda) = T$ atunci $I(NEG_\lambda(\alpha)) = T$, respectiv dacă $I(\lambda) = F$ atunci $I(POS_\lambda(\alpha)) = T$.

Presupunem $I(\lambda) = T$. Fie $k \in NEG_\lambda(\alpha)$ arbitrară.

$a)$ dacă $k \in \alpha_\lambda^0$, cum $\alpha_\lambda^0 \subset S(\alpha)$ şi $I(S(\alpha)) = T$, rezultă $I(k) = T$.

$b)$ dacă $k \in \left\{k^* \setminus (\neg\lambda) \mid k^* \in \alpha_\lambda^-\right\}$, fie $k^* \in \alpha_\lambda^-$ astfel încât $k = k^* \setminus (\neg\lambda)$, deci $k^* = k \vee (\neg\lambda)$. Deoarece $k^* \in \alpha_\lambda^- \subset S(\alpha)$ rezultă $I(k^*) = T$, deci

$$I(k^*) = I(k) \vee I((\neg\lambda)) = I(k).$$

Obţinem astfel,

$$I \in \bigcap_{k \in NEG_\lambda(\alpha)} \aleph(k) = \aleph(NEG_\lambda(\alpha)).$$

Un argument similar conduce la concluzia $I \in \aleph(POS_\lambda(\alpha))$ în cazul în care $I(\lambda) = F$.

Presupunem că există un literal λ astfel încât cel puţin una dintre mulţimile de clauze $POS_\lambda(\alpha)$, $NEG_\lambda(\alpha)$ este validabilă.

Presupunem că $POS_\lambda(\alpha)$ este validabilă şi fie $I \in \aleph(POS_\lambda(\alpha))$.

Definim funcţia de adevăr $\hbar : V \to \{T, F\}$ astfel,

$a)$ dacă $\lambda \in V$ atunci $\forall p \in V, \hbar(p) = \begin{cases} I(p), & \text{dacă } p \neq \lambda, \\ F, & \text{dacă } p = \lambda. \end{cases}$

$b)$ dacă $\lambda \in \neg V$ atunci $\forall p \in V, \hbar(p) = \begin{cases} I(p), & \text{dacă } (\neg p) \neq \lambda, \\ T, & \text{dacă } (\neg p) = \lambda. \end{cases}$

Evident, rezultă $I(\hbar)(\lambda) = F$ şi pentru orice k clauză astfel încât $k \rangle \lambda \langle$ şi $k \rangle (\neg\lambda) \langle$, $I(\hbar)(k) = I(k)$.

Fie $k \in S(\alpha)$ arbitrară.

$a)$ dacă $k \in \alpha_\lambda^0$, atunci $I(\hbar)(k) = I(k)$ şi cum $\alpha_\lambda^0 \subset POS_\lambda(\alpha)$ rezultă $I(k) = T$, deci $I(\hbar)(k) = T$.

$b)$ dacă $k \in \alpha_\lambda^-$ atunci $k \equiv k_1 \vee (\neg\lambda)$ deci $I(\hbar)(k) = I(\hbar)(k_1) \vee I(\hbar)((\neg\lambda)) = I(\hbar)(k_1) \vee T = T$.

$c)$ dacă $k \in \alpha_\lambda^+$ atunci $k \setminus \lambda \in POS_\lambda(\alpha)$ deci $I(k \setminus \lambda) = T$.

Deoarece $S(\alpha)$ este reprezentare clauzală liberă de tautologii

$$(k \setminus \lambda) \rangle \lambda \langle \text{ şi } (k \setminus \lambda) \rangle (\neg\lambda) \langle ,$$

deci

$$I(\hbar)(k \setminus \lambda) = I(k \setminus \lambda).$$

Rezultă $I(\hbar)(k \setminus \lambda) = T$. Evident, $k \equiv (k \setminus \lambda) \vee \lambda$, deci

$$I(\hbar)(k) = I(\hbar)(k \setminus \lambda) \vee I(\hbar)(\lambda) = T \vee I(\hbar)(\lambda) = T.$$

Obţinem în final,

$$I(\hbar) \in \bigcap_{k \in S(\alpha)} \aleph(k) = \aleph(S(\alpha))$$

deci $S(\alpha)$ este validabilă. ■

Un argument similar conduce la aceeaşi concluzie în ipoteza că $NEG_\lambda(\alpha)$ este validabilă.

Corolarul 4.2.1 *Dacă $S(\alpha)$ este o reprezentare clauzală liberă de tautologii atunci*

$$\aleph(S(\alpha)) \subset \bigcap_{\lambda\ \in V \cup \neg V} (\aleph(NEG_\lambda(\alpha)) \cup \aleph(POS_\lambda(\alpha)))$$

Demonstraţie. Justificarea relaţiei este imediată şi rezultă din demonstraţia Teoremei 4.2.1. ■

Corolarul 4.2.2 (Regula clauzei unitare) *Fie $S(\alpha)$ reprezentare clauzală liberă de tautologii. Dacă λ este clauză unitară, atunci $S(\alpha)$ este validabilă dacă şi numai dacă $NEG_\lambda(\alpha)$ este validabilă.*

Demonstraţie. Deoarece λ este clauză unitară, $\lambda \in \alpha_\lambda^+$.Rezultă $\lambda \setminus \lambda = \Box \in POS_\lambda(\alpha)$ deci cum clauza vidă este invalidabilă, rezultă $POS_\lambda(\alpha)$ invalidabilă. Dacă $S(\alpha)$ este validabilă, utilizând rezultatul stabilit de Teorema 4.2.1 rezultă $NEG_\lambda(\alpha)$ este validabilă. Reciproc, dacă $NEG_\lambda(\alpha)$ este validabilă, atunci din Teorema 4.2.1 obţinem că $S(\alpha)$ este validabilă. ■

Corolarul 4.2.3 (Regula literalilor puri) *Fie $S(\alpha)$ reprezentare clauzală liberă de tautologii. Dacă λ este literal pur, atunci $S(\alpha)$ este validabilă dacă şi numai dacă $NEG_\lambda(\alpha)$ este validabilă.*

Demonstraţie. Deoarece λ este literal pur, $\alpha_\lambda^- = \emptyset$ deci

$$NEG_\lambda(\alpha) = \alpha_\lambda^0 \subset POS_\lambda(\alpha).$$

Obţinem,

$$\aleph(POS_\lambda(\alpha)) = \bigcap_{k \in POS_\lambda(\alpha)} \aleph(k) \subset \bigcap_{k \in NEG_\lambda(\alpha)} \aleph(k) = \aleph(NEG_\lambda(\alpha)).$$

Dacă $S(\alpha)$ este validabilă şi $I \in \Im$ este astfel încât $I(S(\alpha)) = T$, atunci din Corolarul 4.2.1 obţinem $I(POS_\lambda(\alpha)) = T$ sau $I(NEG_\lambda(\alpha)) = T$. Deoarece $\aleph(POS_\lambda(\alpha)) \subset \aleph(NEG_\lambda(\alpha))$ rezultă $I(NEG_\lambda(\alpha)) = T$ deci $NEG_\lambda(\alpha)$ este validabilă.

Dacă $NEG_\lambda(\alpha)$ este validabilă atunci din Teorema 4.2.1 obţinem direct concluzia $S(\alpha)$ validabilă. ■

Rezultatul stabilit de teorema de separare Davis-Putnam constituie suportul pentru dezvoltarea unei tehnici de căutare sistematică a unui model pentru o reprezentare clauzală dată.

Fie $S(\alpha)$ reprezentare clauzală liberă de tautologii şi $\lambda_1, \ldots, \lambda_n$ literalii care apar în clauzele din $S(\alpha)$. Fie reprezentările clauzale

$$\{\gamma_{i_1 \ldots i_k} \mid i_1, \ldots, i_k \in \{0,1\},\ 1 \le k \le n,\ \}$$

rezultate astfel,

$$\begin{aligned} \gamma_0 &= NEG_{\lambda_1}(\alpha), \\ \gamma_1 &= POS_{\lambda_1}(\alpha), \\ \gamma_{i_1 \ldots i_{k-1} 0} &= NEG_{\lambda_k}\left(\gamma_{i_1 \ldots i_{k-1}}\right), \\ \gamma_{i_1 \ldots i_{k-1} 1} &= POS_{\lambda_k}\left(\gamma_{i_1 \ldots i_{k-1}}\right), \end{aligned}$$

$k = 2, \ldots, n$.

Deoarece în clauzele reprezentării clauzale $\gamma_{i_1 \ldots i_k}$ nu mai apar literalii $\lambda_1, \ldots, \lambda_k$ obţinem că toate reprezentările clauzale $\gamma_{i_1 \ldots i_n}, i_1, \ldots, i_n \in \{0,1\}$ conţin numai clauze fără literali. Rezultă că pentru orice $i_1, \ldots, i_n \in \{0,1\}, \gamma_{i_1 \ldots i_n} \in \{\{\Box\}, \emptyset\}$.

Dacă $\gamma_{i_1 \ldots i_n} = \{\Box\}$ pentru toţi $i_1, \ldots, i_n \in \{0,1\}$, atunci pe baza concluziei stabilite de Teorema 4.2.1, toate reprezentările clauzale $\gamma_{i_1 \ldots i_{n-1}}$ rezultă invalidabile. Iterând acelaşi argument obţinem că toate reprezentările clauzale $\gamma_{i_1 \ldots i_k}$, $i_1, \ldots, i_k \in \{0,1\}$, $1 \le k \le n$ sunt invalidabile deci în final $S(\alpha)$ rezultă invalidabilă. Dacă există $i_1, \ldots, i_n \in \{0,1\}$ astfel încât $\gamma_{i_1 \ldots i_n} = \emptyset$, atunci din Teorema 4.2.1, $\gamma_{i_1 \ldots i_{n-1}}$ rezultă validabilă. Iterând acelaşi argument, obţinem că toate reprezentările clauzale $\gamma_{i_1 \ldots i_k}$, $1 \le k \le n$ sunt validabile deci în final $S(\alpha)$ rezultă validabilă.

Metoda descrisă corespunde generării unui arbore binar complet cu vârfurile etichetate cu reprezentările clauzale $\gamma_{i_1 \ldots i_k}$, $i_1, \ldots, i_k \in \{0,1\}$, $1 \le k \le n$ şi radăcina etichetată cu $S(\alpha)$. Orice vârf intermediar corespunde unei acţiuni de "separare" a reprezentării clauzale asociate ca etichetă a vârfului în "partea pozitivă (POS)" şi "partea negativă (NEG)" în raport cu un anume literal. Utilizând rezultatele stabilite de Corolarul 4.2.2 şi Corolarul 4.2.3 obţinem că, în cazul în care reprezentarea clauzală corespunzătoare unui vârf, fie conţine o clauză unitară fie există un literal pur, nu mai este necesară generarea ambilor fii ai vârfului, pentru vârful respectiv fiind suficientă generarea unui singur fiu. Într-adevăr, dacă reprezentarea clauzală γ corespunzătoare vârfului conţine o clauză unitară λ, atunci $POS_{\lambda}(\gamma)$ este invalidabilă, deci toate vârfurile din subarborele de rădăcina etichetată prin reprezentarea clauzală $POS_{\lambda}(\gamma)$ vor avea ca etichete reprezentări clauzale invalidabile. Dacă în reprezentarea clauzală γ corespunzătoare vârfului există un literal pur λ atunci întreaga informaţie relativ la validabilitatea reprezentării γ este conţinută de subarborele de rădăcina etichetată $NEG_{\lambda}(\gamma)$.

De asemenea, dacă reprezentarea clauzală, etichetă a unui vârf este formula vidă, atunci $S(\alpha)$ rezultă validabilă deci nu mai este necesară extinderea în continuare a arborelui. Dacă reprezentarea clauzală etichetă a unui vârf conţine clauza vidă, atunci toate reprezentările clauzale etichete ale vârfurilor din subarborele de rădăcină acel vârf vor fi invalidabile deci devine inutilă extinderea arborelui în direcţia acelui vârf.

Descrierea metodei de demonstrare automată rezultată pe baza acestor consideraţii este prezentată în continuare. Simularea generării arborelui etichetat cu reprezentări clauzale este realizată prin menţinerea unei stive T în care sunt reţinute vârfurile arborelui generate dar încă neprelucrate.

```
procedure DvP;
Intrare: Structură de date pentru stocarea reprezentării clauzale S(α)
EliminaTautologii(S(α));
γ ← S(α); T ← ∅; sw ← false;
repeat
if γ = ∅ then
        write ('Validabilă');
        sw ← true
    else
      if □ ∈ γ then
              if T = ∅ then
                      write ('Invalidabilă');
                      sw← true
                  else
                    γ ← TOP(T);
                    POP(T);
              endif
          else
            if (există λ clauză unitară) or (există λ literal pur ) then
                                                          γ ← NEG_λ(γ)
                                                       else
                                                          alege ( λ literal);
                                                          γ ← NEG_λ(γ);
                                                          PUSH(T, POS_λ(γ))
            endif
      endif
endif
until sw
end.
```

Observaţia 4.2.4 Apelul EliminaTautologii($S(\alpha)$) realizează detectarea şi eliminarea clauzelor tautologii din reprezentarea clauzală $S(\alpha)$. Apelul este necesar doar la momentul iniţial, deoarece dacă $S(\alpha)$ este o reprezentare clauzală liberă de tautologii atunci toate reprezentările clauzale generate pe durata căutării iniţiate de procedura DvP sunt reprezentări clauzale libere de tautologii. Apelul alege(λ literal) efectuează selectarea unui literal pentru aplicarea operaţiei de separare reprezentării clauzale curente.

Exemplul 4.2.2 Evoluţia determinată de procedura DvP pentru datele de intrare

$$S(\alpha) = \{k_1, k_2, k_3, k_4, k_5, k_6\}$$

unde

$$\begin{aligned} k_1 &= (\neg p) \vee o, \\ k_2 &= (\neg p) \vee (\neg c), \\ k_3 &= (\neg m) \vee c \vee i, \\ k_4 &= m, \\ k_5 &= p, \\ k_6 &= (\neg i). \end{aligned}$$

este :

Iniţializări: $\gamma \leftarrow \{(\neg p) \vee o, (\neg p) \vee (\neg c), (\neg m) \vee c \vee i, m, p, (\neg i)\}$;
$sw \leftarrow false; T \leftarrow \emptyset$

Iteraţia 1: $\lambda = m$ clauză unitară
$\gamma \leftarrow NEG_m(\gamma) = \{(\neg p) \vee o, (\neg p) \vee (\neg c), c \vee i, p, (\neg i)\}$

Iteraţia 2: $\lambda = p$ clauză unitară
$\gamma \leftarrow NEG_p(\gamma) = \{o, (\neg c), c \vee i, (\neg i)\}$

Iteraţia 3: $\lambda = o$ clauză unitară (literalul o este şi literal pur)
$\gamma \leftarrow NEG_o(\gamma) = \{(\neg c), c \vee i, (\neg i)\}$

Iteraţia 4: $\lambda = (\neg c)$ clauză unitară
$\gamma \leftarrow NEG_{(\neg c)}(\gamma) = \{i, (\neg i)\}$

Iteraţia 5: $\lambda = i$ clauză unitară
$\gamma \leftarrow NEG_i(\gamma) = \{\square\}$

Iteraţia 6: $\square \in \gamma$ şi $T = \emptyset \Rightarrow$write $('invalidalibil\breve{a}')$, sw$\leftarrow$ $true$
$\Rightarrow$ STOP.

Observaţia 4.2.5 Deoarece algoritmul decide că reprezentarea clauzală $S(\alpha)$ este invalidabilă, rezultă $H \vdash \beta$ unde $\beta = ((m \wedge p) \rightarrow i)$

$$H = \{\alpha_1 = (p \rightarrow (o \wedge (\neg c))), \alpha_2 = ((m \wedge (\neg c)) \rightarrow i)\}$$

(Exemplul **??**)

Exemplul 4.2.3 Evoluţia determinată de procedura DvP pentru datele de intrare

$$S(\alpha) = \{k_1, k_2, k_3, k_4\}$$

unde

$$\begin{aligned} k_1 &= a \vee (\neg b) \vee c, \\ k_2 &= (\neg a) \vee (\neg c), \\ k_3 &= (\neg c) \vee b, \\ k_4 &= (\neg b) \vee a. \end{aligned}$$

este:

Iniţializări: $\gamma \leftarrow \{a \vee (\neg b) \vee c, (\neg a) \vee (\neg c), (\neg c) \vee b, (\neg b) \vee a\}$;
$sw \leftarrow false; T \leftarrow \emptyset$,

Iteraţia 1: Nu există clauză unitară şi nici literal pur; apelul alege (λ literal) selectează $\lambda = a$
$\gamma \leftarrow NEG_a(\gamma) = \{(\neg c), (\neg c) \vee b\}$
$T \leftarrow POS_a(\gamma) = \{(\neg b) \vee c, (\neg c) \vee b, (\neg b)\}$,

Iteraţia 2: $\lambda = (\neg c)$ clauză unitară (literalul $(\neg c)$ este şi literal pur)
$\gamma \leftarrow NEG_{(\neg c)}(\gamma) = \emptyset$,

Iteraţia 3: $\gamma = \emptyset \Rightarrow$ decizia terminală 'validabilă'.

4.3 Metoda bazată pe principiul rezoluţiei

Rezoluţia a fost introdusă ca regulă de inferenţă de către J.A.Robinson în 1965 şi constituie sistemul deductiv cel mai frecvent utilizat în demonstrarea automată.

Reamintim că rezoluţia este o regulă de inferenţă derivată, reprezentată convenţional prin $\frac{(\alpha\to\beta),((\neg\alpha)\to\gamma)}{(\beta\vee\gamma)}REZ$ (Aplicaţia 1.4.9).

Observaţia 4.3.1 Deoarece

$$((\alpha \to \beta) \to (((\neg\alpha) \to \gamma) \to (\beta \vee \gamma))) \in T_h$$

aplicând teorema deducţiei obţinem

$$\{(\alpha \to \beta), ((\neg\alpha) \to \gamma)\} \vdash (\beta \vee \gamma)$$

sau echivalent

$$\{(\alpha \to \beta), ((\neg\alpha) \to \gamma)\} \vdash \{(\beta \vee \gamma)\}$$

şi care din punct de vedere semantic revine la

$$\{(\alpha \to \beta), ((\neg\alpha) \to \gamma)\} \vDash \{(\beta \vee \gamma)\}$$

adică

$$\aleph((\alpha \to \beta)) \cap \aleph(((\neg\alpha) \to \gamma)) \subset \aleph((\beta \vee \gamma)).$$

Deoarece

$$\aleph((\alpha \to \beta)) = (\Im \setminus \aleph(\alpha)) \cup \aleph(\beta) = \aleph((\neg\alpha)) \cup \aleph(\beta) = \aleph(((\neg\alpha) \vee \beta))$$

şi

$$\aleph(((\neg\alpha) \to \gamma)) = (\Im \setminus \aleph((\neg\alpha))) \cup \aleph(\gamma) = \aleph(\alpha) \cup \aleph(\gamma) = \aleph((\alpha \vee \gamma))$$

obţinem,

$$\aleph(((\neg\alpha) \vee \beta)) \cap \aleph((\alpha \vee \gamma)) \subset \aleph((\beta \vee \gamma))$$

adică

$$\{((\neg\alpha) \vee \beta), (\alpha \vee \gamma)\} \vDash \{(\beta \vee \gamma)\}$$

ceea ce este echivalent cu

$$\{((\neg\alpha) \vee \beta), (\alpha \vee \gamma)\} \vdash (\beta \vee \gamma).$$

Definiţia 4.3.1 *Fie k_1, k_2 clauze şi λ literal. Clauzele k_1, k_2 sunt λ-rezolubile dacă $k_1\langle\lambda\rangle$ şi k_2 $\langle(\neg\lambda)\rangle$. Clauza $rez_\lambda(k_1, k_2) = (k_1 \setminus \lambda) \vee (k_2 \setminus (\neg\lambda))$ este λ-rezolventa perechii de clauze (k_1, k_2) . Clauzele k_1, k_2 se numesc clauze parentale ale rezolventei.*

Observaţia 4.3.2 În particular, pentru $\alpha = \lambda$, $\beta = (k_2 \setminus (\neg\lambda))$, $\gamma = (k_1 \setminus \lambda)$, din observaţia precedentă rezultă

$$\{k_1, k_2\} \vDash rez_\lambda(k_1, k_2)$$

sau echivalent,

$$\aleph(k_1) \cap \aleph(k_2) \subset \aleph(rez_\lambda(k_1, k_2)).$$

Evident, dacă niciuna din clauzele k_1, k_2 nu este tautologie atunci

$$rez_\lambda(k_1, k_2)\rangle\lambda\langle$$

şi

$$rez_\lambda(k_1, k_2)\rangle(\neg\lambda)\langle.$$

Exemplul 4.3.1 Fie

$$\begin{aligned} k_1 &= a \vee (\neg b) \vee c, \\ k_2 &= (\neg a) \vee (\neg c), \\ k_3 &= (\neg b) \vee (\neg a). \end{aligned}$$

Clauzele k_1, k_2 sunt a-rezolubile şi c-rezolubile.

$$\begin{aligned} rez_a(k_1, k_2) &= (\neg b) \vee c \vee (\neg c) \\ rez_c(k_1, k_2) &= a \vee (\neg b) \vee (\neg a) \end{aligned}$$

Clauzele k_1, k_3 sunt a-rezolubile

$$rez_a(k_1, k_3) = (\neg b) \vee c \vee (\neg b) \equiv c \vee (\neg b)$$

Clauzele k_2, k_3 nu sunt rezolubile.

Din exemplele considerate rezultă că este posibil ca rezolventa a două clauze să fie clauză tautologie în condiţiile în care ambele clauze parentale nu sunt tautologii. De asemenea, este posibil ca în rezolventa unei perechi de clauze să fie generate duplicate ale unuia sau mai mulţi literali.

Definiţia 4.3.2 *Fie $S(\alpha)$ o reprezentare clauzală liberă de tautologii şi λ literal. Rezoluţia $REZ_\lambda(\alpha)$ în raport cu literalul λ este reprezentarea clauzală,*

$$REZ_\lambda(\alpha) = \alpha_\lambda^0 \cup \left\{rez_\lambda(k_1, k_2) \mid k_1 \in \alpha_\lambda^+, k_2 \in \alpha_\lambda^- \right\}$$

Exemplul 4.3.2 Dacă

$$S(\alpha) = \{(\neg p) \vee o, (\neg p) \vee (\neg c), (\neg m) \vee c \vee i, m, p, (\neg i)\}$$

atunci,

$$\begin{aligned} REZ_p(\alpha) &= \{(\neg m) \vee c \vee i, m, (\neg i), o, (\neg c)\}, \\ REZ_c(REZ_p(\alpha)) &= \{(\neg m) \vee i, m, (\neg i), o\}, \\ REZ_i(REZ_c(REZ_p(\alpha))) &= \{(\neg m), m, o\}, \\ REZ_m(REZ_i(REZ_c(REZ_p(\alpha)))) &= \{\square, o\}. \end{aligned}$$

Teorema 4.3.1 (Teorema de bază a rezoluţiei) . *Fie $S(\alpha)$ o reprezentare clauzală. Dacă $S(\alpha)$ este validabilă, atunci pentru orice literal λ, $REZ_\lambda(\alpha)$ este validabilă. Dacă există un literal λ astfel încât $REZ_\lambda(\alpha)$ este validabilă, atunci $S(\alpha)$ este validabilă.*

Demonstraţie. Presupunem că $S(\alpha)$ este validabilă şi fie λ literal arbitrar. Fie $I \in \aleph(S(\alpha))$ arbitrară, deci $I \in \bigcap\limits_{k\in S(\alpha)} \aleph(k)$.

Pentru orice $k \in REZ_\lambda(\alpha)$ obţinem,

$a)$ dacă $k \in \alpha_\lambda^0$, deoarece $\alpha_\lambda^0 \subset S(\alpha)$ rezultă $I(k) = T$

$b)$ dacă $k = rez_\lambda(k_1, k_2)$ pentru anume $k_1 \in \alpha_\lambda^+, k_2 \in \alpha_\lambda^-$,atunci cum $\{k_1, k_2\} \vDash \mathrm{rez}_\lambda(k_1, k_2)$, rezultă $I(k) = T$.

Obţinem astfel

$$I \in \bigcap_{k\in REZ_\lambda(\alpha)} \aleph(k) = \aleph(REZ_\lambda(\alpha)),$$

deci $REZ_\lambda(\alpha)$ este validabilă. În particular, deoarece pentru orice $I \in \aleph(S(\alpha))$ a rezultat $I \in \aleph(REZ_\lambda(\alpha))$, obţinem

$$\aleph(S(\alpha)) \subset \aleph(REZ_\lambda(\alpha)).$$

Presupunem că există λ literal astfel încât $REZ_\lambda(\alpha)$ este validabilă şi fie

$$I \in \aleph(REZ_\lambda(\alpha))$$

arbitrară.

Dacă $I \in \aleph(POS_\lambda(\alpha))$ atunci utilizând rezultatul stabilit de Teorema 4.2.1 rezultă $S(\alpha)$ validabilă. Dacă $I \notin \aleph(POS_\lambda(\alpha))$ atunci există $k \in POS_\lambda(\alpha)$ astfel încât $I(k) = F$. În acest caz, deoarece $\alpha_\lambda^0 \subset REZ_\lambda(\alpha)$, obţinem $k = k_1 \setminus \lambda$ pentru anume $k_1 \in \alpha_\lambda^+$.

Fie $k^* \in NEG_\lambda(\alpha)$ arbitrară.

$a)$ dacă $k^* \in \alpha_\lambda^0 \subset REZ_\lambda(\alpha)$ atunci $I(k^*) = T$.

$b)$ dacă $k^* = k_2 \setminus (\neg\lambda)$ pentru anume $k_2 \in \alpha_\lambda^-$, atunci k_1, k_2 sunt $\lambda-$rezolubile şi $rez_\lambda(k_1, k_2) \in REZ_\lambda(\alpha)$ deci $I(rez_\lambda(k_1, k_2)) = T$.

Obţinem,

$$\begin{aligned} I(rez_\lambda(k_1, k_2)) &= I((k_1 \setminus \lambda) \vee (k_2 \setminus (\neg\lambda))) = \\ &= I((k_1 \setminus \lambda)) \vee I((k_2 \setminus (\neg\lambda))) = \\ &= I(k) \vee I(k^*) = F \vee I(k^*) = I(k^*) \end{aligned}$$

deci $I(k^*) = T$

Rezultă

$$I \in \bigcap_{k^*\in NEG_\lambda(\alpha)} \aleph(k^*) = \aleph(NEG_\lambda(\alpha))$$

deci $NEG_\lambda(\alpha)$ este validabilă. Utilizând rezultatul stabilit de Teorema 4.2.1 rezultă $S(\alpha)$ validabilă. ■

În particular, a rezultat că pentru orice $I \in \aleph(REZ_\lambda(\alpha))$, $I \in \aleph(POS_\lambda(\alpha))$ sau $I \in \aleph(NEG_\lambda(\alpha))$, deci

$$\aleph(REZ_\lambda(\alpha)) \subset \aleph(POS_\lambda(\alpha)) \cup \aleph(NEG_\lambda(\alpha)).$$

Corolarul 4.3.1 *Dacă $S(\alpha)$ este o reprezentare clauzală, atunci*

$$\aleph(S(\alpha)) \subset \bigcap_{\lambda\ \in V\cup\neg V} \aleph(REZ_\lambda(\alpha)) \subset \bigcap_{\lambda\ \in V\cup\neg V} (\aleph(NEG_\lambda(\alpha)) \cup \aleph(POS_\lambda(\alpha))).$$

Demonstraţie. Din argumentele utilizate în demonstraţia Teoremei 4.3.1 a rezultat că pentru orice λ literal,

$$\aleph(S(\alpha)) \subset \aleph(REZ_\lambda(\alpha)),$$

deci

$$\aleph(S(\alpha)) \subset \bigcap_{\lambda\ \in V\cup\neg V} \aleph(REZ_\lambda(\alpha)).$$

De asemenea, pentru fiecare λ literal, a rezultat relaţia

$$\aleph(REZ_\lambda(\alpha)) \subset \aleph(POS_\lambda(\alpha)) \cup \aleph(NEG_\lambda(\alpha)),$$

deci

$$\bigcap_{\lambda\ \in V\cup\neg V} \aleph(REZ_\lambda(\alpha)) \subset \bigcap_{\lambda\ \in V\cup\neg V} (\aleph(NEG_\lambda(\alpha)) \cup \aleph(POS_\lambda(\alpha))).$$

■

Definiţia 4.3.3 *Fie $S(\alpha)$ o reprezentare clauzală liberă de tautologii. Secvenţa de clauze $k_1, \ldots, k_n$ este o $S(\alpha)-$derivare rezolutivă dacă pentru orice i, $1 \leq i \leq n$, este îndeplinită una din condiţiile,*

(i) $k_i \in S(\alpha)$

(ii) există $1 \leq j, p \leq i$ şi există λ literal astfel încât $k_i = rez_\lambda(k_j, k_p)$

$S(\alpha)$-derivarea rezolutivă $k_1, \ldots, k_n$ este o $S(\alpha)$-respingere rezolutivă dacă $k_n = \square$.

Teorema 4.3.2 (Teorema de completitudine a rezoluţiei) . *Fie $S(\alpha)$ o reprezentare clauzală. Atunci $S(\alpha)$ este invalidabilă dacă şi numai dacă există o $S(\alpha)$-respingere rezolutivă.*

Demonstraţie. Demonstrăm că pentru orice $S(\alpha)$-derivare rezolutivă $k_1, \ldots, k_n$, $S(\alpha) \models k_i$, $1 \leq i \leq n$.

Evident, $k_1 \in S(\alpha)$, deci cum

$$\aleph(S(\alpha)) = \bigcap_{k\in S(\alpha)} \aleph(k) \subset \aleph(k_1),$$

rezultă

$$S(\alpha) \models k_1.$$

Presupunem $S(\alpha) \models k_i$, $\forall i$, $1 \leq i \leq j \leq n-1$.

Dacă $k_{j+1} \in S(\alpha)$ atunci din

$$\aleph(S(\alpha)) = \bigcap_{k\in S(\alpha)} \aleph(k) \subset \aleph(k_{j+1})$$

obţinem $S(\alpha) \models k_1$.

Dacă există i, p, $1 \leq i, p \leq j$ şi λ literal astfel încât $k_{j+1} = rez_\lambda(k_i, k_p)$, utilizând ipoteza inductivă, obţinem

$$\aleph(S(\alpha)) \subset \aleph(k_i) \cap \aleph(k_p) \subset \aleph(rez_\lambda(k_i, k_p))$$

deci

$$S(\alpha) \models k_{j+1}.$$

Presupunem că există o $S(\alpha)$-derivare rezolutivă $k_1, \ldots, k_n = \square$. Din

$$\aleph(S(\alpha)) \subset \aleph(\square) = \emptyset$$

obţinem evident $\aleph(S(\alpha)) = \emptyset$ deci $S(\alpha)$ este invalidabilă.

Reciproc, presupunem că $S(\alpha)$ este invalidabilă. Fără restrângerea generalităţii, putem presupune că $S(\alpha)$ este liberă de tautologii.

Fie $\lambda_1, \ldots, \lambda_n$ literalii care au ocurenţe în clauzele mulţimii $S(\alpha)$. Considerăm secvenţa de reprezentări clauzale,

$$S_0(\alpha) = S(\alpha), S_i(\alpha) = REZ_{\lambda_i}(S_{i-1}(\alpha)),\ i = 1, \ldots, n$$

Presupunem de asemenea că $S_i(\alpha)$, $i = 1, \ldots, n$ sunt libere de tautologii.

Rezultă că pentru fiecare $i = 1, \ldots, n$, literalii $\lambda_1, \ldots, \lambda_i$ nu au ocurenţe în clauzele mulţimii $S_i(\alpha)$, deci $S_n(\alpha) = \varnothing$ sau $S_n(\alpha) = \{\square\}$.

Dacă $S_n(\alpha) = \varnothing$ atunci $S_{n-1}(\alpha)$ este validabilă. Din Teorema 4.3.1 rezultă $S_i(\alpha)$ validabilă, $i = 0, \ldots, n$ deci $S(\alpha)$ este validabilă.

Analog, dacă $S_n(\alpha) = \{\square\}$ atunci $S_{n-1}(\alpha)$ este invalidabilă deci iterând acelaşi argument obţinem $S_i(\alpha)$ invalidabilă, $i = 0, \ldots, n$.

În ipoteza că $S(\alpha)$ este invalidabilă rezultă $S_n(\alpha) = \{\square\}$.

Dacă

$$S_i(\alpha) = \left\{\gamma_i^1, \ldots, \gamma_i^{m_i}\right\}\ i = 0, \ldots, n$$

, evident

$$m_n = 1 \text{ şi } \gamma_n^1 = \square.$$

$$\gamma_1^1, \ldots, \gamma_1^{m_1}, \gamma_2^1, \ldots, \gamma_2^{m_2}, \ldots, \gamma_{n-1}^1, \ldots, \gamma_{n-1}^{m_{n-1}}, \gamma_n^1$$

este o $S(\alpha)$-derivare rezolutivă. Deoarece $\gamma_n^1 = \square$, obţinem în final că

$$\gamma_1^1, \ldots, \gamma_1^{m_1}, \gamma_2^1, \ldots, \gamma_2^{m_2}, \ldots, \gamma_{n-1}^1, \ldots, \gamma_{n-1}^{m_{n-1}}, \gamma_n^1$$

este o $S(\alpha)$-respingere rezolutivă. ■

Rezultatul stabilit de Teorema 4.3.2 constituie suportul unui sistem de demonstrare automată pentru verificarea validabilităţii reprezentărilor clauzale. În esenţă, metoda revine la tentativa de obţinere a unei respingeri rezolutive prin substituirea reprezentării clauzale curente printr-o reprezentare clauzală cu numar de literali strict mai mic, dar posibil cu mai multe clauze. Deoarece prin luarea rezolventelor este posibilă generarea de clauze tautologii, în scopul menţinerii reprezentărilor clauzale libere de tautologii este necesar ca la fiecare etapă să fie aplicat un test pentru detectarea şi eliminarea rezolventelor tautologii nou generate. Spre deosebire de tehnica Davis-Putnam, implementarea demonstratorului rezolutiv nu necesită menţinerea unei structuri de date suplimentare. Similar metodei Davis-Putnam, aplicarea prioritară a regulilor literalilor puri şi respectiv a clauzei unitare, este o modalitate de prevenire a creşterii excesive a numărului de clauze.

Descrierea algoritmului de demonstrare automată rezolutiv este:

```
procedure DemRez;
Intrare: Structură de date pentru stocarea reprezentării clauzale S(α)
EliminaTautologii(S(α));
EliminaDuplicate(S(α));
γ ← S(α); sw ← false;
repeat
if γ = ∅ then
              write ('validabilă');
              sw← true
         else
            if □ ∈ γ then
                          write ('invalidabilă');
                          sw← true
                     else
                         if (există λ clauza unitară) or (există λ literal pur) then
                                                                              γ ← NEG_λ(γ)
                                                                          else
                                                                              alege (λ literal);
                                                                              γ ← REZ_λ(γ);
                                                                              EliminaTautologii(γ);
                                                                              EliminaDuplicate(γ);
                         endif
            endif
endif
until sw
end.
```

Observaţia 4.3.3 Similar algoritmului DvP, apelul EliminaTautologii(γ) detectează şi elimină clauzele tautologii din reprezentarea clauzală curentă γ (iniţial $\gamma = S(\alpha)$) . Apelul este necesar la fiecare etapă în care noua reprezentare clauzală este calculată ca rezoluţie a reprezentării clauzale curente. Apelul EliminaDuplicate(γ) detectează şi elimină duplicatele literalilor din fiecare clauză a reprezentării clauzale curente. Apelul alege(λ literal) efectuează selectarea unui literal pentru aplicarea rezoluţiei.

Exemplul 4.3.3 Procedura DemRez aplicată reprezentării clauzale

$$S(\alpha) = \{(\neg a) \vee (\neg c) \vee b,\ (\neg b) \vee c,\ (\neg c) \vee a,\ a \vee (\neg b) \vee (\neg c)\},$$

determină următoarea evoluţie:

Iniţializare: $\gamma \leftarrow S(\alpha)$, $sw \leftarrow false$,

Iteraţia 1: Nu există clauze unitare şi nici literali puri.
Apelul alege(λ literal) $\Rightarrow \lambda = a$
$\gamma \leftarrow REZ_a(\gamma) = \{(\neg b) \vee c,\ (\neg c) \vee b \vee (\neg c), (\neg c) \vee b \vee (\neg b) \vee (\neg c)\}$
Apelul EliminaTautologii(γ) $\Rightarrow$ elimină clauza tautologie $(\neg c) \vee b \vee (\neg b) \vee (\neg c)$
$\gamma \leftarrow \{(\neg b) \vee c,\ (\neg c) \vee b \vee (\neg c)\}$
Apelul EliminaDuplicate(γ) $\Rightarrow$ elimină duplicatul literalului $(\neg c)$ din clauza $(\neg c) \vee b \vee (\neg c)$
$\gamma \leftarrow \{(\neg b) \vee c, (\neg c) \vee b\}$,

Iteraţia 2: Nu există clauze unitare şi nici literali puri.
Apelul alege(λ literal) $\Rightarrow \lambda = b$
$\gamma \leftarrow REZ_b(\gamma) = \{c \vee (\neg c)\}$
Apelul EliminaTautologii(γ) $\Rightarrow$ elimină clauza tautologie $c \vee (\neg c)$
$\gamma \leftarrow \emptyset$
Apelul EliminaDuplicate(γ) $\Rightarrow$ nu determină modificări,

Iteraţia 3: $\gamma = \emptyset \Rightarrow$ write('validabilă'), sw$\leftarrow$ *true* $\Rightarrow$STOP.

4.4 Metoda arborilor semantici

Metoda arborilor semantici este un algoritm relativ eficient pentru verificarea validabilităţii formulelor limbajului calculului cu propoziţii. Similar metodelor precendente, metoda iniţiază căutarea sistematică a unui model pentru formula căreia îi este aplicată. Ideea care stă la baza metodei este simplă şi intuitivă şi anume este bazată pe observaţia că o mulţime finită de literali este invalidabilă dacă şi numai dacă mulţimea conţine cel puţin o pereche de literali complementari. Deoarece orice structură simbolică de tipul formulă conţine un număr finit de literali, verificarea validabilităţii unei formule revine în ultimă instanţă la o tehnică de construcţie a uneia sau mai multe mulţimi de literali corelate semantic cu formula considerată.

De exemplu, fie formula $\alpha = (p \wedge ((\neg q) \vee (\neg p)))$.

Pentru $I \in \Im$ arbitrară, $I(\alpha) = T$ dacă şi numai dacă $I(p) = T$ şi

$$I(((\neg q) \vee (\neg p))) = T.$$

Deoarece, $I(((\neg q) \vee (\neg p))) = T$ dacă şi numai dacă $I((\neg q)) = T$ sau $I((\neg p)) = T$. Rezultă $I(\alpha) = T$ dacă şi numai dacă $I(p) = T$ şi $I((\neg q)) = T$ deci $\aleph(\alpha) = \aleph(\{p, (\neg q)\})$. În acest mod, validabilitatea formulei α corespunde validabilităţii mulţimii de literali $\{p, (\neg q)\}$, care este evident validabilă. Deci α este validabilă şi

$$\aleph(\alpha) = \{I \mid I \in \Im,\ I(p) = T, I(q) = F\}.$$

Procedând în mod similar pentru formula,

$$\beta = ((p \vee q) \wedge ((\neg p) \wedge (\neg q))),$$

pentru $I \in \Im$ arbitrară obţinem, $I(\beta) = T$ dacă şi numai dacă $I((p \vee q)) = T$ şi $I(((\neg p) \wedge (\neg q))) = T$. Rezultă $I(\beta) = T$ dacă şi numai dacă este indeplinită cel puţin una din condiţiile,

a) $I(p) = I((\neg p)) = I((\neg q)) = T$,

b) $I(q) = I((\neg p)) = I((\neg q)) = T$,

adică dacă şi numai dacă cel puţin una din mulţimile

$$\{p, (\neg p), (\neg q)\}, \{q, (\neg p), (\neg q)\}$$

este validabilă. Evident, ambele mulţimi conţin câte o pereche de literali complementari deci sunt invalidabile, ceea ce în final implică, β invalidabilă.

Implementarea unei metode de căutare sistematică este facilitată de utilizarea unor reprezentări tabelare şi/sau grafice. În cele ce urmează, opţiunea este pentru structuri de reprezentare de tip arborescenţe etichetate, sistemul de etichete fiind gestionat astfel:

rădăcina arborelui este etichetată cu formula dată, vârfurile interioare sunt etichetate cu formulele generate pe durata procesului de căutare respectiv vârfurile terminale au ca etichete mulţimi de literali a caror validabilitate este testată. Vârfurile terminale cu etichete mulţimi invalidabile sunt marcate “$\times$” (vârf închis), respectiv vârfurile terminale cu etichete mulţimi validabile sunt marcate “$\bigcirc$” (vârf deschis). Arborele etichetat rezultat se numeşte *arbore semantic.*

Definiţia 4.4.1 *Un arbore semantic este* complet *dacă toate vârfurile terminale sunt marcate “$\times$” sau “$\bigcirc$”. Un arbore semantic complet este* închis *dacă toate vârfurile terminale sunt marcate “$\times$”. Dacă cel puţin un vârf terminal al unui arbore semantic complet este marcat “$\bigcirc$” atunci arborele este* deschis.

Exemplul 4.4.1 *a) Arborele semnatic corespunzător formulei* $\alpha = (p \wedge ((\neg q) \vee (\neg p)))$ *este*

$$
\begin{array}{ccc}
 & (p \wedge ((\neg q) \vee (\neg p))) & \\
 & \downarrow & \\
 & \{p, ((\neg q) \vee (\neg p))\} & \\
 & \swarrow \searrow & \\
\{p, (\neg q)\} & & \{p, (\neg p)\} \\
\bigcirc & & \times
\end{array}
$$

b) Pentru formula $\beta = ((p \vee q) \wedge ((\neg p) \wedge (\neg q)))$ *arborele semantic este următorul*

$$
\begin{array}{ccc}
 & ((p \vee q) \wedge ((\neg p) \wedge (\neg q))) & \\
 & \downarrow & \\
 & \{(p \vee q), ((\neg p) \wedge (\neg q))\} & \\
 & \downarrow & \\
 & \{(p \vee q), (\neg p), (\neg q)\} & \\
 & \swarrow \searrow & \\
\{p, (\neg p), (\neg q)\} & & \{q, (\neg p), (\neg q)\} \\
\times & & \times
\end{array}
$$

Definiţia 4.4.2 *Fie* $\gamma \in FORM$. *Spunem că* γ *este o* $\alpha - formul\breve{a}$ *dacă este îndeplinită una din condiţiile:*

(i) există $\delta \in FORM$ *astfel încât* $\gamma = (\neg (\neg \delta))$,

(ii) există $\delta, \eta \in FORM$ *astfel încât* $\gamma = (\delta \wedge \eta)$,

(iii) există $\delta, \eta \in FORM$ *astfel încât* $\gamma = (\neg (\delta \vee \eta))$,

(iv) există $\delta, \eta \in FORM$ *astfel încât* $\gamma = (\neg (\delta \rightarrow \eta))$,

(v) există $\delta, \eta \in FORM$ *astfel încât* $\gamma = (\delta \leftrightarrow \eta)$.

Definiţia 4.4.3 *Fie* $\gamma \in FORM$. *Spunem că* γ *este o* $\beta - formul\breve{a}$ *dacă este îndeplinită una din condiţiile:*

(i) există $\delta, \eta \in FORM$ *astfel încât* $\gamma = (\delta \vee \eta)$,

(ii) există $\delta, \eta \in FORM$ *astfel încât* $\gamma = (\neg (\delta \wedge \eta))$,

(iii) există $\delta, \eta \in FORM$ *astfel încât* $\gamma = (\delta \to \eta)$,

(iv) există $\delta, \eta \in FORM$ *astfel încât* $\gamma = (\neg(\delta \leftrightarrow \eta))$.

Generarea unui tablou semantic este bazată pe următoarele două tipuri de reguli:

1. $\alpha - reguli$ ($Reguli\ pentru\ \alpha - formule$)

γ	γ_1	γ_2
$(\neg(\neg\delta))$	δ	
$(\delta \wedge \eta)$	δ	η
$(\neg(\delta \vee \eta))$	$(\neg\delta)$	$(\neg\eta)$
$(\neg(\delta \to \eta))$	δ	$(\neg\eta)$
$(\delta \leftrightarrow \eta)$	$(\delta \to \eta)$	$(\eta \to \delta)$

2. $\beta - reguli$ ($Reguli\ pentru\ \beta - formule$)

γ	γ_1	γ_2
$(\delta \vee \eta)$	δ	η
$(\neg(\delta \wedge \eta))$	$(\neg\delta)$	$(\neg\eta)$
$(\delta \to \eta)$	$(\neg\delta)$	η
$(\neg(\delta \leftrightarrow \eta))$	$(\neg(\delta \to \eta))$	$(\neg(\eta \to \delta))$

Observaţia 4.4.1 Mulţimea $FORM$ este partajată în mulţimea $\alpha - formulelor$ şi respectiv mulţimea $\beta - formulelor$. Pentru orice formulă, regula aplicabilă există şi este unică.

Procedura pentru construcţia unui arbore semantic complet

Fie γ formula a cărei validabilitate este testată.

Iniţializare: T arbore cu un singur vârf (rădăcina) etichetat $\{\gamma\}$,

End C: toate vârfurile terminale ale arborelui T sunt marcate "$\times$" sau "$\bigcirc$",

P1. Selectează un vârf terminal neetichetat n. Fie $U(n)$ eticheta vârfului n,

P2. Dacă toate formulele din $U(n)$ sunt literali, atunci dacă există cel puţin o pereche de literali complementari în $U(n)$ marchează vârful n cu "$\times$", altfel marchează cu "$\bigcirc$",

P3. Dacă există cel puţin o formulă în $U(n)$ care nu este literal, atunci selectează o formulă ϑ care nu este literal,

a) Dacă ϑ este o $\alpha - formulă$, atunci extinde arborele T adăugând un nou vârf n_1 ca fiu al vârfului n. Etichetează vârful n_1 :
   - $U(n_1) \leftarrow (U(n) \setminus \{\vartheta\}) \cup \{\vartheta_1\}$ dacă $\vartheta = (\neg(\neg\vartheta_1))$,
   - $U(n_1) \leftarrow (U(n) \setminus \{\vartheta\}) \cup \{\vartheta_1, \vartheta_2\}$ unde ϑ_1, ϑ_2 rezultă prin aplicarea $\alpha - regulei$ asociate formulei ϑ.

b) Dacă ϑ este o $\beta - formulă$, atunci extinde arborele T adăugând două noi vârfuri n_1, n_2 ca fii ai vârfului n. Etichetează vârfurile n_1, n_2:
   - $U(n_1) \leftarrow (U(n) \setminus \{\vartheta\}) \cup \{\vartheta_1\}$,
   - $U(n_2) \leftarrow (U(n) \setminus \{\vartheta\}) \cup \{\vartheta_2\}$

   unde ϑ_1, ϑ_2 rezultă prin aplicarea $\beta - regulei$ asociate formulei ϑ.

4.5 Aplicaţii

Exemplul 4.5.1 *Să se determine forma normal conjunctivă (CNF) pentru formula:*

$$\neg(\neg p \vee q) \vee (r \to \neg s)$$

Soluţie
1. $\neg(\neg p \vee q) \vee (\neg r \vee \neg s)$
2. $(\neg\neg p \wedge \neg q) \vee (\neg r \vee \neg s)$
3. $(p \wedge \neg q) \vee (\neg r \vee \neg s)$
4. $(p \vee \neg r \vee \neg s) \wedge (\neg q \vee \neg r \vee \neg s)$.

Exemplul 4.5.2 *Reduceţi la CNF formulele*
a) $(\neg p \to q) \to (q \to \neg r)$
b) $p \vee (\neg q \wedge (r \to \neg p))$
Soluţie
a)
1. $\neg(\neg p \to q) \vee (q \to \neg r)$
2. $\neg(p \vee q) \vee (\neg q \vee \neg r)$
3. $(\neg p \wedge \neg q) \vee (\neg q \vee \neg r)$
4. $(\neg p \vee \neg q \vee \neg r) \wedge (\neg q \vee \neg r)$.
b)
1. $p \vee (\neg q \wedge (\neg r \vee \neg p))$
2. $p \vee (\neg q \wedge \neg(r \wedge p))$
3. $(p \vee \neg q) \wedge (p \vee \neg(r \wedge p))$
4. $(p \vee \neg q) \wedge (p \vee \neg r \vee \neg q)$

Exemplul 4.5.3 *Să se calculeze mulţimile* α^+_λ, α^-_λ, α^0_λ, $POS_\lambda(\alpha)$, $NEG_\lambda(\alpha)$ *unde* $\lambda = \beta$ *respectiv* $\lambda = \neg\delta$ *iar*

$$S(\alpha) = \{\neg\beta \vee \delta \vee \gamma,\ \beta \vee \eta \vee \neg\delta,\ \delta \vee \beta \vee \theta,\ \neg\delta,\ \gamma \vee \neg\eta \vee \neg\delta, \delta, \beta \vee \eta \vee \theta,\ \neg\beta \vee \neg\delta\}.$$

Soluţie
Fie $\lambda = \beta$. *În acest caz, obţinem mulţimile:*

$$\begin{aligned}
\alpha^+_\beta &= \{\beta \vee \eta \vee \neg\delta,\ \delta \vee \beta \vee \theta,\ \beta \vee \eta \vee \theta\}\\
\alpha^-_\beta &= \{\neg\beta \vee \delta \vee \gamma,\ \neg\beta \vee \neg\delta\}\\
\alpha^0_\beta &= \{\neg\delta,\ \gamma \vee \neg\eta \vee \neg\delta, \delta, \}\\
POS_\beta(\alpha) &= \{\neg\delta,\ \gamma \vee \neg\eta \vee \neg\delta, \delta, \} \cup \{\eta \vee \neg\delta,\ \delta \vee \theta,\ \eta \vee \theta\} =\\
&= \{\neg\delta,\ \gamma \vee \neg\eta \vee \neg\delta, \delta, \eta \vee \neg\delta,\ \delta \vee \theta,\ \eta \vee \theta\}\\
NEG_\beta(\alpha) &= \{\neg\delta,\ \gamma \vee \neg\eta \vee \neg\delta, \delta, \} \cup \{\delta \vee \gamma,\ \neg\delta\} =\\
&= \{\neg\delta,\ \gamma \vee \neg\eta \vee \neg\delta, \delta,\ \delta \vee \gamma,\ \neg\delta\}.
\end{aligned}$$

Pentru $\lambda = \neg\delta$ *obţinem următoarele mulţimi:*

$$\begin{aligned}
\alpha^+_{\neg\delta} &= \{\beta \vee \eta \vee \neg\delta,\ \neg\delta,\ \gamma \vee \neg\eta \vee \neg\delta\}\\
\alpha^-_{\neg\delta} &= \{\neg\beta \vee \delta \vee \gamma,\ \delta \vee \beta \vee \theta,\ \delta, \}\\
\alpha^0_{\neg\delta} &= \{\beta \vee \eta \vee \theta\}\\
POS_{\neg\delta}(\alpha) &= \{\beta \vee \eta \vee \theta, \} \cup \{\beta \vee \eta,\ \Box,\ \gamma \vee \neg\eta\} = \{\beta \vee \eta \vee \theta,\ \beta \vee \eta,\ \Box,\ \gamma \vee \neg\eta\}\\
NEG_{\neg\delta}(\alpha) &= \{\beta \vee \eta \vee \theta, \} \cup \{\neg\beta \vee \gamma,\ \beta \vee \theta,\ \Box\} = \{\beta \vee \eta \vee \theta,\ \neg\beta \vee \gamma,\ \beta \vee \theta,\ \Box\}.
\end{aligned}$$

Exemplul 4.5.4 *Să se calculeze mulţimile* α_λ^+, α_λ^-, α_λ^0, $POS_\lambda(\alpha)$, $NEG_\lambda(\alpha)$, $REZ_\lambda(\alpha)$ *unde* $\lambda = \neg\beta$ *respectiv* $\lambda = \delta$ *iar*

$$S(\alpha) = \{\delta \vee \neg\gamma \vee \beta \vee \eta,\ \neg\beta \vee \eta \vee \gamma,\ \delta \vee \neg\beta \vee \gamma,\ \neg\beta,\ \gamma \vee \neg\delta \vee \eta\}.$$

Soluţie

Pentru $\lambda = \neg\beta$ *obţinem următoarele mulţimi:*

$$\begin{aligned}
\alpha_{\neg\beta}^+ &= \{\neg\beta \vee \eta \vee \gamma,\ \delta \vee \neg\beta \vee \gamma, \neg\beta\} \\
\alpha_{\neg\beta}^- &= \{\delta \vee \neg\gamma \vee \beta \vee \eta\} \\
\alpha_{\neg\beta}^0 &= \{\gamma \vee \neg\delta \vee \eta\} \\
POS_{\neg\beta}(\alpha) &= \{\gamma \vee \neg\delta \vee \eta\} \cup \{\eta \vee \gamma,\ \delta \vee \gamma, \square\} = \{\gamma \vee \neg\delta \vee \eta,\ \eta \vee \gamma,\ \delta \vee \gamma, \square\} \\
NEG_{\neg\beta}(\alpha) &= \{\gamma \vee \neg\delta \vee \eta\} \cup \{\delta \vee \neg\gamma \vee \eta\} = \{\gamma \vee \neg\delta \vee \eta,\ \delta \vee \neg\gamma \vee \eta\} \\
REZ_{\neg\beta}(\alpha) &= \{\gamma \vee \neg\delta \vee \eta\} \cup \{\eta \vee \gamma \vee \delta \vee \neg\gamma \vee \eta,\ \delta \vee \gamma \vee \delta \vee \neg\gamma \vee \eta, \square \vee \delta \vee \neg\gamma \vee \eta\} = \\
&= \{\gamma \vee \neg\delta \vee \eta,\ \eta \vee \gamma \vee \delta \vee \neg\gamma \vee \eta,\ \delta \vee \gamma \vee \delta \vee \neg\gamma \vee \eta, \delta \vee \neg\gamma \vee \eta\}.
\end{aligned}$$

Fie acum $\lambda = \delta$. *În această situaţie avem*

$$\begin{aligned}
\alpha_\delta^+ &= \{\delta \vee \neg\gamma \vee \beta \vee \eta,\ \delta \vee \neg\beta \vee \gamma\} \\
\alpha_\delta^- &= \{\gamma \vee \neg\delta \vee \eta\} \\
\alpha_\delta^0 &= \{\neg\beta \vee \eta \vee \gamma,\ \neg\beta\} \\
POS_\delta(\alpha) &= \{\neg\beta \vee \eta \vee \gamma,\ \neg\beta\} \cup \{\neg\gamma \vee \beta \vee \eta,\ \neg\beta \vee \gamma\} = \\
&= \{\neg\beta \vee \eta \vee \gamma,\ \neg\beta,\ \neg\gamma \vee \beta \vee \eta,\ \neg\beta \vee \gamma\} \\
NEG_\delta(\alpha) &= \{\neg\beta \vee \eta \vee \gamma,\ \neg\beta\} \cup \{\gamma \vee \eta\} = \{\neg\beta \vee \eta \vee \gamma,\ \neg\beta,\ \gamma \vee \eta\} \\
REZ_\delta(\alpha) &= \{\neg\beta \vee \eta \vee \gamma,\ \neg\beta\} \cup \{\neg\gamma \vee \beta \vee \eta \vee \gamma \vee \eta,\ \neg\beta \vee \gamma \vee \gamma \vee \eta\} = \\
&= \{\neg\beta \vee \eta \vee \gamma,\ \neg\beta,\ \neg\gamma \vee \beta \vee \eta \vee \gamma \vee \eta,\ \neg\beta \vee \gamma \vee \gamma \vee \eta\}.
\end{aligned}$$

Exemplul 4.5.5 *Fie* $\lambda = \eta$ *şi* $S(\alpha) = \{\beta \vee \eta \vee \gamma,\ \neg\beta \vee \eta \vee \theta,\ \neg\eta,\ \gamma \vee \neg\eta,\ \theta \vee \beta \vee \neg\eta\}$. *Calculaţi mulţimile* α_λ^+, α_λ^-, α_λ^0, $POS_\lambda(\alpha)$, $NEG_\lambda(\alpha)$, $REZ_\lambda(\alpha)$.

Soluţie

$$\begin{aligned}
\alpha_\eta^+ &= \{\beta \vee \eta \vee \gamma,\ \neg\beta \vee \eta \vee \theta\} \\
\alpha_\eta^- &= \{\neg\eta,\ \gamma \vee \neg\eta,\ \theta \vee \beta \vee \neg\eta\} \\
\alpha_\eta^0 &= \emptyset \\
POS_\eta(\alpha) &= \emptyset \cup \{\beta \vee \gamma,\ \neg\beta \vee \theta\} = \{\beta \vee \gamma,\ \neg\beta \vee \theta\} \\
NEG_\eta(\alpha) &= \emptyset \cup \{\square,\ \gamma,\ \theta \vee \beta\} = \{\square,\ \gamma,\ \theta \vee \beta\} \\
REZ_\eta(\alpha) &= \emptyset \cup \{\beta \vee \gamma \vee \square, \beta \vee \gamma, \beta \vee \gamma \vee \theta, \neg\beta \vee \theta \vee \square, \neg\beta \vee \theta \vee \gamma, \neg\beta \vee \theta \vee \beta\} = \\
&= \{\beta \vee \gamma \vee \square, \beta \vee \gamma, \beta \vee \gamma \vee \theta, \neg\beta \vee \theta \vee \square, \neg\beta \vee \theta \vee \gamma, \neg\beta \vee \theta \vee \beta\}.
\end{aligned}$$

Exemplul 4.5.6 *Să se aplice algoritmul Davis-Putnam următoarei reprezentări clauzale*

$$S(\alpha) = \{k_1,\ k_2,\ k_3,\ k_4,\ k_5,\ k_6\}$$

unde

$$\begin{aligned}
k_1 &= \neg p \vee q \vee \neg s\\
k_2 &= \neg r \vee \neg q\\
k_3 &= p \vee r\\
k_4 &= p\\
k_5 &= r\\
k_6 &= s.
\end{aligned}$$

Soluţie

Iniţializări: $\gamma \leftarrow \{\neg p \vee q \vee \neg s,\ \neg r \vee \neg q,\ p \vee r,\ p,\ r,\ s\}$;
$sw \leftarrow false; T \leftarrow \emptyset$,

Iteraţia 1: $\lambda = p$ *clauză unitară*
$\gamma \leftarrow NEG_p(\gamma) = \{\neg r \vee \neg q,\ r,\ s,\ q \vee \neg s\}$;

Iteraţia 2: $\lambda = r$ *clauză unitară*
$\gamma \leftarrow NEG_r(\gamma) = \{s,\ q \vee \neg s,\ \neg q\}$

Iteraţia 3: $\gamma = s$ *clauză unitară*
$\gamma \leftarrow NEG_s(\gamma) = \{\neg q,\ q\}$

Iteraţia 4: $\gamma = \neg q$ *clauză unitară*
$\gamma \leftarrow NEG_{\neg q}(\gamma) = \{\square\}$

Iteraţia 5: $\square \in \gamma$ *şi* $T = \emptyset \Rightarrow$ *write "**Invalidabilă**"*, $sw \leftarrow true$
$\Rightarrow$ *STOP.*

Exemplul 4.5.7 *Fie repreztarea clauzală*

$$S(\alpha) = \{k_1,\ k_2,\ k_3,\ k_4,\ k_5,\ k_6,\ k_7,\ k_8,\ k_9\}$$

unde

$$\begin{aligned}
k_1 &= p \vee q \vee r\\
k_2 &= p \vee \neg q \vee \neg r\\
k_3 &= p \vee \neg w\\
k_4 &= \neg r \vee \neg u\\
k_5 &= u \vee x\\
k_6 &= \neg r \vee \neg q \vee \neg w\\
k_7 &= r \vee \neg q \vee \neg p\\
k_8 &= \neg x \vee u\\
k_9 &= \neg u \vee q.
\end{aligned}$$

Soluţie

Iniţializări: $\gamma \leftarrow S(\alpha)$, $sw \leftarrow false; T \leftarrow \emptyset$;

Iteraţia 1: $\lambda = \neg w$ *literal pur*
$\gamma \leftarrow NEG_{\neg w}(\gamma) = \{p \vee q \vee r,\ p \vee \neg q \vee \neg r,\ \neg r \vee \neg u,\ u \vee x,\ r \vee \neg q \vee \neg p,\ \neg x \vee u,\ \neg u \vee q\}$;

Iteraţia 2: *Nu există clauze unitare şi nici literali puri.*
alege literal: $\lambda = p$
$T \leftarrow POS_p(\gamma) = \{\neg r \vee \neg u,\, u \vee x,\, \neg x \vee u,\, \neg u \vee q,\, \neg q \vee \neg r,\, q \vee r\}$
$\gamma \leftarrow NEG_p(\gamma) = \{\neg r \vee \neg u,\, u \vee x,\, \neg x \vee u,\, \neg u \vee q,\, \neg q \vee r\}$

Iteraţia 3: *Nu există clauze unitare şi nici literali puri.*
alege literal: $\lambda = q$
$T \leftarrow POS_q(\gamma) = \{\neg r \vee \neg u,\, u \vee x,\, \neg x \vee u,\, \neg u\}$
$\gamma \leftarrow NEG_q(\gamma) = \{\neg r \vee \neg u,\, u \vee x,\, \neg x \vee u,\, r\}$

Iteraţia 4: $\lambda = r$ *clauză unitară*
$\gamma \leftarrow NEG_r(\gamma) = \{u \vee x,\, \neg x \vee u,\, \neg u\}$

Iteraţia 5: $\lambda = \neg u$ *clauză unitară*
$\gamma \leftarrow NEG_{\neg u}(\gamma) = \{x,\, \neg x\}$

Iteraţia 6: $\lambda = x$ *clauză unitară*
$\gamma \leftarrow NEG_x(\gamma) = \{\square\}$

Iteraţia 7: $\square \in \gamma$
$\gamma \leftarrow TOP(T) = \{\neg r \vee \neg u,\, u \vee x,\, \neg x \vee u,\, \neg u\}$
$POP(T)$

Iteraţia 8: $\lambda = \neg u$ *clauză unitară*
$\gamma \leftarrow NEG_{\neg u}(\gamma) = \{x, \neg x\}$

Iteraţia 9: $\lambda = x$ *clauză unitară*
$\gamma \leftarrow NEG_x(\gamma) = \{\square\}$

Iteraţia 10: $\square \in \gamma$
$\gamma \leftarrow TOP(T) = \{\neg r \vee \neg u,\, u \vee x,\, \neg x \vee u,\, \neg u \vee q,\, \neg q \vee \neg r,\, q \vee r\}$
$POP(T)$

Iteraţia 11: *Nu există clauze unitare şi nici literali puri.*
alege literal: $\lambda = q$
$T \leftarrow POS_q(\gamma) = \{\neg r \vee \neg u,\, u \vee x,\, \neg x \vee u,\, \neg u,\, r\}$
$\gamma \leftarrow NEG_q(\gamma) = \{\neg r \vee \neg u,\, u \vee x,\, \neg x \vee u,\, \neg r\}$

Iteraţia 12: $\lambda = \neg r$ *literal pur*
$\gamma \leftarrow NEG_{\neg r}(\gamma) = \{u \vee x,\, \neg x \vee u\}$

Iteraţia 13: $\lambda = u$ *literal pur*
$\gamma \leftarrow NEG_u(\gamma) = \emptyset$

Iteraţia 13: $\lambda = \emptyset \Rightarrow$ *write "**Validabilă** ", sw* $\leftarrow$ *true*
$\Rightarrow$ *STOP.*

Exemplul 4.5.8 *Să se determine forma normal conjunctivă (CNF) pentru formula dată şi să se aplice algoritmul Davis-Putnam pentru formula*

$$\alpha = (\neg a \to b) \to ((c \to \neg d) \to (\neg a \to d)).$$

Soluţie

Forma normal conjunctivă se obţine astfel:
1. $\neg(a \vee b) \vee (\neg(\neg c \vee \neg d) \vee (a \vee d))$
2. $(\neg a \wedge \neg b) \vee ((c \wedge d) \vee (a \vee d))$
3. $(\neg a \wedge \neg b) \vee ((c \vee a \vee d) \wedge (d \vee a))$
4. $((\neg a \wedge \neg b) \vee (c \vee a \vee d)) \wedge ((\neg a \wedge \neg b) \vee (d \vee a))$
5. $(\neg a \vee c \vee a \vee d) \wedge (\neg b \vee c \vee a \vee d) \wedge (\neg a \vee d \vee a) \wedge (\neg b \vee d \vee a)$.

Fie reprezentarea clauzală

$$S(\alpha) = \{\neg a \vee c \vee a \vee d,\ \neg b \vee c \vee a \vee d,\ \neg a \vee d \vee a,\ \neg b \vee d \vee a\}.$$

Evoluţia determinată de procedura DvP pentru datele de intrare

$$S(\alpha) = \{\neg b \vee c \vee a \vee d,\ \neg b \vee d \vee a\} \text{ (au fost eliminate tautologiile)}$$

este:

Iniţializări: $\gamma \leftarrow S(\alpha)$, $sw \leftarrow false$, $T \leftarrow \emptyset$;

Iteraţia 1: $\lambda = a$ *literal pur*
$\gamma \leftarrow NEG_a(\gamma) = \emptyset$

Iteraţia 2: $\lambda = \emptyset \Rightarrow$ *write "**Validabilă** ",* $sw \leftarrow true$
$\Rightarrow$ *STOP.*

Exemplul 4.5.9 *Să se aplice algoritmul bazat pe rezoluţie următoarei reprezentări clauzale*

$$S(\alpha) = \{a \vee b \vee \neg c,\ \neg b \vee d \vee \neg a,\ a \vee c \vee \neg b,\ \neg a \vee \neg c \vee b\}.$$

Soluţie

Iniţializări: $\gamma \leftarrow S(\alpha)$, $sw \leftarrow false$;

Iteraţia 1: $\lambda = d$ *literal pur*
$\gamma \leftarrow NEG_d(\gamma) = \{a \vee b \vee \neg c,\ a \vee c \vee \neg b,\ \neg a \vee \neg c \vee b\}$

Iteraţia 2: Nu există clauze unitare şi nici literali puri.
alege literal: $\lambda = a$
$\gamma \leftarrow REZ_a(\gamma) = \{b \vee \neg c \vee \neg c \vee b,\ c \vee \neg b \vee \neg c \vee b\}$
EliminaTautologii(γ*):* $\gamma \leftarrow \{\neg c \vee b\}$

Iteraţia 3: $\lambda = b$ *(literal pur)*
$\gamma \leftarrow NEG_b(\gamma) = \emptyset$

Iteraţia 4: $\lambda = \emptyset \Rightarrow$ *write "**validabilă** ",* $sw \leftarrow true$
$\Rightarrow$ *STOP.*

Exemplul 4.5.10 *Să se determine forma normal conjunctivă (CNF) pentru formula dată şi să se aplice algoritmul bazat pe rezoluţie pentru formula*

$$\alpha = (\beta \vee \delta) \rightarrow (\neg(\gamma \wedge \delta) \rightarrow (\neg\gamma \rightarrow \beta))$$

Soluţie

Forma normal conjunctivă ataşată formulei date este:

1. $(\beta \vee \delta) \to (\neg(\gamma \wedge \delta) \to (\gamma \vee \beta))$
2. $(\beta \vee \delta) \to ((\gamma \wedge \delta) \vee (\gamma \vee \beta))$
3. $\neg(\beta \vee \delta) \vee ((\gamma \vee \beta) \wedge (\delta \vee \gamma \vee \beta))$
4. $(\neg\beta \wedge \neg\delta) \vee ((\gamma \vee \beta) \wedge (\delta \vee \gamma \vee \beta))$
5. $((\neg\beta \wedge \neg\delta) \vee (\gamma \vee \beta)) \wedge ((\neg\beta \wedge \neg\delta) \vee (\delta \vee \gamma \vee \beta))$
6. $(\neg\beta \vee \gamma \vee \beta) \wedge (\neg\delta \vee \gamma \vee \beta) \wedge (\neg\beta \vee \delta \vee \gamma \vee \beta) \wedge (\neg\delta \vee \delta \vee \gamma \vee \beta)$

În continuare vom aplica algoritmul bazat pe principiul rezoluţiei următoarei reprezentări clauzale:

$$S(\alpha) = \{\neg\beta \vee \gamma \vee \beta,\ \neg\delta \vee \gamma \vee \beta,\ \neg\beta \vee \delta \vee \gamma \vee \beta,\ \neg\delta \vee \delta \vee \gamma \vee \beta\}$$

Apelul EliminaTautologii($S(\alpha)$) duce la

$$S(\alpha) = \{\neg\delta \vee \gamma \vee \beta\}$$

Iniţializări: $\gamma \leftarrow S(\alpha)$, $sw \leftarrow false$;

Iteraţia 1: $\lambda = \neg\delta$ *literal pur*
$\gamma \leftarrow NEG_b(\gamma) = \emptyset$

Iteraţia 2: $\lambda = \emptyset \Rightarrow$ *write "**validabilă** ", sw* $\leftarrow$ *true*
$\Rightarrow$ *STOP.*

Exemplul 4.5.11 *Să se determine forma normal conjunctivă (CNF) pentru formula dată şi să se aplice algoritmul bazat pe rezoluţie pentru formula*

$$\alpha = ((a \to \neg b) \leftrightarrow (\neg c \to d))$$

Soluţie

Forma normal conjunctivă ataşată formulei date este:

1. $((a \to \neg b) \to (\neg c \to d)) \wedge ((\neg c \to d) \to (a \to \neg b))$
2. $(\neg(\neg a \vee \neg b) \vee (c \vee d)) \wedge (\neg(c \vee d) \vee (\neg a \vee \neg b))$
3. $((a \wedge b) \vee (c \vee d)) \wedge ((\neg c \wedge \neg d) \vee (\neg a \vee \neg b))$
4. $(a \vee c \vee d) \wedge (b \vee c \vee d) \wedge (\neg c \vee \neg a \vee \neg b) \wedge (\neg d \vee \neg a \vee \neg b)$

În cele ce urmează vom descrie algoritmul bazat pe principiul rezoluţiei având drept reprezentare clauzală următoarea mulţime:

$$S(\alpha) = \{a \vee c \vee d,\ b \vee c \vee d,\ \neg c \vee \neg a \vee \neg b,\ \neg d \vee \neg a \vee \neg b\}$$

Iniţializări: $\gamma \leftarrow S(\alpha)$, $sw \leftarrow false$;

Iteraţia 1: *Nu există clauze unitare şi nici literali puri.*
alege literal: $\lambda = c$
$\gamma \leftarrow REZ_c(\gamma) = \{a \vee d \vee \neg a \vee \neg b,\ b \vee d \vee \neg a \vee \neg b\}$
apelul EliminaTautologii(γ) duce la: $\gamma = \emptyset$

Iteraţia 2: $\lambda = \emptyset \Rightarrow$ *write "**validabilă** ", sw* $\leftarrow$ *true*
$\Rightarrow$ *STOP.*

Exemplul 4.5.12 *Să se construiască arborii semantici pentru formulele de mai jos:*
a) $(\neg(a \vee b) \wedge (\neg(a \leftrightarrow b)))$

$$
\begin{array}{ccc}
 & (\neg(a \vee b) \wedge (\neg(a \leftrightarrow b))) & \\
 & \downarrow & \\
 & \{(\neg(a \vee b), \neg(a \leftrightarrow b))\} & \\
 & \downarrow & \\
 & \{(\neg a), (\neg b), \neg(a \leftrightarrow b))\} & \\
 & \swarrow \searrow & \\
\{(\neg a), (\neg b), \neg(a \to b)\} & & \{(\neg a), (\neg b), \neg(b \to a)\} \\
\downarrow & & \downarrow \\
\{(\neg a), (\neg b), a, (\neg b)\} & & \{(\neg a), (\neg b), b, (\neg a)\} \\
\downarrow & & \downarrow \\
\times & & \times
\end{array}
$$

b) $(a \wedge ((\neg b) \vee (\neg a)))$

$$
\begin{array}{ccc}
 & (a \wedge ((\neg b) \vee (\neg a))) & \\
 & \downarrow & \\
 & \{a, ((\neg b) \vee (\neg a))\} & \\
 & \swarrow \searrow & \\
\{a, (\neg b)\} & & \{a, (\neg a)\} \\
\downarrow & & \downarrow \\
\bigcirc & & \times
\end{array}
$$

c) $(\neg(a \to b) \vee (\neg(b \wedge a)))$

$$
\begin{array}{cccc}
 & (\neg(a \to b) \vee (\neg(b \wedge a))) & & \\
 & \swarrow \searrow & & \\
\{\neg(a \to b)\} & & \multicolumn{2}{c}{\{\neg(b \wedge a)\}} \\
\downarrow & & \multicolumn{2}{c}{\swarrow \searrow} \\
\{a, (\neg b)\} & & \{(\neg b)\} & \{(\neg a)\} \\
\downarrow & & \downarrow & \downarrow \\
\bigcirc & & \bigcirc & \bigcirc
\end{array}
$$

4.6 Exerciţii

Exerciţiul 4.6.1 *Să se stabilească forma normală conjunctivă (CNF) pentru expresiile logice:*
1) $(a \to b) \to c$;
2) $(a \to b) \vee (b \to a)$;
3) $(\neg a \to (a \to b))$;
4) $(a \to (b \to c)) \to (a \to (c \to b))$;
5) $(a \to b) \to ((b \to c) \to (a \to c))$;

Exerciţiul 4.6.2 *Reduceţi la CNF următoarele formule:*
1. $p \to (q \wedge r)$
2. $(p \vee q) \to r$

3. $\neg(\neg p \vee q) \vee (r \rightarrow \neg s)$
4. $\neg((p \rightarrow (q \rightarrow r))) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$
5. $\neg(((a \rightarrow b) \rightarrow a) \rightarrow a)$
6. $\neg(a \vee (a \rightarrow b))$
7. $((x \rightarrow y) \rightarrow (z \rightarrow \neg x)) \rightarrow (\neg y \rightarrow \neg z)$
8. $((((x \rightarrow y) \rightarrow \neg x) \rightarrow \neg y) \rightarrow \neg z) \rightarrow z$
9. $(x \rightarrow (y \rightarrow z)) \rightarrow ((x \rightarrow \neg z) \rightarrow (x \rightarrow \neg y))$
10. $(\neg x \rightarrow \neg y)\neg((y \wedge z) \rightarrow (x \wedge z))$
11. $((x \rightarrow y) \rightarrow \neg x) \rightarrow (x \rightarrow (y \wedge x))$
12. $(z \rightarrow x) \rightarrow (\neg(y \vee z) \rightarrow x)$
13. $\neg((x \wedge y) \rightarrow x) \vee (x \wedge (y \vee z))$
14. $\neg(x \wedge (y \vee z)) \rightarrow \neg((x \wedge y) \vee z)$.

Exerciţiul 4.6.3 *Fie următoarea formulă* $\alpha \in FORM$ *cu* $p_1, ..., p_6 \in V$*:*

$$\alpha = \neg((p_1 \vee (p_2 \rightarrow p_3)) \wedge \neg(p_4 \vee p_5)) \vee p_6.$$

Să se scrie forma normală conjunctivă expresiei date.

Exerciţiul 4.6.4 *Să se convertească următoarele formule în CNF:*
1) $f_1 : (a \wedge b) \leftrightarrow (c \rightarrow (a \rightarrow \neg b))$;
2) $f_2 : (\neg a \rightarrow (\neg b \rightarrow c) \leftrightarrow d)$;
3) $f_3 : a \rightarrow (b \rightarrow (c \vee \neg b))$;
4) $f_4 : (\neg a \rightarrow \neg b) \rightarrow (c \rightarrow a)$;
5) $f_5 : (\neg b \rightarrow a) \rightarrow (\neg a \rightarrow c)$;
6) $f_6 : (a \vee b) \rightarrow ((c \vee d) \rightarrow (\neg d \rightarrow b))$;
7) $f_7 : (b \rightarrow \neg a) \rightarrow ((\neg c \vee a) \rightarrow b)$;
8) $f_8 : \neg(\neg(a \vee \neg b) \wedge \neg(a \vee b))$;
9) $f_9 : (p \vee (q \wedge \neg a)) \wedge (\neg p \vee a \vee \neg b)$;
10) $f_{10} : (a \wedge \neg\neg b) \vee (\neg c \wedge d)$.

Exerciţiul 4.6.5 *Se consideră următoarele clauze:*

$$\{\neg q \vee r,\ p \vee r \vee \neg p,\ p \vee \neg r,\ p \vee q \vee r,\ r \vee \neg p, p \vee \neg r \vee r,\ p \vee \neg r \vee \neg p,\ \neg q\}.$$

a) Identificaţi tautologiile, clauzele unitare şi literalii puri.
b) Folosind subpunctul anterior, aplicaţi algoritmul lui Davis-Putnam.

Exerciţiul 4.6.6 *Să se calculeze mulţimile* α_λ^+, α_λ^-, α_λ^0, $POS_\lambda(\alpha)$, $NEG_\lambda(\alpha)$ *pentru următoarele reprezentări clauzele:*

a) $S(\alpha) = \{a \vee b \vee \neg c,\ \neg b \vee d \vee \neg a,\ a \vee c \vee \neg b,\ \neg a \vee \neg c \vee b\}$ *şi* $\lambda_1 = a$*, respectiv* $\lambda_2 = \neg d$.

b) $S(\alpha) = \{\neg a \vee b \vee c,\ \neg b \vee d \vee \neg e \vee a,\ \neg a \vee \neg c \vee d \vee e,\ b \vee c \vee a \vee e\}$ *şi* $\lambda_1 = \neg b$*, respectiv* $\lambda_2 = \neg e$.

Exerciţiul 4.6.7 *Să se aplice algoritmul Davis-Putnam următoarelor formule:*
1) $(\neg p \vee q) \wedge (\neg q) \wedge p$
2) $(p \vee q) \wedge (r \vee q) \wedge (\neg r) \wedge (\neg q)$
3) $p \wedge q \wedge r$
4) $(p \vee q) \wedge (\neg p \vee q) \wedge r$
5) $(p \vee q) \wedge (\neg q)$
6) $(p \vee q) \wedge (\neg p \vee q) \wedge (\neg r \vee \neg q) \wedge (r \vee \neg q)$.

Exerciţiul 4.6.8 *Aplicând algortimul lui Davis-Putnam, demonstraţi că următoarea formulă este validabilă:*

$$(((q \to p) \land (p \to q)) \to (\neg q \land \neg r)) \lor (((r \to p) \land (q \to s)) \to ((p \to r) \to (r \land s))).$$

Exerciţiul 4.6.9 *Să se arate că* $\alpha = (p \lor q \lor \neg r) \land (p \lor \neg q) \land \neg p \land r \land u$ *este invalidabilă.*

Exerciţiul 4.6.10 *Să se arate că* $\alpha = (p \lor q) \land \neg p \land (\neg p \lor q \lor \neg r)$ *este realizabilă.*

Exerciţiul 4.6.11 *Să se arate că* $\alpha = (p \lor \neg q) \land (\neg p \lor q) \land (q \lor \neg r) \land (\neg q \lor \neg r)$ *este realizabilă.*

Exerciţiul 4.6.12 *Se consideră următoarele clauzele:*

$$C_1 : \neg a \lor q$$
$$C_2 : \neg b \lor c \lor \neg a$$
$$C_3 : q \lor b \lor c$$
$$C_4 : d \lor q \lor \neg b.$$

Determinaţi următoarele rezolvente:

$R_1 :\ rez_a(C_1, C_2)$

$R_2 :\ rez_{\neg a}(R_1, C_1)$

$R_3 :\ rez_b(C_2, C_4)$

$R_4 :\ rez_q(R_2, C_4)$

$R_5 :\ rez_q(C_1, C_3)$

$R_6 :\ rez_b(R_5, R_3)$

$R_7 :\ rez_q(R_6, C_1)$

Exerciţiul 4.6.13 *Aplicând procedura DemRez, demonstraţi că următoarea reprezentare clauzală este invalidabilă:*

$$S = \{p \lor q \lor r,\ \neg p \lor r,\ \neg q,\ \neg r\}.$$

Exerciţiul 4.6.14 *Aplicaţi algoritmul DemRez următoarei reprezentări clauzale:*

$$S = \{p \lor q,\ \neg q \lor r,\ \neg p \lor q,\ \neg r\}.$$

Exerciţiul 4.6.15 *Să se determine forma normal conjunctivă (CNF) şi să se aplice algoritmul Davis-Putnam pentru formula*

$$\theta = (\neg(\alpha \land \beta)) \leftrightarrow (\neg \gamma \to \eta).$$

Exerciţiul 4.6.16 *Să se determine forma normal conjunctivă (CNF) şi să se aplice algoritmul bazat pe rezoluţie pentru formula*

$$\theta = (\neg(\alpha \land \beta)) \leftrightarrow (\neg \gamma \to \eta).$$

Exerciţiul 4.6.17 *Să se calculeze mulţimile* α_λ^+, α_λ^-, α_λ^0, $POS_\lambda(\alpha)$, $NEG_\lambda(\alpha)$ *pentru*

a) $S(\alpha) = \{\beta \vee \omega \vee \neg\theta,\ \neg\omega \vee \gamma \vee \neg\beta,\ \beta \vee \theta \vee \neg\omega,\ \neg\beta \vee \neg\theta \vee \omega\}$ *şi* $\lambda_1 = \beta$, *respectiv* $\lambda_2 = \neg\gamma$

b) $S(\alpha) = \{\neg\gamma \vee \theta \vee \psi,\ \neg\theta \vee \beta \vee \neg\delta \vee \gamma,\ \neg\gamma \vee \neg\psi \vee \beta \vee \delta,\ \theta \vee \psi \vee \gamma \vee \delta\}$ *şi* $\lambda_1 = \neg\theta$, *respectiv* $\lambda_2 = \neg\delta$.

Exerciţiul 4.6.18 *Să se aplice algoritmul Davis-Putnam pentru reprezentările clauzale* $S(\alpha)$ *considerate la exerciţiul anterior.*

Exerciţiul 4.6.19 *Să se construiască arborii semantici pentru următoarele formule:*

1) $(a \wedge b) \leftrightarrow (a \to (a \to \neg b))$;
2) $(\neg b \to (\neg a \to b) \leftrightarrow b)$;
3) $a \to (b \to (a \vee \neg b))$;
4) $(\neg a \to \neg b) \to \neg(a \to b)$;
5) $(\neg b \to a) \to (\neg a \to c)$;
6) $(a \vee b) \to ((a \vee b) \to (\neg b \to a))$;
7) $(b \to \neg a) \to ((\neg a \vee \neg b) \to b)$;
8) $\neg(\neg(a \vee \neg b) \wedge \neg(a \vee b))$;
9) $(p \vee (q \wedge \neg p)) \wedge (\neg p \vee q \vee \neg p)$;
10) $(a \wedge \neg b) \vee (a \to b)$.

Modele de exerciţii

1.1 Model 1

Exerciţiul 1.1.1 *Se consideră formula*

$$\alpha = ((\neg a \vee (b \wedge \neg c)) \leftrightarrow (a \vee (\neg b \rightarrow \neg(c \wedge a))))$$

şi substituţia $\sigma = \{(x \vee \neg m)|\alpha,\ (d \wedge \neg t)|a,\ (q \vee p)|m,\ a|q\}$.

Să se determine: secvenţa generativă formule (SGF) pentru formula α*; tabelul de adevăr pentru formula* α*; arborele de structură pentru formula* α*;* $\alpha\sigma$*- rezultatul aplicării substituţiei* σ *pentru formula* α *şi arborele de structură asociat lui* $\alpha\sigma$*.*

Exerciţiul 1.1.2 *Se consideră formula* $\alpha = (b \vee (\neg a)) \rightarrow ((\neg b \rightarrow \neg a) \vee (a \rightarrow b))$.

a) *Să se verifice validabilitatea formulei* α *prin aplicarea metodei arborilor semantici.*

b) *Să se determine rezultatul aplicării funcţiei de interpretare* $I(\alpha)$ *asupra formulei* α.

Exerciţiul 1.1.3 a) *Să se verifice dacă următorul secvent este demonstrabil:*

$$S = \{(\alpha \vee (\neg\beta)),\ (\beta \vee (\gamma \wedge \theta))\} \Rightarrow \{\neg\alpha \rightarrow (\theta \wedge \gamma)\}.$$

b) *Să se calculeze mulţimile* α_λ^+, α_λ^-, α_λ^0, $POS_\lambda(\alpha)$, $NEG_\lambda(\alpha)$, $REZ_\lambda(\alpha)$ *unde* $\lambda = \beta$, *respectiv* $\lambda = \neg\delta$, *iar*

$$S(\alpha) = \{\neg\gamma \vee \beta \vee \neg\eta,\ \neg\beta \vee \delta \vee \neg\gamma,\ \neg\delta,\ \beta,\ \theta \vee \beta,\ \delta \vee \beta \vee \neg\theta,\ \gamma \vee \eta \vee \neg\delta\}.$$

Exerciţiul 1.1.4 *Să se determine forma normală conjunctivă (CNF) şi să se aplice algoritmul bazat pe rezoluţie pentru formula* $\alpha = ((b \rightarrow (\neg a)) \leftrightarrow (\neg c \rightarrow d))$.

1.2 Model 2

Exerciţiul 1.2.1 *Se consideră formula*

$$\alpha = (\neg(a \wedge (\neg b)) \vee (\neg a \rightarrow c))) \rightarrow (\neg(\neg a \vee b) \rightarrow (c \vee a))$$

şi substituţia $\sigma = \{(x \vee \neg m)|\alpha,\ (m \wedge n)|a,\ (q \vee p)|m,\ a|q\}$.

Să se determine: secvenţa generativă formule (SGF) pentru formula α*; tabelul de adevăr pentru formula* α*; arborele de structură pentru formula* α*;* $\alpha\sigma$*- rezultatul aplicării substituţiei* σ *pentru formula* α *şi arborele de structură asociat lui* $\alpha\sigma$*.*

Exerciţiul 1.2.2 *Se consideră formula* $\alpha = (\neg(a \wedge (\neg b))) \to (\neg(a \to b) \to (\neg b \vee a))$.

a) Să se verifice validabilitatea formulei α *prin aplicarea metodei arborilor semantici.*

b) Să se determine rezultatul aplicării funcţiei de interpretare $I(\alpha)$ *asupra formulei* α.

Exerciţiul 1.2.3 *a) Să se verifice dacă următorul secvent este demonstrabil:*

$$S = \{(a \vee (b \to c)), (a \to (\neg c))\} \Rightarrow \{\neg(d \vee (\neg b)) \to (\neg c)\}.$$

b) Să se calculeze mulţimile α_λ^+, α_λ^-, α_λ^0, $POS_\lambda(\alpha)$, $NEG_\lambda(\alpha)$, $REZ_\lambda(\alpha)$ *unde* $\lambda = \eta$, *respectiv* $\lambda = \neg\theta$, *iar*

$$S(\alpha) = \{\neg\gamma \vee \beta \vee \neg\delta,\ \neg\beta \vee \eta \vee \neg\gamma,\ \neg\theta,\ \beta,\ \theta \vee \beta \vee \neg\eta,\ \delta \vee \beta \vee \neg\theta,\ \gamma \vee \eta \vee \neg\delta\}.$$

Exerciţiul 1.2.4 *Să se determine forma normală conjunctivă (CNF) şi să se aplice algoritmul Davis-Putnam pentru formula* $\alpha = ((\neg a \vee b)) \leftrightarrow (d \to c))$.

1.3 Model 3

Exerciţiul 1.3.1 *Se consideră formula*

$$\theta = ((\alpha \vee \beta) \to (\neg\alpha \vee (\gamma \wedge \delta)) \to (\neg\beta \to (\gamma \vee \delta)))$$

şi substituţia $\sigma = \{(a \vee b)|\alpha,\ (\alpha \vee \beta)|a,\ (p \wedge q)|m,\ \delta|b\}$.

Să se determine: secvenţa generativă formule (SGF) pentru formula α*; arborele de structură pentru formula* α*;* $\alpha\sigma$*- rezultatul aplicării substituţiei* σ *pentru formula* α*, arborele de structură asociat lui* $\alpha\sigma$*, funcţia de adâncime* $h(\theta)$ *pentru formula dată dar şi pentru formula obţinută în urma aplicării substituţiei* $\theta\sigma$.

Exerciţiul 1.3.2 *Folosind axiomelele, să se identifice demonstraţii formale pentru*

a) $(\neg(a \wedge \neg b) \to b) \to ((\neg c \vee d) \to ((a \wedge \neg b) \vee b))$;

b) $((\neg a \wedge c) \to (\neg b)) \to (b \to (\neg c \vee a))$;

c) $((\neg b \wedge a) \leftrightarrow c) \to ((\neg a \vee b) \to c)$.

Exerciţiul 1.3.3 *Verificaţi fără tabele de adevăr dacă următoarele formule sunt echivalente:*

$$a \vee (b \to c) \text{ şi } b \to (a \vee c).$$

Exerciţiul 1.3.4 *Să se aplice algoritmul Davis-Putnam şi algoritmul bazat pe rezoluţie următoarei reprezentări clauzale*

$$S = \{p \vee \neg q,\ \neg p \vee q,\ q \vee \neg r,\ s,\ \neg s \vee \neg q \vee \neg r,\ s \vee r\}.$$