1 Sintaxa limbajului

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

2 Sistemul deducţiei naturale (Gentzen)

Definiţia 1.6.1. Se numeşte secvent o pereche de mulţimi de formule (H, Γ) în care apar numai conective din mulţimea $L \setminus \{\leftrightarrow\}$. Prima componentă a perechii se numeşte *antecedent*, respectiv cea de a doua componentă se numeşte *succedent*. Convenţional secventul (H, Γ) este reprezentat $H \Longrightarrow \Gamma$.

Definiţia 1.6.2. Secventul $H \Longrightarrow \Gamma$ este *secvent axiomă*, dacă

$H \cap \Gamma \neq \emptyset$.

Definiţia 1.6.3. 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 1.6.4. Regulile de inferenţă Gentzen sunt:

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

$G2.$ $\dfrac{H \cup \{\alpha, \beta\} \Longrightarrow \Gamma}{H \cup \{(\alpha \wedge \beta)\} \Longrightarrow \Gamma}$, (*regula conjuncţie stânga*)

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

$G4.$ $\dfrac{H \cup \{\beta\} \Longrightarrow \Gamma \ , \ H \Longrightarrow \Gamma \cup \{\alpha\}}{H \cup \{(\alpha \rightarrow \beta)\} \Longrightarrow \Gamma}$, (*regula implicaţie stânga*)

$G5.$ $\dfrac{H \cup \{\alpha\} \Longrightarrow \Gamma}{H \Longrightarrow \Gamma \cup \{(\neg\alpha)\}}$, (*regula negaţie dreapta*)

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

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

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

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

Observaţie Regulile Gentzen definesc modalităţile de "descompunere" în formule de adâncime mai mică a formulelor selectate din mulţimile

antecedent/succedent. Se observă, că 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 1.6.1. 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 1.6.4 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,

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

$\iota\iota$) 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 1.6.5 Arborele de deducţie T este un *arbore de demonstraţie* pentru secventul etichetă a vârfului rădăcină, dacă orice vârf terminal are ca etichetă un secvent axiomă.

Definiţia 1.6.6 Arborele de deducţie T este un *arbore încheiat*, dacă toate vârfurile terminale au etichete secvenţi încheiaţi.

Definiţia 1.6.7 Arborele 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 1.6.8 Secventul S este demonstrabil, notat $\vdash S$, dacă există T *arbore de demonstraţie* cu S etichetă a rădăcinii.

Exemplul 1.6.2 Fie secventul

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

Arborele T reprezentat în *Figura* 1.6.1. 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 $\vdash S$.

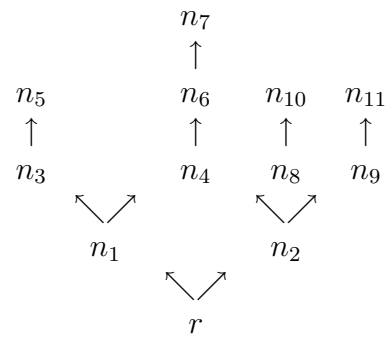


Figura 1.6.1.

Secvenţii etichete ale vârfurilor arborelui T sunt : $\varphi(r) = S$,
$\varphi(n_1) = \{\beta, ((\neg\alpha) \to \gamma)\} \Longrightarrow \{(\beta \vee \gamma)\}$,
$\varphi(n_2) = \{((\neg\alpha) \to \gamma)\} \Longrightarrow \{\alpha, (\beta \vee \gamma)\}$,
$\varphi(n_3) = \{\beta, \gamma\} \Longrightarrow \{(\beta \vee \gamma)\}$,
$\varphi(n_4) = \{\beta\} \Longrightarrow \{(\neg\alpha), (\beta \vee \gamma)\}$,
$\varphi(n_5) = \{\beta, \gamma\} \Longrightarrow \{\beta, \gamma\}$,
$\varphi(n_6) = \{\alpha, \beta\} \Longrightarrow \{(\beta \vee \gamma)\}$,
$\varphi(n_7) = \{\alpha, \beta\} \Longrightarrow \{\beta, \gamma\}$,
$\varphi(n_8) = \{\gamma\} \Longrightarrow \{\alpha, (\beta \vee \gamma)\}$,
$\varphi(n_9) = \emptyset \Longrightarrow \{(\neg\alpha), \alpha, (\beta \vee \gamma)\}$,
$\varphi(n_{10}) = \{\gamma\} \Longrightarrow \{\alpha, \beta, \gamma\}$,
$\varphi(n_{11}) = \{\alpha\} \Longrightarrow \{\alpha, (\beta \vee \gamma)\}$.

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:

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

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

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

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* 1.6.2; 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 $G4$;

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

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

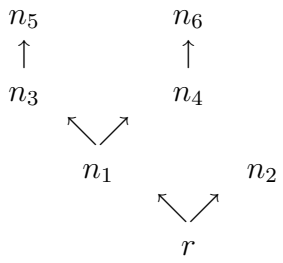
3. $\partial T = \{n_3, n_4, n_2\}$; $n = n_3$; formula selectată $(\beta \vee \gamma)$; regula $G7$;

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

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

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

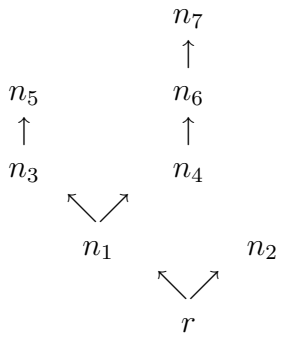
T :



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

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

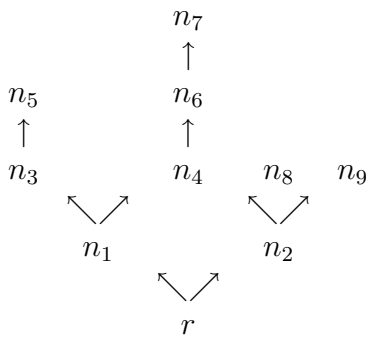
T :



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

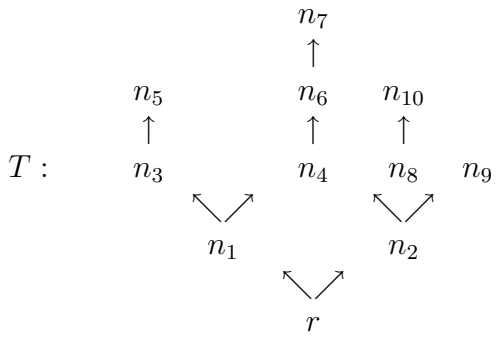
6. $\partial T = \{n_5, n_7, n_2\}$; $n = n_2$; formula selectată $((\neg\alpha) \to \gamma)$; regula selectată $G4$;

T :



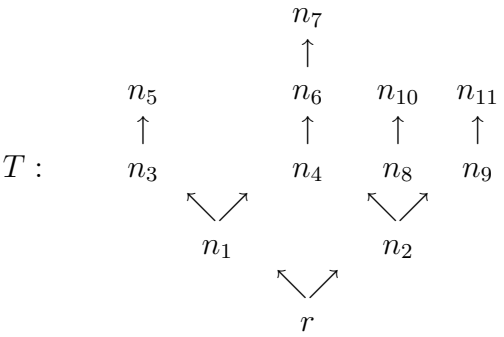
$\varphi(n_8) = \{\gamma\} \Longrightarrow \{\alpha, (\beta \vee \gamma)\}$, $\varphi(n_9) = \emptyset \Longrightarrow \{(\neg\alpha), \alpha, (\beta \vee \gamma)\}$.

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



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

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



$\varphi(n_{11}) = \{\alpha\} \Longrightarrow \{\alpha, (\beta \vee \gamma)\}$ 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 procesul de expandare se încheie cu decizia " S este secvent demonstrabil".

3 Semantica limbajului calculului cu propoziţii

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

$$\begin{array}{lcll} \neg & : & \{T,F\} \to \{T,F\} & (\textit{negaţie}) \\ \wedge & : & \{T,F\}^2 \to \{T,F\} & (\textit{conjuncţie}) \\ \vee & : & \{T,F\}^2 \to \{T,F\} & (\textit{disjuncţie}) \\ \to & : & \{T,F\}^2 \to \{T,F\} & (\textit{implicacţie}) \\ \leftrightarrow & : & \{T,F\}^2 \to \{T,F\} & (\textit{echivalenţă}) \end{array}$$

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

,

$\rightarrow$	T	F
T	T	F
F	T	T

,

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

.

Numim funcţie de adevăr orice funcţie $h : V \rightarrow \{T, F\}$.

Lema 1.7.1. Fie h o funcţie de adevăr. Atunci există şi este unică $I(h) : FORM \rightarrow \{T, F\}$, astfel încât următoarele cerinţe să fie îndeplinite:

1. pentru orice $\alpha \in V$, $I(h)(\alpha) = h(\alpha)$;
2. pentru orice $\alpha, \beta \in FORM$,
   - 2.1 $I(h)((\neg\alpha)) = \neg I(h)(\alpha)$;
   - 2.2 $I(h)((\alpha \wedge \beta)) = I(h)(\alpha) \wedge I(h)(\beta)$;
   - 2.3 $I(h)((\alpha \vee \beta)) = I(h)(\alpha) \vee I(h)(\beta)$;
   - 2.4 $I(h)((\alpha \rightarrow \beta)) = I(h)(\alpha) \rightarrow I(h)(\beta)$;
   - 2.5 $I(h)((\alpha \leftrightarrow \beta)) = I(h)(\alpha) \leftrightarrow I(h)(\beta)$.

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 $1, 2.1 - 2.5$. 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* 1.7.1. 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.

Observaţie Dacă $I : FORM \to \{T, F\}$ verifică cerinţele din enunţul

Lemei 1.7.1, atunci există şi este unică funcţia de adevăr h, astfel încât

$I = I(h)$. Într-adevăr, utilizând rezultatul stabilit de *Lema* 1.7.1, pentru restricţia

$h = I|_{FORM}$, obţinem concluzia $I = I(h)$.

Definiţia 1.7.1. Formula α este validabilă, dacă $\aleph(\alpha) \neq \emptyset$. Dacă $\aleph(\alpha) = \emptyset$, atunci α este invalidabilă. Termenii, frecvent utilizaţi, sinonimi cu invalidabil, sunt insatisfiabil, nerealizabil, respectiv logic fals.

Definiţia 1.7.2. Formula α este tautologie, dacă $\aleph(\alpha) = \Im$.

Definiţia 1.7.3. 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).

Observaţie Din *Definiţia* 1.7.3 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 1.7.2. Fie $H \subset FORM$ finită, $H = \{\alpha_1, ..., \alpha_n\}$.

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 $I((\alpha_1 \wedge \alpha_2)) = T$, 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 $H = \{\alpha_1, ..., \alpha_k, \alpha_{k+1}\}$.

Utilizând observaţia precedentă şi ipoteza inductivă obţinem

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

Observaţie Deoarece pentru orice $\alpha \in FORM$, şi $I \in \Im$,

$I((\neg\alpha)) = \neg I(\alpha)$, rezultă $I(\alpha) = T$, dacă şi numai dacă

$I((\neg\alpha)) = F$, deci $\aleph((\neg\alpha)) = \Im \setminus \aleph(\alpha)$.

Lema 1.7.3. Dacă $\gamma_1 = \beta_1$, $\gamma_k = (\gamma_{k-1} \vee \beta_k)$, $2 \leq k \leq n$, unde

$\{\beta_1, ..., \beta_n\} \subset FORM$, atunci $\aleph(\gamma_n) = \bigcup_{k=1}^{n} \aleph(\beta_k)$.

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
$\{\beta_1, ..., \beta_k\} \subset FORM$, $2 \leq k \leq n$.

Fie $\{\beta_1, ..., \beta_{n+1}\} \subset FORM$; $\gamma_{n+1} = (\gamma_n \vee \beta_{n+1})$. Utilizând ipoteza inductivă, obţinem

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

Definiţia 1.7.4. Spunem că formulele α, β sunt semantic echivalente (notat $\alpha \equiv \beta$), dacă $\aleph(\alpha) = \aleph(\beta)$.

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

Lema 1.7.4. 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.

Observaţie 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 1.7.5 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 1.7.1 (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 $\overline{\alpha_2}$:

$$\begin{aligned}
I\left(\overline{\alpha_2}\sigma\right) &= I\left(\left(\left(\alpha \to \left(\alpha \to \beta\right)\right) \to \left(\alpha \to \beta\right)\right)\right) \\
&= \neg I\left(\left(\alpha \to \left(\alpha \to \beta\right)\right)\right) \vee I\left(\alpha \to \beta\right) \\
&= \neg\left(\neg I\left(\alpha\right) \vee \neg I\left(\alpha\right) \vee I\left(\beta\right)\right) \vee \left(\neg I\left(\alpha\right) \vee I\left(\beta\right)\right) \\
&= \neg\left(\neg I\left(\alpha\right) \vee I\left(\beta\right)\right) \vee \left(\neg I\left(\alpha\right) \vee I\left(\beta\right)\right) \\
&= \left(I\left(\alpha\right) \wedge \neg I\left(\beta\right)\right) \vee \left(\neg I\left(\alpha\right) \vee I\left(\beta\right)\right) \\
&= \left(I\left(\alpha\right) \vee \neg I\left(\alpha\right) \vee I\left(\beta\right)\right) \wedge \left(\neg I\left(\beta\right) \vee \neg I\left(\alpha\right) \vee I\left(\beta\right)\right) = T.
\end{aligned}$$

3. Axioma $\overline{\alpha_3}$:

$$\begin{aligned}
I\left(\overline{\alpha_3}\sigma\right) &= I\left(\left(\left(\alpha \to \beta\right) \to \left(\left(\beta \to \gamma\right) \to \left(\alpha \to \gamma\right)\right)\right)\right) \\
&= \neg I\left(\left(\alpha \to \beta\right)\right) \vee \left(\neg I\left(\left(\beta \to \gamma\right)\right) \vee I\left(\left(\alpha \to \gamma\right)\right)\right) \\
&= \neg\left(\neg I\left(\alpha\right) \vee I\left(\beta\right)\right) \vee \left(\neg\left(\neg I\left(\beta\right) \vee I\left(\gamma\right)\right) \vee \left(\neg I\left(\alpha\right) \vee I\left(\gamma\right)\right)\right) \\
&= \left(I\left(\alpha\right) \wedge \neg I\left(\beta\right)\right) \vee \left(\left(I\left(\beta\right) \wedge \neg I\left(\gamma\right)\right) \vee \left(\neg I\left(\alpha\right) \vee I\left(\gamma\right)\right)\right)
\end{aligned}$$

$$\begin{aligned}
&= \left(I\left(\alpha\right) \wedge \neg I\left(\beta\right)\right) \vee \left(\left(I\left(\beta\right) \vee \neg I\left(\alpha\right) \vee I\left(\gamma\right)\right) \wedge \right. \\
&\qquad \left. \left(\neg I\left(\gamma\right) \vee \neg I\left(\alpha\right) \vee I\left(\gamma\right)\right)\right) \\
&= \left(I\left(\alpha\right) \wedge \neg I\left(\beta\right)\right) \vee \left(\left(I\left(\beta\right) \vee \neg I\left(\alpha\right) \vee I\left(\gamma\right)\right) \wedge T\right) \\
&= \left(I\left(\alpha\right) \wedge \neg I\left(\beta\right)\right) \vee \left(I\left(\beta\right) \vee \neg I\left(\alpha\right) \vee I\left(\gamma\right)\right) \\
&= \left(I\left(\alpha\right) \vee I\left(\beta\right) \vee \neg I\left(\alpha\right) \vee I\left(\gamma\right)\right) \wedge \\
&\qquad \left(\neg I\left(\beta\right) \vee I\left(\beta\right) \vee \neg I\left(\alpha\right) \vee I\left(\gamma\right)\right) \\
&= T \wedge T = T
\end{aligned}$$

4. Axioma $\overline{\alpha_4}$:

$$\begin{aligned}
I\left(\overline{\alpha_4}\sigma\right) &= I\left(\left(\left(\alpha \leftrightarrow \beta\right) \to \left(\alpha \to \beta\right)\right)\right) \\
&= \neg I\left(\left(\alpha \leftrightarrow \beta\right)\right) \vee I\left(\left(\alpha \to \beta\right)\right) \\
&= \neg\left(I\left(\left(\alpha \to \beta\right)\right) \wedge I\left(\left(\beta \to \alpha\right)\right)\right) \vee I\left(\left(\alpha \to \beta\right)\right) \\
&= \neg I\left(\left(\alpha \to \beta\right)\right) \vee \neg I\left(\left(\beta \to \alpha\right)\right) \vee I\left(\left(\alpha \to \beta\right)\right) = T
\end{aligned}$$

5. Axioma $\overline{\alpha_5}$:

$$\begin{aligned}
I\left(\overline{\alpha_5}\sigma\right) &= I\left(\left(\left(\alpha \leftrightarrow \beta\right) \to \left(\beta \to \alpha\right)\right)\right) \\
&= \neg I\left(\left(\alpha \leftrightarrow \beta\right)\right) \vee I\left(\left(\beta \to \alpha\right)\right) \\
&= \neg\left(I\left(\left(\alpha \to \beta\right)\right) \wedge I\left(\left(\beta \to \alpha\right)\right)\right) \vee I\left(\left(\beta \to \alpha\right)\right) \\
&= \neg I\left(\left(\alpha \to \beta\right)\right) \vee \neg I\left(\left(\beta \to \alpha\right)\right) \vee I\left(\left(\beta \to \alpha\right)\right) = T
\end{aligned}$$

6. Axioma $\overline{\alpha_6}$:

$$\begin{aligned}
I\left(\overline{\alpha_6}\sigma\right) &= I\left(\left(\left(\alpha \to \beta\right) \to \left(\left(\beta \to \alpha\right) \to \left(\alpha \leftrightarrow \beta\right)\right)\right)\right) \\
&= \neg I\left(\left(\alpha \to \beta\right)\right) \vee \left(\neg I\left(\left(\beta \to \alpha\right)\right) \vee I\left(\left(\alpha \leftrightarrow \beta\right)\right)\right) \\
&= \neg I\left(\left(\alpha \to \beta\right)\right) \vee \left(\neg I\left(\left(\beta \to \alpha\right)\right) \vee \right. \\
&\qquad \left. \left(I\left(\left(\alpha \to \beta\right)\right) \wedge I\left(\left(\beta \to \alpha\right)\right)\right)\right) \\
&= \neg I\left(\left(\alpha \to \beta\right)\right) \vee \left(\left(\neg I\left(\left(\beta \to \alpha\right)\right) \vee I\left(\left(\alpha \to \beta\right)\right)\right) \wedge \right. \\
&\qquad \left. \left(\neg I\left(\left(\beta \to \alpha\right)\right) \vee I\left(\left(\beta \to \alpha\right)\right)\right)\right) \\
&= \neg I\left(\left(\alpha \to \beta\right)\right) \vee \left(\left(\neg I\left(\left(\beta \to \alpha\right)\right) \vee I\left(\left(\alpha \to \beta\right)\right)\right) \wedge T\right) \\
&= \neg I\left(\left(\alpha \to \beta\right)\right) \vee \left(\neg I\left(\left(\beta \to \alpha\right)\right) \vee I\left(\left(\alpha \to \beta\right)\right)\right) = T
\end{aligned}$$

7. Axioma $\overline{\alpha_7}$:

$$\begin{aligned}
I\left(\overline{\alpha_7}\sigma\right) &= I\left(\left(\left(\left(\neg\alpha\right) \to \left(\neg\beta\right)\right) \to \left(\beta \to \alpha\right)\right)\right) \\
&= \neg I\left(\left(\left(\neg\alpha\right) \to \left(\neg\beta\right)\right)\right) \vee I\left(\left(\beta \to \alpha\right)\right) \\
&= \neg\left(\neg I\left(\left(\neg\alpha\right)\right) \vee I\left(\left(\neg\beta\right)\right)\right) \vee \left(\neg I\left(\beta\right) \vee I\left(\alpha\right)\right) \\
&= \neg\left(I\left(\alpha\right) \vee \neg I\left(\beta\right)\right) \vee \left(\neg I\left(\beta\right) \vee I\left(\alpha\right)\right) \\
&= \left(\neg I\left(\alpha\right) \wedge I\left(\beta\right)\right) \vee \left(\neg I\left(\beta\right) \vee I\left(\alpha\right)\right) \\
&= \left(\neg I\left(\alpha\right) \vee \neg I\left(\beta\right) \vee I\left(\alpha\right)\right) \wedge \left(I\left(\beta\right) \vee \neg I\left(\beta\right) \vee I\left(\alpha\right)\right) \\
&= T \wedge T = T
\end{aligned}$$

8. Axioma $\overline{\alpha_8}$:

$$\begin{aligned}
I\left(\overline{\alpha_8}\sigma\right) &= I\left(\left(\left(\alpha \vee \beta\right) \leftrightarrow \left(\left(\neg\alpha\right) \to \beta\right)\right)\right) \\
&= I\left(\left(\left(\alpha \vee \beta\right) \to \left(\left(\neg\alpha\right) \to \beta\right)\right)\right) \wedge \\
&\qquad \wedge I\left(\left(\left(\left(\neg\alpha\right) \to \beta\right) \to \left(\alpha \vee \beta\right)\right)\right) \\
&= \left(\neg I\left(\left(\alpha \vee \beta\right)\right) \vee I\left(\left(\left(\neg\alpha\right) \to \beta\right)\right)\right) \wedge \\
&\qquad \wedge \left(\neg I\left(\left(\left(\neg\alpha\right) \to \beta\right)\right) \vee I\left(\left(\alpha \vee \beta\right)\right)\right) \\
&= \left(\neg I\left(\left(\alpha \vee \beta\right)\right) \vee \left(\neg I\left(\left(\neg\alpha\right)\right) \vee I\left(\beta\right)\right)\right) \wedge \\
&\qquad \wedge \left(\neg\left(\neg I\left(\left(\neg\alpha\right)\right) \vee I\left(\beta\right)\right) \vee I\left(\left(\alpha \vee \beta\right)\right)\right) \\
&= \left(\neg I\left(\left(\alpha \vee \beta\right)\right) \vee \left(I\left(\alpha\right) \vee I\left(\beta\right)\right)\right) \wedge \\
&\qquad \wedge \left(\neg\left(I\left(\alpha\right) \vee I\left(\beta\right)\right) \vee I\left(\left(\alpha \vee \beta\right)\right)\right) \\
&= \left(\neg I\left(\left(\alpha \vee \beta\right)\right) \vee I\left(\left(\alpha \vee \beta\right)\right)\right) \wedge \\
&\qquad \wedge \left(\neg I\left(\left(\alpha \vee \beta\right)\right) \vee I\left(\left(\alpha \vee \beta\right)\right)\right) \\
&= T
\end{aligned}$$

9. Axioma $\overline{\alpha_9}$:

$$\begin{aligned} I\left(\overline{\alpha_9}\sigma\right) &= I\left(\left(\left(\alpha \wedge \beta\right) \leftrightarrow \left(\neg\left(\left(\neg\alpha\right) \vee \left(\neg\beta\right)\right)\right)\right)\right) \\ &= I\left(\left(\left(\alpha \wedge \beta\right) \rightarrow \left(\neg\left(\left(\neg\alpha\right) \vee \left(\neg\beta\right)\right)\right)\right) \wedge \right. \\ &\qquad \left. I\left(\left(\left(\neg\left(\left(\neg\alpha\right) \vee \left(\neg\beta\right)\right)\right) \rightarrow \left(\alpha \wedge \beta\right)\right)\right)\right. \\ &= \left(\neg I\left(\left(\alpha \wedge \beta\right)\right) \vee I\left(\left(\neg\left(\left(\neg\alpha\right) \vee \left(\neg\beta\right)\right)\right)\right) \wedge \right. \\ &\qquad \left. \left(\neg I\left(\left(\neg\left(\left(\neg\alpha\right) \vee \left(\neg\beta\right)\right)\right)\right) \vee I\left(\left(\alpha \wedge \beta\right)\right)\right)\right. \end{aligned}$$

$$\begin{aligned} &= \left(\neg I\left(\left(\alpha \wedge \beta\right)\right) \vee \neg\left(I\left(\left(\neg\alpha\right)\right) \vee I\left(\left(\neg\beta\right)\right)\right)\right) \wedge \\ &\qquad \left(\left(I\left(\left(\neg\alpha\right)\right) \vee I\left(\left(\neg\beta\right)\right)\right) \vee I\left(\left(\alpha \wedge \beta\right)\right)\right) \\ &= \left(\neg I\left(\left(\alpha \wedge \beta\right)\right) \vee \neg\left(\neg I\left(\alpha\right) \vee \neg I\left(\beta\right)\right)\right) \wedge \\ &\qquad \left(\left(\neg I\left(\alpha\right) \vee \neg I\left(\beta\right)\right) \vee I\left(\left(\alpha \wedge \beta\right)\right)\right) \\ &= \left(\neg I\left(\left(\alpha \wedge \beta\right)\right) \vee \neg\neg\left(I\left(\alpha\right) \wedge I\left(\beta\right)\right)\right) \wedge \\ &\qquad \left(\neg\left(I\left(\alpha\right) \wedge I\left(\beta\right)\right) \vee I\left(\left(\alpha \wedge \beta\right)\right)\right) \\ &= \left(\neg I\left(\left(\alpha \wedge \beta\right)\right) \vee I\left(\left(\alpha \wedge \beta\right)\right)\right) \wedge \\ &\qquad \left(\neg I\left(\left(\alpha \wedge \beta\right)\right) \vee I\left(\left(\alpha \wedge \beta\right)\right)\right) \\ &= T \wedge T = T \end{aligned}$$

În concluzie, orice exemplu de axiomă este tautologie.

Să observăm că pentru orice formule α, β,

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

Într-adevăr, dacă $I \in \aleph\left(\alpha\right) \cap \aleph\left(\left(\alpha \rightarrow \beta\right)\right)$, atunci $I\left(\left(\left(\alpha \rightarrow \beta\right)\right)\right) = I\left(\alpha\right) = T$, şi

$$\begin{aligned} I\left(\left(\left(\alpha \rightarrow \beta\right)\right)\right) &= I\left(\left(\left(\neg\alpha\right) \vee \beta\right)\right) = \neg I\left(\alpha\right) \vee I\left(\beta\right) \\ &= F \vee I\left(\beta\right) = I\left(\beta\right), \end{aligned}$$

deci $I \in \aleph\left(\beta\right)$.

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 $\alpha_1, ..., \alpha_n, \alpha_{n+1} = \alpha$ demonstraţie formală. Evident, pentru orice k, $1 \leq k \leq n$, $\alpha_1, ..., \alpha_k$ este demonstraţie formală, deci, conform ipotezei inductive, formulele α_k, $k = 1, ..., n$ 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.

Corolarul 1.7.1

1. $T_h \neq FORM$
2. Pentru orice $\alpha \in FORM$ cel mult una din formulele α, $(\neg\alpha)$ este teoremă.
3. Dacă $\alpha \in T_h$, atunci $(\neg\alpha) \notin T_h$.
4. Dacă α este formulă logic falsă, atunci $\aleph(\alpha) = \emptyset$.

Demonstraţie Într-adevăr, pentru orice $\alpha \in V$, $\aleph(\alpha) \neq \Im$, deci

$T_h \cap V = \emptyset$.

Pentru orice $I \in \Im$, şi orice $\alpha \in FORM$, $I(\alpha) = \neg I((\neg\alpha))$, deci cel mult una din formulele α, $(\neg\alpha)$ este demonstrabilă.

Dacă $\alpha \in T_h$, atunci pentru orice $I \in \Im$, $I((\neg\alpha)) = F$, deci

$(\neg\alpha) \notin T_h$.

În particular obţinem că dacă α este formulă logic falsă, atunci $\aleph(\alpha) = \emptyset$.

În continuare vom stabili proprietatea de completitudine a limbajului calculului cu propoziţii, reciprocă a proprietăţii stabilite de *Teorema* 1.7.1.

Definiţia 1.7.5. Mulţimea compatibilă de formule H este un sistem deductiv, dacă $T(H) = H$.

Observaţii

1. Din *Definiţia* 1.7.5. rezultă că o mulţime compatibilă de formule H este un sistem deductiv, dacă şi numai dacă H este punct fix al operatorului de deductibilitate.
2. Deoarece $T(T(H)) = T(H)$, rezultă că, dacă H este o mulţime compatibilă de formule, atunci $T(H)$ este sistem deductiv.
3. Deoarece $T_h \neq FORM$ şi $T(T_h) = T_h$, rezultă că T_h este sistem deductiv, deci mulţimea sistemelor deductive este nevidă. In plus, deoarece pentru orice $H \subset FORM, T_h \subset T(H)$, rezultă că T_h este cel mai mic punct fix al operatorului de deductibilitate în mulţimea sistemelor deductive.

În continuare notăm cu $\mathcal{D}$ mulţimea sistemelor deductive.

Lema 1.7.7. Mulţimea compatibilă de formule H este sistem deductiv, dacă şi numai dacă îndeplineşte următoarele condiţii:

ι) $T_h \subset H$,

$\iota\iota$) H este stabilă modus ponens, în sensul că, pentru orice $\alpha, \beta \in FORM$,dacă $\{\alpha, (\alpha \to \beta)\} \subset H$, atunci $\beta \in H$.

Demonstraţie Presupunem că mulţimea compatibilă H îndeplineşte condiţiile din enunţ. Demonstrăm incluziunea $T(H) \subset H$ prin inducţie asupra $H-$ secvenţei deductive.

Dacă β este o $H-$ secvenţă deductivă, atunci $\beta \in T_h \cup H = H$ adică $\beta \in H$. Presupunem că pentru orice formulă β care admite o $H-$ secvenţă deductivă de lungime mai mică sau cel mult egală cu n, rezultă $\beta \in H$. Fie $\beta_1, ..., \beta_n, \beta$ o $H-$ secvenţă deductivă de lungime $n+1$. Evident, ipoteza inductivă este aplicabilă pentru orice j, $1 \leq j \leq n$, deci $\{\beta_1, ..., \beta_n\} \subset H$. Concluzia $\beta \in H$ rezultă imediat, dacă $\beta \in T_h \cup H$. Dacă există i, j, $1 \leq i, j \leq n$ şi $\beta_j = (\beta_i \to \beta)$, atunci $\beta_i, \beta_j \in H$ deci, utilizând $(\iota\iota)$, rezultă $\beta \in H$.

Orice sistem deductiv H îndeplineşte condiţia (ι) din enunţ, deoarece $T_h \subset T(H) = H$.

Demonstrăm că H este stabilă modus ponens.

Fie $\alpha \in H$, $(\alpha \to \beta) \in H$. Evident α, $(\alpha \to \beta), \beta$ este o $H-$ secvenţă deductivă, deci $\beta \in T(H) = H$.

Observaţie Pentru orice mulţime nevidă de indici I, dacă $D_\iota \in \mathcal{D}$ pentru orice $\iota \in I$, atunci $\bigcap_{\iota \in I} D_\iota \in \mathcal{D}$.

Lema 1.7.8. Mulţimea sistemelor deductive $\mathcal{D}$ este inductiv ordonată de relaţia de incluziune.

Demonstraţie Fie $(D_\iota)_{\iota \in I}$ mulţime de sisteme deductive total ordonată de relaţia de incluziune, unde I este o mulţime nevidă de indici. Demonstrăm că $D = \bigcup_{\iota \in I} D_\iota$ este sistem deductiv. Evident, deoarece $I \neq \emptyset$, $T_h \subset D$. Dacă $\alpha \in D, (\alpha \to \beta) \in D$, fie $\iota_1, \iota_2 \in I$, astfel încât $\alpha \in D_{\iota_1}$, $(\alpha \to \beta) \in D_{\iota_2}$. Fie $\iota \in \{\iota_1, \iota_2\}$ astfel încât $D_{\iota_j} \subset D_\iota$, $j = 1, 2$.

Utilizând *Lema* 1.7.7., obţinem $\beta \in D_\iota \subset D$, deci D este stabilă modus ponens.

Dacă D este mulţime incompatibilă, atunci $D \vdash \perp$. Deoarece deductibilitatea este o proprietate de caracter finit, există mulţimea finită de formule $\{\alpha_1, ..., \alpha_n\} \subset D$, astfel încât $\{\alpha_1, ..., \alpha_n\} \vdash \perp$, deci există $\iota_j \in I$, astfel încât $\alpha_j \in D_{\iota_j}$, $j = 1, ..., n$.

Pentru $\iota \in \{\iota_1, ..., \iota_n\}$, astfel încât $D_{\iota_j} \subset D_\iota$, $j = 1, ..., n$ rezultă $D_\iota \vdash \perp$ ceea ce este evident o contradicţie. Rezultă D mulţime compatibilă deci, conform concluziei stabilite de *Lema* 1.7.7., rezultă în final $D \in \mathcal{D}$.

Corolarul 1.7.2 Mulţimea sistemelor deductive maximale $\mathcal{D}_{\max}$ este nevidă.

Concluzia este o consecinţă imediată a *Lemei Zorn* şi a rezultatului stabilit de *Lema* 1.7.8.

Teorema 1.7.2 (Teorema Lindenbaum Tarski). Orice sistem deductiv este inclus într-un sistem deductiv maximal.

Demonstraţie Fie $D \in \mathcal{D}$. Evident, mulţimea

$$\mathcal{D}(D) = \{D' \mid D' \in \mathcal{D},\ D \subset D'\}$$

este inductiv ordonată faţă de relaţia de incluziune, deci există elemente maximale în mulţimea $\mathcal{D}(D)$. Pentru $\overline{D}$ element maximal în $\mathcal{D}(D)$ demonstrăm că $\overline{D} \in \mathcal{D}_{\max}$.

Dacă $\overline{D} \notin \mathcal{D}_{\max}$, fie $\Gamma \in \mathcal{D}$, astfel încât $\overline{D} \nsubseteq \Gamma$. Evident obţinem $\Gamma \in \mathcal{D}(D)$, deci $\overline{D}$ nu este element maximal în $\mathcal{D}(D)$.
Rezultă $D \subset \overline{D} \in \mathcal{D}_{\max}$.

Definiţia 1.7.6. Sistemul deductiv D este complet, dacă pentru orice
$\alpha \in FORM$, $\alpha \in D$ sau $(\neg\alpha) \in D$.

Observaţie Din consideraţiile precedente a rezultat $T(T_h) = T_h$. Pentru orice propoziţie elementară p, $p \notin T_h$ şi $(\neg p) \notin T_h$, deci T_h este sistem deductiv, dar nu este sistem deductiv complet.

Lema 1.7.9. Orice sistem deductiv este maximal, dacă şi numai dacă este sistem deductiv complet.

Demonstraţie Fie $D \in \mathcal{D}$. Presupunem $D \in \mathcal{D}_{\max}$. Deoarece D este mulţime compatibilă, pentru orice formulă α, cel puţin una din mulţimile $D \cup \{\alpha\}$,
$D \cup \{(\neg\alpha)\}$ este compatibilă.

Dacă $D \cup \{\alpha\}$ este compatibilă, atunci $T(D \cup \{\alpha\}) \in \mathcal{D}$. Rezultă

$$\left.\begin{array}{c} D \subset D \cup \{\alpha\} \subset T(D \cup \{\alpha\}) \\ D \in \mathcal{D} \\ T(D \cup \{\alpha\}) \in \mathcal{D} \end{array}\right\} \Rightarrow D = T(D \cup \{\alpha\}),$$

deci $\alpha \in D$.

Analog, dacă $D \cup \{(\neg\alpha)\}$ este compatibilă, rezultă $(\neg\alpha) \in D$, deci D este sistem deductiv complet.

Presupunem că D este sistem deductiv complet. Dacă $D \notin \mathcal{D}_{\max}$, utilizând concluzia stabilită de *Teorema* 1.7.2, există $\Gamma \in \mathcal{D}_{\max}$, astfel încât $D \subsetneq \Gamma$. Pentru $\alpha \in \Gamma \setminus D$, deoarece D este sistem deductiv complet obţinem

$(\neg\alpha) \in D$. Obţinem $\{\alpha, (\neg\alpha)\} \subset \Gamma$, deci Γ nu este mulţime compatibilă. Rezultă că dacă D este sistem deductiv complet, atunci $D \in \mathcal{D}_{\max}$.

Lema 1.7.10. Orice sistem deductiv este intersecţia sistemelor deductive maximale în care este inclus.

Demonstraţie Fie $D \in \mathcal{D}$. Notăm

$$\mathcal{D}_{\max}(D) = \{D' \mid D' \in \mathcal{D}_{\max},\ D \subset D'\}$$

şi fie $\Gamma = \bigcap_{D' \in \mathcal{D}_{\max}} D'$. Evident, $\Gamma \in \mathcal{D}$ şi $D \subset \Gamma$. Fie $\alpha \in \Gamma$ arbitrar. Deoarece D este mulţime compatibilă, cel puţin una din mulţimile $D \cup \{\alpha\}$, $D \cup \{(\neg\alpha)\}$ este compatibilă.

Dacă $D \cup \{(\neg\alpha)\}$ este compatibilă, atunci $T(D \cup \{(\neg\alpha)\}) \in \mathcal{D}$ şi fie $\Delta \in \mathcal{D}_{\max}$, astfel încât $T(D \cup \{(\neg\alpha)\}) \subset \Delta$.

Rezultă $\{\alpha, (\neg\alpha)\} \subset \Delta$ ceea ce contrazice proprietatea de compatibilitate a mulţimii Δ. Obţinem astfel că $D \cup \{(\neg\alpha)\}$ este incompatibilă, deci $\alpha \in T(D \cup \{(\neg\alpha)\})$. Aplicând teorema deducţiei, rezultă $D \vdash ((\neg\alpha) \rightarrow \bot)$.

Fie $D-$ secvenţa deductivă

$$\begin{aligned}
\gamma_1 &= ((\neg\alpha) \rightarrow \bot) \in T(D) \\
\gamma_2 &= (((\neg\alpha) \rightarrow \bot) \rightarrow ((\neg\,\bot) \rightarrow \alpha)) \in T_h \\
\gamma_3 &= ((\neg\,\bot) \rightarrow \alpha), \quad \frac{\gamma_1, \gamma_2}{\gamma_3} MP \\
\gamma_4 &= (\neg\,\bot) \in T_h \\
\gamma_5 &= \alpha, \quad \frac{\gamma_4, \gamma_3}{\gamma_5} MP.
\end{aligned}$$

Rezultă $\alpha \in T(D) = D$, deci $\Gamma \subset D$.

În concluzie $\Gamma = D$ ceea ce demonstrează proprietatea afirmată în enunţ.

Lema 1.7.11. Fie $D \in \mathcal{D}_{\max}$. Pentru orice formule α, β sunt îndeplinite proprietăţile:

1. $\alpha \in D$, dacă şi numai dacă $(\neg\alpha) \notin D$.
2. $(\alpha \rightarrow \beta) \in D$, dacă şi numai dacă $\alpha \notin D$ sau $\beta \in D$.
3. $(\alpha \vee \beta) \in D$, dacă şi numai dacă $\alpha \in D$ sau $\beta \in D$.
4. $(\alpha \wedge \beta) \in D$, dacă şi numai dacă $\alpha \in D$ şi $\beta \in D$.
5. $(\alpha \leftrightarrow \beta) \in D$, dacă şi numai dacă $\{\alpha, \beta\} \subset D$ sau $\{\alpha, \beta\} \cap D = \emptyset$.

Demonstraţie Fie $D \in \mathcal{D}$ şi $\alpha,\ \beta \in FORM$.

1. Dacă $\alpha \in D$, cum D este mulţime compatibilă, rezultă $(\neg\alpha) \notin D$. Reciproc, dacă $(\neg\alpha) \notin D$, deoarece D este sistem deductiv complet, rezultă $\alpha \in D$.

2. Presupunem $(\alpha \to \beta) \in D$.

Dacă $\alpha \in D$, cum D este stabilă modus ponens, rezultă $\beta \in D$. Rezultă că dacă $(\alpha \to \beta) \in D$, atunci $\alpha \notin D$ sau $\beta \in D$.

Presupunem $\alpha \notin D$, deci cum D este sistem deductiv complet, rezultă $(\neg\alpha) \in D$.

Considerăm $D-$ secvenţa deductivă,

$$\begin{aligned} \gamma_1 &= ((\neg\alpha) \to (\alpha \to \beta)) \in T_h \\ \gamma_2 &= (\neg\alpha) \in D \\ \gamma_3 &= (\alpha \to \beta), \quad \frac{\gamma_2, \gamma_1}{\gamma_3} MP \end{aligned}$$

deci $(\alpha \to \beta) \in T(D) = D$.

Dacă $\beta \in D$, atunci fie $D-$ secvenţa deductivă

$$\begin{aligned} \gamma_1 &= (\beta \to (\alpha \to \beta)) \in T_h \\ \gamma_2 &= \beta \in D \\ \gamma_3 &= (\alpha \to \beta), \quad \frac{\gamma_2, \gamma_1}{\gamma_3} MP, \end{aligned}$$

deci $(\alpha \to \beta) \in T(D) = D$.

3. Presupunem $(\alpha \vee \beta) \in D$. Din $D-$ secvenţa deductivă

$$\begin{aligned} \gamma_1 &= (\alpha \vee \beta) \in D \\ \gamma_2 &= ((\alpha \vee \beta) \to ((\neg\alpha) \to \beta)) \in T_h \\ \gamma_3 &= ((\neg\alpha) \to \beta), \quad \frac{\gamma_1, \gamma_2}{\gamma_3} MP, \end{aligned}$$

rezultă $((\neg\alpha) \to \beta) \in T(D) = D$. Utilizând proprietăţile (1) şi (2), obţinem în continuare $(\neg\alpha) \notin D$ sau $\beta \in D$, adică $\alpha \in D$ sau $\beta \in D$.

Dacă $\alpha \in D$, atunci considerăm $D-$ secvenţa deductivă, $\alpha, (\alpha \to (\alpha \vee \beta)), (\alpha \vee \beta)$, respectiv dacă $\beta \in D$, atunci considerăm $D-$ secvenţa deductivă $\beta, (\beta \to (\alpha \vee \beta)), (\alpha \vee \beta)$.

În ambele cazuri rezultă $(\alpha \vee \beta) \in T(D) = D$.

4.Dacă $(\alpha \wedge \beta) \in D$, atunci evident

$$(\alpha \wedge \beta), ((\alpha \wedge \beta) \to \alpha), \alpha, ((\alpha \wedge \beta) \to \beta), \beta$$

este o $D-$ secvenţă deductivă, deci $\{\alpha, \beta\} \subset T(D) = D$.

Presupunem $\{\alpha, \beta\} \subset D$. Deoarece

$$(\alpha \to (\beta \to (\alpha \wedge \beta))) \in T_h \quad \text{(aplicaţia 1.4.13.)},$$

secvenţa $\alpha, \beta, (\alpha \to (\beta \to (\alpha \wedge \beta))), (\beta \to (\alpha \wedge \beta)), (\alpha \wedge \beta)$ este o

$D-$ secvenţă deductivă, deci $(\alpha \wedge \beta) \in T(D) = D$.

5. Presupunem $(\alpha \leftrightarrow \beta) \in D$. Deoarece $((\alpha \leftrightarrow \beta) \to (\alpha \to \beta)) \in T_h$ şi $((\alpha \leftrightarrow \beta) \to (\beta \to \alpha)) \in T_h$, rezultă $(\alpha \to \beta) \in D$ şi $(\beta \to \alpha) \in D$.

Dacă $\alpha \in D$, cum D este stabilă modus ponens, rezultă

$\beta \in T(D) = D$, deci $\{\alpha, \beta\} \subset D$.

Dacă $\alpha \notin D$, atunci aplicând proprietatea (2), din $(\beta \to \alpha) \in D$, rezultă $\beta \notin D$, deci $\{\alpha, \beta\} \cap D = \emptyset$.

Reciproc, să presupunem că $\{\alpha, \beta\} \subset D$ sau $\{\alpha, \beta\} \cap D = \emptyset$.

Dacă $\{\alpha, \beta\} \subset D$, atunci fie $D-$ secvenţa deductivă

$$\begin{aligned}
\gamma_1 &= (\beta \to (\alpha \to \beta)) \in T_h \\
\gamma_2 &= \beta \in D \\
\gamma_3 &= (\alpha \to \beta), \quad \frac{\gamma_2, \gamma_1}{\gamma_3} MP \\
\gamma_4 &= (\alpha \to (\beta \to \alpha)) \in T_h \\
\gamma_5 &= \alpha \in D \\
\gamma_6 &= (\beta \to \alpha), \quad \frac{\gamma_5, \gamma_4}{\gamma_6} MP \\
\gamma_7 &= (\alpha \leftrightarrow \beta), \quad \frac{\gamma_3, \gamma_6}{\gamma_7} IE.
\end{aligned}$$

Rezultă $\{(\alpha \to \beta), (\beta \to \alpha), (\alpha \leftrightarrow \beta)\} \subset D$, deci $(\alpha \leftrightarrow \beta) \in T(D) = D$.

Dacă $\{\alpha, \beta\} \cap D = \emptyset$, deoarece D este sistem deductiv complet, rezultă $\{(\neg\alpha), (\neg\beta)\} \subset D$. Pe baza rezultatului precedent, obţinem

$$\{((\neg\alpha) \to (\neg\beta)), ((\neg\beta) \to (\neg\alpha)), ((\neg\alpha) \leftrightarrow (\neg\beta))\} \subset D.$$

Deoarece D este stabilă modus ponens şi

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

respectiv

$$(((\neg\beta) \to (\neg\alpha)) \to (\alpha \to \beta)) \in T_h,$$

rezultă $\{(\alpha \to \beta), (\beta \to \alpha)\} \subset D$, deci $(\alpha \leftrightarrow \beta) \in T(D) = D$. În concluzie, dacă $\{\alpha, \beta\} \subset D$ sau $\{\alpha, \beta\} \cap D = \emptyset$, atunci $(\alpha \leftrightarrow \beta) \in D$.

Definiţia 1.7.7. Se numeşte validare mulţimea $D_I = \{\alpha \mid I \in \aleph(\alpha)\}$ unde I este o interpretare.

Lema 1.7.12. Orice validare este sistem deductiv complet.

Demonstraţie Fie $D_I = \{\alpha \mid I \in \aleph(\alpha)\}$ validare.

Evident pentru orice $\alpha \in FORM$, $\alpha \in D_I$ sau $(\neg\alpha) \in D_I$.

Din *Teorema* 1.7.1, pentru orice $\alpha \in T_h$, $I \in \aleph(\alpha)$, deci $\alpha \in D_I$, adică $T_h \subset D_I$.

Demonstrăm că D_I este stabilă modus ponens. Fie $\alpha, (\alpha \to \beta) \in D_I$, deci $I(\alpha) = I((\alpha \to \beta)) = T$.

Deoarece $I((\alpha \to \beta)) = \neg I(\alpha) \vee I(\beta) = I(\beta)$, rezultă $I(\beta) = T$, adică $\beta \in D_I$.

Din proprietatea de a fi stabilă modus ponens şi $T_h \subset D_I$ obţinem evident $T(D_I) = D_I$. Compatibilitatea mulţimii D_I rezultă din argumentul următor: deoarece $I(\bot) = F$ pentru orice $I \in \Im$ obţinem $\bot \notin D_I$ pentru orice $I \in \Im$, deci $T(D_I) = D_I \neq FORM$. Proprietăţile stabilite justifică concluzia că pentru orice $I \in \Im$, D_I este sistem deductiv complet. Utilizând concluzia *Lemei* 1.7.9., rezultă că pentru orice $I \in \Im$, D_I este sistem deductiv maximal.

Lema 1.7.13. Orice sistem deductiv complet este validare.

Demonstraţie Fie D sistem deductiv complet.

Definim $I : FORM \to \{T, F\}$ prin

$$\forall \alpha \in FORM, \quad I(\alpha) = \begin{cases} T & \text{dacă } \alpha \in D \\ F & \text{dacă } \alpha \notin D \end{cases}$$

Deoarece D este sistem deductiv complet, funcţia I este bine definită şi $I \in \Im$. Evident, $D = D_I$. Din proprietăţile operaţiilor logice $\neg, \wedge, \vee, \to, \leftrightarrow$ pe mulţimea $\{T, F\}$ şi concluzia stabilită de *Lema* 1.7.11. rezultă că pentru orice formule α, β,

1. $I((\neg\alpha)) = \neg I(\alpha)$,
2. $I((\alpha \to \beta)) = I(\alpha) \to I(\beta)$,
3. $I((\alpha \vee \beta)) = I(\alpha) \vee I(\beta)$,
4. $I((\alpha \wedge \beta)) = I(\alpha) \wedge I(\beta)$,
5. $I((\alpha \leftrightarrow \beta)) = I(\alpha) \leftrightarrow I(\beta)$.

Într-adevăr,

1. $I((\neg\alpha)) = T$, dacă şi numai dacă $(\neg\alpha) \in D$, dacă şi numai dacă $\alpha \notin D$, dacă şi numai dacă $I(\alpha) = F$ dacă şi numai dacă $\neg I(\alpha) = T$, deci
$I((\neg\alpha)) = \neg I(\alpha)$.

2. $I((\alpha \to \beta)) = T$, dacă şi numai dacă $(\alpha \to \beta) \in D$, dacă şi numai dacă $\alpha \notin D$ sau $\beta \in D$, dacă şi numai dacă $I(\alpha) = F$ sau $I(\beta) = T$, dacă şi numai dacă $I(\alpha) \to I(\beta) = T$, deci

$I((\alpha \to \beta)) = I(\alpha) \to I(\beta)$.

3. $I((\alpha \vee \beta)) = T$, dacă şi numai dacă $(\alpha \vee \beta) \in D$, dacă şi numai dacă $\alpha \in D$ sau $\beta \in D$, dacă şi numai dacă $I(\alpha) = T$ sau $I(\beta) = T$, dacă şi numai dacă $I(\alpha) \vee I(\beta) = T$, deci $I((\alpha \vee \beta)) = I(\alpha) \vee I(\beta)$.

4. $I((\alpha \wedge \beta)) = T$, dacă şi numai dacă $(\alpha \wedge \beta) \in D$, dacă şi numai dacă $\alpha \in D$ şi $\beta \in D$, dacă şi numai dacă $I(\alpha) = T$ şi $I(\beta) = T$, dacă şi numai dacă $I(\alpha) \wedge I(\beta) = T$, deci $I((\alpha \wedge \beta)) = I(\alpha) \wedge I(\beta)$.

5. $I((\alpha \leftrightarrow \beta)) = T$, dacă şi numai dacă $(\alpha \leftrightarrow \beta) \in D$, dacă şi numai dacă $\{\alpha, \beta\} \subset D$ sau $\{\alpha, \beta\} \cap D = \emptyset$, dacă şi numai dacă $I(\alpha) = I(\beta) = T$ sau $I(\alpha) = I(\beta) = F$, dacă şi numai dacă

$I(\alpha) \leftrightarrow I(\beta) = T$, deci $I((\alpha \leftrightarrow \beta)) = I(\alpha) \leftrightarrow I(\beta)$.

Teorema 1.7.3 (Teorema de completitudine a limbajului calculului cu propoziţii)

Orice formulă demonstrabilă a limbajului calculului cu propoziţii este tautologie.

Demonstraţie Justificarea afirmaţiei din enunţ rezultă pe baza concluziilor stabilite de lemele precedente.

Într-adevăr, deoarece $T_h \in \mathcal{D}$ şi $T_h \subset D$ pentru orice $D \in \mathcal{D}$, din

Lema 1.7.10. obţinem $T_h = \bigcap_{D \in \mathcal{D}_{\max}} D$.

Din *Lema* 1.7.9. rezultă $\bigcap_{D \in \mathcal{D}_{\max}} D = \bigcap_{\substack{D \in \mathcal{D} \\ D \text{ complet}}} D$ deci, în continuare, pe baza concluziilor stabilite de *Lema* 1.7.12. şi *Lema* 1.7.13., obţinem

$$\bigcap_{\substack{D \in \mathcal{D} \\ D \text{ complet}}} D = \bigcap_{\substack{D \in \mathcal{D} \\ D \text{ validare}}} D = \bigcap_{I \in \Im} D_I$$

În final rezultă $T_h = \bigcap_{I \in \Im} D_I$, deci pentru orice formulă α, $\alpha \in T_h$, dacă şi numai dacă $\forall I \in \Im, I(\alpha) = T$ deci, dacă şi numai dacă α este tautologie.

Corolarul 1.7.3 Pentru orice formule α, β, $(\alpha \to \beta) \in T_h$,dacă şi numai dacă $\aleph(\alpha) \subset \aleph(\beta)$.

Demonstraţie Presupunem $(\alpha \to \beta) \in T_h$ deci $\aleph((\alpha \to \beta)) = \Im$. Pentru orice $I \in \aleph(\alpha)$, obţinem $I(\alpha) = I((\alpha \to \beta)) = T$ din care rezultă evident $I(\beta) = T$, deci $I \in \aleph(\beta)$. Presupunem $\aleph(\alpha) \subset \aleph(\beta)$.

Pentru $I \in \Im$ arbitrară,

$$I((\alpha \to \beta)) = I((\neg\alpha)) \vee I(\beta) = \neg I(\alpha) \vee I(\beta).$$

Dacă $I(\alpha) = T$, atunci $I \in \aleph(\alpha)$, deci $I \in \aleph(\beta)$, ceea ce implică evident $I((\alpha \to \beta)) = T$.

Dacă $I(\alpha) = F$, atunci $\neg I(\alpha) = T$, deci $I((\alpha \to \beta)) = T$.

Rezultă că $(\alpha \to \beta)$ este tautologie, deci $(\alpha \to \beta) \in T_h$.

Definiţia 1.7.8. Fie $H, \Gamma \subset FORM$. Spunem că Γ este consecinţă semantică a mulţimii de formule H (notat $H \vDash \Gamma$), dacă

$$\aleph(H) \subset \bigcup_{\beta \in \Gamma} \aleph(\beta).$$

Observaţie Dacă H este o mulţime inconsistentă de formule, atunci $H \vDash \Gamma$ pentru orice $\Gamma \subset FORM$.

Rezultatul formulat de următoarea teoremă stabileşte corespondenţa dintre conceptul de ordin sintactic de deductibilitate globală şi conceptul de consecinţă semantică în cazul particular când ambele mulţimi, mulţimea premiselor şi mulţimea concluzie sunt finite.

Teorema 1.7.4. Dacă $H = \{\alpha_1, ..., \alpha_n\}$, $\Gamma = \{\beta_1, ..., \beta_m\}$, atunci $H \vdash \Gamma$, dacă şi numai dacă $H \vDash \Gamma$.

Demonstraţie Justificarea afirmaţiei din enunţ poate fi obţinută în mai multe moduri. Proprietatea poate fi demonstrată în cazul general în care mulţimile H, Γ nu sunt în mod necesar finite (*Teorema* 1.7.7.), utilizând concluzia Teoremei de compacitate (*Teorema* 1.7.5.). În cazul particular în care ambele mulţimi H, Γ sunt finite, afirmaţia din enunţ rezultă din următorul argument:

Din *Teorema* 1.5.3, $\{\alpha_1, ..., \alpha_n\} \vdash \{\beta_1, ..., \beta_m\}$, dacă şi numai dacă
$\left\{\bigwedge_{i=1}^{n} \alpha_i\right\} \vdash \bigvee_{j=1}^{m} \beta_j$.

De asemenea, din teorema deducţiei,
$\left\{\bigwedge_{i=1}^{n} \alpha_i\right\} \vdash \bigvee_{j=1}^{m} \beta_j$, dacă şi numai dacă $\left(\bigwedge_{i=1}^{n} \alpha_i \to \bigvee_{j=1}^{m} \beta_j\right) \in T_h$.

Conform *Corolarului* 1.7.3., obţinem $H \vdash \Gamma$, dacă şi numai dacă
$\aleph\left(\bigwedge_{i=1}^{n} \alpha_i\right) \subset \aleph\left(\bigvee_{j=1}^{m} \beta_j\right)$.

Utilizând rezultatele stabilite de *Lema* 1.7.2. şi *Lema* 1.7.3., obţinem
$\aleph(H) = \aleph\left(\bigwedge_{i=1}^{n} \alpha_i\right); \aleph\left(\bigvee_{j=1}^{m} \beta_j\right) = \bigcup_{j=1}^{m} \aleph(\beta_j)$, deci $H \vdash \Gamma$, dacă şi numai dacă

$H \vDash \Gamma$.

Corolarul 1.7.4. Dacă $H = \{\alpha_1, ..., \alpha_n\}$, $\Gamma = \{\beta_1, ..., \beta_m\}$ sunt mulţimi de formule, atunci $H \vDash \Gamma$, dacă şi numai dacă formula

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

este invalidabilă.

Demonstraţie Deoarece $\aleph((\neg\beta_j)) = \Im \setminus \aleph(\beta_j)$, evident incluziunea $\aleph\left(\bigwedge_{i=1}^{n} \alpha_i\right) \subset \aleph\left(\bigvee_{j=1}^{m} \beta_j\right)$ are loc dacă şi numai dacă

$$\aleph\left(\bigwedge_{i=1}^{n} \alpha_i\right) \cap \left(\Im \setminus \aleph\left(\bigvee_{j=1}^{m} \beta_j\right)\right) = \emptyset,$$

deci dacă şi numai dacă $\left(\bigcap_{i=1}^{n} \aleph(\alpha_i)\right) \cap \left(\Im \setminus \bigcup_{j=1}^{m} \aleph(\beta_j)\right) = \emptyset$.

Obţinem în final concluzia $H \vDash \Gamma$, dacă şi numai dacă

$$\left(\bigcap_{i=1}^{n} \aleph(\alpha_i)\right) \cap \left(\bigcap_{j=1}^{m} \aleph((\neg\beta_j))\right) = \emptyset,$$

deci dacă şi numai dacă formula $\left(\bigwedge_{i=1}^{n} \alpha_i\right) \wedge \left(\bigwedge_{j=1}^{m} (\neg\beta_j)\right)$ este invalidabilă.

Rezultatul stabilit de *Corolarul* 1.7.4 permite caracterizarea în termeni semantici a deductibilităţii globale în cazul particular când atât mulţimea premiselor, cât şi mulţimea concluzie sunt mulţimi finite şi anume,

$H \vdash \Gamma$, dacă şi numai dacă $\left(\bigwedge_{i=1}^{n} \alpha_i\right) \wedge \left(\bigwedge_{j=1}^{m} (\neg\beta_j)\right)$este invalidabilă.

Cu alte cuvinte, dacă H, Γ sunt mulţimi finite, atunci proprietatea unei mulţimi Γ de a fi global deductibilă pe baza mulţimii de premise H este echivalentă cu proprietatea ca mulţimea concluzie să fie consecinţă semantică a mulţimii

premiselor, respectiv cu proprietatea ca formula rezultată prin conjuncţia formulelor din mulţimea premisă cu conjuncţia negaţiilor formulelor din mulţimea concluzie să fie invalidabilă.

Corolarul 1.7.5 Mulţimea finită H este consistentă, dacă şi numai dacă este compatibilă.

Demonstraţie Conform unei caracterizări stabilite anterior, H este incompatibilă, dacă şi numai dacă $H \vdash \perp$. Din Teorema 1.7.4, $H \vdash \perp$, dacă şi numai dacă $H \models \perp$, deci cum $\aleph(\perp) = \emptyset$, dacă şi numai dacă $\aleph(H) = \emptyset$.

În concluzie, H este incompatibilă, dacă şi numai dacă H este inconsistentă.

În continuare vom stabili caracterul finit al consistenţei (validabilităţii) mulţimilor arbitrare de formule. Proprietatea este referită ca teoremă de compacitate a limbajului calculului cu propoziţii.

Definiţia 1.7.9 Mulţimea de formule H este finit validabilă, dacă orice submulţime finită a ei este consistentă.

Teorema 1.7.5. (Teorema de compacitate a limbajului calculului cu propoziţii). Mulţimea de formule H este consistentă, dacă şi numai dacă este finit validabilă.

Demonstraţie Dacă H este consistentă, atunci pentru orice

$H_0 \subset H$, $\emptyset \neq \aleph(H) \subset \aleph(H_0)$, deci H_0 este de asemenea consistentă.

În particular, dacă H este consistentă, atunci H este finit validabilă.

Demonstrăm că orice mulţime de formule este finit validabilă, dacă şi numai dacă este compatibilă.

Presupunem că H este finit validabilă. Dacă H ar fi incompatibilă, atunci $H \vdash \perp$ şi cum deductibilitatea este o proprietate de caracter finit, există $H_0 \subset H$, H_0 finită, astfel încât $H_0 \vdash \perp$. Utilizând rezultatul stabilit de *Corolarul* 1.7.5., obţinem H_0 inconsistentă, ceea ce ar contrazice ipoteza că H este finit validabilă. Rezultă că dacă H este finit validabilă, atunci H este mulţime compatibilă. Presupunem că H este compatibilă. Orice H_0 submulţime finită a mulţimii H este compatibilă, deci conform rezultatului stabilit de *Corolarul* 1.7.5., H_0 este consistentă. Rezultă că dacă H este compatibilă, atunci H este finit validabilă.

In concluzie, orice mulţime de formule este finit validabilă, dacă şi numai dacă este compatibilă.

Dacă H este finit validabilă, atunci $T(H)$ este sistem deductiv. Conform Teoremei Lindenbaum Tarski, există D sistem deductiv maximal, astfel încât $T(H) \subset D$. Din rezultatele stabilite de *Lema* 1.7.9., *Lema* 1.7.12., *Lema* 1.7.13., există $I \in \Im$ şi $D = D_I$. Obţinem astfel $H \subset T(H) \subset D = D_I$, adică $I \in \aleph(H)$, deci H este validabilă.

Observaţie Conform concluziei Teoremei Lindenbaum Tarski, orice sistem deductiv este inclus într-un sistem deductiv maximal. Construcţia efectivă a unui sistem deductiv maximal care să includă un sistem deductiv dat poate fi realizată pe baza următoarei proceduri.

Deoarece $FORM$ este numărabilă, putem considera $\alpha_1, ..., \alpha_n, ...$ o enumerare oarecare a mulţimii formulelor.

Fie D sistem deductiv. Construim şirul ascendent de mulţimi compatibile $(D_n)_{n\geq 0}$ în modul următor:

$$D_n = \begin{cases} D, & \text{dacă } n = 0 \\ D_{n-1} \cup \{\alpha_n\}, & \text{dacă } D_{n-1} \cup \{\alpha_n\} \text{ este compatibilă} \\ D_{n-1} \cup \{(\neg\alpha)\}, & \text{dacă } D_{n-1} \cup \{\alpha_n\} \text{ este incompatibilă.} \end{cases}$$

Deoarece D este sistem deductiv, rezultă D_0 este mulţime compatibilă, deci cel puţin una din mulţimile $D \cup \{\alpha_1\}$, $D \cup \{(\neg\alpha_1)\}$ este compatibilă. Pe

baza unui argument inductiv, rezultă D_n mulţime compatibilă pentru orice $n \geq 0$.

Fie $\overline{D} = \bigcup_{n=0}^{\infty} D_n$.

Dacă $\overline{D}$ ar fi incompatibilă, atunci $\overline{D} \vdash \bot$. Deoarece deductibilitatea este proprietate de caracter finit, există $\overline{D}_0 \subset \overline{D}$, $\overline{D}_0$ finită, astfel încât $\overline{D}_0 \vdash \bot$. Deoarece $(D_n)_{n\geq 0}$ este şir ascendent, există $m \geq 1$, astfel încât $\overline{D}_0 \subset D_m$, deci $D_m \vdash \bot$, ceea ce evident contrazice proprietatea de compatibilitate a mulţimii D_m. Rezultă $\overline{D}$ mulţime compatibilă. Evident, $T_h \subset D \subset \overline{D}$. Demonstrăm că $\overline{D}$ este stabilă modus ponens. Într-adevăr, dacă $\{\alpha, (\alpha \to \beta)\} \subset \overline{D}$, atunci fie $n \geq 0$ astfel încât $\{\alpha, (\alpha \to \beta)\} \subset D_n$. Dacă $(\neg\alpha) \in D_n$, atunci $(\neg\alpha)$, α, $((\neg\alpha) \to (\alpha \to \bot))$, $(\alpha \to \bot)$, $\bot$ este o D_n- secvenţă deductivă, deci ar rezulta că D_n este mulţime incompatibilă. Rezultă $(\neg\alpha) \notin D_n$.

Fie $0 \leq i, j \leq n$, astfel încât $\alpha \in D_i \setminus D_{i-1}$, $(\alpha \to \beta) \in D_j \setminus D_{j-1}$ unde $D_{-1} = \emptyset$. Presupunem $\beta = \alpha_k$. Dacă $D_k = D_{k-1} \cup \{(\neg\beta)\}$, atunci $\{\alpha, (\alpha \to \beta), (\neg\beta)\} \subset D_m$,unde $m = \max\{n, k\}$.

Evident $\alpha, (\alpha \to \beta), \beta, (\neg\beta), ((\neg\beta) \to (\beta \to \bot)), (\beta \to \bot), \bot$ este D_m- secvenţă deductivă, deci ar rezulta D_m incompatibilă. În concluzie, la pasul k al procedurii descrise, $D_k = D_{k-1} \cup \{\beta\}$, deci $\beta \in \overline{D}$.

Din concluzia *Lemei* 1.7.7 rezulta, $\overline{D}$ sistem deductiv. Evident, pentru orice formulă α, $\alpha \in \overline{D}$ sau $(\neg\alpha) \in \overline{D}$ deci $\overline{D}$ sistem deductiv complet.

Din *Lema* 1.7.9., rezultă că $\overline{D}$ este sistem deductiv maximal şi $D \subset \overline{D}$.

Corespondenţa dintre conceptele de compatibilitate/incompatibilitate şi consistenţă/inconsistenţă este stabilită de următoarea teoremă.

Teorema 1.7.6. Mulţimea $H \subset FORM$ este incompatibilă, dacă şi numai dacă este inconsistentă.

Demonstraţie Dacă H este incompatibilă, atunci $H \vdash \bot$, deci cum proprietatea de deductibilitate este de caracter finit, rezultă că există $H_0 \subset H$, H_0 finită, astfel încât $H_0 \vdash \bot$, deci $H_0 \vDash \bot$.

Deoarece $\aleph(\bot) = \emptyset$, rezultă $\aleph(H_0) = \emptyset$, deci H_0 este inconsistentă.

Evident, $\aleph(H_0) \supseteq \aleph(H)$, deci $\aleph(H) = \emptyset$. Dacă H este inconsistentă, conform teoremei de compacitate există $H_0 \subset H$, H_0 finită şi inconsistentă. Rezultă că pentru orice $\Gamma \subset FORM$, $H_0 \vDash \Gamma$.

În particular, pentru $\Gamma = \{\bot\}$ rezultă $H_0 \vDash \{\bot\}$ deci $H_0 \vdash \bot$.

Obţinem astfel $FORM = T(H_0) \subseteq T(H)$, deci H este incompatibilă.

În continuare, putem stabili o caracterizare a deductibilităţii globale în termeni semantici.

Teorema 1.7.7. Fie H, Γ mulţimi arbitrare de formule. Atunci $H \vdash \Gamma$, dacă şi numai dacă $H \vDash \Gamma$.

Demonstraţie Utilizând rezultatele anterior stabilite, obţinem

$H \vdash \Gamma$ dacă şi numai dacă $H \cup \neg\Gamma \vdash \perp$, deci dacă şi numai dacă $H \cup \neg\Gamma$ este inconsistentă.

Deoarece

$$\begin{aligned}
\aleph(H \cup \neg\Gamma) &= \left(\bigcap_{\alpha\in H} \aleph(\alpha)\right) \cap \left(\bigcap_{\gamma\in\Gamma} \aleph((\neg\gamma))\right) \\
&= \left(\bigcap_{\alpha\in H} \aleph(\alpha)\right) \cap \left(\bigcap_{\gamma\in\Gamma} (\Im \setminus \aleph(\gamma))\right) \\
&= \left(\bigcap_{\alpha\in H} \aleph(\alpha)\right) \cap \left(\Im \setminus \bigcup_{\gamma\in\Gamma} \aleph(\gamma)\right),
\end{aligned}$$

rezultă $\left(\bigcap_{\alpha\in H} \aleph(\alpha)\right) \cap \left(\Im \setminus \bigcup_{\gamma\in\Gamma} \aleph(\gamma)\right) = \emptyset$, adică

$$\left(\bigcap_{\alpha\in H} \aleph(\alpha)\right) \subset \bigcup_{\gamma\in\Gamma} \aleph(\gamma).$$

Rezultă în final concluzia $H \cup \neg\Gamma$ este inconsistentă, dacă şi numai dacă $H \vDash \Gamma$.

Corolarul 1.7.6. $(\alpha \to \beta) \in T_h$ dacă şi numai dacă $\aleph(\alpha) \subset \aleph(\beta)$.

Demonstraţie Presupunem $(\alpha \to \beta) \in T_h$, deci $\aleph((\alpha \to \beta)) = \Im$. Pentru orice $I \in \aleph(\alpha)$, obţinem $I(\alpha) = I((\alpha \to \beta)) = T$, din care rezultă evident $I(\beta) = T$, deci $I \in \aleph(\beta)$.

Presupunem $\aleph(\alpha) \subset \aleph(\beta)$. Fie $I \in \Im$ arbitrară; conform unei observaţii precedente,

$I((\alpha \to \beta)) = I((\neg\alpha)) \vee I(\beta) = \neg I(\alpha) \vee I(\beta)$.

Dacă $I(\alpha) = T$, atunci $I \in \aleph(\alpha)$, deci $I \in \aleph(\beta)$, ceea ce implică evident $I((\alpha \to \beta)) = T$.

Dacă $I(\alpha) = F$, atunci $\neg I(\alpha) = T$, deci $I((\alpha \to \beta)) = T$.

Rezultă că $(\alpha \to \beta)$ este tautologie, deci $(\alpha \to \beta) \in T_h$.

Observaţie Dacă H este o mulţime inconsistentă de formule, atunci $H \vDash \Gamma$ pentru orice $\Gamma \subset FORM$.

4 Semantica sistemului deducţiei naturale

În cadrul acestei secţiuni vor fi stabilite rezultate relativ la calculul cu secvenţi finiţi, majoritatea concluziilor putând fi însă extinse la cazul secvenţilor infiniţi.

Definiţia 1.8.1. Fie $S = H \Longrightarrow \Gamma$ secvent şi $I \in \Im$. Valoarea de adevăr $I(S)$ calculată de interpretarea I pentru secventul S este $I(S) = T$, dacă şi numai dacă există $\alpha \in H$, astfel încât $I(\alpha) = F$ sau există $\gamma \in \Gamma$, astfel încât $I(\gamma) = T$.

Observaţie Din *Definiţia* 1.8.1 rezultă că $I(S) = F$, dacă şi numai dacă pentru orice $\alpha \in H, I(\alpha) = T$ şi pentru orice $\gamma \in \Gamma, I(\gamma) = F$.

Cu alte cuvinte, $I(S) = F$, dacă şi numai dacă

$$\begin{aligned} I \in & \bigcap_{\alpha \in H} \aleph(\alpha) \cap \left(\bigcap_{\gamma \in \Gamma} (\Im \setminus \aleph(\gamma)) \right) = \\ = & \bigcap_{\alpha \in H} \aleph(\alpha) \cap \left(\Im \setminus \left(\bigcup_{\gamma \in \Gamma} \aleph(\gamma) \right) \right). \end{aligned}$$

Definiţia 1.8.2. Secventul S este secvent valid (notat $\models S$), dacă pentru orice $I \in \Im$, $I(S) = T$. Secventul S este secvent falsificabil dacă există $I \in \Im$, astfel încât $I(S) = F$.

Teorema 1.8.1. Fie $S = H \Longrightarrow \Gamma$ secvent arbitrar. Atunci S este secvent valid, dacă şi numai dacă $H \vDash \Gamma$.

Demonstraţie Presupunem că S este secvent valid. Dacă $H = \emptyset$, atunci $\aleph(H) = \Im$. Conform definiţiei, în acest caz rezultă că pentru orice $I \in \Im$ există $\gamma \in \Gamma$, astfel încât $I(\gamma) = T$, deci $I \in \bigcup_{\beta \in \Gamma} \aleph(\beta)$.

Dacă $H \neq \emptyset$, dar $\aleph(H) = \emptyset$, atunci $\aleph(H) \subset \bigcup_{\beta \in \Gamma} \aleph(\beta)$, deci $H \vDash \Gamma$.

Dacă $H \neq \emptyset$ şi $\aleph(H) \neq \emptyset$, atunci cum $\vDash S$, rezultă că pentru orice $I \in \aleph(H)$ există $\gamma \in \Gamma$, astfel încât $I(\gamma) = T$, deci $I \in \bigcup_{\beta \in \Gamma} \aleph(\beta)$, ceea ce evident implică $H \vDash \Gamma$.

Presupunem $H \vDash \Gamma$. Fie $I \in \Im$ arbitrară. Dacă $I(\alpha) = T$ pentru orice $\alpha \in H$, atunci $I \in \aleph(H)$, deci $I \in \bigcup_{\beta \in \Gamma} \aleph(\beta)$.

Rezultă că există $\beta \in \Gamma$, astfel încât $I \in \aleph(\beta)$, deci conform definiţiei obţinem $I(S) = T$.

Dacă există $\alpha \in H$, astfel încât $I(\alpha) = F$, atunci evident se obţine aceeaşi concluzie. Rezultă astfel că pentru orice $I \in \Im$, $I(S) = T$, deci $\vDash S$.

În concluzie, $S = H \Longrightarrow \Gamma$ este secvent valid, dacă şi numai dacă

$\aleph(H) \subset \bigcup_{\beta\in\Gamma} \aleph(\beta)$ sau echivalent, $\bigcap_{\alpha\in H} \aleph(\alpha) \subset \bigcup_{\beta\in\Gamma} \aleph(\beta)$.

Teorema 1.8.2. (Teorema de consistenţă a sistemului deducţiei naturale) Orice secvent demonstrabil este secvent valid.

Demonstraţie Concluzia formulată în enunţ rezultă din următoarele argumente:

1. Orice secvent axiomă este secvent valid.

2. Dacă $\frac{S_1}{S}$ este exemplu de regulă Gentzen, atunci pentru orice $I \in \Im$, $I(S_1) = I(S)$.

3. Dacă $\frac{S_1,S_2}{S}$ este exemplu de regulă Gentzen, atunci pentru orice $I \in \Im$, $I(S_1) \wedge I(S_2) = I(S)$.

4. Demonstrăm prin inducţie asupra înălţimii arborelui de demonstraţie că orice secvent demonstrabil este secvent valid.

1. Fie $S = H \Rightarrow \Gamma$ secvent axiomă, deci $H \cap \Gamma \neq \emptyset$. Fie $\alpha \in H \cap \Gamma$. Pentru $I \in \Im$, arbitrar, dacă $I(\alpha) = T$, cum $\alpha \in \Gamma$ obţinem $I(S) = T$, respectiv dacă $I(\alpha) = F$, cum $\alpha \in H$, conform definiţiei rezultă de asemenea concluzia $I(S) = T$.

Observaţie Justificarea afirmaţiei 1 poate fi efectuată şi astfel: utilizând concluzia *Lemei* 1.5.2., dacă $S = H \Rightarrow \Gamma$ este secvent axiomă, atunci $H \cap \Gamma \neq \emptyset$ şi deci $H \vdash \Gamma$. Rezultă $H \models \Gamma$ deci S este secvent valid.

2. Verificăm că pentru orice $I \in \Im$ şi orice regulă Gentzen de tipul 1, $\frac{S_1}{S}$, $I(S_1) = I(S)$.

Regula $G1$: Fie $S_1 = H \Longrightarrow \Gamma \cup \{\alpha\}$, $S = H \cup \{(\neg\alpha)\} \Longrightarrow \Gamma$.

Din *Teorema* 1.5.4., obţinem $H \cup \{(\neg\alpha)\} \vdash \Gamma$, dacă şi numai dacă $H \vdash \Gamma \cup \{\alpha\}$.

Utilizând concluzia *Teoremei* 1.7.4., obţinem $H \cup \{(\neg\alpha)\} \vdash \Gamma$, dacă şi numai dacă $H \cup \{(\neg\alpha)\} \models \Gamma$ respectiv $H \vdash \Gamma \cup \{\alpha\}$ dacă şi numai dacă $H \models \Gamma \cup \{\alpha\}$.

Rezultă $H \cup \{(\neg\alpha)\} \models \Gamma$, dacă şi numai dacă $H \models \Gamma \cup \{\alpha\}$ sau echivalent, $\left(\bigcap_{\beta\in H} \aleph(\beta)\right) \cap \aleph((\neg\alpha)) \subset \bigcup_{\gamma\in\Gamma} \aleph(\gamma)$.

Deoarece $\aleph((\neg\alpha)) = \Im \setminus \aleph(\alpha)$,

$\left(\bigcap_{\beta\in H} \aleph(\beta)\right) \cap \aleph((\neg\alpha)) \subset \bigcup_{\gamma\in\Gamma} \aleph(\gamma)$ este indeplinită, dacă şi numai dacă

$\bigcap_{\beta\in H} \aleph(\beta) \subset \left(\bigcup_{\gamma\in\Gamma} \aleph(\gamma)\right) \cup \aleph(\alpha)$.

Din definiţie rezultă $I(S_1) = F$, dacă şi numai dacă

$$\begin{aligned} I &\in \left(\bigcap_{\beta\in H} \aleph(\beta)\right) \cap \left(\Im \setminus \left(\left(\bigcup_{\gamma\in\Gamma} \aleph(\gamma)\right) \cup \aleph(\alpha)\right)\right) \\ &= \left(\bigcap_{\beta\in H} \aleph(\beta)\right) \cap \left(\Im \setminus \left(\bigcup_{\gamma\in\Gamma} \aleph(\gamma)\right)\right) \cap (\Im \setminus \aleph(\alpha)) \\ &= \left(\left(\bigcap_{\beta\in H} \aleph(\beta)\right) \cap \aleph((\neg\alpha))\right) \cap \left(\Im \setminus \left(\bigcup_{\gamma\in\Gamma} \aleph(\gamma)\right)\right), \end{aligned}$$

respectiv $I(S) = F$, dacă şi numai dacă

$$I \in \left(\left(\bigcap_{\beta\in H} \aleph(\beta)\right) \cap \aleph((\neg\alpha))\right) \cap \left(\Im \setminus \left(\bigcup_{\gamma\in\Gamma} \aleph(\gamma)\right)\right).$$

Rezultă $I(S_1) = F$, dacă şi numai dacă $I(S) = F$, deci
$I(S_1) = I(S)$.

Regula $G2:$ Fie $S_1 = H \cup \{\alpha, \beta\} \Longrightarrow \Gamma$, $S = H \cup \{(\alpha \wedge \beta)\} \Longrightarrow \Gamma$.

Utilizând un argument similar celui considerat la cazul precedent, obţinem caracterizările,

$H \cup \{\alpha, \beta\} \vdash \Gamma$, dacă şi numai dacă $H \cup \{(\alpha \wedge \beta)\} \vdash \Gamma$,
$H \cup \{\alpha, \beta\} \vdash \Gamma$, dacă şi numai dacă $H \cup \{\alpha, \beta\} \vDash \Gamma$,
$H \cup \{(\alpha \wedge \beta)\} \vdash \Gamma$, dacă şi numai dacă $H \cup \{(\alpha \wedge \beta)\} \vDash \Gamma$.

Obţinem astfel $H \cup \{\alpha, \beta\} \vDash \Gamma$, dacă şi numai dacă
$H \cup \{(\alpha \wedge \beta)\} \vDash \Gamma$ adică, dacă şi numai dacă
$\left(\bigcap_{\delta\in H} \aleph(\delta)\right) \cap \aleph(\beta) \cap \aleph(\alpha) \subset \bigcup_{\gamma\in\Gamma} \aleph(\gamma)$.

Evident, deoarece $\aleph((\alpha \wedge \beta)) = \aleph(\alpha) \cap \aleph(\beta)$, rezultă în continuare, că incluziunea $\left(\bigcap_{\delta\in H} \aleph(\delta)\right) \cap \aleph(\beta) \cap \aleph(\alpha) \subset \bigcup_{\gamma\in\Gamma} \aleph(\gamma)$ are loc, dacă şi numai dacă
$\left(\bigcap_{\delta\in H} \aleph(\delta)\right) \cap \aleph((\alpha \wedge \beta)) \subset \bigcup_{\gamma\in\Gamma} \aleph(\gamma)$.

Rezultă $I(S_1) = F$, dacă şi numai dacă

$$\begin{aligned} I &\in \left(\left(\bigcap_{\delta\in H} \aleph(\delta)\right) \cap \aleph(\beta) \cap \aleph(\alpha)\right) \cap \left(\Im \setminus \bigcup_{\gamma\in\Gamma} \aleph(\gamma)\right) = \\ &= \left(\bigcap_{\delta\in H} \aleph(\delta)\right) \cap \aleph((\alpha \wedge \beta)) \cap \left(\Im \setminus \bigcup_{\gamma\in\Gamma} \aleph(\gamma)\right), \end{aligned}$$

deci dacă şi numai dacă $I(S) = F$. Obţinem astfel, $I(S_1) = I(S)$.

Regula $G5$: Fie $S_1 = H \cup \{\alpha\} \Longrightarrow \Gamma$, $S = H \Longrightarrow \Gamma \cup \{(\neg\alpha)\}$.

Justificarea relaţiei $I(S_1) = I(S)$ poate fi realizată pe baza unui argument analog argumentului considerat pentru stabilirea concluziei relativ la regula $G1$.

Regula $G7$: Fie $S_1 = H \Longrightarrow \Gamma \cup \{\alpha, \beta\}$, $S = H \Longrightarrow \Gamma \cup \{(\alpha \vee \beta)\}$.

Din *Teorema* 1.5.4., $H \vdash \Gamma \cup \{\alpha, \beta\}$, dacă şi numai dacă

$H \vdash \Gamma \cup \{(\alpha \vee \beta)\}$.

Din *Teorema* 1.7.4. rezultă $H \vdash \Gamma \cup \{\alpha, \beta\}$, dacă şi numai dacă

$H \vDash \Gamma \cup \{\alpha, \beta\}$ deci, dacă şi numai dacă

$\bigcap\limits_{\delta \in H} \aleph(\delta) \subset \bigcup\limits_{\gamma \in \Gamma} \aleph(\gamma) \cup \aleph(\alpha) \cup \aleph(\beta)$.

Deoarece $\aleph(\alpha) \cup \aleph(\beta) = \aleph((\alpha \vee \beta))$, relaţia

$\bigcap\limits_{\delta \in H} \aleph(\delta) \subset \bigcup\limits_{\gamma \in \Gamma} \aleph(\gamma) \cup \aleph(\alpha) \cup \aleph(\beta)$

este echivalentă cu

$\bigcap\limits_{\delta \in H} \aleph(\delta) \subset \bigcup\limits_{\gamma \in \Gamma} \aleph(\gamma) \cup \aleph((\alpha \vee \beta))$.

Obţinem $I(S_1) = F$, dacă şi numai dacă

$$\begin{aligned} I \in & \left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \left(\Im \setminus \left(\bigcup_{\gamma \in \Gamma} \aleph(\gamma) \cup \aleph(\alpha) \cup \aleph(\beta)\right)\right) = \\ = & \left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \left(\Im \setminus \left(\bigcup_{\gamma \in \Gamma} \aleph(\gamma) \cup \aleph((\alpha \vee \beta))\right)\right), \end{aligned}$$

dacă şi numai dacă $I(S) = F$, ceea ce evident implică $I(S_1) = I(S)$.

Regula $G8$: Fie $S_1 = H \cup \{\alpha\} \Longrightarrow \Gamma \cup \{\beta\}$,

$S = H \Longrightarrow \Gamma \cup \{(\alpha \to \beta)\}$.

Procedând în mod similar, rezultă $I(S) = F$, dacă şi numai dacă

$$I \in \left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \left(\Im \setminus \left(\bigcup_{\gamma \in \Gamma} \aleph(\gamma) \cup \aleph((\alpha \to \beta))\right)\right)$$

Deoarece $\aleph((\alpha \to \beta)) = (\Im \setminus \aleph(\alpha)) \cup \aleph(\beta)$, obţinem

$$\begin{aligned} & \left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \left(\Im \setminus \left(\bigcup_{\gamma \in \Gamma} \aleph(\gamma) \cup \aleph((\alpha \to \beta))\right)\right) = \\ & = \left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \aleph(\alpha) \cap \left(\Im \setminus \bigcup_{\gamma \in \Gamma} \aleph(\gamma)\right) \cap (\Im \setminus \aleph(\beta)) = \\ & = \left(\bigcap_{\delta \in H} \aleph(\delta) \cap \aleph(\alpha)\right) \cap \left(\Im \setminus \left(\bigcup_{\gamma \in \Gamma} \aleph(\gamma) \cup \aleph(\beta)\right)\right). \end{aligned}$$

Analog, $I(S_1) = F$, dacă şi numai dacă

$$I \in \left(\bigcap_{\delta \in H} \aleph(\delta) \cap \aleph(\alpha)\right) \cap \left(\Im \setminus \left(\bigcup_{\gamma \in \Gamma} \aleph(\gamma) \cup \aleph(\beta)\right)\right),$$

deci $I(S) = F$, dacă şi numai dacă $I(S_1) = F$, ceea ce implică evident $I(S) = I(S_1)$.

3. Verificăm că pentru orice $I \in \Im$ şi orice exemplu de regulă Gentzen de al doilea tip $\frac{S_1, S_2}{S}$, $I(S_1) \wedge I(S_2) = I(S)$.

Regula $G3:$ Fie $S_1 = H \cup \{\alpha\} \Longrightarrow \Gamma$, $S_2 = H \cup \{\beta\} \Longrightarrow \Gamma$, $S = H \cup \{(\alpha \vee \beta)\} \Longrightarrow \Gamma$.

Pe baza unei tehnici de demonstraţie similare celei utilizate în cazurile precedente obţinem $H \cup \{(\alpha \vee \beta)\} \vdash \Gamma$, dacă şi numai dacă $H \cup \{\alpha\} \vdash \Gamma$ şi $H \cup \{\beta\} \vdash \Gamma$.

De asemenea, $H \cup \{(\alpha \vee \beta)\} \vdash \Gamma$, dacă şi numai dacă $H \cup \{(\alpha \vee \beta)\} \vDash \Gamma$, adică $\left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \aleph((\alpha \vee \beta)) \subset \bigcup_{\gamma \in \Gamma} \aleph(\gamma)$.

Deoarece $\aleph((\alpha \vee \beta)) = \aleph(\alpha) \cup \aleph(\beta)$, rezultă evident,

$$\left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \aleph((\alpha \vee \beta)) \cap \left(\Im \setminus \left(\bigcup_{\gamma \in \Gamma} \aleph(\gamma)\right)\right) =$$
$$= \left(\left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \aleph(\alpha) \cap \left(\Im \setminus \left(\bigcup_{\gamma \in \Gamma} \aleph(\gamma)\right)\right)\right) \cup$$
$$\cup \left(\left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \aleph(\beta) \cap \left(\Im \setminus \left(\bigcup_{\gamma \in \Gamma} \aleph(\gamma)\right)\right)\right).$$

Obţinem $I(S) = F$, dacă şi numai dacă

$$I \in \left(\left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \aleph((\alpha \vee \beta))\right) \cap \left(\Im \setminus \bigcup_{\gamma \in \Gamma} \aleph(\gamma)\right),$$

deci $I \in \left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \aleph(\alpha) \cap \left(\Im \setminus \left(\bigcup_{\gamma \in \Gamma} \aleph(\gamma)\right)\right)$ sau

$$I \in \left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \aleph(\beta) \cap \left(\Im \setminus \left(\bigcup_{\gamma \in \Gamma} \aleph(\gamma)\right)\right).$$

Analog, $I(S_1) = F$, dacă şi numai dacă

$$\left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \aleph(\alpha) \cap \left(\Im \setminus \left(\bigcup_{\gamma \in \Gamma} \aleph(\gamma)\right)\right) = \emptyset$$

şi respectiv $I(S_2) = F$, dacă şi numai dacă

$$\left(\bigcap_{\delta\in H}\aleph(\delta)\right)\cap\aleph(\beta)\cap\left(\Im\setminus\left(\bigcup_{\gamma\in\Gamma}\aleph(\gamma)\right)\right)=\emptyset.$$

Obţinem astfel, $I(S) = F$ dacă şi numai dacă $I(S_1) = F$ sau $I(S_2) = F$, deci $I(S) = I(S_1) \wedge I(S_2)$.

Regula $G4$: Fie $S_1 = H \cup \{\beta\} \Longrightarrow \Gamma$, $S_2 = H \Longrightarrow \Gamma \cup \{\alpha\}$,
$S = H \cup \{(\alpha \to \beta)\} \Longrightarrow \Gamma$.
Procedând într-o manieră asemănătoare, obţinem $H\cup\{(\alpha \to \beta)\} \vdash \Gamma$, dacă şi numai dacă $H \cup \{\beta\} \vdash \Gamma$ şi $H \vdash \Gamma \cup \{\alpha\}$, sau echivalent, $H \cup \{(\alpha \to \beta)\} \vDash \Gamma$, dacă şi numai dacă $H \cup \{\beta\} \vDash \Gamma$ şi $H \vDash \Gamma \cup \{\alpha\}$.

$I(S) = F$, dacă şi numai dacă

$$I\in\left(\left(\bigcap_{\delta\in H}\aleph(\delta)\right)\cap\aleph((\alpha\to\beta))\right)\cap\left(\Im\setminus\bigcup_{\gamma\in\Gamma}\aleph(\gamma)\right).$$

Din proprietăţile anterior stabilite rezultă

$$\begin{aligned}\aleph((\alpha\to\beta)) &= \aleph(((\neg\alpha)\vee\beta)) = \aleph((\neg\alpha))\cup\aleph(\beta)\\ &= (\Im\setminus\aleph(\alpha))\cup\aleph(\beta)\end{aligned}$$

deci $I(S) = F$, dacă şi numai dacă

$$\begin{aligned}I &\in \left(\bigcap_{\delta\in H}\aleph(\delta)\right)\cap(\Im\setminus\aleph(\alpha))\cap\left(\Im\setminus\bigcup_{\gamma\in\Gamma}\aleph(\gamma)\right)\\ &= \left(\bigcap_{\delta\in H}\aleph(\delta)\right)\cap\left(\Im\setminus\left(\aleph(\alpha)\cup\bigcup_{\gamma\in\Gamma}\aleph(\gamma)\right)\right)\end{aligned}$$

sau

$$I\in\left(\bigcap_{\delta\in H}\aleph(\delta)\right)\cap\aleph(\beta)\cap\left(\Im\setminus\bigcup_{\gamma\in\Gamma}\aleph(\gamma)\right).$$

Evident însă, $I(S_1) = F$, dacă şi numai dacă

$$I\in\left(\bigcap_{\delta\in H}\aleph(\delta)\right)\cap\aleph(\beta)\cap\left(\Im\setminus\bigcup_{\gamma\in\Gamma}\aleph(\gamma)\right)$$

respectiv $I(S_2) = F$, dacă şi numai dacă

$$I\in\left(\bigcap_{\delta\in H}\aleph(\delta)\right)\cap\left(\Im\setminus\left(\aleph(\alpha)\cup\bigcup_{\gamma\in\Gamma}\aleph(\gamma)\right)\right).$$

Rezultă $I(S) = F$, dacă şi numai dacă $I(S_1) = F$ sau $I(S_2) = F$, deci $I(S) = I(S_1) \wedge I(S_2)$.

Regula $G6$: Fie $S_1 = H \Longrightarrow \Gamma \cup \{\alpha\}$, $S_2 = H \Longrightarrow \Gamma \cup \{\beta\}$, $S = H \Longrightarrow \Gamma \cup \{(\alpha \wedge \beta)\}$.

Aplicând aceeaşi tehnică de demonstraţie, obţinem $I(S) = F$, dacă şi numai dacă

$$I \in \left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \left(\Im \setminus \left(\aleph((\alpha \wedge \beta)) \cup \bigcup_{\gamma \in \Gamma} \aleph(\gamma)\right)\right).$$

Evident,

$$\begin{aligned}\left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \left(\Im \setminus \left(\aleph((\alpha \wedge \beta)) \cup \bigcup_{\gamma \in \Gamma} \aleph(\gamma)\right)\right) = \\ = \left(\left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \left(\Im \setminus \left(\aleph(\alpha) \cup \bigcup_{\gamma \in \Gamma} \aleph(\gamma)\right)\right)\right) \cup \\ \cup \left(\left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \left(\Im \setminus \left(\aleph(\beta) \cup \bigcup_{\gamma \in \Gamma} \aleph(\gamma)\right)\right)\right).\end{aligned}$$

De asemenea, $I(S_1) = F$, dacă şi numai dacă

$$I \in \left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \left(\Im \setminus \left(\aleph(\alpha) \cup \bigcup_{\gamma \in \Gamma} \aleph(\gamma)\right)\right),$$

respectiv $I(S_2) = F$, dacă şi numai dacă

$$I \in \left(\left(\bigcap_{\delta \in H} \aleph(\delta)\right) \cap \left(\Im \setminus \left(\aleph(\beta) \cup \bigcup_{\gamma \in \Gamma} \aleph(\gamma)\right)\right)\right).$$

Rezultă astfel $I(S) = F$, dacă şi numai dacă $I(S_1) = F$ sau $I(S_2) = F$, deci $I(S) = I(S_1) \wedge I(S_2)$.

4. Demonstrăm prin inducţie asupra înălţimii arborelui de demonstraţie că orice secvent demonstrabil este secvent valid.

Pentru $T(S)$ arbore de demonstraţie, notăm cu r vârful rădăcină şi respectiv cu $h(T(S))$ înălţimea lui.

Fie S secvent demonstrabil, astfel încât $h(T(S)) = 1$. Pe baza definiţiei, rezultă că S este secvent axiomă, deci S este secvent valid.

Presupunem că orice secvent care admite un arbore de demonstraţie de înălţime mai mică sau egală cu n este secvent valid. Fie S secvent, astfel încât $h(T(S)) = n + 1$.

a) Dacă $od(r) = 1$, fie n_1 unicul fiu al rădăcinii şi $\varphi(n_1) = S_1$.

Evident $T_1 = (V(T) \setminus \{r\}, E(T) \setminus \{r\ n_1\})$ este de asemenea un arbore de demonstraţie, $T_1 = T(S_1)$ şi $h(T(S_1)) = n$. Pe baza ipotezei inductive obţinem că S_1 este secvent valid. Deoarece $\frac{S_1}{S}$ este exemplu de regulă Gentzen, pentru orice $I \in \Im$, $I(S) = I(S_1) = T$, deci S este secvent valid.

b) Dacă $od(r) = 2$, fie n_1, n_2 fiii vârfului rădăcina; $\varphi(n_i) = S_i$, $i = 1, 2$. Evident, arborii $T_i = (V(T) \setminus \{r\}, E(T) \setminus \{r\ n_i\})$, $i = 1, 2$ sunt arbori de demonstraţie, $T_i = T(S_i)$ şi $h(T(S_i)) \leq n$, $i = 1, 2$. Din ipoteza inductivă rezultă că $S_i, i = 1, 2$ sunt secvenţi valizi.

Deoarece $\frac{S_1, S_2}{S}$ este exemplu de regula Gentzen, pentru orice $I \in \Im$,

$I(S) = I(S_1) \wedge I(S_2) = T$, deci S este secvent valid.

Observaţie Dacă $\frac{H_1 \Longrightarrow \Gamma_1}{H \Longrightarrow \Gamma}$ este un exemplu de regulă Gentzen de primul tip, atunci $H \models \Gamma$ dacă şi numai dacă $H_1 \models \Gamma_1$.
Dacă $\frac{H_1 \Longrightarrow \Gamma_1, H_2 \Longrightarrow \Gamma_2}{H \Longrightarrow \Gamma}$ este un exemplu de regulă Gentzen de al doilea tip, atunci $H \models \Gamma$, dacă şi numai dacă $H_i \models \Gamma_i$, $i = 1, 2$.

Observaţie Fie $S = H \Rightarrow \Gamma$ secvent încheiat şi nu este secvent axiomă, deci $H \cap \Gamma = \emptyset$, $H \cup \Gamma \subset V$. Considerăm funcţia de adevăr $\hbar : V \to \{T, F\}$ definită prin,

$$\forall p \in V,\ \hbar(p) = \begin{cases} T, \text{ dacă } p \in V \setminus \Gamma \\ F, \text{ altfel} \end{cases}$$

Evident,$\forall \alpha \in H$, $I(\hbar)(\alpha) = T$ şi $\forall \gamma \in \Gamma$, $I(\hbar)(\gamma) = F$, deci

$I(\hbar)(S) = F$.

Cu alte cuvinte, orice secvent încheiat şi care nu este secvent axiomă este falsificabil (nu este secvent valid).

Teorema 1.8.3. (Teorema de completitudine a sistemului deducţiei naturale) Orice secvent valid este secvent demonstrabil .

Demonstraţie Fie S secvent valid şi $T(S)$ arbore de deducţie încheiat pentru S. Dacă toate etichetele vârfurilor terminale ale arborelui $T(S)$ sunt secvenţi axiomă, atunci $T(S)$ este arbore de demonstraţie pentru S, deci S este secvent demonstrabil. Presupunem că există cel puţin un vârf terminal a cărui etichetă S_1 nu este secvent axiomă; fie acesta n_1. Conform observaţiei precedente, există cel puţin o interpretare I, astfel încât $I(S_1) = F$. Fie $n_1, n_2, ..., n_p = r$ drumul de la vârful n_1 la rădăcina arborelui; $\varphi(n_k) = S_k$, $k = 1, ..., p$. Utilizând proprietăţile stabilite in demonstraţia *Teoremei* 1.8.2., rezultă $I(S_k) = F, k = 1, ..., p$, deci $I(S) = F$, ceea ce evident contrazice ipoteza că S este secvent valid.

În concluzie, arborele încheiat $T(S)$ este arbore de demonstraţie pentru S.

5 Verificarea automată a validabilităţii formulelor

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 rezultatele stabilite în cadrul secţiunilor precedente permit reducerea problemei verificării, dacă $H \vdash \Gamma$ la verificarea
validabilităţii/invalidabilităţii unei singure formule. Într-adevăr, dacă $H = \{\alpha_1, ..., \alpha_n\}, \Gamma = \{\beta_1, ..., \beta_m\}$, atunci conform rezultatului stabilit de *Teorema* 1.5.3., $H \vdash \Gamma$, dacă şi numai dacă $\left(\bigwedge_{i=1}^{n} \alpha_i \to \bigvee_{j=1}^{m} \beta_j\right)$ 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 $\{\alpha_1, ..., \alpha_n\} \vdash \{\beta_1, ..., \beta_m\}$ 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 1.9.1. Fie $a, b \in V$, $\alpha = (a \to (b \to (a \wedge b)))$. Deoarece $n = 2$

vor fi generate 2^2 funcţii de adevăr.

$\hbar(a)$	$\hbar(b)$	$I(\hbar)(a \wedge b)$	$I(\hbar)\begin{pmatrix} b \rightarrow \\ (a \wedge b) \end{pmatrix}$	$I(\hbar)\begin{pmatrix} a \rightarrow \\ (b \rightarrow (a \wedge b)) \end{pmatrix}$
T	T	T	T	T
T	F	F	T	T
F	T	F	F	T
F	F	F	T	T

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

5.1 1.9.1. Normalizarea CNF a formulelor limbajului calculului cu propoziţii.

Definiţia 1.9.1.1. Un literal este o propoziţie elementară sau negaţia unei propoziţii elementare. Mulţimea literalilor este notată $V \cup \neg V$.

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

Observaţie Pentru $k \neq \square$, 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 $\aleph(\square)$ revine la reuniune după mulţimea vidă de indici, deci $\aleph(\square) = \emptyset$. Cu alte cuvinte, clauza vidă este invalidabilă.

Definiţia 1.9.1.3. Se numeşte formulă normalizată CNF (**C**onjunctive **N**ormal **F**orm) o structură simbolică de tip conjunctiv $\alpha = \bigwedge_{i=1}^{n} k_i$, unde $\{k_1, ..., k_n\}$ este o mulţime de clauze. Mulţimea clauzelor

$S(\alpha) = \{k_1, ..., k_n\}$ se numeşte reprezentare clauzală pentru formula α. Formula CNF corespunzătoare mulţimii vide de clauze se numeşte formula vidă şi este notată $\emptyset$.

Observaţie 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 1.9.1.4. 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 1.9.1.1. 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 de lungime minimă pentru α.

Rezultă că mulţimea subformulelor lui α este

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

Observaţie 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 1.9.1.1. 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 $((\beta \to \gamma) \wedge (\gamma \to \beta))$;

$$(\beta \leftrightarrow \gamma) \equiv ((\beta \to \gamma) \wedge (\gamma \to \beta)).$$

$T2$. Eliminarea conectivei "$\to$": Fiecare subformulă de tipul $(\beta \to \gamma)$ se substituie cu $((\neg\beta) \vee \gamma)$;

$$(\beta \to \gamma) \equiv ((\neg\beta) \vee \gamma).$$

$T3$. Se aduc negaţiile în faţa literalilor:

ι) Fiecare subformulă de tipul $(\neg(\beta \vee \gamma))$ se substituie cu $((\neg\beta) \wedge (\neg\gamma))$;

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

$\iota\iota$) 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 β;

$$(\neg(\neg\beta)) \equiv \beta.$$

$T5$. 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 1.9.1.2. Fie $a, b \in V$, $\alpha = (((\neg b) \to (\neg a)) \longleftrightarrow (a \to b))$.

Mulţimea subformulelor formulei α este

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

Aplicarea transformării $T1$ determină substituirea subformulei

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

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 subformulelor:

$$\begin{array}{ccc} (((\neg b) \to (\neg a)) \to (a \to b)) & \text{prin} & ((\neg((\neg b) \to (\neg a))) \vee (a \to b)) \\ ((a \to b) \to ((\neg b) \to (\neg a))) & \text{prin} & ((\neg(a \to b)) \vee ((\neg b) \to (\neg a))) \\ (a \to b) & \text{prin} & ((\neg a) \vee b) \\ ((\neg b) \to (\neg a)) & \text{prin} & ((\neg(\neg b)) \vee (\neg a)). \end{array}$$

Rezultă

$$\alpha_2 = \left(\begin{array}{l} ((\neg((\neg(\neg b)) \vee (\neg a))) \vee ((\neg a) \vee b)) \wedge \\ \quad \wedge ((\neg((\neg a) \vee b)) \vee ((\neg(\neg b)) \vee (\neg a))) \end{array} \right).$$

Aplicarea transformării $T3$ determină substituirea subformulelor:

$$\begin{array}{ccc} (\neg((\neg(\neg b))\vee(\neg a))) & \text{prin} & ((\neg(\neg(\neg b)))\wedge(\neg(\neg a))) \\ (\neg((\neg a)\vee b)) & \text{prin} & ((\neg(\neg a))\wedge(\neg b))\,. \end{array}$$

Rezultă

$$\alpha_3 = \left(\begin{array}{l} (((\neg(\neg(\neg b)))\wedge(\neg(\neg a)))\vee((\neg a)\vee b))\wedge \\ \quad \wedge(((\neg(\neg a))\wedge(\neg b))\vee((\neg(\neg b))\vee(\neg a))) \end{array} \right).$$

Aplicarea transformării $T4$ determină substituirea subformulelor:

$$\begin{array}{ccc} (\neg(\neg b)) & \text{prin} & b \\ (\neg(\neg a)) & \text{prin} & a. \end{array}$$

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 subformulelor:

$(((\neg b)\wedge a)\vee((\neg a)\vee b))$ prin $(((\neg b)\vee((\neg a)\vee b))\wedge(a\vee((\neg a)\vee b)))$
$((a\wedge(\neg b))\vee(b\vee(\neg a)))$ prin $((a\vee(b\vee(\neg a)))\wedge((\neg b)\vee(b\vee(\neg a))))\,.$

Rezultă reprezentarea normalizată CNF pentru formula α,

$$\alpha' = \left(\begin{array}{l} (((\neg b)\vee((\neg a)\vee b))\wedge(a\vee((\neg a)\vee b)))\wedge \\ \quad \wedge((a\vee(b\vee(\neg a)))\wedge((\neg b)\vee(b\vee(\neg a)))) \end{array} \right)$$

şi $\alpha \equiv \alpha'$.

Reprezentarea clauzală este

$$S(\alpha) = \left\{ \begin{array}{l} k_1 = ((\neg b)\vee((\neg a)\vee b))\,,\ k_2 = (a\vee((\neg a)\vee b))\,, \\ k_3 = (a\vee(b\vee(\neg a)))\,,\ k_4 = ((\neg b)\vee(b\vee(\neg a))) \end{array} \right\}.$$

Exemplul 1.9.1.3. Fie $m, p, o, c, i \in V, \beta = ((m\wedge p)\to i)$

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

Conform rezultatelor stabilite în secţiunile precedente, verificarea deductibilităţii formulei β sub familia de ipoteze H, corespunde verificării proprietăţii de a fi tautologie a formulei $((\alpha_1\wedge\alpha_2)\to\beta)$ sau echivalent la demonstrarea invalidabilităţii formulei

$$\gamma = ((\alpha_1\wedge\alpha_2)\wedge(\neg\beta))\,.$$

Deoarece numărul propoziţiilor elementare $n = 5$, verificările pot fi efectuate după modelul din *Exemplul* 1.9.1. pe baza generării a 2^5 funcţii de adevăr.

Dată fiind structura de tip conjunctiv, o reprezentare normalizată CNF a formulei γ este $\gamma' = \left((\alpha_1' \wedge \alpha_2') \wedge (\neg\beta)'\right)$, unde $\alpha_1', \alpha_2', (\neg\beta)'$ sunt reprezentări normalizate CNF corespunzătoare respectiv formulelor $\alpha_1, \alpha_2, (\neg\beta)$. De asemenea, $S(\gamma) = S(\alpha_1) \cup S(\alpha_2) \cup S((\neg\beta))$ este o reprezentare clauzală pentru γ, unde $S(\alpha_1)$, $S(\alpha_2)$, $S((\neg\beta))$ sunt reprezentări clauzale respectiv pentru $\alpha_1, \alpha_2, (\neg\beta)$.

Aplicând construcţia descrisă în demonstraţia *Teoremei* 1.9.1.1., rezultă

$$\alpha_1 = (p \rightarrow (o \wedge (\neg c))) \equiv ((\neg p) \vee (o \wedge (\neg c))) \equiv \\ \equiv (((\neg p) \vee o) \wedge ((\neg p) \vee (\neg c)))$$

deci

$$S(\alpha_1) = \{k_1 = ((\neg p) \vee o),\ k_2 = ((\neg p) \vee (\neg c))\}.$$

$$\alpha_2 = ((m \wedge (\neg c)) \rightarrow i) \equiv ((\neg(m \wedge (\neg c))) \vee i) \equiv \\ \equiv (((\neg m) \vee (\neg(\neg c))) \vee i) \equiv (((\neg m) \vee c) \vee i)$$

deci

$$S(\alpha_2) = \{k_3 = (((\neg m) \vee c) \vee i)\}.$$

$$(\neg\beta) = (\neg((m \wedge p) \rightarrow i)) \equiv (\neg(m \wedge p))(\neg((\neg(m \wedge p)) \vee i)) \equiv \\ \equiv (\neg(\neg(m \wedge p))) \wedge (\neg i) \equiv ((m \wedge p) \wedge (\neg i))$$

deci

$$S((\neg\beta)) = \{k_4 = m,\ k_5 = p,\ k_6 = (\neg i)\}.$$

Pentru formula γ rezultă reprezentarea clauzală

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

Observaţie Deoarece pentru orice formulă α, $(\alpha \wedge \alpha) \equiv \alpha$, $(\alpha \vee \alpha) \equiv \alpha$, obţinem următoarele reguli de simplificare:

1. Prin eliminarea duplicatelor literalilor din clauza k rezultă clauza k' şi $\aleph(k) = \aleph(k')$.

2. Prin eliminarea duplicatelor clauzelor într-o reprezentare clauzală $S(\alpha)$ rezultă reprezentarea clauzală $S'(\alpha)$ şi

$\aleph(\alpha) = \aleph(S(\alpha)) = \aleph(S'(\alpha))$.

De asemenea, deoarece pentru orice formule α, β, $(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$, $(\alpha \vee \beta) \equiv (\beta \vee \alpha)$, rezultă

3. Dacă $\alpha = \bigwedge\limits_{i=1}^{n} k_i; n \geq 1$ este o reprezentare normalizată CNF, atunci pentru orice σ permutare a mulţimii $\{1, 2, ..., n\}$, $\alpha \equiv \alpha_\sigma$, unde $\alpha_\sigma = \bigwedge\limits_{i=1}^{n} k_{\sigma(i)}$.

4. Fie clauza $k = \bigvee\limits_{i=1}^{n} L_i$; $n \geq 1$. Pentru orice σ permutare a mulţimii $\{1, 2, ..., n\}$, $k \equiv k_\sigma$, unde $k_\sigma = \bigvee\limits_{i=1}^{n} L_{\sigma(i)}$.

Bazat pe aceste observaţii, în continuare vom presupune că nu există duplicate ale literalilor în clauze şi nici duplicate ale aceleiaşi clauze într-o reprezentare clauzală. De asemenea, putem permuta arbitrar literalii unei clauze, respectiv clauzele într-o reprezentare normalizată CNF. Pentru simplificarea notaţiilor convenim să eliminăm parantezele în expresiile clauzelor, cu excepţia reprezentărilor literalilor negativi.

Lema 1.9.1.1

1) Clauza $k = \bigvee\limits_{i=1}^{n} L_i; n \geq 1$ este tautologie, dacă şi numai dacă există cel puţin o pereche de literali complementari în mulţimea $\{L_1, ..., L_n\}$.

2) Dacă $S(\alpha)$ este o reprezentare clauzală, $k \in S(\alpha)$ şi k este tautologie, atunci $\aleph(S(\alpha)) = \aleph(S(\alpha) \setminus \{k\})$.

Demonstraţie

1) Presupunem că $\{p, (\neg p)\} \subset \{L_1, ..., L_n\}$, $p \in V$.
Rezultă $n \geq 2$. Fie σ o permutare a mulţimii $\{1, 2, ..., n\}$, astfel încât $L_{\sigma(1)} = p$, $L_{\sigma(2)} = (\neg p)$.

Conform observaţiei precedente, $k \equiv k_\sigma$, unde
$k_\sigma = \bigvee\limits_{i=1}^{n} L_{\sigma(i)} = L_{\sigma(1)} \vee L_{\sigma(2)} \vee k'$, $k' = \bigvee\limits_{i=3}^{n} L_{\sigma(i)}$.

Pentru orice $I \in \Im$ obţinem

$I(k) = I(k_\sigma) = I(p) \vee I((\neg p)) \vee I(k') = T$, deci k este tautologie. Dacă nu există literali complementari în mulţimea $\{L_1, ..., L_n\}$, fie $\{p_1, ..., p_k\}$ şi $\{(\neg q_1), ..., (\neg q_r)\}$ submulţimile literalilor pozitivi şi respectiv negativi;

$$\{p_1, ..., p_k\} \cup \{(\neg q_1), ..., (\neg q_r)\} = \{L_1, ..., L_n\}$$

Definim funcţia de adevăr $\hbar : V \to \{T, F\}$, $\forall s \in V$,

$$\hbar(s) = \begin{cases} F, \text{ dacă } s \in \{p_1, ..., p_k\} \\ T, \text{ dacă } s \in V \setminus \{p_1, ..., p_k\}. \end{cases}$$

Evident, $I(\hbar)(L_i) = F$, $1 \leq i \leq n$ deci $I(\hbar)(k) = F$. Rezultă că pentru orice clauză tautologie $k = \bigvee_{i=1}^{n} L_i$; $n \geq 1$, există cel puţin o pereche de literali complementari în mulţimea $\{L_1, ..., L_n\}$.

2) Fie $S(\alpha) = \{k_1, ..., k_n\}$ reprezentare clauzală, astfel încât există k clauză tautologie, $k \in S(\alpha)$. Deoarece $\aleph(k) = \Im$, rezultă

$$\aleph(S(\alpha)) = \bigcap_{k^* \in S(\alpha)} \aleph(k^*) = \aleph(k) \cap \left(\bigcap_{k^* \in S(\alpha) \setminus \{k\}} \aleph(k^*) \right) = \bigcap_{k^* \in S(\alpha) \setminus \{k\}} \aleph(k^*) = \aleph(S(\alpha) \setminus \{k\}).$$

Observaţie Concluziile stabilite de *Lema* 1.9.1.1. permit detectarea şi eliminarea clauzelor tautologii dintr-o reprezentare clauzală, ceea ce în particular oferă posibilitatea presupunerii, fără restrângerea generalităţii, că reprezentările clauzale nu conţin tautologii. O reprezentare clauzală care nu conţine tautologii este referită ca reprezentare clauzală liberă de tautologii.

Obţinerea unei reprezentări clauzale corespunzătoare unei formule date α poate fi realizată prin aplicarea procedurii descrise în secţiunea 1.6 pentru construirea unui arbore de deducţie încheiat pentru secventul

$S = \emptyset \Rightarrow \{\alpha\}$.

Teorema 1.9.1.2. Fie $T = (V(T), E(T))$ arbore încheiat pentru

$S = \emptyset \Rightarrow \{\alpha\}$ şi fie $\partial T = \{n_1, ..., n_p\}$ mulţimea vârfurilor terminale din T (frontiera arborelui). Pentru fiecare vârf $n_i \in \partial T$, $1 \leq i \leq p$, considerăm clauza

$k_i = \bigvee_{j=1}^{l_i} \left(\neg s_j^{(i)} \right) \vee \bigvee_{j=1}^{r_i} d_j^{(i)}$, unde

$\varphi(n_i) = \left\{ s_1^{(i)}, ..., s_{l_i}^{(i)} \right\} \Rightarrow \left\{ d_1^{(i)}, ..., d_{r_i}^{(i)} \right\}$

este secventul etichetă a vârfului n_i, $1 \leq i \leq p$.

Mulţimea $S(\alpha) = \{k_1, ..., k_p\}$ este o reprezentare clauzală pentru α şi

$\aleph(S(\alpha)) = \aleph(\alpha)$.

Demonstraţie Dacă T este arbore de demonstraţie, atunci $\varphi(n_i)$, $1 \leq i \leq p$ sunt secvenţi axiomă, deci din *Lema* 1.9.1.1. rezultă că toate clauzele

k_i, $1 \leq i \leq p$ sunt tautologii, deci $\aleph(S(\alpha)) = \Im$.

Din *Teorema* 1.8.2. rezultă S este secvent valid, deci $\aleph(S) = \Im$. În continuare, utilizând rezultatul stabilit de *Teorema* 1.8.1 obţinem $\emptyset \models \{\alpha\}$, deci $\emptyset \models \alpha$, adică $\aleph(\alpha) = \Im$. În concluzie, dacă T este arbore de demonstraţie atunci $\aleph(S(\alpha)) = \aleph(\alpha) = \Im$. Dacă T nu este arbore de demonstraţie, atunci T este arbore contraexemplu.

Din argumentele utilizate în demonstraţiile *Teoremei* 1.8.2. şi

Teoremei 1.8.3. rezultă în particular că orice interpretare, care falsifică secventul etichetă a unui vârf terminal, falsifică secventul etichetă a rădăcinii şi reciproc orice interpretare, care falsifică secventul etichetă a rădăcinii, falsifică cel puţin un secvent etichetă a unui vârf terminal. Cu alte cuvinte, rezultă $\aleph(S) = \bigcap_{i=1}^{p} \aleph(\varphi(n_i))$.

Deoarece pentru orice $I \in \Im$, $I(\varphi(n_i)) = T$, dacă şi numai dacă există j, $1 \leq j \leq l_i$ şi $I\left(s_j^{(i)}\right) = F$ sau există j, $1 \leq j \leq r_i$ şi $I\left(d_j^{(i)}\right) = T$ rezultă, $I(\varphi(n_i)) = T$, dacă şi numai dacă $I(k_i) = T$.

Obţinem astfel, $\aleph(S(\alpha)) = \bigcap_{i=1}^{p} \aleph(\varphi(n_i))$, deci $\aleph(S) = \aleph(S(\alpha))$.

Utilizând un argument similar obţinem, $\aleph(S) = \aleph(\emptyset \Rightarrow \{\alpha\}) = \aleph(\alpha)$, deci în final rezultă $\aleph(S(\alpha)) = \aleph(\alpha)$.

Corolarul 1.9.1.1 Utilizând rezultatul stabilit de *Lema* 1.9.1.1 rezultă că, dacă $\partial T = \{n_1, ..., n_p\}$ este frontiera unui arbore încheiat pentru secventul

$S = \emptyset \Rightarrow \{\alpha\}$ şi $\left\{n_{i_1}, ..., n_{i_q}\right\}$ sunt vârfurile din ∂T ale căror etichete nu sunt secvenţi axiomă, atunci $S'(\alpha) = \left\{k_{i_1}, ..., k_{i_q}\right\}$ este o reprezentare clauzală liberă de tautologii pentru formula α.

Exemplul 1.9.1.3. Fie $a, b, c \in V$ şi formula

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

În *Figura* 1.9.1. este reprezentat un arbore încheiat pentru secventul

$S = \emptyset \Rightarrow \{\alpha\}$.

$T = (\{r, n_1, ..., n_{12}\}, E(T))$

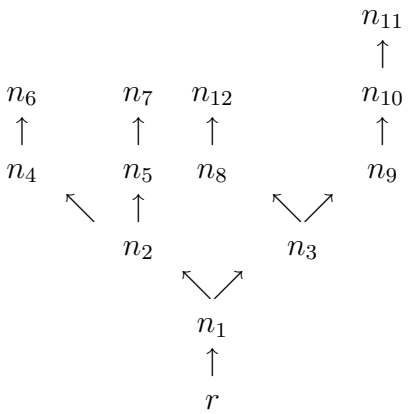


Figura 1.9.1.

unde

$\varphi(r) = \emptyset \Rightarrow \{(((a \to b) \vee ((\neg a) \to c)) \to (b \vee c))\}$

$\varphi(n_1) = \{((a \to b) \vee ((\neg a) \to c))\} \Rightarrow \{(b \vee c)\}$
$\varphi(n_2) = \{(a \to b)\} \Rightarrow \{(b \vee c)\}$
$\varphi(n_3) = \{((\neg a) \to c)\} \Rightarrow \{(b \vee c)\}$
$\varphi(n_4) = \{b\} \Rightarrow \{(b \vee c)\}$
$\varphi(n_5) = \emptyset \Rightarrow \{a, (b \vee c)\}$
$\varphi(n_6) = \{b\} \Rightarrow \{b, c\}$ (secvent axiomă)
$\varphi(n_7) = \emptyset \Rightarrow \{a, b, c\}$
$\varphi(n_8) = \{c\} \Rightarrow \{(b \vee c)\}$
$\varphi(n_9) = \emptyset \Rightarrow \{(\neg a), (b \vee c)\}$
$\varphi(n_{10}) = \{a\} \Rightarrow \{(b \vee c)\}$
$\varphi(n_{11}) = \{a\} \Rightarrow \{b, c\}$
$\varphi(n_{12}) = \{c\} \Rightarrow \{b, c\}$ (secvent axiomă).

Mulţimea frontieră este $\partial T = \{n_6, n_7, n_{11}, n_{12}\}$, deci
$S(\alpha) = \{k_1, k_2, k_3, k_4\}$, $S'(\alpha) = \{k_2, k_3\}$ unde,
$k_1 = (\neg b) \vee b \vee c$ (clauză tautologie) − asociată vârfului n_6
$k_2 = a \vee b \vee c$−asociată vârfului n_7
$k_3 = (\neg a) \vee b \vee c$−asociată vârfului n_{11}
$k_4 = (\neg c) \vee b \vee c$ (clauză tautologie) − asociată vârfului n_{12}.

Cu alte cuvinte, $\alpha = (((a \to b) \vee ((\neg a) \to c)) \to (b \vee c)) \equiv \alpha'$,unde
$\alpha' = (((a \vee b) \vee c) \wedge (((\neg a) \vee b) \vee c))$ este formă normală conjunctivă.
Rezultă $\aleph(\alpha) = \aleph(((a \vee b) \vee c)) \cap \aleph((((\neg a) \vee b) \vee c)) = \aleph((b \vee c))$.

5.2 1.9.2 Metoda Davis-Putnam pentru verificarea validabilităţii formulelor limbajului calculului cu propoziţii.

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 1.9.2.1 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.

Observaţie 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 1.9.2.2 Fie $S(\alpha)$ o reprezentare clauzală liberă de tautologii şi λ literal. Fie submulţimile de clauze

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

Observaţie 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, α_{λ}^{+}, α_{λ}^{-}, α_{λ}^{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 1.9.2.1 Fie $S(\alpha) = \{k_1, k_2, k_3, k_4, k_5, k_6\}$, unde $k_1 = (\neg p) \vee o$, $k_2 = (\neg p) \vee (\neg c)$, $k_3 = (\neg m) \vee c \vee i$, $k_4 = m$, $k_5 = p$, $k_6 = (\neg i)$.

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

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

Observaţie Dacă $S(\alpha)$ 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 1.9.2.3 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 1.9.2.1 (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 $(k \setminus \lambda) \rangle \lambda \langle$ 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 1.9.2.1. Dacă $S(\alpha)$ este o reprezentare clauzală liberă de tautologii, atunci $\aleph(S(\alpha)) \subset \bigcap_{\lambda \in V \cup \neg V} (\aleph(NEG_\lambda(\alpha)) \cup \aleph(POS_\lambda(\alpha)))$.

Justificarea relaţiei este imediată şi rezultă din demonstraţia *Teoremei* 1.9.2.1.

Corolarul 1.9.2.2. (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

$\lambda \setminus \lambda = \square \in POS_{\lambda}(\alpha)$, deci cum clauza vidă este invalidabilă, rezultă $POS_{\lambda}(\alpha)$ invalidabilă. Dacă $S(\alpha)$ este validabilă, utilizând rezultatul stabilit de *Teorema* 1.9.2.1., rezultă

$NEG_{\lambda}(\alpha)$ este validabilă. Reciproc, dacă $NEG_{\lambda}(\alpha)$ este validabilă, atunci din *Teorema* 1.9.2.1 obţinem că $S(\alpha)$ este validabilă.

Corolarul 1.9.2.3 (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* 1.9.2.1 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* 1.9.2.1 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 $\lambda_1, ..., \lambda_n$ literalii care apar în clauzele din $S(\alpha)$. Fie reprezentările clauzale
$\{\gamma_{i_1...i_k} \mid i_1, ..., i_k \in \{0,1\},\ 1 \leq k \leq n,\ \}$ rezultate astfel,

$$\begin{aligned} \gamma_0 &= NEG_{\lambda_1}(\alpha), \gamma_1 = POS_{\lambda_1}(\alpha), \gamma_{i_1...i_{k-1}0} = \\ &= NEG_{\lambda_k}\left(\gamma_{i_1...i_{k-1}}\right), \gamma_{i_1...i_{k-1}1} = POS_{\lambda_k}\left(\gamma_{i_1...i_{k-1}}\right),\ k = 2, ..., n. \end{aligned}$$

Deoarece în clauzele reprezentării clauzale $\gamma_{i_1...i_k}$ nu mai apar literalii

$\lambda_1, ..., \lambda_k$, obţinem că toate reprezentările clauzale

$\gamma_{i_1...i_n}, i_1, ..., i_n \in \{0,1\}$ conţin numai clauze fără literali. Rezultă că pentru orice
$i_1, ..., i_n \in \{0,1\}, \gamma_{i_1...i_n} \in \{\{\square\}, \emptyset\}$.

Dacă $\gamma_{i_1...i_n} = \{\square\}$ pentru toţi $i_1, ..., i_n \in \{0, 1\}$, atunci pe baza concluziei stabilite de *Teorema* 1.9.2.1, toate reprezentările clauzale $\gamma_{i_1...i_{n-1}}$ rezultă invalidabile. Iterând acelaşi argument, obţinem că toate reprezentările clauzale $\gamma_{i_1...i_k}$, $i_1, ..., i_k \in \{0, 1\}$, $1 \leq k \leq n$ sunt invalidabile, deci în final $S(\alpha)$ rezultă invalidabilă. Dacă există $i_1, ..., i_n \in \{0, 1\}$, astfel încât $\gamma_{i_1...i_n} = \emptyset$, atunci din *Teorema* 1.9.2.1, $\gamma_{i_1...i_{n-1}}$ rezultă validabilă. Iterând acelaşi argument, obţinem că toate reprezentările clauzale $\gamma_{i_1...i_k}$, $1 \leq k \leq n$ sunt validabile, deci în final $S(\alpha)$ rezultă validabilă.

Metoda descrisă corespunde generării unui arbore binar complet cu vârfurile etichetate cu reprezentările clauzale $\gamma_{i_1...i_k}$, $i_1, ..., i_k \in \{0, 1\}$, $1 \leq k \leq n$ 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* 1.9.2.2 şi *Corolarul* 1.9.2.3, 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 clauza 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
```

Observaţie 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 1.9.2.2. 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 $('invalidabilă')$, sw$\leftarrow true$
$\Rightarrow$stop

Observaţie Deoarece algoritmul decide că reprezentarea clauzală $S(\alpha)$ este invalidabilă, rezultă $H \vdash \beta$, unde

$$\begin{aligned} \beta &= ((m \wedge p) \rightarrow i), \\ H &= \{\alpha_1 = (p \rightarrow (o \wedge (\neg c))), \alpha_2 = ((m \wedge (\neg c)) \rightarrow i)\} \end{aligned}$$

(*Exemplul* 1.9.1.3)

Exemplul 1.9.2.3 .Evoluţia determinată de procedura DvP pentru datele de intrare $S(\alpha) = \{k_1, k_2, k_3, k_4\}$, unde

$$k_1 = a \vee (\neg b) \vee c,\ k_2 = (\neg a) \vee (\neg c),\ k_3 = (\neg c) \vee b,\ k_4 = (\neg b) \vee a$$

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'

5.3 1.9.3 Metoda bazată pe principiul rezoluţiei pentru verificarea validabilităţii formulelor limbajului calculului cu propoziţii.

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 $\frac{(\alpha\rightarrow\beta),((\neg\alpha)\rightarrow\gamma)}{(\beta\vee\gamma)} REZ$ (*Aplicaţia* 1.4.9).

Observaţie Deoarece $((\alpha \rightarrow \beta) \rightarrow (((\neg\alpha) \rightarrow \gamma) \rightarrow (\beta \vee \gamma))) \in T_h$, aplicând teorema deducţiei obţinem $\{(\alpha \rightarrow \beta), ((\neg\alpha) \rightarrow \gamma)\} \vdash (\beta \vee \gamma)$ sau echivalent $\{(\alpha \rightarrow \beta), ((\neg\alpha) \rightarrow \gamma)\} \vdash \{(\beta \vee \gamma)\}$ şi care din punct de vedere semantic revine la $\{(\alpha \rightarrow \beta), ((\neg\alpha) \rightarrow \gamma)\} \vDash \{(\beta \vee \gamma)\}$,

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

Deoarece $\aleph((\alpha \rightarrow \beta)) = (\Im \setminus \aleph(\alpha)) \cup \aleph(\beta) = \aleph((\neg\alpha)) \cup \aleph(\beta) =$
$\aleph(((\neg\alpha) \vee \beta))$ şi
$\aleph(((\neg\alpha) \rightarrow \gamma)) = (\Im \setminus \aleph((\neg\alpha))) \cup \aleph(\gamma) = \aleph(\alpha) \cup \aleph(\gamma) = \aleph((\alpha \vee \gamma))$
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 1.9.3.1 Fie k_1, k_2 clauze şi λ literal. Clauzele k_1, k_2 sunt $\lambda-$rezolubile, dacă $k_1\langle\lambda\rangle$ şi $k_2\ \langle(\neg\lambda)\rangle$. Clauza $\text{rez}_\lambda(k_1, k_2) = (k_1 \setminus \lambda) \vee (k_2 \setminus (\neg\lambda))$ este

$\lambda-$rezolventa perechii de clauze (k_1, k_2). Clauzele k_1, k_2 se numesc clauze parentale ale rezolventei.

Observaţie În particular, pentru $\alpha = \lambda$, $\beta = (k_2 \setminus (\neg\lambda))$,

$\gamma = (k_1 \setminus \lambda)$, din observaţia precedentă rezultă $\{k_1, k_2\} \vDash \text{rez}_\lambda(k_1, k_2)$ sau echivalent, $\aleph(k_1) \cap \aleph(k_2) \subset \aleph(rez_\lambda(k_1, k_2))$. Evident, dacă nici una 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 1.9.3.1 Fie $k_1 = a \vee (\neg b) \vee c$, $k_2 = (\neg a) \vee (\neg c)$,
$k_3 = (\neg b) \vee (\neg a)$.
Clauzele k_1, k_2 sunt $a-$rezolubile şi $c-$rezolubile.
$\text{rez}_a(k_1, k_2) = (\neg b) \vee c \vee (\neg c)$
$\text{rez}_c(k_1, k_2) = a \vee (\neg b) \vee (\neg a)$
Clauzele k_1, k_3 sunt $a-$rezolubile

$\text{rez}_a(k_1, k_3) = (\neg b) \vee c \vee (\neg b) \equiv c \vee (\neg b)$

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 1.9.3.2 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 1.9.3.2 Dacă

$$S(\alpha) = \{(\neg p) \vee o, (\neg p) \vee (\neg c), (\neg m) \vee c \vee i, m, p, (\neg i)\},$$

atunci,

$REZ_p(\alpha) = \{(\neg m) \vee c \vee i, m, (\neg i), o, (\neg c)\}$

$REZ_c(REZ_p(\alpha)) = \{(\neg m) \vee i, m, (\neg i), o\}$

$REZ_i(REZ_c(REZ_p(\alpha))) = \{(\neg m), m, o\}$

$REZ_m(REZ_i(REZ_c(REZ_p(\alpha)))) = \{\square, o\}$.

Teorema 1.9.3.1 (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 $I \in \bigcap_{k \in S(\alpha)} \aleph(k)$.

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 $\{k_1, k_2\} \vDash \text{rez}_\lambda(k_1, k_2)$, 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 1.9.2.1, rezultă $S(\alpha)$ validabilă. Dacă $I \notin \aleph(POS_\lambda(\alpha))$, atunci există $k \in POS_\lambda(\alpha)$, astfel încât $I(k) = F$. In 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 $k^* = k_2 \backslash (\neg\lambda)$ 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

$$I(rez_\lambda(k_1, k_2)) = I((k_1 \setminus \lambda) \vee (k_2 \setminus (\neg\lambda))) =$$
$$= I((k_1 \setminus \lambda)) \vee I((k_2 \setminus (\neg\lambda))) = I(k) \vee I(k^*) = F \vee I(k^*) = I(k^*),$$

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* 1.9.2.1, 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 1.9.3.1 Dacă $S(\alpha)$ este o reprezentare clauzală, atunci

$$\begin{aligned} \aleph(S(\alpha)) &\subset \bigcap_{\lambda \in V \cup \neg V} \aleph(REZ_\lambda(\alpha)) \subset \\ &\subset \bigcap_{\lambda \in V \cup \neg V} (\aleph(NEG_\lambda(\alpha)) \cup \aleph(POS_\lambda(\alpha))). \end{aligned}$$

Demonstraţie Din argumentele utilizate în demonstraţia *Teoremei* 1.9.3.1 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 1.9.3.3 Fie $S(\alpha)$ o reprezentare clauzală liberă de tautologii.

Secvenţa de clauze $k_1, ..., k_n$ este o $S(\alpha)-$derivare rezolutivă dacă pentru orice i, $1 \leq i \leq n$, este îndeplinită una din condiţiile:

ι) $k_i \in S(\alpha)$;

$\iota\iota$) 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 $k_1, ..., k_n$ este o $S(\alpha)-$respingere rezolutivă, dacă $k_n = \square$.

Teorema 1.9.3.2 (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 $k_1, ..., k_n$, $S(\alpha) \models k_i$, $1 \leq i \leq n$.

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 $k_1, ..., k_n = \square$.

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 $\lambda_1, ..., \lambda_n$ literalii care au ocurenţe în clauzele mulţimii $S(\alpha)$. Considerăm secvenţa de reprezentări clauzale $S_0(\alpha) = S(\alpha)$,

$S_i(\alpha) = REZ_{\lambda_i}(S_{i-1}(\alpha))$, $i = 1, ..., n$.

Presupunem de asemenea că $S_i(\alpha)$, $i = 1, ..., n$ sunt libere de tautologii.

Rezultă că pentru fiecare $i = 1, ..., n$, literalii $\lambda_1, ..., \lambda_i$ 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* 1.9.3.1 rezultă $S_i(\alpha)$ validabilă, $i = 0, ..., n$, 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, $i = 0, ..., n$.

În ipoteza că $S(\alpha)$ este invalidabilă rezultă $S_n(\alpha) = \{\square\}$.

Dacă $S_i(\alpha) = \{\gamma_i^1, ..., \gamma_i^{m_i}\}$. $i = 0, ..., n$, unde $m_n = 1$ şi $\gamma_n^1 = \square$, atunci $\gamma_1^1, ..., \gamma_1^{m_1}$, $\gamma_2^1, ..., \gamma_2^{m_2}$, ..., $\gamma_{n-1}^1, ..., \gamma_{n-1}^{m_{n-1}}$, γ_n^1 este o

$S(\alpha)$ −derivare rezolutivă.

Deoarece $\gamma_n^1 = \square$, obţinem în final că

$\gamma_1^1, ..., \gamma_1^{m_1}$, $\gamma_2^1, ..., \gamma_2^{m_2}$, ..., $\gamma_{n-1}^1, ..., \gamma_{n-1}^{m_{n-1}}$, $\gamma_n^1 = \square$

este o $S(\alpha)$ −respingere rezolutivă.

Rezultatul stabilit de *Teorema* 1.9.3.2 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 număr 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 $S(\alpha)$;
EliminaTautologii($S(\alpha)$);
EliminaDuplicate($S(\alpha)$);
$\gamma \leftarrow S(\alpha)$; $sw \leftarrow false$;
repeat
if $\gamma = \emptyset$ then
write ($'validabilă'$);
sw$\leftarrow true$
else
if $\square \in \gamma$ then
write ($'invalidabilă'$);
sw$\leftarrow true$
else
if ($există$ λ clauza unitară) *or*
($există$ λ literal pur)
then
$\gamma \leftarrow NEG_{\lambda}(\gamma)$
else
alege(λ literal);
$\gamma \leftarrow REZ_{\lambda}(\gamma)$;
EliminaTautologii(γ);
EliminaDuplicate(γ);
endif
endif
endif
until sw
end

Observaţie 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 1.9.3.3 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 $(\neg c) \vee b \vee (\neg c)$

$\gamma \leftarrow \{(\neg b) \vee c, (\neg c) \vee b\}$.

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.

5.4 1.9.4 Demonstratorul Gentzen pentru verificarea validabilităţii formulelor limbajului calculului cu propoziţii.

Principalul dezavantaj comun metodei Davis-Putnam şi metodei bazate pe rezoluţie în verificarea unei deducţii $H \vdash \Gamma$, unde $H = \{\alpha_1, ..., \alpha_n\}$, $\Gamma = \{\beta_1, ..., \beta_m\}$ rezidă din necesitatea preprocesării formulei $\gamma = \bigwedge_{i=1}^{n} \alpha_i \wedge \bigwedge_{j=1}^{m} (\neg \beta_j)$ pentru obţinerea reprezentării clauzale $S(\gamma)$. Conform rezultatelor stabilite în secţiunile precedente, problema verificării $H \vdash \Gamma$ este echivalentă cu $H \vDash \Gamma$

şi revine la verificarea validabilităţii formulei $\gamma = \bigwedge_{i=1}^{n} \alpha_i \wedge \bigwedge_{j=1}^{m} (\neg\beta_j)$. Din prezentarea efectuată în secţiunea 1.6, calculul cu secvenţi este aplicabil numai în cazul în care formulele din mulţimile antecedent şi succedent conţin în exclusivitate conective din mulţimea $L \setminus \{\leftrightarrow\}$, în timp ce formulele din $H \cup \Gamma$ pot eventual să conţină şi conectiva "$\leftrightarrow$".

Aplicând formulelor din H, Γ transformarea constând în substituirea construcţiilor de tipul $(\beta \leftrightarrow \gamma)$ subformule ale formulelor din $H \cup \Gamma$ respectiv prin construcţiile semantic echivalente
$((\beta \rightarrow \gamma) \wedge (\gamma \rightarrow \beta))$, rezultă mulţimile H_1, Γ_1.
Evident $\aleph(H) = \aleph(H_1)$ şi $\aleph(\Gamma) = \aleph(\Gamma_1)$, deci $H \vDash \Gamma$, dacă şi numai dacă $H_1 \vDash \Gamma_1$.

Deoarece $H \vdash \Gamma$, dacă şi numai dacă $H \vDash \Gamma$, rezultă $H \vdash \Gamma$, dacă şi numai dacă $H_1 \vdash \Gamma_1$.

Concluzia *Teoremei* 1.8.1 permite reformularea problemei $H \vDash \Gamma$ în termenii validităţii secventului $S = H \Rightarrow \Gamma$, respectiv concluziile

Teoremelor 1.8.2 şi 1.8.3 stabilind corespondenţa dintre proprietăţile de validitate şi demonstrabilitate ale unui secvent finit.

Concluzionând, obţinem că $\{\alpha_1, ..., \alpha_n\} \vdash \{\beta_1, ..., \beta_m\}$ dacă şi numai dacă secventul $S = \{\alpha_1, ..., \alpha_n\} \Rightarrow \{\beta_1, ..., \beta_m\}$ este secvent demonstrabil şi ca atare, problema verificării $\{\alpha_1, ..., \alpha_n\} \vdash \{\beta_1, ..., \beta_m\}$ poate fi rezolvată pe baza unei tentative de construcţie a unui arbore de demonstraţie pentru secventul S.

Tentativa de construcţie a unui arbore de demonstratie $T(S)$ pentru un secvent dat S este realizată prin expandări succesive aplicate vârfurilor terminale etichetate cu secvenţi care nu sunt încheiaţi din arborele de deducţie curent T. La momentul iniţial, arborele de deducţie are numai vârful rădăcină etichetat cu S. Pasul de expandare este aplicat vârfului n_1 selectat de pe frontiera ∂T a arborelui curent generat T şi constă în selectarea unei formule $\alpha \notin V$ din antecedentul sau succedentul secventului etichetă S_1 a vârfului şi identificarea regulei Gentzen asociate. Dacă regula Gentzen este de primul tip, fie aceasta $\frac{S_2}{S_1}$, atunci arborele T este extins prin adăugarea unui vârf nou n_2 ca fiu al vârfului n_1 cu eticheta $\varphi(n_2) = S_2$. Dacă S_2 este secvent încheiat şi nu este axiomă, atunci noul arbore este un arbore contraexemplu pentru S, deci S nu este secvent valid şi în consecinţă, $\{\alpha_1, ..., \alpha_n\} \nvdash \{\beta_1, ..., \beta_m\}$.

Dacă regula Gentzen este de al doilea tip, fie aceasta $\frac{S_2, S_3}{S_1}$, atunci arborele T este extins prin adăugarea noilor vârfuri n_2, n_3 ca fii ai vârfului n_1, etichetele noilor vârfuri fiind $\varphi(n_2) = S_2$ şi respectiv $\varphi(n_3) = S_3$. Dacă cel puţin unul dintre secvenţii S_2, S_3 este secvent încheiat şi nu este axiomă, atunci noul arbore este arbore contraexemplu pentru S, deci $\{\alpha_1, ..., \alpha_n\} \nvdash \{\beta_1, ..., \beta_m\}$.

Selectarea vârfului pentru aplicarea pasului de expandare este efectuată astfel:

ι) Dacă toate vârfurile din ∂T au etichete secvenţi axiomă, atunci T este un arbore de demonstraţie pentru S, deci $\{\alpha_1, ..., \alpha_n\} \vdash \{\beta_1, ..., \beta_m\}$ şi se decide terminarea procesului de expandare.

$\iota\iota$) În caz contrar, selectează pentru aplicarea pasului de expandare unul din vârfurile din ∂T a cărui etichetă nu este secvent axiomă.

Observaţie Presupunem că tentativa de construcţie a unui arbore de demonstraţie pentru secventul S determină în final un arbore contraexemplu T. Dacă n este vârful generat astfel încât $\varphi(n)$ este secvent încheiat şi nu este axiomă, atunci pentru orice interpretare I astfel încât $I(\varphi(n)) = F$, rezultă $I(S) = F$. Cu alte cuvinte, interpretarea I "demonstrează" $\{\alpha_1, ..., \alpha_n\} \nvdash \{\beta_1, ..., \beta_m\}$, deci invalidează (falsifică) afirmaţia $\{\alpha_1, ..., \alpha_n\} \vdash \{\beta_1, ..., \beta_m\}$.

Exemplul 1.9.4.1 Fie $H \vdash \Gamma$, unde $p, q, r, s, t \in V$,

$$H = \{(p \wedge (\neg q)), (p \to q), (t \to r), (p \wedge s)\}, \Gamma = \{t\}.$$

Aplicarea demonstratorului Gentzen secventului $S = H \Rightarrow \Gamma$ determină următoarea evoluţie:

Iniţializare: $T : r$, $\varphi(r) = S$.

P1: $\partial T = \{r\}$, r vârf selectat, $(p \wedge (\neg q))$formulă selectată, regula $G2$

$$T : \begin{array}{c} n_1 \\ \uparrow \\ r \end{array} \qquad \varphi(n_1) = \{p, (\neg q), (p \to q), (t \to r), (p \wedge s)\} \Rightarrow \{t\}$$

P2: $\partial T = \{n_1\}$, n_1 vârf selectat,$(\neg q)$formulă selectată, regula $G1$

$$T : \begin{array}{c} n_2 \\ \uparrow \\ n_1 \\ \uparrow \\ r \end{array} \qquad \varphi(n_2) = \{p, (p \to q), (t \to r), (p \wedge s)\} \Rightarrow \{q, t\}$$

P3: $\partial T = \{n_1\}$, n_2 vârf selectat,$(p \to q)$formulă selectată, regula $G4$

$$T : \begin{array}{ccc} n_3 & & n_4 \\ & \nwarrow \quad \nearrow & \\ & n_2 & \\ & \uparrow & \\ & n_1 & \\ & \uparrow & \\ & r & \end{array} \qquad \begin{array}{l} \varphi(n_3) = \{p, q, (t \to r), (p \wedge s)\} \Rightarrow \{q, t\} \\ \varphi(n_4) = \{p, (t \to r), (p \wedge s)\} \Rightarrow \{p, q, t\} \end{array}$$

P4: $\partial T = \{n_3, n_4\}$ Ambele vârfuri au etichete secvenţi axiomă,deci T arbore de demonstraţie pentru S.

Exemplul 1.9.4.2 Fie $H \vdash \Gamma$, unde $p, q, r, s \in V$,

$$H = \{(((\neg p) \to q) \to ((\neg r) \to s))\}, \ \Gamma = \emptyset.$$

Aplicarea demonstratorului Gentzen secventului $S = H \Rightarrow \Gamma$ determină următoarea evoluţie:

Iniţializare: $T : r$, $\varphi(r) = \{(((\neg p) \to q) \to ((\neg r) \to s))\} \Rightarrow \emptyset$.

P1: $\partial T = \{r\}$, r vârf selectat, $(((\neg p) \to q) \to ((\neg r) \to s))$ formulă selectată, regula $G4$

$$T: \begin{array}{ccc} n_1 & & n_2 \\ & \nwarrow\!\!\nearrow & \\ & r & \end{array} \qquad \begin{array}{l} \varphi(n_1) = \{((\neg r) \to s)\} \Rightarrow \emptyset \\ \varphi(n_2) = \emptyset \Rightarrow \{((\neg p) \to q)\} \end{array}$$

P2: $\partial T = \{n_1, n_2\}$, n_1 vârf selectat, $((\neg r) \to s)$ formulă selectată, regula $G4$

$$T: \begin{array}{ccccc} n_3 & & n_4 & & \\ & \nwarrow\!\!\nearrow & & & \\ & n_1 & & n_2 & \\ & & \nwarrow\!\!\nearrow & & \\ & & r & & \end{array} \qquad \begin{array}{l} \varphi(n_3) = \{s\} \Rightarrow \emptyset \\ \varphi(n_4) = \emptyset \Rightarrow \{(\neg r)\}. \end{array}$$

Vârful nou generat n_3 are eticheta secvent încheiat şi care nu este axiomă, deci

$$T \text{ este arbore contraexemplu} \Rightarrow \{(((\neg p) \to q) \to ((\neg r) \to s))\} \nvdash \emptyset.$$

5.5 1.9.5 Metoda arborilor semantici pentru verificarea validabilităţii formulelor limbajului calculului cu propoziţii.

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 arborescent 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 căror 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 1.9.5.1. 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 "◯", atunci arborele este deschis.

Exemplul 1.9.5.1

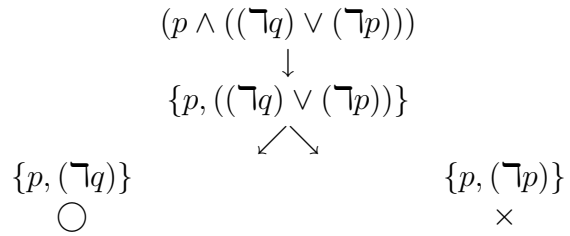


Figura 1.9.5.1. Arbore semantic pentru $\alpha = (p \wedge ((\neg q) \vee (\neg p)))$.

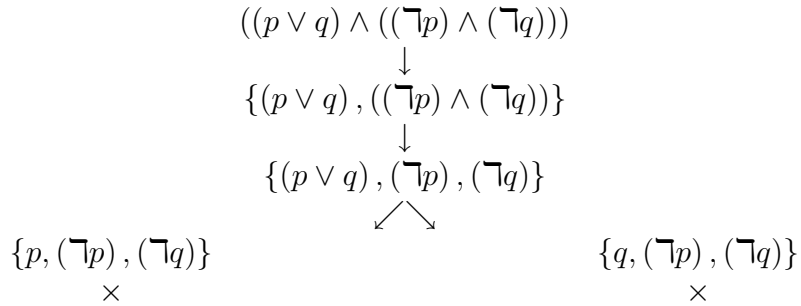


Figura 1.9.5.2. Arbore semantic pentru $\beta = ((p \vee q) \wedge ((\neg p) \wedge (\neg q)))$.

Definiţia 1.9.5.2. Fie $\gamma \in FORM$. Spunem că γ este o $\alpha - formulă$, dacă este îndeplinită una din condiţiile:

ι) există $\delta \in FORM$, astfel încât $\gamma = (\neg(\neg\delta))$

$\iota\iota$) există $\delta, \eta \in FORM$, astfel încât $\gamma = (\delta \wedge \eta)$

$\iota\iota\iota$) există $\delta, \eta \in FORM$, astfel încât $\gamma = (\neg(\delta \vee \eta))$

$\iota\nu$) există $\delta, \eta \in FORM$, astfel încât $\gamma = (\neg(\delta \rightarrow \eta))$

ν) există $\delta, \eta \in FORM$, astfel încât $\gamma = (\delta \leftrightarrow \eta)$.

Definiţia 1.9.5.3. Fie $\gamma \in FORM$. Spunem că γ este o $\beta - formulă$, dacă este îndeplinită una din condiţiile:

ι) există $\delta, \eta \in FORM$, astfel încât $\gamma = (\delta \vee \eta)$

$\iota\iota$) există $\delta, \eta \in FORM$, astfel încât $\gamma = (\neg(\delta \wedge \eta))$

$\iota\iota\iota$) există $\delta, \eta \in FORM$, astfel încât $\gamma = (\delta \rightarrow \eta)$

$\iota\nu$) 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
$(\neg(\delta \vee \eta))$	$(\neg\delta)$	$(\neg\eta)$
$(\neg(\delta \rightarrow \eta))$	δ	$(\neg\eta)$
$(\delta \leftrightarrow \eta)$	$(\delta \rightarrow \eta)$	$(\eta \rightarrow \delta)$

2. $\beta - reguli$ ($Reguli\ pentru\ \beta - formule$)

γ	γ_1	γ_2
$(\delta \vee \eta)$	δ	η
$(\neg(\delta \wedge \eta))$	$(\neg\delta)$	$(\neg\eta)$
$(\delta \rightarrow \eta)$	$(\neg\delta)$	η
$(\neg(\delta \leftrightarrow \eta))$	$(\neg(\delta \rightarrow \eta))$	$(\neg(\eta \rightarrow \delta))$

Observaţie Mulţimea $FORM$ este partajată în mulţimea *α-formulelor* şi respectiv mulţimea *β-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 $\{\gamma\}$

Condiţia de terminare 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 *α-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 *β-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 *β-regulei* asociate formulei ϑ.

Observaţie Procedura descrisă este un algoritm. Justificarea este imediată şi rezultă din faptul că mulţimea subformulelor unei formule este finită.

Teorema 1.9.5.1 (Teoremele de consistenţă-completitudine pentru metoda arborilor semantici)

Fie T un arbore semantic complet pentru formula ϑ. Formula ϑ este invalidabilă, dacă şi numai dacă T este arbore închis.

Demonstraţie.

Teorema de consistenţă: Dacă arborele semantic complet T pentru formula ϑ este arbore închis, atunci ϑ este invalidabilă.

Demonstrăm că dacă arborele T construit pentru formula ϑ este arbore închis, atunci ϑ este invalidabilă.

Demonstrăm prin inducţie asupra înălţimii $h(n)$ a subarborelui $T(n)$ de rădăcina vârful n din arborele T că, dacă $T(n)$ este închis, atunci $U(n)$ este invalidabilă.

Dacă $h(n) = 0$, atunci n este vârf terminal şi cum $T(n)$ este subarbore închis, există o pereche de literali complementari în $U(n)$, deci $U(n)$ este invalidabilă. Dacă $h(n) \geq 1$, atunci etichetele succesorilor vârfului n sunt generate prin aplicarea unei $\alpha - reguli$ sau a unei $\beta - reguli$.

Cazul 1: Dacă la nivelul vârfului a fost aplicată o $\alpha - regulă$, atunci $U(n) = U_0 \cup \{(\alpha_1 \wedge \alpha_2)\}$, unde U_0 este o mulţime, posibil vidă, de literali, $U(n') = U_0 \cup \{\alpha_1, \alpha_2\}$.

$$n : U(n) = U_0 \cup \{(\alpha_1 \wedge \alpha_2)\}$$
$$\downarrow$$
$$n' : U(n') = U_0 \cup \{\alpha_1, \alpha_2\}$$

Figura 1.9.5.2 (*a*)

Deoarece $h(n') = h(n) - 1$, din ipoteza inductivă rezultă $U(n')$ invalidabilă. Pentru $I \in \Im$ arbitrară, fie $\alpha' \in U(n')$, $I(\alpha') = F$. Atunci
- există $\alpha_0 \in U_0 \subset U(n')$ şi $I(\alpha_0) = F$, sau
- $I(\alpha_1) = F$ sau $I(\alpha_2) = F$ deci $I((\alpha_1 \wedge \alpha_2)) = F$.

În fiecare caz rezultă că există $\alpha \in U(n)$ şi $I(\alpha) = F$, deci $U(n)$ este invalidabilă.

Cazul 2: Dacă la nivelul vârfului a fost aplicată o $\beta - regula$, atunci $U(n) = U_0 \cup \{(\beta_1 \vee \beta_2)\}$ unde U_0 este o mulţime, posibil vidă, de literali,

$U(n') = U_0 \cup \{\beta_1\}; U(n'') = U_0 \cup \{\beta_2\}$

$$n : U(n) = U_0 \cup \{(\beta_1 \vee \beta_2)\}$$
$$\swarrow \quad \searrow$$
$$n' : U(n') = U_0 \cup \{\beta_1\} \qquad n'' : U(n'') = U_0 \cup \{\beta_2\}$$

Figura 1.9.5.2 (b)

Deoarece $h(n') \leq h(n) - 1$, $h(n'') \leq h(n) - 1$, din ipoteza inductivă rezultă că ambele mulţimi $U(n')$,$U(n'')$ sunt invalidabile. Fie $I \in \Im$ arbitrară.

Atunci,

• există $\alpha_0 \in U_0 \subset U(n') \cap U(n'')$ si $I(\alpha_0) = F$, sau

• $I(\alpha_0) = T$ pentru orice $\alpha_0 \in U_0$, caz în care deoarece $U(n')$,$U(n'')$ sunt invalidabile, rezultă $I(\beta_1) = I(\beta_2) = F$ deci $I((\beta_1 \vee \beta_2)) = F$, ceea ce evident implică $U(n)$ mulţime invalidabilă.

Teorema de completitudine: Dacă formula ϑ este invalidabilă, atunci arborele semantic complet T pentru formula ϑ este arbore închis.

Demonstraţie

Din construcţie rezultă evident că, dacă la $P3$ procedura a aplicat fomulei selectate o $\alpha - formulă$, atunci $\aleph(U(n)) = \aleph(U(n_1))$. Dacă formulei selectate i-a fost aplicată o $\beta - formulă$,atunci $\aleph(U(n)) = \aleph(U(n_1)) \cup \aleph(U(n_2))$.

Presupunem că frontiera ∂T a arborelui semantic complet T corespunzător formulei ϑ conţine cel puţin un vârf deschis n_1. Fie $P : n_1, ..., n_k = r$ drumul de la vârful terminal n_1 la rădăcina r în arborele T. Deoarece $\aleph(U(n_1)) \neq \emptyset$, pentru orice $I \in \aleph(U(n_1))$, utilizând observaţia precedentă, rezultă $I \in \aleph(U(n_i))$, $i = 1, ..., k$ deci $I(\vartheta) \neq \emptyset$, adică ϑ este validabilă. Obţinem astfel că dacă ϑ este invalidabilă, atunci prin aplicarea procedurii descrise rezultă arborele semantic complet T cu toate vârfurile terminale închise, deci T este arbore închis.

Corolarul 1.9.5.1 Formula ϑ este validabilă, dacă şi numai dacă arborele semantic complet construit prin aplicarea procedurii descrise este deschis.

Corolarul 1.9.5.2 Formula ϑ este validă, dacă şi numai dacă arborele semantic complet construit pentru $(\neg\vartheta)$ este închis.

Într-adevăr, ϑ este validă, dacă şi numai dacă $(\neg\vartheta)$ este invalidabilă, dacă şi numai dacă arborele semantic complet construit pentru $(\neg\vartheta)$ este închis.

Corolarul 1.9.5.3 Procedura descrisă este o procedură de decizie pentru verificarea validităţii în limbajul calculului cu propoziţii.

Observaţie Pentru $n_1 \in \partial T$ vârf deschis, construcţia unei interpretări $I \in \aleph(U(n_1))$ este imediată şi anume:dacă

$U(n_1) = \{p_1, ..., p_k, (\neg q_1), ..., (\neg q_s)\}$, $p_1, ..., p_k$, $q_1, ..., q_s \in V$,

atunci pentru orice funcţie de adevăr $\hbar : V \to \{T, F\}$, astfel încât
$\hbar(p_i) = T$, $i = 1, ..., k$, $\hbar(q_j) = F$, $j = 1, ..., s$, obţinem $I(\hbar) \in \aleph(U(n_1))$.

Cu alte cuvinte, construcţia din demonstraţia teoremei precedente defineşte numai parţial o interpretare model pentru formula ϑ în sensul că valorile pentru anumiţi literali pot eventual să rămână neprecizate.

De exemplu, pentru formula $\vartheta = (p \vee (q \wedge (\neg q)))$ arborele semantic complet este

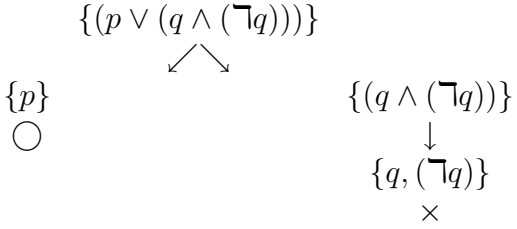


Figura 1.9.5.3

deci orice interpretare I astfel încât $I(p) = T$ este model pentru ϑ indiferent de valoarea $I(q)$.

Convenim să numim drum deschis un drum de la rădăcină la un vârf terminal deschis într-un arbore semantic complet.

În continuare vom demonstra că orice drum deschis defineşte complet valorile unei funcţii de adevăr pe mulţimea literalilor care au ocurenţe în formula considerată. În acest scop este necesar calculul reuniunii mulţimilor de formule etichete corespunzătoare vârfurilor unui drum deschis. Următoarea definiţie caracterizează o astfel de mulţime de formule.

Definiţia 1.9.5.4 Mulţimea de formule U este o mulţime Hintikka dacă îndeplineşte următoarele condiţii:

1. Pentru orice p$\in V$, p$\notin U$ sau $(\neg p) \notin U$.
2. Dacă $\gamma \in U$ şi $\gamma = (\gamma_1 \wedge \gamma_2)$,atunci $\gamma_1 \in U$ şi $\gamma_2 \in U$.
3. Dacă $\beta \in U$ şi $\beta = (\beta_1 \vee \beta_2)$, atunci $\beta_1 \in U$ sau $\beta_2 \in U$.

Din punct de vedere intuitiv, cerinţele impuse în

Definiţia 1.9.5.4 asigură absenţa contradicţiilor şi de asemenea, într-o mulţime Hintikka să existe suficiente formule pentru ca valorile de adevăr asociate propoziţiilor elementare să inducă o interpretare în care toate formulele mulţimii să rezulte valide.

Pentru exemplul din *Figura* 1.9.5.3 mulţimea

$$U = \{p, (p \vee (q \wedge (\neg q)))\}$$

este o mulţime Hintikka.

Teorema 1.9.5.2 Fie T un arbore (tablou) semantic complet. Dacă $P : n_1, ..., n_k = r$ este drumul de la vârful terminal deschis n_1 la rădăcina r în arborele T, atunci $U = \bigcup_{i=1}^{k} U(n_i)$ este o mulţime Hintikka.

Demonstraţie Fie $m \in U$ literal. Fie n_j cel mai apropiat vârf de rădăcină pe drumul P, astfel încât $m \in U(n_j)$. Deoarece nu există reguli de descompunere pentru literali, rezultă $m \in U(n_p)$, $p = j, j-1..., 1$ deci $m \in U(n_1)$. Cu alte cuvinte toţi literalii din U aparţin mulţimii $U(n_1)$. Deoarece n_1 este vârf deschis, $U(n_1)$ nu conţine literali complementari deci condiţia 1 din *Definiţia* 1.9.5.4 este îndeplinită.

Presupunem $\gamma \in U$ şi $\gamma = (\gamma_1 \wedge \gamma_2)$. Deoarece T este arbore semantic complet, există j, $2 \leq j \leq k$, astfel încât la nivelul vârfului n_j formula γ a fost prelucrată prin aplicarea unei $\alpha - reguli$. Rezultă $od_T(n_j) = 1$ şi deci n_{j-1} este unicul succesor al vârfului n_j în arborele T, ceea ce implică $\{\gamma_1, \gamma_2\} \subset U(n_{j-1}) \subset U$. Analog, dacă $\gamma \in U$ şi $\gamma = (\gamma_1 \vee \gamma_2)$ şi n_j este vârful la nivelul căruia a fost prelucrată formula γ prin aplicarea unei $\beta - reguli$, atunci $od_T(n_j) = 2$. Deoarece n_{j-1} este succesor al vârfului n_j în arborele T, rezultă $\gamma_1 \in U(n_{j-1})$ sau $\gamma_2 \in U(n_{j-1})$, deci $\gamma_1 \in U$ sau $\gamma_2 \in U$. Rezultă $U = \bigcup_{i=1}^{k} U(n_i)$ este mulţime Hintikka.

Rezultatul stabilit de următoarea teoremă permite construirea unui model pentru o formulă dată ϑ pe baza mulţimii Hintikka reuniunea mulţimilor de formule etichete ale vârfurilor dintr-un drum deschis al unui arbore semantic complet.

Teorema 1.9.5.3 (Teorema Hintikka pentru limbajul calculului cu propoziţii) Orice U mulţime Hintikka este validabilă.

Demonstraţie Fie $A = \{p_1, ..., p_m\}$ submulţimea propoziţiilor elementare care au ocurenţe în formulele mulţimii U. Fie funcţia de adevăr $\hbar : V \to \{T, F\}$, definită prin:

$$\hbar(p) = \begin{cases} T, \text{ dacă } p \in U \text{ sau } (\neg p) \notin U \\ F, \text{ dacă } (\neg p) \in U \\ arbitrar \text{ , altfel.} \end{cases}$$

Aşa cum este ilustrat în exemplul de mai sus, este posibil ca o propoziţie elementară q să aibă ocurenţe în formulele din U, dar nici q şi nici $(\neg q)$ să nu aparţină mulţimii U. Pentru aceste propoziţii elementare, ca şi pentru propoziţiile elementare care nu au ocurenţe în formulele din U, valoarea funcţiei $\hbar$ este arbitrară.

Deoarece U este mulţime Hintikka, fiecare variabilă propoziţională care are ocurenţe în formulele mulţimii U are asociată o valoare de adevăr, deci funcţia $\hbar$ este bine definită.

Demonstrăm prin inducţie structurală că pentru orice $\gamma \in U$, $I(\hbar)(\gamma) = T$, unde $I(\hbar)$ este interpretarea indusă de $\hbar$.

• Proprietatea este evidentă pentru γ propoziţie elementară.

Pentru $\gamma = (\neg p), p \in V$, deoarece U este mulţime Hintikka, rezultă $p \notin U$ deci $\hbar(p) = F$, ceea ce evident implică $I(\hbar)(\gamma) = T$.

• Dacă $\gamma = (\gamma_1 \wedge \gamma_2)$, atunci $\gamma_1 \in U$, $\gamma_2 \in U$. Din ipoteza inductivă, $I(\hbar)(\gamma_1) = I(\hbar)(\gamma_2) = T$, deci $I(\hbar)(\gamma) = T$.

• Dacă $\gamma = (\gamma_1 \vee \gamma_2)$, atunci $\gamma_1 \in U$ sau $\gamma_2 \in U$. Din ipoteza inductivă, $I(\hbar)(\gamma_1) = T$ sau $I(\hbar)(\gamma_2) = T$, deci $I(\hbar)(\gamma) = T$.

În concluzie $I(\hbar) \in \aleph(U)$, deci U este validabilă.

Observaţie Proprietatea de completitudine a metodei arborilor semantici pentru calculul cu propoziţii este o consecinţă imediată a *Teoremei* 1.9.5.3.

Într-adevăr, dacă arborele semantic complet T pentru formula ϑ este deschis şi $P : n_1, ..., n_k = r$ este drum deschis de la vârful terminal n_1 la rădăcina r, atunci $U = \bigcup_{i=1}^{k} U(n_i)$ este o mulţime Hintikka, deci este validabilă, ceea ce evident implică ϑ validabilă.

Observaţie Construcţia efectivă a unui arbore (tablou) semantic poate fi simplificată pe baza următoarelor observaţii:

1. Dacă la nivelul unui vârf intermediar n, există $\alpha \in FORM$, astfel încât $\{\alpha, (\neg\alpha)\} \subset U(n)$, atunci toate drumurile terminal-rădăcină rezultate prin extinderea drumului de la rădăcină la vârful n vor fi drumuri închise. În consecinţă, dacă n este vârful selectat şi există α, astfel încât $\{\alpha, (\neg\alpha)\} \subset U(n)$, atunci nu mai este necesară extinderea arborelui semantic prin expandarea vârfului n.

2. Implementarea procedurii de generare a unui arbore semantic complet poate fi substanţial eficientizată, dacă etichetele vârfurilor arborelui sunt reprezentate printr-un sistem de pointeri către locaţiile de memorie unde sunt stocate formulele componente.

3. Procedura de generare a unui arbore semantic complet poate fi accelerată, dacă opţiunile asupra vârfului respectiv asupra formulei, ce urmează să fie procesate la pasul următor, sunt efectuate utilizând informaţie euristică. De exemplu, pentru prevenirea apariţiilor duplicatelor unor formule se poate opta pentru aplicarea preferenţială a $\alpha - regulilor$.

Justificarea corectitudinii simplificărilor propuse rezultă fără dificultate din consideraţiile teoretice efectuate în cadrul acestei secţiuni.