



Logică Matematică și Computațională

Anul I, Semestrul II 2026

Laurențiu Leuștean

Pagina web: <https://cs.unibuc.ro/courses/lmc/>



LOGICA PROPOZIȚIONALĂ

Limbajul logicii propoziționale este bazat pe **propoziții** sau **enunțuri declarative**, despre care se poate argumenta în principiu că sunt **adevărate** sau **false**.

Propoziții declarative

- ▶ Suma numerelor 2 și 4 este 6.
- ▶ Mihai Eminescu a fost un scriitor român.
- ▶ Maria a reacționat violent la acuzațiile lui Ion.
- ▶ Orice număr natural par > 2 este suma a două numere prime. (Conjectura lui Goldbach).
- ▶ Andrei este deștept.
- ▶ Marțienilor le place pizza.

Propoziții care nu sunt declarative

- ▶ Poți să îmi dai, te rog, pâinea?
- ▶ Pleacă!

Considerăm anumite propoziții ca fiind **atomice** și le notăm

$p, q, r, \dots$ sau $p_1, p_2, p_3, \dots$

Exemple: p =Numărul 2 este par. q =Mâine plouă. r =Sunt obosit.

Pornind de la propozițiile atomice, putem crea propoziții complexe (notate $\varphi, \psi, \chi, \dots$) folosind conectorii logici $\neg$ (negația), $\rightarrow$ (implicația), $\vee$ (disjuncția), $\wedge$ (conjuncția), $\leftrightarrow$ (echivalența).

Exemple:

$\neg p$ = Numărul 2 **nu** este par.

$p \vee q$ = Numărul 2 este par **sau** mâine plouă.

$p \wedge q$ = Numărul 2 este par **și** mâine plouă.

$p \rightarrow q$ = **Dacă** numărul 2 este par, **atunci** mâine plouă.

$p \leftrightarrow q$ = Numărul 2 este par **dacă și numai dacă** mâine plouă.

Putem aplica repetat conectorii pentru a obține propoziții și mai complexe. Pentru a elimina ambiguitățile, folosim parantezele (,).

Exemplu: $\varphi = (p \wedge q) \rightarrow ((\neg r) \vee q)$



Exemplu:

Fie propoziția:

φ = *Dacă este frig afară, trebuie să îmi iau paltonul.*

Considerăm propozițiile atomice

p = *Este frig afară.* q = *Trebuie să îmi iau paltonul.*

Atunci $\varphi = p \rightarrow q$. Cine este $\neg\varphi$?

$\neg\varphi = p \wedge (\neg q)$ = *Este frig afară și nu trebuie să îmi iau paltonul.*



Exemplu:

Fie propoziția:

φ = *Dacă trenul întârzie și nu sunt taxiuri la gară, atunci Ion întârzie la întâlnire.*

Considerăm propozițiile atomice

p = *Trenul întârzie.*

q = *Sunt taxiuri la gară.*

r = *Ion întârzie la întâlnire.*

Atunci $\varphi = (p \wedge (\neg q)) \rightarrow r$.

Presupunem că φ, p sunt adevărate și r este falsă (deci $\neg r$ este adevărată). Ce putem spune despre q ? **q este adevărată.**



Definiția 1.1

Limbajul logicii propoziționale LP este format din:

- ▶ o mulțime numărabilă $V = \{v_n \mid n \in \mathbb{N}\}$ de *variabile*;
 - ▶ conectori logici: $\neg$ (se citește *non*), $\rightarrow$ (se citește *implică*)
 - ▶ paranteze: $(,)$.
- Mulțimea *Sim* a *simbolurilor* lui LP este

$$Sim := V \cup \{\neg, \rightarrow, (,)\}.$$

- Notăm variabilele cu $v, u, w, v_0, v_1, v_2, \dots$



Definiția 1.2

Mulțimea *Expr* a *expresiilor* lui LP este mulțimea tuturor șirurilor finite de simboluri ale lui LP.

- ▶ Expresia vidă se notează λ .
- ▶ **Lungimea** unei expresii θ este numărul simbolurilor din θ . Sim^n este mulțimea șirurilor de simboluri ale lui LP de lungime n .
- ▶ Prin convenție, $Sim^0 = \{\lambda\}$. Atunci $Expr = \bigcup_{n \in \mathbb{N}} Sim^n$.

Exemple:

$(((((v_7, v_1 \neg \rightarrow (v_2), \neg v_1 v_2, ((v_1 \rightarrow v_2) \rightarrow (\neg v_1))), (\neg(v_1 \rightarrow v_2))).$



Operația de bază pentru expresii este **concatenarea**: dacă $\varphi = \varphi_0 \dots \varphi_{k-1}$ și $\psi = \psi_0 \dots \psi_{l-1}$ sunt expresii, atunci concatenarea lor, notată $\varphi\psi$, este expresia $\varphi_0 \dots \varphi_{k-1}\psi_0 \dots \psi_{l-1}$.

Definiția 1.3

Fie $\theta = \theta_0\theta_1 \dots \theta_{k-1}$ o expresie a lui LP, unde $\theta_i \in \text{Sim}$ pentru orice $i \in \{0, 1, \dots, k-1\}$.

- ▶ Dacă $0 \leq i \leq j \leq k-1$, atunci expresia $\theta_i \dots \theta_j$ se numește (i, j) -**subexpresia** lui θ ;
- ▶ Spunem că o expresie ψ **apare** în θ dacă există $0 \leq i \leq j \leq k-1$ a.î. ψ este (i, j) -subexpresia lui θ .

Definiția formulelor este un exemplu de **definiție inductivă**.

Definiția 1.4

Formulele lui LP sunt expresiile lui LP definite astfel:

- (F0) Orice variabilă propozițională este formulă.
- (F1) Dacă φ este formulă, atunci $(\neg\varphi)$ este formulă.
- (F2) Dacă φ și ψ sunt formule, atunci $(\varphi \rightarrow \psi)$ este formulă.
- (F3) Numai expresiile obținute aplicând regulile (F0), (F1), (F2) sunt formule.

Notății: Mulțimea formulelor se notează **Form**. Notăm formulele cu $\varphi, \psi, \chi, \dots$

- ▶ Orice formulă se obține aplicând regulile (F0), (F1), (F2) de un număr finit de ori.
- ▶ $Form \subseteq Expr$. Formulele sunt expresiile "bine formate".



Exemple:

- ▶ $v_1 \neg \rightarrow (v_2)$, $\neg v_1 v_2$ nu sunt formule.
- ▶ $((v_1 \rightarrow v_2) \rightarrow (\neg v_1))$, $(\neg(v_1 \rightarrow v_2))$ sunt formule.

Citire unică (Unique readability)

Dacă φ este o formulă, atunci **exact** una din următoarele alternative are loc:

- ▶ $\varphi = v$, unde $v \in V$;
- ▶ $\varphi = (\neg\psi)$, unde ψ este formulă;
- ▶ $\varphi = (\psi \rightarrow \chi)$, unde ψ, χ sunt formule.

Mai mult, scrierea lui φ sub una din aceste forme este unică.

Propoziția 1.5 (Principiul inducției pe formule)

Fie P o proprietate. Presupunem că:

- (0) Orice variabilă are proprietatea P .
- (1) Pentru orice formulă φ , dacă φ are proprietatea P , atunci și $(\neg\varphi)$ are proprietatea P .
- (2) Pentru orice formule φ, ψ , dacă φ și ψ au proprietatea P , atunci $(\varphi \rightarrow \psi)$ are proprietatea P .

Atunci orice formulă φ are proprietatea P .

Dem.: Pentru orice formulă φ , notăm cu $c(\varphi)$ numărul conectorilor logici care apar în φ . Pentru orice $n \in \mathbb{N}$ definim proprietatea $Q(n)$ astfel:

$Q(n)$ e adevărată ddacă orice formulă φ cu $c(\varphi) \leq n$ are proprietatea P .

Demonstrăm prin inducție că $Q(n)$ este adevărată pentru orice $n \in \mathbb{N}$.



Principiul inducției pe formule

Pasul inițial. $Q(0)$ este adevărată, deoarece pentru orice formulă φ , $c(\varphi) \leq 0 \iff c(\varphi) = 0 \iff \varphi = v$, cu $v \in V$ și, conform ipotezei (0), v are proprietatea **P**.

Ipoteza de inducție. Fie $n \in \mathbb{N}$. Presupunem că $Q(n)$ este adevărată.

Pasul de inducție. Demonstrăm că $Q(n+1)$ este adevărată. Fie φ o formulă cu $c(\varphi) \leq n+1$. Avem trei cazuri:

- ▶ $\varphi = v \in V$. Atunci φ are proprietatea **P**, conform (0).
- ▶ $\varphi = (\neg\psi)$, unde ψ este formulă. Atunci $c(\psi) = c(\varphi) - 1 \leq n$, deci, conform ipotezei de inducție, ψ are proprietatea **P**.
Aplicînd ipoteza (1), rezultă că φ are proprietatea **P**.
- ▶ $\varphi = (\psi \rightarrow \chi)$, unde ψ, χ sunt formule. Atunci $c(\psi), c(\chi) \leq c(\varphi) - 1 \leq n$, deci, conform ipotezei de inducție, ψ și χ au proprietatea **P**. Rezultă din (2) că φ are proprietatea **P**.

Așadar, $Q(n)$ este adevărată pentru orice $n \in \mathbb{N}$. Deoarece pentru orice formulă φ există $N \in \mathbb{N}$ a.î. $c(\varphi) \leq N$, rezultă că orice formulă φ are proprietatea **P**. □



Principiul inducției pe formule

Propoziția 1.6 (Principiul inducției pe formule - variantă alternativă)

Fie Γ o mulțime de formule care are următoarele proprietăți:

- ▶ $V \subseteq \Gamma$;
- ▶ Γ este închisă la $\neg$, adică $\varphi \in \Gamma$ implică $(\neg\varphi) \in \Gamma$;
- ▶ Γ este închisă la $\rightarrow$, adică $\varphi, \psi \in \Gamma$ implică $(\varphi \rightarrow \psi) \in \Gamma$.

Atunci $\Gamma = \text{Form}$.

Dem.: Definim următoarea proprietate **P**: pentru orice formulă φ ,
 φ are proprietatea **P** dacă $\varphi \in \Gamma$.

Conform definiției lui Γ , rezultă că sunt satisfăcute ipotezele (0), (1), (2) din Principiul inducției pe formule (Propoziția 1.5), deci îl putem aplica pentru a obține că orice formulă are proprietatea **P**, deci orice formulă φ este în Γ . Așadar, $\Gamma = \text{Form}$. □



Definiția 1.7

Fie φ o formulă a lui LP. O **subformulă** a lui φ este orice formulă ψ care apare în φ .

Notăție: Mulțimea subformulelor lui φ se notează $\text{SubForm}(\varphi)$.

Exemplu:

Fie $\varphi = ((v_1 \rightarrow v_2) \rightarrow (\neg v_1))$. Atunci

$$\text{SubForm}(\varphi) = \{v_1, v_2, (v_1 \rightarrow v_2), (\neg v_1), \varphi\}.$$

Conectorii derivați $\vee$ (se citește **sau**), $\wedge$ (se citește **și**), $\leftrightarrow$ (se citește **dacă și numai dacă**) sunt introduși prin abrevierile:

$$(\varphi \vee \psi) \quad := \quad ((\neg\varphi) \rightarrow \psi)$$

$$(\varphi \wedge \psi) \quad := \quad (\neg(\varphi \rightarrow (\neg\psi)))$$

$$(\varphi \leftrightarrow \psi) \quad := \quad ((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)).$$

Convenții

- ▶ În practică, renunțăm la parantezele exterioare, le punem numai atunci când sunt necesare. Astfel, scriem $\neg\varphi$, $\varphi \rightarrow \psi$, dar scriem $(\varphi \rightarrow \psi) \rightarrow \chi$.
- ▶ Pentru a mai reduce din folosirea parantezelor, presupunem că
 - $\neg$ are precedența mai mare decât ceilalți conectori;
 - $\wedge, \vee$ au precedență mai mare decât $\rightarrow, \leftrightarrow$.

Prin urmare, formula $((\varphi \rightarrow (\psi \vee \chi)) \wedge ((\neg\psi) \leftrightarrow (\psi \vee \chi)))$ va fi scrisă $(\varphi \rightarrow \psi \vee \chi) \wedge (\neg\psi \leftrightarrow \psi \vee \chi)$.



Principiul recursiei pe formule

Propoziția 1.8 (Principiul recursiei pe formule)

Fie A o mulțime și funcțiile

$$G_0 : V \rightarrow A, \quad G_{\neg} : A \rightarrow A, \quad G_{\rightarrow} : A \times A \rightarrow A.$$

Atunci există o unică funcție

$$F : \text{Form} \rightarrow A$$

care satisface următoarele proprietăți:

(R0) $F(v) = G_0(v)$ pentru orice variabilă $v \in V$.

(R1) $F(\neg\varphi) = G_{\neg}(F(\varphi))$ pentru orice formulă φ .

(R2) $F(\varphi \rightarrow \psi) = G_{\rightarrow}(F(\varphi), F(\psi))$ pentru orice formule φ, ψ .

Principiul recursiei pe formule se folosește pentru a da **definiții recursive** ale diverselor funcții asociate formulelor.

Exemplu:

Fie $c : \text{Form} \rightarrow \mathbb{N}$ definită astfel: pentru orice formulă φ ,
 $c(\varphi)$ este numărul conectorilor logici care apar în φ .

O definiție recursivă a lui c este următoarea:

$$\begin{aligned}c(v) &= 0 && \text{pentru orice variabilă } v \\c(\neg\varphi) &= c(\varphi) + 1 && \text{pentru orice formulă } \varphi \\c(\varphi \rightarrow \psi) &= c(\varphi) + c(\psi) + 1 && \text{pentru orice formule } \varphi, \psi.\end{aligned}$$

În acest caz, $A = \mathbb{N}$, $G_0 : V \rightarrow A$, $G_0(v) = 0$,

$$\begin{aligned}G_{\neg} : \mathbb{N} &\rightarrow \mathbb{N}, && G_{\neg}(n) = n + 1, \\G_{\rightarrow} : \mathbb{N} \times \mathbb{N} &\rightarrow \mathbb{N}, && G_{\rightarrow}(m, n) = m + n + 1.\end{aligned}$$



Principiul recursiei pe formule

Notăție:

Pentru orice formulă φ , notăm cu $Var(\varphi)$ mulțimea variabilelor care apar în φ .

Observație

Mulțimea $Var(\varphi)$ poate fi definită și recursiv.

Dem.: Exercițiu.



SEMANTICA LP

Valori de adevăr

Folosim următoarele notații pentru cele două valori de adevăr:

1 pentru **adevărat** și 0 pentru **fals**. Prin urmare, mulțimea valorilor de adevăr este $\{0, 1\}$.

Definim următoarele operații pe $\{0, 1\}$ folosind **tabelele de adevăr**.

$$\neg : \{0, 1\} \rightarrow \{0, 1\},$$

p	$\neg p$
0	1
1	0

Se observă că $\neg p = 1 \iff p = 0$.

$$\rightarrow : \{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\},$$

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

Se observă că $p \rightarrow q = 0 \iff p = 1$ și $q = 0$.



Operațiile $\vee : \{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$, $\wedge : \{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$ și $\leftrightarrow : \{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$ se definesc astfel:

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

p	q	$p \leftrightarrow q$
0	0	1
0	1	0
1	0	0
1	1	1

Observație

Pentru orice $p, q \in \{0, 1\}$, $p \vee q = \neg p \rightarrow q$, $p \wedge q = \neg(p \rightarrow \neg q)$ și $p \leftrightarrow q = (p \rightarrow q) \wedge (q \rightarrow p)$.

Dem.: Exercițiu.



Definiția 1.9

O *evaluare* (sau *interpretare*) este o funcție $e : V \rightarrow \{0, 1\}$.

Teorema 1.10

Pentru orice evaluare $e : V \rightarrow \{0, 1\}$ există o unică funcție

$$e^+ : \text{Form} \rightarrow \{0, 1\}$$

care verifică următoarele proprietăți:

- ▶ $e^+(v) = e(v)$ pentru orice $v \in V$;
- ▶ $e^+(\neg\varphi) = \neg e^+(\varphi)$ pentru orice $\varphi \in \text{Form}$;
- ▶ $e^+(\varphi \rightarrow \psi) = e^+(\varphi) \rightarrow e^+(\psi)$ pentru orice $\varphi, \psi \in \text{Form}$.

Dem.: Aplicăm Principiul recursiei pe formule (Propoziția 1.8) cu $A = \{0, 1\}$, $G_0 = e$, $G_{\neg} : \{0, 1\} \rightarrow \{0, 1\}$, $G_{\neg}(p) = \neg p$ și $G_{\rightarrow} : \{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$, $G_{\rightarrow}(p, q) = p \rightarrow q$. □



Propoziția 1.11

Dacă $e : V \rightarrow \{0, 1\}$ este o evaluare, atunci pentru orice formule φ, ψ ,

$$e^+(\varphi \vee \psi) = e^+(\varphi) \vee e^+(\psi),$$

$$e^+(\varphi \wedge \psi) = e^+(\varphi) \wedge e^+(\psi),$$

$$e^+(\varphi \leftrightarrow \psi) = e^+(\varphi) \leftrightarrow e^+(\psi).$$

Dem.: Exercițiu.



Propoziția 1.12

Pentru orice formulă φ și orice evaluări $e_1, e_2 : V \rightarrow \{0, 1\}$,

$$(*) \quad e_1(v) = e_2(v) \text{ pentru orice } v \in \text{Var}(\varphi) \implies e_1^+(\varphi) = e_2^+(\varphi).$$

Dem.: Definim următoarea proprietate **P**: pentru orice formulă φ ,

φ are proprietatea **P** ddacă pentru orice evaluări
 $e_1, e_2 : V \rightarrow \{0, 1\}$, φ satisface (*).

Demonstrăm că orice formulă φ are proprietatea **P** folosind Principiul inducției pe formule. Avem următoarele cazuri:

► $\varphi = v$. Atunci $e_1^+(v) = e_1(v) = e_2(v) = e_2^+(v)$.



Propoziția 1.12

Pentru orice formulă φ și orice evaluări $e_1, e_2 : V \rightarrow \{0, 1\}$,

$$(*) \quad e_1(v) = e_2(v) \text{ pentru orice } v \in \text{Var}(\varphi) \implies e_1^+(\varphi) = e_2^+(\varphi).$$

Dem.: (continuare)

- $\varphi = \neg\psi$ și ψ satisface **P**. Fie $e_1, e_2 : V \rightarrow \{0, 1\}$ a.î.
 $e_1(v) = e_2(v)$ pentru orice $v \in \text{Var}(\varphi)$. Deoarece
 $\text{Var}(\varphi) = \text{Var}(\psi)$, rezultă că $e_1(v) = e_2(v)$ pentru orice
 $v \in \text{Var}(\psi)$. Așadar, aplicând **P** pentru ψ , obținem că
 $e_1^+(\psi) = e_2^+(\psi)$. Rezultă că

$$e_1^+(\varphi) = \neg e_1^+(\psi) = \neg e_2^+(\psi) = e_2^+(\varphi),$$

deci φ satisface **P**.



Propoziția 1.12

Pentru orice formulă φ și orice evaluări $e_1, e_2 : V \rightarrow \{0, 1\}$,

$$(*) \quad e_1(v) = e_2(v) \text{ pentru orice } v \in \text{Var}(\varphi) \implies e_1^+(\varphi) = e_2^+(\varphi).$$

Dem.: (continuare)

- $\varphi = \psi \rightarrow \chi$ și ψ, χ satisfac **P**. Fie $e_1, e_2 : V \rightarrow \{0, 1\}$ a.î. $e_1(v) = e_2(v)$ pentru orice $v \in \text{Var}(\varphi)$. Deoarece $\text{Var}(\psi) \subseteq \text{Var}(\varphi)$ și $\text{Var}(\chi) \subseteq \text{Var}(\varphi)$, rezultă că $e_1(v) = e_2(v)$ pentru orice $v \in \text{Var}(\psi)$ și pentru orice $v \in \text{Var}(\chi)$. Așadar, aplicând **P** pentru ψ și χ , obținem că $e_1^+(\psi) = e_2^+(\psi)$ și $e_1^+(\chi) = e_2^+(\chi)$. Rezultă că

$$e_1^+(\varphi) = e_1^+(\psi) \rightarrow e_1^+(\chi) = e_2^+(\psi) \rightarrow e_2^+(\chi) = e_2^+(\varphi),$$

deci φ satisface **P**.



Fie φ o formulă.

Definiția 1.13

- ▶ O evaluare $e : V \rightarrow \{0, 1\}$ este **model** al lui φ dacă $e^+(\varphi) = 1$. **Notăție:** $e \models \varphi$.
- ▶ φ este **satisfiabilă** dacă admite un model.
- ▶ Dacă φ nu este satisfiabilă, spunem și că φ este **nesatisfiabilă** sau **contradictorie**.
- ▶ φ este **tautologie** dacă orice evaluare este model al lui φ .
Notăție: $\models \varphi$.

Notăție: Mulțimea tuturor modelelor lui φ se notează $Mod(\varphi)$.

Propoziția 1.14

- (i) φ este tautologie ddacă $\neg\varphi$ este nesatisfiabilă.
- (ii) φ este nesatisfiabilă ddacă $\neg\varphi$ este tautologie.

Dem.: Exercițiu.

Fie φ o formulă arbitrară și $\text{Var}(\varphi) = \{x_1, x_2, \dots, x_k\}$. Pentru orice evaluare $e : V \rightarrow \{0, 1\}$, $e^+(\varphi)$ depinde doar de $e(x_1), \dots, e(x_k)$, conform Propoziției 1.12.

Așadar, $e^+(\varphi)$ depinde doar de restricția lui e la $\{x_1, x_2, \dots, x_k\}$:

$$e' : \{x_1, \dots, x_k\} \rightarrow \{0, 1\}, \quad e'(x_i) = e(x_i).$$

Sunt 2^k astfel de funcții posibile $e'_1, e'_2, \dots, e'_{2^k}$. Asociem fiecăreia o linie într-un tabel:

x_1	x_2	$\dots$	x_k	$\dots$ subformule ale lui $\varphi \dots$	φ
$e'_1(x_1)$	$e'_1(x_2)$	$\dots$	$e'_1(x_k)$	$\dots$	$e_1^{+}(\varphi)$
$e'_2(x_1)$	$e'_2(x_2)$	$\dots$	$e'_2(x_k)$	$\dots$	$e_2^{+}(\varphi)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
$e'_{2^k}(x_1)$	$e'_{2^k}(x_2)$	$\dots$	$e'_{2^k}(x_k)$	$\dots$	$e_{2^k}^{+}(\varphi)$

Pentru orice i , $e_i^{+}(\varphi)$ se definește similar cu Teorema 1.10.

φ este tautologie ddacă pentru orice $i \in \{1, \dots, 2^k\}$, $e_i^{+}(\varphi) = 1$

φ este satisfiabilă ddacă există $i \in \{1, \dots, 2^k\}$ a.î. $e_i^{+}(\varphi) = 1$



Exemplu:

Fie

$$\varphi = v_1 \rightarrow (v_2 \rightarrow (v_1 \wedge v_2)).$$

Vrem să demonstrăm că $\models \varphi$.

$$\text{Var}(\varphi) = \{v_1, v_2\}.$$

v_1	v_2	$v_1 \wedge v_2$	$v_2 \rightarrow (v_1 \wedge v_2)$	φ
0	0	0	1	1
0	1	0	0	1
1	0	0	1	1
1	1	1	1	1

Definiția 1.15

Fie φ, ψ două formule. Spunem că

- ▶ φ este **consecință semantică** a lui ψ dacă $\text{Mod}(\psi) \subseteq \text{Mod}(\varphi)$. **Notăție:** $\psi \models \varphi$.
- ▶ φ și ψ sunt **(logic) echivalente** dacă $\text{Mod}(\psi) = \text{Mod}(\varphi)$.
Notăție: $\varphi \sim \psi$.

Observație

Relația $\sim$ este o relație de echivalență pe mulțimea *Form* a formulelor lui *LP*.

Propoziția 1.16

Fie φ, ψ formule. Atunci

- (i) $\psi \models \varphi$ ddacă $\models \psi \rightarrow \varphi$.
- (ii) $\psi \sim \varphi$ ddacă $(\psi \models \varphi \text{ și } \varphi \models \psi)$ ddacă $\models \psi \leftrightarrow \varphi$.

Dem.: Exercițiu.



Propoziția 1.17

Pentru orice formule φ, ψ, χ ,

terțul exclus $\models \varphi \vee \neg \varphi$ (1)

modus ponens $\varphi \wedge (\varphi \rightarrow \psi) \models \psi$ (2)

afirmarea concluziei $\psi \models \varphi \rightarrow \psi$ (3)

contradicția $\models \neg(\varphi \wedge \neg \varphi)$ (4)

dubla negație $\varphi \sim \neg \neg \varphi$ (5)

contrapозиția $\varphi \rightarrow \psi \sim \neg \psi \rightarrow \neg \varphi$ (6)

negarea premisei $\neg \varphi \models \varphi \rightarrow \psi$ (7)

modus tollens $\neg \psi \wedge (\varphi \rightarrow \psi) \models \neg \varphi$ (8)

tranzitivitatea implicației $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \chi) \models \varphi \rightarrow \chi$ (9)



Tautologii, consecințe semantice și echivalențe

legile lui de Morgan $\neg(\varphi \vee \psi) \sim \neg\varphi \wedge \neg\psi$ (10)

$$\neg(\varphi \wedge \psi) \sim \neg\varphi \vee \neg\psi \quad (11)$$

exportarea și importarea $\varphi \rightarrow (\psi \rightarrow \chi) \sim \varphi \wedge \psi \rightarrow \chi$ (12)

idempotența $\varphi \sim \varphi \wedge \varphi \sim \varphi \vee \varphi$ (13)

slăbirea $\models \varphi \wedge \psi \rightarrow \varphi \quad \models \varphi \rightarrow \varphi \vee \psi$ (14)

comutativitatea $\varphi \wedge \psi \sim \psi \wedge \varphi \quad \varphi \vee \psi \sim \psi \vee \varphi$ (15)

asociativitatea $\varphi \wedge (\psi \wedge \chi) \sim (\varphi \wedge \psi) \wedge \chi$ (16)

$$\varphi \vee (\psi \vee \chi) \sim (\varphi \vee \psi) \vee \chi \quad (17)$$

absorbția $\varphi \vee (\varphi \wedge \psi) \sim \varphi$ (18)

$$\varphi \wedge (\varphi \vee \psi) \sim \varphi \quad (19)$$

distributivitatea $\varphi \wedge (\psi \vee \chi) \sim (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$ (20)

$$\varphi \vee (\psi \wedge \chi) \sim (\varphi \vee \psi) \wedge (\varphi \vee \chi) \quad (21)$$



Tautologii, consecințe semantice și echivalențe

$$\varphi \rightarrow \psi \wedge \chi \sim (\varphi \rightarrow \psi) \wedge (\varphi \rightarrow \chi) \quad (22)$$

$$\varphi \rightarrow \psi \vee \chi \sim (\varphi \rightarrow \psi) \vee (\varphi \rightarrow \chi) \quad (23)$$

$$\varphi \wedge \psi \rightarrow \chi \sim (\varphi \rightarrow \chi) \vee (\psi \rightarrow \chi) \quad (24)$$

$$\varphi \vee \psi \rightarrow \chi \sim (\varphi \rightarrow \chi) \wedge (\psi \rightarrow \chi) \quad (25)$$

$$\varphi \rightarrow (\psi \rightarrow \chi) \sim \psi \rightarrow (\varphi \rightarrow \chi) \sim (\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi) \quad (26)$$

$$\neg \varphi \sim \varphi \rightarrow \neg \varphi \sim (\varphi \rightarrow \psi) \wedge (\varphi \rightarrow \neg \psi) \quad (27)$$

$$\varphi \rightarrow \psi \sim \neg \varphi \vee \psi \sim \neg(\varphi \wedge \neg \psi) \quad (28)$$

$$\varphi \vee \psi \sim \varphi \vee (\neg \varphi \wedge \psi) \sim (\varphi \rightarrow \psi) \rightarrow \psi \quad (29)$$

$$\varphi \leftrightarrow (\psi \leftrightarrow \chi) \sim (\varphi \leftrightarrow \psi) \leftrightarrow \chi \quad (30)$$

$$\models (\varphi \rightarrow \psi) \vee (\neg \varphi \rightarrow \psi) \quad (31)$$

$$\models (\varphi \rightarrow \psi) \vee (\varphi \rightarrow \neg \psi) \quad (32)$$

$$\models \neg \varphi \rightarrow (\neg \psi \leftrightarrow (\psi \rightarrow \varphi)) \quad (33)$$

$$\models (\varphi \rightarrow \psi) \rightarrow (((\varphi \rightarrow \chi) \rightarrow \psi) \rightarrow \psi) \quad (34)$$

Dem.: Exercițiu.

Demonstrăm (1): $\models \varphi \vee \neg\varphi$.

Fie $e : V \rightarrow \{0, 1\}$ o evaluare arbitrară. Trebuie să arătăm că $e^+(\varphi \vee \neg\varphi) = 1$. Observăm că $e^+(\varphi \vee \neg\varphi) = e^+(\varphi) \vee \neg e^+(\varphi)$. Putem demonstra că $e^+(\varphi) \vee \neg e^+(\varphi) = 1$ în două moduri.

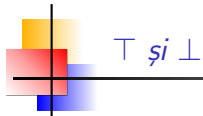
I. Folosim tabelele de adevăr.

$e^+(\varphi)$	$\neg e^+(\varphi)$	$e^+(\varphi) \vee \neg e^+(\varphi)$
0	1	1
1	0	1

II. Raționăm direct.

Avem două cazuri:

- ▶ $e^+(\varphi) = 1$. Atunci $\neg e^+(\varphi) = 0$ și, prin urmare, $e^+(\varphi) \vee \neg e^+(\varphi) = 1$.
- ▶ $e^+(\varphi) = 0$. Atunci $\neg e^+(\varphi) = 1$ și, prin urmare, $e^+(\varphi) \vee \neg e^+(\varphi) = 1$.



$\top$ și $\perp$

De multe ori este convenabil să avem o tautologie canonică și o formulă nesatisfiabilă canonică.

Observație

$v_0 \rightarrow v_0$ este tautologie și $\neg(v_0 \rightarrow v_0)$ este nesatisfiabilă.

Dem.: Exercițiu.

Notății

Notăm $v_0 \rightarrow v_0$ cu $\top$ și o numim **adevărul**. Notăm $\neg(v_0 \rightarrow v_0)$ cu $\perp$ și o numim **falsul**.

- ▶ φ este tautologie ddacă $\varphi \sim \top$.
- ▶ φ este nesatisfiabilă ddacă $\varphi \sim \perp$.

Definiția 1.18

Pentru orice formule φ, χ, χ' , definim

$\varphi_\chi(\chi') :=$ expresia obținută din φ prin înlocuirea tuturor aparițiilor lui χ cu χ' .

$\varphi_\chi(\chi')$ se numește **substituția lui χ cu χ' în φ** . Spunem și că $\varphi_\chi(\chi')$ este o **instanță de substituție** a lui φ .

- ▶ $\varphi_\chi(\chi')$ este de asemenea formulă.
- ▶ $\varphi_\varphi(\chi') = \chi'$.
- ▶ Dacă χ nu este subformulă a lui φ , atunci $\varphi_\chi(\chi') = \varphi$.

Exemple:

Fie $\varphi = (v_1 \rightarrow v_2) \rightarrow \neg(v_1 \rightarrow v_2)$.

- ▶ $\chi = v_1 \rightarrow v_2, \chi' = v_4. \quad \varphi_\chi(\chi') = v_4 \rightarrow \neg v_4$
- ▶ $\chi = v_1, \chi' = \neg\neg v_2. \quad \varphi_\chi(\chi') = (\neg\neg v_2 \rightarrow v_2) \rightarrow \neg(\neg\neg v_2 \rightarrow v_2)$
- ▶ $\chi = v_1 \rightarrow v_2, \chi' = v_4 \vee v_1. \quad \varphi_\chi(\chi') = (v_4 \vee v_1) \rightarrow \neg(v_4 \vee v_1)$



Propoziția 1.19

Pentru orice formule φ, χ, χ' ,

$$\chi \sim \chi' \text{ implică } \varphi \sim \varphi_{\chi}(\chi').$$

Propoziția 1.20

Pentru orice formule φ, ψ, χ și orice variabilă $v \in V$,

- ▶ $\varphi \sim \psi$ implică $\varphi_v(\chi) \sim \psi_v(\chi)$.
- ▶ Dacă φ este tautologie atunci și $\varphi_v(\chi)$ este tautologie.
- ▶ Dacă φ este nesatisfiabilă, atunci și $\varphi_v(\chi)$ este nesatisfiabilă.



Notății

Scriem $\varphi \wedge \psi \wedge \chi$ în loc de $(\varphi \wedge \psi) \wedge \chi$. Similar, scriem $\varphi \vee \psi \vee \chi$ în loc de $(\varphi \vee \psi) \vee \chi$.

Fie $\varphi_1, \varphi_2, \dots, \varphi_n$ formule. Pentru $n \geq 3$, notăm

$$\varphi_1 \wedge \dots \wedge \varphi_n := ((\dots (\varphi_1 \wedge \varphi_2) \wedge \varphi_3) \wedge \dots \wedge \varphi_{n-1}) \wedge \varphi_n$$

$$\varphi_1 \vee \dots \vee \varphi_n := ((\dots (\varphi_1 \vee \varphi_2) \vee \varphi_3) \vee \dots \vee \varphi_{n-1}) \vee \varphi_n.$$

- ▶ $\varphi_1 \wedge \dots \wedge \varphi_n$ se mai scrie și $\bigwedge_{i=1}^n \varphi_i$ sau $\bigwedge_{i=1}^n \varphi_i$.
- ▶ $\varphi_1 \vee \dots \vee \varphi_n$ se mai scrie și $\bigvee_{i=1}^n \varphi_i$ sau $\bigvee_{i=1}^n \varphi_i$.



Conjunții și disjunții finite

Propoziția 1.21

Pentru orice evaluare $e : V \rightarrow \{0, 1\}$,

- ▶ $e^+(\varphi_1 \wedge \dots \wedge \varphi_n) = 1$ *ddacă* $e^+(\varphi_i) = 1$ *pentru orice* $i \in \{1, \dots, n\}$.
- ▶ $e^+(\varphi_1 \vee \dots \vee \varphi_n) = 1$ *ddacă* $e^+(\varphi_i) = 1$ *pentru un* $i \in \{1, \dots, n\}$.

Dem.: Exercițiu.

Propoziția 1.22

$$\begin{aligned}\neg(\varphi_1 \vee \dots \vee \varphi_n) &\sim \neg\varphi_1 \wedge \dots \wedge \neg\varphi_n \\ \neg(\varphi_1 \wedge \dots \wedge \varphi_n) &\sim \neg\varphi_1 \vee \dots \vee \neg\varphi_n\end{aligned}$$

Dem.: Exercițiu.



FORMA NORMALĂ CONJUNCTIVĂ / DISJUNCTIVĂ

Definiția 1.23

Un **literal** este o

- ▶ variabilă (în care caz spunem că este **literal pozitiv**) sau
- ▶ negația unei variabile (în care caz spunem că este **literal negativ**).

Exemple: v_1, v_2, v_{10} literali pozitivi; $\neg v_0, \neg v_{100}$ literali negativi

Convenție: $\bigvee_{i=1}^1 \varphi_i = \varphi_1$ și $\bigwedge_{i=1}^1 \varphi_i = \varphi_1$.

Definiția 1.24

O formulă φ este în **formă normală disjunctivă (FND)** dacă φ este o disjuncție de conjuncții de literali.

Așadar, φ este în FND ddacă $\varphi = \bigvee_{i=1}^n \left(\bigwedge_{j=1}^{k_i} L_{i,j} \right)$, unde fiecare $L_{i,j}$ este literal.

Definiția 1.25

O formulă φ este în **formă normală conjunctivă (FNC)** dacă φ este o conjuncție de disjuncții de literali.

Așadar, φ este în FNC ddacă $\varphi = \bigwedge_{i=1}^n \left(\bigvee_{j=1}^{k_i} L_{i,j} \right)$, unde fiecare $L_{i,j}$ este literal.

Exemple

- ▶ $(v_0 \vee v_1) \wedge (v_3 \vee v_5) \wedge (\neg v_{20} \vee \neg v_{15} \vee \neg v_{34})$ este în FNC
- ▶ $(\neg v_9 \wedge v_1) \vee v_{24} \vee (v_2 \wedge \neg v_1 \wedge v_2)$ este în FND
- ▶ $v_1 \wedge \neg v_5 \wedge v_4$ este atât în FND cât și în FNC
- ▶ $\neg v_{10} \vee v_{20} \vee v_4$ este atât în FND cât și în FNC
- ▶ $(v_1 \vee v_2) \wedge ((v_1 \wedge v_3) \vee (v_4 \wedge v_5))$ nu este nici în FND, nici în FNC



Forma normală conjunctivă / disjunctivă

Notăție: Dacă L este literal, atunci $L^c := \begin{cases} \neg v & \text{dacă } L = v \in V \\ v & \text{dacă } L = \neg v. \end{cases}$

Propoziția 1.26

- (i) Fie φ o formulă în FNC, $\varphi = \bigwedge_{i=1}^n \left(\bigvee_{j=1}^{k_i} L_{i,j} \right)$. Atunci $\neg\varphi \sim \bigvee_{i=1}^n \left(\bigwedge_{j=1}^{k_i} L_{i,j}^c \right)$, o formulă în FND.
- (ii) Fie φ o formulă în FND, $\varphi = \bigvee_{i=1}^n \left(\bigwedge_{j=1}^{k_i} L_{i,j} \right)$. Atunci $\neg\varphi \sim \bigwedge_{i=1}^n \left(\bigvee_{j=1}^{k_i} L_{i,j}^c \right)$, o formulă în FNC.

Dem.: Exercițiu.



Funcția asociată unei formule

Exemplu: Arătați că $\models v_1 \rightarrow (v_2 \rightarrow v_1 \wedge v_2)$.

v_1	v_2	$v_1 \rightarrow (v_2 \rightarrow v_1 \wedge v_2)$
0	0	1
0	1	1
1	0	1
1	1	1

Acest tabel definește o funcție $F : \{0, 1\}^2 \rightarrow \{0, 1\}$

ε_1	ε_2	$F(\varepsilon_1, \varepsilon_2)$
0	0	1
0	1	1
1	0	1
1	1	1

Funcția asociată unei formule

Fie φ o formulă și $Var(\varphi) = \{v_{i_1}, v_{i_2}, \dots, v_{i_n}\}$, unde $n \geq 1$ și $0 \leq i_1 < i_2 < \dots < i_n$.

Fie $(\varepsilon_1, \dots, \varepsilon_n) \in \{0, 1\}^n$. Definim $e_{\varepsilon_1, \dots, \varepsilon_n} : Var(\varphi) \rightarrow \{0, 1\}$ astfel:

$$e_{\varepsilon_1, \dots, \varepsilon_n}(v_{i_k}) = \varepsilon_k \quad \text{pentru orice } k \in \{1, \dots, n\}.$$

Definim $e_{\varepsilon_1, \dots, \varepsilon_n}^+(\varphi) \in \{0, 1\}$ astfel:

$$e_{\varepsilon_1, \dots, \varepsilon_n}^+(\varphi) := e^+(\varphi),$$

unde $e : V \rightarrow \{0, 1\}$ este orice evaluare care extinde $e_{\varepsilon_1, \dots, \varepsilon_n}$, adică, $e(v_{i_k}) = e_{\varepsilon_1, \dots, \varepsilon_n}(v_{i_k}) = \varepsilon_k$ pentru orice $k \in \{1, \dots, n\}$.

Conform Propoziției 1.12, definiția nu este ambiguă.

Definiția 1.27

Funcția asociată lui φ este $F_\varphi : \{0, 1\}^n \rightarrow \{0, 1\}$, definită astfel:

$$F_\varphi(\varepsilon_1, \dots, \varepsilon_n) = e_{\varepsilon_1, \dots, \varepsilon_n}^+(\varphi) \text{ pentru orice } (\varepsilon_1, \dots, \varepsilon_n) \in \{0, 1\}^n.$$

Așadar, F_φ este funcția definită de tabela de adevăr pentru φ .



Propoziția 1.28

- (i) Fie φ o formulă. Atunci
 - 1.1 $\models \varphi$ ddacă F_φ este funcția constantă 1.
 - 1.2 φ este nesatisfiabilă ddacă F_φ este funcția constantă 0.
- (ii) Fie φ, ψ două formule astfel încât $\text{Var}(\varphi) = \text{Var}(\psi)$. Atunci
 - 2.1 $\varphi \models \psi$ ddacă $F_\varphi \leq F_\psi$.
 - 2.2 $\varphi \sim \psi$ ddacă $F_\varphi = F_\psi$.
- (iii) Există formule diferite φ, ψ a.î. $F_\varphi = F_\psi$.

Definiția 1.29

O **funcție booleană** este o funcție $F : \{0, 1\}^n \rightarrow \{0, 1\}$, unde $n \geq 1$.
Spunem că n este **numărul variabilelor** lui F .

Exemplu: Pentru orice formulă φ , F_φ este funcție Booleană cu n variabile, unde $n = |\text{Var}(\varphi)|$.

Teorema 1.30

Fie $n \geq 1$ și $H : \{0, 1\}^n \rightarrow \{0, 1\}$ o funcție booleană arbitrară.
Atunci există o formulă φ în FND a.î. $H = F_\varphi$.

Dem.: Dacă $H(\varepsilon_1, \dots, \varepsilon_n) = 0$ pentru orice $(\varepsilon_1, \dots, \varepsilon_n) \in \{0, 1\}^n$,
luăm $\varphi := \bigvee_{i=1}^n (v_i \wedge \neg v_i)$. Avem că $\text{Var}(\varphi) = \{v_1, \dots, v_n\}$, așadar,
 $F_\varphi : \{0, 1\}^n \rightarrow \{0, 1\}$. Cum $v_i \wedge \neg v_i$ este nesatisfiabilă pentru orice i , rezultă că φ este de asemenea nesatisfiabilă. Deci, F_φ este funcția constantă 0.

Altcumva, mulțimea

$$T := H^{-1}(1) = \{(\varepsilon_1, \dots, \varepsilon_n) \in \{0, 1\}^n \mid H(\varepsilon_1, \dots, \varepsilon_n) = 1\}$$

este nevidă.

Considerăm formula

$$\varphi := \bigvee_{(\varepsilon_1, \dots, \varepsilon_n) \in T} \left(\bigwedge_{\varepsilon_i=1} v_i \wedge \bigwedge_{\varepsilon_i=0} \neg v_i \right).$$

Deoarece $\text{Var}(\varphi) = \{v_1, \dots, v_n\}$, avem că $F_\varphi : \{0, 1\}^n \rightarrow \{0, 1\}$.

Demonstrăm că pentru orice $(\delta_1, \dots, \delta_n) \in \{0, 1\}^n$, avem că

$$F_\varphi(\delta_1, \dots, \delta_n) = 1 \iff H(\delta_1, \dots, \delta_n) = 1,$$

de unde va rezulta imediat că $H = F_\varphi$.

Caracterizarea funcțiilor booleene

Avem că $F_\varphi(\delta_1, \dots, \delta_n) = 1 \iff e_{\delta_1, \dots, \delta_n}^+(\varphi) = 1$
 $\iff e^+(\varphi) = 1$ pentru orice $e : V \rightarrow \{0, 1\}$ a.î. $e(v_i) = \delta_i$
pentru orice $i \in \{1, \dots, n\}$.

Fie $e : V \rightarrow \{0, 1\}$ a.î. $e(v_i) = \delta_i$ pentru orice $i \in \{1, \dots, n\}$.

$$\begin{aligned} e^+(\varphi) = 1 &\iff \bigvee_{(\varepsilon_1, \dots, \varepsilon_n) \in T} (\bigwedge_{\varepsilon_i=1} e(v_i) \wedge \bigwedge_{\varepsilon_i=0} \neg e(v_i)) = 1 \\ &\iff \bigvee_{(\varepsilon_1, \dots, \varepsilon_n) \in T} (\bigwedge_{\varepsilon_i=1} \delta_i \wedge \bigwedge_{\varepsilon_i=0} \neg \delta_i) = 1 \\ &\iff \text{există } (\varepsilon_1, \dots, \varepsilon_n) \in T \text{ a.î.} \\ &\quad \bigwedge_{\varepsilon_i=1} \delta_i = 1 \wedge \bigwedge_{\varepsilon_i=0} \neg \delta_i = 1 \\ &\iff \text{există } (\varepsilon_1, \dots, \varepsilon_n) \in T \text{ a.î. } \delta_i = \varepsilon_i \\ &\quad \text{pentru orice } i \in \{1, \dots, n\} \\ &\iff (\delta_1, \dots, \delta_n) \in T \\ &\iff H(\delta_1, \dots, \delta_n) = 1. \end{aligned}$$

Prin urmare, $F_\varphi(\delta_1, \dots, \delta_n) = 1 \iff H(\delta_1, \dots, \delta_n) = 1$. □



Teorema 1.31

Fie $n \geq 1$ și $H : \{0, 1\}^n \rightarrow \{0, 1\}$ o funcție booleană arbitrară.
Atunci există o formulă ψ în FNC a.î. $H = F_\psi$.

Dem.: Dacă $H(\varepsilon_1, \dots, \varepsilon_n) = 1$ pentru orice $(\varepsilon_1, \dots, \varepsilon_n) \in \{0, 1\}^n$,
atunci luăm

$$\psi := \bigwedge_{i=1}^n (v_i \vee \neg v_i).$$

Avem că $\text{Var}(\varphi) = \{v_1, \dots, v_n\}$, așadar, $F_\varphi : \{0, 1\}^n \rightarrow \{0, 1\}$.
Cum $v_i \vee \neg v_i$ este tautologie pentru orice i , rezultă că φ este de asemenea tautologie. Deci, F_φ este funcția constantă 1.



Caracterizarea funcțiilor booleene

Altcumva, mulțimea

$$F := H^{-1}(0) = \{(\varepsilon_1, \dots, \varepsilon_n) \in \{0, 1\}^n \mid H(\varepsilon_1, \dots, \varepsilon_n) = 0\}$$

este nevidă.

Considerăm formula

$$\psi := \bigwedge_{(\varepsilon_1, \dots, \varepsilon_n) \in F} \left(\bigvee_{\varepsilon_i=1} \neg v_i \vee \bigvee_{\varepsilon_i=0} v_i \right).$$

Deoarece $\text{Var}(\varphi) = \{v_1, \dots, v_n\}$, avem că $F_\varphi : \{0, 1\}^n \rightarrow \{0, 1\}$.

Demonstrăm că pentru orice $(\delta_1, \dots, \delta_n) \in \{0, 1\}^n$, avem că

$$F_\varphi(\delta_1, \dots, \delta_n) = 0 \iff H(\delta_1, \dots, \delta_n) = 0,$$

de unde va rezulta imediat că $H = F_\varphi$.

Avem că $F_\varphi(\delta_1, \dots, \delta_n) = 0 \iff e_{\delta_1, \dots, \delta_n}^+(\varphi) = 0$
 $\iff e^+(\varphi) = 0$ pentru orice $e : V \rightarrow \{0, 1\}$ a.î. $e(v_i) = \delta_i$
pentru orice $i \in \{1, \dots, n\}$.

Fie $e : V \rightarrow \{0, 1\}$ a.î. $e(v_i) = \delta_i$ pentru orice $i \in \{1, \dots, n\}$.

$$\begin{aligned} e^+(\varphi) = 0 &\iff \bigwedge_{(\varepsilon_1, \dots, \varepsilon_n) \in F} (\bigvee_{\varepsilon_i=1} \neg e(v_i) \vee \bigvee_{\varepsilon_i=0} e(v_i)) = 0 \\ &\iff \bigwedge_{(\varepsilon_1, \dots, \varepsilon_n) \in F} (\bigvee_{\varepsilon_i=1} \neg \delta_i \vee \bigvee_{\varepsilon_i=0} \delta_i) = 0 \\ &\iff \text{există } (\varepsilon_1, \dots, \varepsilon_n) \in F \text{ a.î.} \\ &\quad \bigvee_{\varepsilon_i=1} \neg \delta_i \vee \bigvee_{\varepsilon_i=0} \delta_i = 0 \\ &\iff \text{există } (\varepsilon_1, \dots, \varepsilon_n) \in F \text{ a.î. } \delta_i = \varepsilon_i \\ &\quad \text{pentru orice } i \in \{1, \dots, n\} \\ &\iff (\delta_1, \dots, \delta_n) \in F \\ &\iff H(\delta_1, \dots, \delta_n) = 0. \end{aligned}$$

Prin urmare, $F_\varphi(\delta_1, \dots, \delta_n) = 0 \iff H(\delta_1, \dots, \delta_n) = 0$.





Caracterizarea funcțiilor Booleene

Exemplu: Fie $H : \{0, 1\}^3 \rightarrow \{0, 1\}$ descrisă prin tabelul:

ε_1	ε_2	ε_3	$H(\varepsilon_1, \varepsilon_2, \varepsilon_3)$	
0	0	0	0	$D_1 = v_1 \vee v_2 \vee v_3$
0	0	1	0	$D_2 = v_1 \vee v_2 \vee \neg v_3$
0	1	0	1	$C_1 = \neg v_1 \wedge v_2 \wedge \neg v_3$
0	1	1	0	$D_3 = v_1 \vee \neg v_2 \vee \neg v_3$
1	0	0	1	$C_2 = v_1 \wedge \neg v_2 \wedge \neg v_3$
1	0	1	1	$C_3 = v_1 \wedge \neg v_2 \wedge v_3$
1	1	0	1	$C_4 = v_1 \wedge v_2 \wedge \neg v_3$
1	1	1	1	$C_5 = v_1 \wedge v_2 \wedge v_3$

$\varphi = C_1 \vee C_2 \vee C_3 \vee C_4 \vee C_5$ în FND a.î. $H = F_\varphi$.

$\psi = D_1 \wedge D_2 \wedge D_3$ în FNC a.î. $H = F_\psi$.



Teorema 1.32

Orice formulă φ este echivalentă cu o formulă φ^{FND} în FND și cu o formulă φ^{FNC} în FNC.

Dem.:

Fie $n = |\text{Var}(\varphi)|$ și $F_\varphi : \{0, 1\}^n \rightarrow \{0, 1\}$ funcția booleană asociată. Aplicând Teorema 1.30 cu $H := F_\varphi$, obținem o formulă φ^{FND} în FND a.î. $F_\varphi = F_{\varphi^{FND}}$. Așadar, conform Propoziției 1.28.(ii), $\varphi \sim \varphi^{FND}$.

Similar, aplicând Teorema 1.31 cu $H := F_\varphi$, obținem o formulă φ^{FNC} în FNC a.î. $F_\varphi = F_{\varphi^{FNC}}$. Prin urmare, $\varphi \sim \varphi^{FNC}$. □



Forma normală conjunctivă / disjunctivă

Algoritm pentru a aduce o formulă la FNC/FND:

Pasul 1. Se înlocuiesc implicațiile și echivalențele, folosind:

$$\varphi \rightarrow \psi \sim \neg\varphi \vee \psi \quad \text{și} \quad \varphi \leftrightarrow \psi \sim (\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi).$$

Pasul 2. Se înlocuiesc dublele negații, folosind $\neg\neg\psi \sim \psi$, și se aplică regulile De Morgan pentru a înlocui

$$\neg(\varphi \vee \psi) \text{ cu } \neg\varphi \wedge \neg\psi \quad \text{și} \quad \neg(\varphi \wedge \psi) \text{ cu } \neg\varphi \vee \neg\psi.$$

Pasul 3. Pentru FNC, se aplică distributivitatea lui $\vee$ față de $\wedge$, pentru a înlocui

$$\varphi \vee (\psi \wedge \chi) \text{ cu } (\varphi \vee \psi) \wedge (\varphi \vee \chi) \quad \text{și} \quad (\psi \wedge \chi) \vee \varphi \text{ cu } (\psi \vee \varphi) \wedge (\chi \vee \varphi).$$

Pentru FND, se aplică distributivitatea lui $\wedge$ față de $\vee$, pentru a înlocui

$$\varphi \wedge (\psi \vee \chi) \text{ cu } (\varphi \wedge \psi) \vee (\varphi \wedge \chi) \quad \text{și} \quad (\psi \vee \chi) \wedge \varphi \text{ cu } (\psi \wedge \varphi) \vee (\chi \wedge \varphi).$$

Exemplu

Considerăm formula $\varphi := (\neg v_0 \rightarrow \neg v_2) \rightarrow (v_0 \rightarrow v_2)$.

Avem

$$\begin{aligned}\varphi &\sim \neg(\neg v_0 \rightarrow \neg v_2) \vee (v_0 \rightarrow v_2) && \text{Pasul 1} \\ &\sim \neg(\neg\neg v_0 \vee \neg v_2) \vee (v_0 \rightarrow v_2) && \text{Pasul 1} \\ &\sim \neg(\neg\neg v_0 \vee \neg v_2) \vee (\neg v_0 \vee v_2) && \text{Pasul 1} \\ &\sim \neg(v_0 \vee \neg v_2) \vee (\neg v_0 \vee v_2) && \text{Pasul 2} \\ &\sim (\neg v_0 \wedge \neg\neg v_2) \vee (\neg v_0 \vee v_2) && \text{Pasul 2} \\ &\sim (\neg v_0 \wedge v_2) \vee \neg v_0 \vee v_2 && \text{Pasul 2}\end{aligned}$$

Putem lua $\varphi^{FND} := (\neg v_0 \wedge v_2) \vee \neg v_0 \vee v_2$.

Pentru a obține FNC, continuăm cu Pasul 3:

$$\begin{aligned}\varphi &\sim (\neg v_0 \wedge v_2) \vee (\neg v_0 \vee v_2) \\ &\sim (\neg v_0 \vee \neg v_0 \vee v_2) \wedge (v_2 \vee \neg v_0 \vee v_2).\end{aligned}$$

Putem lua $\varphi^{FNC} := (\neg v_0 \vee \neg v_0 \vee v_2) \wedge (v_2 \vee \neg v_0 \vee v_2)$. Se observă, folosind idempotența și comutativitatea lui $\vee$, că $\varphi^{FNC} \sim \neg v_0 \vee v_2$.



CLAUZE ȘI REZOLUȚIE

Definiția 1.33

O **clauză** este o mulțime finită de literali:

$$C = \{L_1, \dots, L_n\}, \text{ unde } L_1, \dots, L_n \text{ sunt literali.}$$

Dacă $n = 0$, obținem clauza vidă $\square := \emptyset$.

O clauză nevidă este considerată implicit o disjuncție.

Definiția 1.34

Fie C o clauză și $e : V \rightarrow \{0, 1\}$. Spunem că **e este model al lui C** sau că **e satisface C** și scriem $e \models C$ dacă există $L \in C$ a.î. $e \models L$.

Definiția 1.35

O clauză C se numește

- (i) **satisfiabilă** dacă are un model.
- (ii) **validă** dacă orice evaluare $e : V \rightarrow \{0, 1\}$ este model al lui C .

Definiția 1.36

O clauză C este **trivială** dacă există un literal L a.î. $L \in C$ și $L^c \in C$.

Propoziția 1.37

- (i) Orice clauză nevidă este satisfiabilă.
- (ii) Clauza vidă $\square$ este nesatisfiabilă.
- (iii) O clauză este validă ddacă este trivială.

Dem.: Exercițiu.

Notăm $\text{Var}(C) := \{x \in V \mid x \in C \text{ sau } \neg x \in C\}$.

Dacă $x \in \text{Var}(C)$, spunem ca x **apare în** C .

- $\text{Var}(C) = \emptyset$ ddacă $C = \square$.

$\mathcal{S} = \{C_1, \dots, C_m\}$ este o mulțime finită de clauze.

Dacă $m = 0$, obținem mulțimea vidă de clauze $\emptyset$.

$\mathcal{S}$ este considerată implicit ca o formulă în FNC: conjuncție de disjuncții ale literalilor din fiecare clauză.

Definiția 1.38

Fie $e : V \rightarrow \{0, 1\}$. Spunem că *e este model al lui $\mathcal{S}$ sau că e satisface $\mathcal{S}$* și scriem $e \models \mathcal{S}$ dacă $e \models C_i$ pentru orice $i \in \{1, \dots, m\}$.

Definiția 1.39

$\mathcal{S}$ se numește

- (i) *satisfiabilă* dacă are un model.
- (ii) *validă* dacă orice evaluare $e : V \rightarrow \{0, 1\}$ este model al lui $\mathcal{S}$.



Propoziția 1.40

- ▶ Dacă $\mathcal{S}$ conține clauza vidă $\square$, atunci $\mathcal{S}$ este nesatisfiabilă.
- ▶ $\emptyset$ este validă.

Dem.: Exercițiu.

Notăm $\text{Var}(\mathcal{S}) := \bigcup_{C \in \mathcal{S}} \text{Var}(C)$.

Dacă $x \in \text{Var}(\mathcal{S})$, spunem ca x **apare în** $\mathcal{S}$.

- ▶ $\text{Var}(\mathcal{S}) = \emptyset$ ddacă ($\mathcal{S} = \emptyset$ sau $\mathcal{S} = \{\square\}$).



Exemplu

$\mathcal{S} = \{\{v_1, \neg v_3\}, \{\neg v_3, v_3\}, \{v_2, v_1\}, \{v_2, \neg v_1, v_3\}\}$ este satisfiabilă.

Dem.: Considerăm $e : V \rightarrow \{0, 1\}$ a.î. $e(v_1) = e(v_2) = 1$. Atunci $e \models \mathcal{S}$. □

Exemplu

$\mathcal{S} = \{\{\neg v_1, v_2\}, \{\neg v_3, \neg v_2\}, \{v_1\}, \{v_3\}\}$ este nesatisfiabilă.

Dem.: Presupunem că $\mathcal{S}$ are un model e . Atunci $e(v_1) = e(v_3) = 1$ și, deoarece $e \models \{\neg v_3, \neg v_2\}$, trebuie să avem $e(v_2) = 0$. Rezultă că $e(v_2) = e^+(\neg v_1) = 0$, deci e nu satisface $\{\neg v_1, v_2\}$. Am obținut o contradicție. □



Unei formule φ în FNC îi asociem o mulțime finită de clauze $\mathcal{S}_\varphi$ astfel:

Fie

$$\varphi := \bigwedge_{i=1}^n \left(\bigvee_{j=1}^{k_i} L_{i,j} \right),$$

unde fiecare $L_{i,j}$ este literal. Pentru orice i , fie C_i clauza obținută considerând toți literalii $L_{i,j}, j \in \{1, \dots, k_i\}$ distincți. Fie $\mathcal{S}_\varphi$ mulțimea tuturor clauzelor $C_i, i \in \{1, \dots, n\}$ distincte.

$\mathcal{S}_\varphi$ se mai numește și **forma clauzală** a lui φ .

Propoziția 1.41

Pentru orice evaluare $e : V \rightarrow \{0, 1\}$, $e \models \varphi$ dacă $e \models \mathcal{S}_\varphi$.



Unei mulțimi finite de clauze $\mathcal{S}$ îi asociem o formulă $\varphi_{\mathcal{S}}$ în FNC astfel:

- ▶ $C = \{L_1, \dots, L_n\}, n \geq 1 \mapsto \varphi_C := L_1 \vee L_2 \vee \dots \vee L_n.$
- ▶ $\square \mapsto \varphi_{\square} := v_0 \wedge \neg v_0.$

Fie $\mathcal{S} = \{C_1, \dots, C_m\}$ o mulțime nevidă de clauze. Formula asociată lui $\mathcal{S}$ este

$$\varphi_{\mathcal{S}} := \bigwedge_{i=1}^m \varphi_{C_i}.$$

Formula asociată mulțimii vide de clauze este $\varphi_{\emptyset} := v_0 \vee \neg v_0.$

Formula $\varphi_{\mathcal{S}}$ nu este unic determinată, depinde de ordinea în care se scriu elementele în clauze și în $\mathcal{S}$, dar se observă imediat că: $\mathcal{S} = \mathcal{S}'$ implică $\varphi_{\mathcal{S}} \sim \varphi_{\mathcal{S}'}.$

Propoziția 1.42

Pentru orice evaluare $e : V \rightarrow \{0, 1\}$, $e \models \mathcal{S}$ dacă și numai dacă $e \models \varphi_{\mathcal{S}}.$



Definiția 1.43

Fie C_1, C_2 două clauze. O clauză R se numește **rezolvent** al clauzelor C_1, C_2 dacă există un literal L a.î. $L \in C_1, L^c \in C_2$ și

$$R = (C_1 \setminus \{L\}) \cup (C_2 \setminus \{L^c\}).$$

Regula Rezoluției

$$\text{Rez} \quad \frac{C_1, C_2}{(C_1 \setminus \{L\}) \cup (C_2 \setminus \{L^c\})}, \quad L \in C_1, L^c \in C_2$$

Notăm cu **Res**(C_1, C_2) mulțimea rezolvenților clauzelor C_1, C_2 .

- ▶ Rezoluția a fost introdusă de **Blake** (1937) și dezvoltată de **Davis, Putnam** (1960) și **Robinson** (1965).
- ▶ Multe demonstratoare automate de teoreme folosesc rezoluția. Limbajul PROLOG este bazat pe rezoluție.



Exemplu

$C_1 = \{v_1, v_2, \neg v_5\}$, $C_2 = \{v_1, \neg v_2, v_{100}, v_5\}$.

- ▶ Luăm $L := \neg v_5$. Atunci $L \in C_1$ și $L^c = v_5 \in C_2$. Prin urmare, $R = \{v_1, v_2, \neg v_2, v_{100}\}$ este rezolvent al clauzelor C_1, C_2 .
- ▶ Dacă luăm $L' := v_2$, atunci $L' \in C_1$ și $L'^c = \neg v_2 \in C_2$. Prin urmare, $R' = \{v_1, \neg v_5, v_{100}, v_5\}$ este rezolvent al clauzelor C_1, C_2 .

Exemplu

$C_1 = \{v_7\}$, $C_2 = \{\neg v_7\}$. Atunci clauza vidă $\square$ este rezolvent al clauzelor C_1, C_2 .



Fie S o mulțime finită de clauze.

Definiția 1.44

O *derivare prin rezoluție din S* sau o *S -derivare prin rezoluție* este o secvență $C_1, C_2, \dots, C_n$ de clauze a.î. pentru fiecare $i \in \{1, \dots, n\}$, una din următoarele condiții este satisfăcută:

- (i) C_i este o clauză din S ;
- (ii) există $j, k < i$ a.î. C_i este rezolvent al clauzelor C_j, C_k .

Definiția 1.45

Fie C o clauză. O *derivare prin rezoluție a lui C din S* este o S -derivare prin rezoluție $C_1, C_2, \dots, C_n$ a.î. $C_n = C$.



Exemplu

Fie

$$\mathcal{S} = \{\{\neg v_1, v_2\}, \{\neg v_2, \neg v_3, v_4\}, \{v_1\}, \{v_3\}, \{\neg v_4\}\}.$$

O derivare prin rezoluție a clauzei vide $\square$ din $\mathcal{S}$ este următoarea:

C_1	$=$	$\{\neg v_4\}$	$C_1 \in \mathcal{S}$
C_2	$=$	$\{\neg v_2, \neg v_3, v_4\}$	$C_2 \in \mathcal{S}$
C_3	$=$	$\{\neg v_2, \neg v_3\}$	C_3 rezolvent al clauzelor C_1, C_2
C_4	$=$	$\{v_3\}$	$C_4 \in \mathcal{S}$
C_5	$=$	$\{\neg v_2\}$	C_5 rezolvent al clauzelor C_3, C_4
C_6	$=$	$\{\neg v_1, v_2\}$	$C_6 \in \mathcal{S}$
C_7	$=$	$\{\neg v_1\}$	C_7 rezolvent al clauzelor C_5, C_6
C_8	$=$	$\{v_1\}$	$C_8 \in \mathcal{S}$
C_9	$=$	$\square$	C_9 rezolvent al clauzelor C_7, C_8 .



Notăm $Res(S) := \bigcup_{C_1, C_2 \in S} Res(C_1, C_2)$.

Propoziția 1.46

Pentru orice evaluare $e : V \rightarrow \{0, 1\}$,

$$e \models S \quad \Rightarrow \quad e \models Res(S).$$

Dem.: Dacă $Res(S) = \emptyset$, atunci este validă, deci $e \models Res(S)$.
Presupunem că $Res(S)$ este nevidă și fie $R \in Res(S)$. Atunci există clauze $C_1, C_2 \in S$ și un literal L a.î. $L \in C_1, L^c \in C_2$ și $R = (C_1 \setminus \{L\}) \cup (C_2 \setminus \{L^c\})$. Avem două cazuri:

- ▶ $e \models L$. Atunci $e \not\models L^c$. Deoarece $e \models C_2$, există $U \in C_2, U \neq L^c$ a.î. $e \models U$. Deoarece $U \in R$, obținem că $e \models R$.
- ▶ $e \not\models L$. Deoarece $e \models C_1$, există $U \in C_1, U \neq L$ a.î. $e \models U$. Deoarece $U \in R$, obținem că $e \models R$. □



Teorema 1.47 (Teorema de corectitudine a rezoluției)

Dacă $\square$ se derivează prin rezoluție din $\mathcal{S}$, atunci $\mathcal{S}$ este nesatisfiabilă.

Dem.: Fie $C_1, C_2, \dots, C_n = \square$ o $\mathcal{S}$ -derivare prin rezoluție a lui $\square$. Presupunem că $\mathcal{S}$ este satisfiabilă și fie $e \models \mathcal{S}$.

Demonstrăm prin inducție după i că:

pentru orice $1 \leq i \leq n$, $e \models C_i$.

Pentru $i = n$, obținem că $e \models \square$, ceea ce este o contradicție.

Cazul $i = 1$ este evident, deoarece $C_1 \in \mathcal{S}$.

Presupunem că $e \models C_j$ pentru orice $j < i$. Avem două cazuri:

- ▶ $C_i \in \mathcal{S}$. Atunci $e \models C_i$.
- ▶ există $j, k < i$ a.î. $C_i \in \text{Res}(C_j, C_k)$. Deoarece, conform ipotezei de inducție, $e \models \{C_j, C_k\}$ aplicăm Propoziția 1.46 pentru a conclud că $e \models C_i$.



Algoritmul Davis-Putnam (DP)

Intrare: $\mathcal{S}$ mulțime finită nevidă de clauze netriviiale.

$i := 1, \mathcal{S}_1 := \mathcal{S}.$

Pi.1 Fie x_i o variabilă care apare în $\mathcal{S}_i$. Definim

$$\mathcal{T}_i^1 := \{C \in \mathcal{S}_i \mid x_i \in C\}, \quad \mathcal{T}_i^0 := \{C \in \mathcal{S}_i \mid \neg x_i \in C\}.$$

Pi.2 **if** $(\mathcal{T}_i^1 \neq \emptyset \text{ și } \mathcal{T}_i^0 \neq \emptyset)$ **then**

$$\mathcal{U}_i := \{(C_1 \setminus \{x_i\}) \cup (C_0 \setminus \{\neg x_i\}) \mid C_1 \in \mathcal{T}_i^1, C_0 \in \mathcal{T}_i^0\}.$$

else $\mathcal{U}_i := \emptyset.$

Pi.3 Definim

$$\begin{aligned} \mathcal{S}'_{i+1} &:= (\mathcal{S}_i \setminus (\mathcal{T}_i^0 \cup \mathcal{T}_i^1)) \cup \mathcal{U}_i; \\ \mathcal{S}_{i+1} &:= \mathcal{S}'_{i+1} \setminus \{C \in \mathcal{S}'_{i+1} \mid C \text{ trivială}\}. \end{aligned}$$

Pi.4 **if** $\mathcal{S}_{i+1} = \emptyset$ **then** $\mathcal{S}$ este satisfiabilă.

else if $\square \in \mathcal{S}_{i+1}$ **then** $\mathcal{S}$ este nesatisfiabilă.

else $\{i := i + 1; \text{ go to Pi.1}\}.$



Algoritmul Davis-Putnam (DP)

$\mathcal{S} = \{\{v_1, \neg v_3\}, \{v_2, v_1\}, \{v_2, \neg v_1, v_3\}\}$. $i := 1$, $\mathcal{S}_1 := \mathcal{S}$.

P1.1 $x_1 := v_3$; $\mathcal{T}_1^1 := \{\{v_2, \neg v_1, v_3\}\}$; $\mathcal{T}_1^0 := \{\{v_1, \neg v_3\}\}$.

P1.2 $\mathcal{U}_1 := \{\{v_2, \neg v_1, v_1\}\}$.

P1.3 $\mathcal{S}'_2 := \{\{v_2, v_1\}, \{v_2, \neg v_1, v_1\}\}$; $\mathcal{S}_2 := \{\{v_2, v_1\}\}$.

P1.4 $i := 2$ and go to P2.1.

P2.1 $x_2 := v_2$; $\mathcal{T}_2^1 := \{\{v_2, v_1\}\}$; $\mathcal{T}_2^0 := \emptyset$.

P2.2 $\mathcal{U}_2 := \emptyset$.

P2.3 $\mathcal{S}_3 := \emptyset$.

P2.4 $\mathcal{S}$ este satisfiabilă.



Algoritmul Davis-Putnam (DP)

$\mathcal{S} = \{\{\neg v_1, v_2, \neg v_4\}, \{\neg v_3, \neg v_2\}, \{v_1, v_3\}, \{v_1\}, \{v_3\}, \{v_4\}\}.$

$i := 1, \mathcal{S}_1 := \mathcal{S}.$

P1.1 $x_1 := v_1; \mathcal{T}_1^1 := \{\{v_1, v_3\}, \{v_1\}\}; \mathcal{T}_1^0 := \{\{\neg v_1, v_2, \neg v_4\}\}.$

P1.2 $\mathcal{U}_1 := \{\{v_3, v_2, \neg v_4\}, \{v_2, \neg v_4\}\}.$

P1.3 $\mathcal{S}_2 := \{\{\neg v_3, \neg v_2\}, \{v_3\}, \{v_4\}, \{v_3, v_2, \neg v_4\}, \{v_2, \neg v_4\}\}.$

P1.4 $i := 2$ and go to P2.1.

P2.1. $x_2 := v_2; \mathcal{T}_2^1 := \{\{v_3, v_2, \neg v_4\}, \{v_2, \neg v_4\}\}; \mathcal{T}_2^0 := \{\{\neg v_3, \neg v_2\}\}.$

P2.2 $\mathcal{U}_2 := \{\{v_3, \neg v_4, \neg v_3\}, \{\neg v_4, \neg v_3\}\}.$

P2.3 $\mathcal{S}_3 := \{\{v_3\}, \{v_4\}, \{\neg v_4, \neg v_3\}\}.$

P2.4 $i := 3$ and go to P3.1.

P3.1 $x_3 := v_3; \mathcal{T}_3^1 := \{\{v_3\}\}; \mathcal{T}_3^0 := \{\{\neg v_4, \neg v_3\}\}.$

P3.2. $\mathcal{U}_3 := \{\{\neg v_4\}\}.$ P3.3 $\mathcal{S}_4 := \{\{v_4\}, \{\neg v_4\}\}.$

P3.4 $i := 4$ and go to P4.1.

P4.1 $x_4 := v_4; \mathcal{T}_4^1 := \{\{v_4\}\}; \mathcal{T}_4^0 := \{\{\neg v_4\}\}.$

P4.2 $\mathcal{U}_4 := \{\square\}.$ P4.3 $\mathcal{S}_5 := \{\square\}.$

P4.4 $\mathcal{S}$ nu este satisfiabilă.



Propoziția 1.48

Fie $n := |\text{Var}(\mathcal{S})|$. Atunci algoritmul DP se termină după cel mult n pași.

Dem.: Se observă imediat că pentru orice i ,

$$\text{Var}(\mathcal{S}_{i+1}) \subseteq \text{Var}(\mathcal{S}_i) \setminus \{x_i\} \subsetneq \text{Var}(\mathcal{S}_i).$$

Prin urmare, $n = |\text{Var}(\mathcal{S}_1)| > |\text{Var}(\mathcal{S}_2)| > |\text{Var}(\mathcal{S}_3)| > \dots \geq 0$.



Fie $N \leq n$ numărul de pași după care se termină DP. Atunci

$\mathcal{S}_{N+1} = \emptyset$ sau $\square \in \mathcal{S}_{N+1}$.



Propoziția 1.49

Pentru orice $i \leq N$,

$$S_{i+1} \text{ este satisfiabilă} \iff S_i \text{ este satisfiabilă}.$$

Dem.: Exercițiu suplimentar.

Teorema 1.50

Algoritmul DP este corect și complet, adică,

$$S \text{ este nesatisfiabilă} \text{ ddacă } \square \in S_{N+1}.$$

Dem.: Aplicăm Propoziția 1.49. Obținem că $S = S_1$ este nesatisfiabilă ddacă S_{N+1} este nesatisfiabilă ddacă $\square \in S_{N+1}$.





SINTAXA LP

Folosim un **sistem deductiv** de tip Hilbert pentru LP .

Axiomele logice

Mulțimea Axm a **axiomelor** lui LP constă din toate formulele de forma:

$$(A1) \quad \varphi \rightarrow (\psi \rightarrow \varphi)$$

$$(A2) \quad (\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$$

$$(A3) \quad (\neg\psi \rightarrow \neg\varphi) \rightarrow (\varphi \rightarrow \psi),$$

unde φ , ψ și χ sunt formule.

Regula de deducție

Pentru orice formule φ, ψ ,

din φ și $\varphi \rightarrow \psi$ se inferă ψ (**modus ponens** sau **(MP)**):

$$\frac{\varphi, \varphi \rightarrow \psi}{\psi}.$$



Fie Γ o mulțime de formule. Definiția Γ -teoremelor este un nou exemplu de **definiție inductivă**.

Definiția 1.51

Γ -teoremele sunt formulele lui *LP* definite astfel:

- (T0) Orice axiomă este Γ -teoremă.*
- (T1) Orice formulă din Γ este Γ -teoremă.*
- (T2) Dacă φ și $\varphi \rightarrow \psi$ sunt Γ -teoreme, atunci ψ este Γ -teoremă.*
- (T3) Numai formulele obținute aplicând regulile (T0), (T1), (T2) sunt Γ -teoreme.*

Dacă φ este Γ -teoremă, atunci spunem și că φ este **dedusă din ipotezele Γ** .



Notății

$Thm(\Gamma)$	$:=$	mulțimea Γ -teoremelor	Thm	$:=$	$Thm(\emptyset)$
$\Gamma \vdash \varphi$	$:\Leftrightarrow$	φ este Γ -teoremă	$\vdash \varphi$	$:\Leftrightarrow$	$\emptyset \vdash \varphi$
$\Gamma \vdash \Delta$	$:\Leftrightarrow$	$\Gamma \vdash \varphi$ pentru orice $\varphi \in \Delta$.			

Definiția 1.52

O formulă φ se numește **teoremă** a lui LP dacă $\vdash \varphi$.

Reformulăm condițiile (T0), (T1), (T2) folosind notația $\vdash$:

Propoziția 1.53

- (i) dacă φ este axiomă, atunci $\Gamma \vdash \varphi$;
- (ii) dacă $\varphi \in \Gamma$, atunci $\Gamma \vdash \varphi$;
- (iii) dacă $\Gamma \vdash \varphi$ și $\Gamma \vdash \varphi \rightarrow \psi$, atunci $\Gamma \vdash \psi$.

Definiția Γ -teoremelor dă naștere la metoda de demonstrație prin **inducție după Γ -teoreme**.

Versiunea 1

Fie **P** o proprietate a formulelor. Demonstrăm că orice Γ -teoremă satisface **P** astfel:

- (i) Demonstrăm că orice axiomă are proprietatea **P** .
- (ii) Demonstrăm că orice formulă din Γ are proprietatea **P** .
- (iii) Demonstrăm că dacă φ și $\varphi \rightarrow \psi$ au proprietatea **P** , atunci ψ are proprietatea **P** .

Versiunea 2

Fie Σ o mulțime de formule. Demonstrăm că $Thm(\Gamma) \subseteq \Sigma$ astfel:

- (i) Demonstrăm că orice axiomă este în Σ .
- (ii) Demonstrăm că orice formulă din Γ este în Σ .
- (iii) Demonstrăm că dacă $\varphi \in \Sigma$ și $\varphi \rightarrow \psi \in \Sigma$, atunci $\psi \in \Sigma$.



Propoziția 1.54

Fie Γ, Δ mulțimi de formule.

(i) Dacă $\Gamma \subseteq \Delta$, atunci $Thm(\Gamma) \subseteq Thm(\Delta)$, adică, pentru orice formulă φ ,

$\Gamma \vdash \varphi$ implică $\Delta \vdash \varphi$.

(ii) $Thm \subseteq Thm(\Gamma)$, adică, pentru orice formulă φ ,

$\vdash \varphi$ implică $\Gamma \vdash \varphi$.

(iii) Dacă $\Gamma \vdash \Delta$, atunci $Thm(\Delta) \subseteq Thm(\Gamma)$, adică, pentru orice formulă φ ,

$\Delta \vdash \varphi$ implică $\Gamma \vdash \varphi$.

(iv) $Thm(Thm(\Gamma)) = Thm(\Gamma)$, adică, pentru orice formulă φ ,

$Thm(\Gamma) \vdash \varphi$ ddacă $\Gamma \vdash \varphi$.



Definiția 1.55

O Γ -demonstrație (*demonstrație din ipotezele Γ*) este o secvență de formule $\theta_1, \dots, \theta_n$ a.î. pentru fiecare $i \in \{1, \dots, n\}$, una din următoarele condiții este satisfăcută:

- (i) θ_i este axiomă;
- (ii) $\theta_i \in \Gamma$;
- (iii) există $k, j < i$ a.î. $\theta_k = \theta_j \rightarrow \theta_i$.

O $\emptyset$ -demonstrație se va numi simplu *demonstrație*.

Lema 1.56

Dacă $\theta_1, \dots, \theta_n$ este o Γ -demonstrație, atunci

$$\Gamma \vdash \theta_i \text{ pentru orice } i \in \{1, \dots, n\}.$$



Definiția 1.57

Fie φ o formulă. O Γ -demonstrație a lui φ sau *demonstrație a lui φ din ipotezele Γ* este o Γ -demonstrație $\theta_1, \dots, \theta_n$ a.î. $\theta_n = \varphi$. În acest caz, n se numește *lungimea* Γ -demonstrației.

Propoziția 1.58

Fie Γ o mulțime de formule și φ o formulă. Atunci $\Gamma \vdash \varphi$ ddacă există o Γ -demonstrație a lui φ .



Propoziția 1.59

Pentru orice mulțime de formule Γ și orice formulă φ ,

$\Gamma \vdash \varphi$ dacă există o submulțime finită Σ a lui Γ a.î. $\Sigma \vdash \varphi$.

Dem.: " $\Leftarrow$ " Fie $\Sigma \subseteq \Gamma$, Σ finită a.î. $\Sigma \vdash \varphi$. Aplicând Propoziția 1.54.(i) obținem că $\Gamma \vdash \varphi$.

" $\Rightarrow$ " Presupunem că $\Gamma \vdash \varphi$. Conform Propoziției 1.58, φ are o Γ -demonstrație $\theta_1, \dots, \theta_n = \varphi$. Fie

$$\Sigma := \Gamma \cap \{\theta_1, \dots, \theta_n\}.$$

Atunci Σ este finită, $\Sigma \subseteq \Gamma$ și $\theta_1, \dots, \theta_n = \varphi$ este o Σ -demonstrație a lui φ , deci $\Sigma \vdash \varphi$.




$$\vdash \varphi \rightarrow \varphi$$

Propoziția 1.60

Pentru orice formulă φ , $\vdash \varphi \rightarrow \varphi$.

Dem.:

- (1) $\vdash (\varphi \rightarrow ((\varphi \rightarrow \varphi) \rightarrow \varphi)) \rightarrow ((\varphi \rightarrow (\varphi \rightarrow \varphi)) \rightarrow (\varphi \rightarrow \varphi))$
(A2) (cu $\varphi, \psi := \varphi \rightarrow \varphi, \chi := \varphi$) și Propoziția 1.53.(i)
- (2) $\vdash \varphi \rightarrow ((\varphi \rightarrow \varphi) \rightarrow \varphi)$
(A1) (cu $\varphi, \psi := \varphi \rightarrow \varphi$) și Propoziția 1.53.(i)
- (3) $\vdash (\varphi \rightarrow (\varphi \rightarrow \varphi)) \rightarrow (\varphi \rightarrow \varphi)$
(1), (2) și Propoziția 1.53.(iii). Scriem de obicei (MP): (1), (2)
- (4) $\vdash \varphi \rightarrow (\varphi \rightarrow \varphi)$
(A1) (cu $\varphi, \psi := \varphi$) și Propoziția 1.53.(i)
- (5) $\vdash \varphi \rightarrow \varphi$
(MP): (3), (4)



Teorema deducției

Teorema 1.61 (Teorema deducției)

Fie $\Gamma \subseteq \text{Form}$ și $\varphi, \psi \in \text{Form}$. Atunci

$$\Gamma \cup \{\varphi\} \vdash \psi \text{ ddacă } \Gamma \vdash \varphi \rightarrow \psi.$$

Dem.: Exercițiu suplimentar.

Teorema deducției este un instrument foarte util pentru a arăta că o formulă e teoremă.



Propoziția 1.62

Pentru orice formule φ, ψ, χ ,

$$\vdash (\varphi \rightarrow \psi) \rightarrow ((\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \chi)). \quad (35)$$

Dem.: Folosind teorema deducției observăm că

$$\begin{aligned} & \vdash (\varphi \rightarrow \psi) \rightarrow ((\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \chi)) \\ & \quad \Updownarrow \\ & \{\varphi \rightarrow \psi\} \vdash (\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \chi) \\ & \quad \Updownarrow \\ & \{\varphi \rightarrow \psi, \psi \rightarrow \chi\} \vdash \varphi \rightarrow \chi \\ & \quad \Updownarrow \\ & \{\varphi \rightarrow \psi, \psi \rightarrow \chi, \varphi\} \vdash \chi. \end{aligned}$$



În acest fel am reformulat ceea ce aveam de demonstrat. A demonstra teorema inițială este echivalent cu a demonstra

$$\{\varphi \rightarrow \psi, \psi \rightarrow \chi, \varphi\} \vdash \chi.$$

- | | | |
|-----|--|---------------------------|
| (1) | $\{\varphi \rightarrow \psi, \psi \rightarrow \chi, \varphi\} \vdash \varphi$ | Propoziția 1.53.(ii) |
| (2) | $\{\varphi \rightarrow \psi, \psi \rightarrow \chi, \varphi\} \vdash \varphi \rightarrow \psi$ | Propoziția 1.53.(ii) |
| (3) | $\{\varphi \rightarrow \psi, \psi \rightarrow \chi, \varphi\} \vdash \psi$ | (MP): (1), (2) |
| (4) | $\{\varphi \rightarrow \psi, \psi \rightarrow \chi, \varphi\} \vdash \psi \rightarrow \chi$ | Propoziția 1.53.(ii) |
| (5) | $\{\varphi \rightarrow \psi, \psi \rightarrow \chi, \varphi\} \vdash \chi$ | (MP): (3), (4). $\square$ |



Propoziția 1.63

Pentru orice mulțime de formule Γ și orice formule φ, ψ, χ ,

$$\Gamma \vdash \varphi \rightarrow \psi \text{ și } \Gamma \vdash \psi \rightarrow \chi \Rightarrow \Gamma \vdash \varphi \rightarrow \chi.$$

Dem.:

- | | | |
|-----|---|--|
| (1) | $\Gamma \vdash \varphi \rightarrow \psi$ | ipoteză |
| (2) | $\Gamma \vdash (\varphi \rightarrow \psi) \rightarrow ((\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \chi))$ | P.1.62 și P.1.54.(ii) |
| (3) | $\Gamma \vdash (\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \chi)$ | (MP): (1), (2) |
| (4) | $\Gamma \vdash \psi \rightarrow \chi$ | ipoteză |
| (5) | $\Gamma \vdash \varphi \rightarrow \chi$ | (MP): (3), (4). □ |



Propoziția 1.64

Pentru orice formule φ, ψ, χ ,

$$\vdash (\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow (\psi \rightarrow (\varphi \rightarrow \chi)) \quad (36)$$

Dem.: Exercițiu.

Propoziția 1.65

Pentru orice mulțime de formule Γ și orice formule φ, ψ, χ ,

$$\Gamma \cup \{\neg\psi\} \vdash \neg(\varphi \rightarrow \varphi) \Rightarrow \Gamma \vdash \psi.$$

Dem.: Exercițiu.



Propoziția 1.66

Pentru orice formule φ, ψ ,

$$\{\psi, \neg\psi\} \vdash \varphi \quad (37)$$

$$\vdash \neg\psi \rightarrow (\psi \rightarrow \varphi) \quad (38)$$

$$\vdash \psi \rightarrow (\neg\psi \rightarrow \varphi) \quad (39)$$

$$\vdash \neg\neg\varphi \rightarrow \varphi \quad (40)$$

$$\vdash \varphi \rightarrow \neg\neg\varphi \quad (41)$$

$$\vdash (\varphi \rightarrow \psi) \rightarrow (\neg\psi \rightarrow \neg\varphi) \quad (42)$$

$$\{\psi, \neg\varphi\} \vdash \neg(\psi \rightarrow \varphi) \quad (43)$$

$$\vdash (\varphi \rightarrow \neg\varphi) \rightarrow \neg\varphi \quad (44)$$

$$\vdash (\neg\varphi \rightarrow \varphi) \rightarrow \varphi \quad (45)$$

Dem.: Exercițiu.



Propoziția 1.67

Pentru orice mulțime de formule Γ și orice formule φ, ψ ,

$$\Gamma \cup \{\psi\} \vdash \varphi \text{ și } \Gamma \cup \{\neg\psi\} \vdash \varphi \Rightarrow \Gamma \vdash \varphi.$$

Dem.:

- | | | |
|-----|---|---------------------|
| (1) | $\Gamma \cup \{\psi\} \vdash \varphi$ | ipoteză |
| (2) | $\Gamma \vdash \psi \rightarrow \varphi$ | Teorema deducției |
| (3) | $\Gamma \cup \{\neg\psi\} \vdash \varphi$ | ipoteză |
| (4) | $\Gamma \vdash \neg\psi \rightarrow \varphi$ | Teorema deducției |
| (5) | $\Gamma \vdash (\psi \rightarrow \varphi) \rightarrow (\neg\varphi \rightarrow \neg\psi)$ | (42) și P.1.54.(ii) |
| (6) | $\Gamma \vdash \neg\varphi \rightarrow \neg\psi$ | (MP): (2), (5) |
| (7) | $\Gamma \vdash \neg\varphi \rightarrow \varphi$ | (6), (4) și P. 1.63 |
| (8) | $\Gamma \vdash (\neg\varphi \rightarrow \varphi) \rightarrow \varphi$ | (45) și P.1.54.(ii) |
| (9) | $\Gamma \vdash \varphi$ | (MP): (7), (8). |



Propoziția 1.68

Pentru orice formule φ, ψ ,

$$\{\varphi \wedge \psi\} \vdash \varphi \quad (46)$$

$$\{\varphi \wedge \psi\} \vdash \psi \quad (47)$$

$$\{\varphi, \psi\} \vdash \varphi \wedge \psi \quad (48)$$

$$\{\varphi, \psi\} \vdash \chi \quad \text{dacă} \quad \{\varphi \wedge \psi\} \vdash \chi \quad (49)$$

$$\vdash \varphi \wedge \psi \leftrightarrow \psi \wedge \varphi \quad (50)$$

Dem.: Exercițiu.



SINTAXA și SEMANTICA

Teorema 1.69 (Teorema de corectitudine (Soundness Theorem))

Orice teoremă este tautologie:

$$\vdash \varphi \Rightarrow \models \varphi$$

pentru orice $\varphi \in \text{Form}$.

Dem.: Fie

$\Sigma :=$ mulțimea tuturor tautologiilor lui LP .

Trebuie să demonstrăm că $\text{Thm} \subseteq \Sigma$. O facem prin inducție după teoreme.

- ▶ Axiomele sunt în Σ (**exercițiu**).
- ▶ Demonstrăm acum că Σ este închisă la modus ponens.

Presupunem că $\varphi, \varphi \rightarrow \psi \in \Sigma$, adică, $\models \varphi$ și $\models \varphi \rightarrow \psi$, deci $\models \varphi \wedge (\varphi \rightarrow \psi)$. Aplicăm (2) pentru a obține că $\models \psi$, adică, $\psi \in \Sigma$. □



Fie $e : V \rightarrow \{0, 1\}$ o evaluare și $v \in V$ o variabilă.

Definim

$$v^e = \begin{cases} v & \text{dacă } e(v) = 1 \\ \neg v & \text{dacă } e(v) = 0. \end{cases}$$

Așadar, $e^+(v^e) = 1$.

Pentru orice mulțime $W = \{x_1, \dots, x_k\}$ de variabile, notăm

$$W^e = \{v^e \mid v \in W\} = \{x_1^e, x_2^e, \dots, x_k^e\}.$$

Pentru orice $a \in \{0, 1\}$, definim evaluarea $e_{v \mapsto a} : V \rightarrow \{0, 1\}$ prin

$$e_{v \mapsto a}(x) = \begin{cases} e(x) & \text{dacă } x \neq v \\ a & \text{dacă } x = v. \end{cases}$$

Propoziția 1.70

Fie $e : V \rightarrow \{0, 1\}$ o evaluare. Pentru orice formulă φ ,

- (i) Dacă $e^+(\varphi) = 1$, atunci $\text{Var}(\varphi)^e \vdash \varphi$.
- (ii) Dacă $e^+(\varphi) = 0$, atunci $\text{Var}(\varphi)^e \vdash \neg\varphi$.

Dem.: Prin inducție după formule. Avem următoarele cazuri:

- ▶ $\varphi = v$. Atunci $\text{Var}(\varphi)^e = \{v^e\}$ și $e^+(v) = e(v)$.
Dacă $e(v) = 1$, atunci $v^e = v$, deci, $\{v^e\} \vdash v$.
Dacă $e(v) = 0$, atunci $v^e = \neg v$, deci, $\{v^e\} \vdash \neg v$.
- ▶ $\varphi = \neg\psi$. Atunci $\text{Var}(\varphi) = \text{Var}(\psi)$, deci $\text{Var}(\varphi)^e = \text{Var}(\psi)^e$.
Dacă $e^+(\varphi) = 1$, atunci $e^+(\psi) = 0$, deci, conform ipotezei de inducție pentru ψ , $\text{Var}(\psi)^e \vdash \neg\psi$, adică, $\text{Var}(\varphi)^e \vdash \varphi$.
Dacă $e^+(\varphi) = 0$, atunci $e^+(\psi) = 1$, deci, conform ipotezei de inducție pentru ψ , $\text{Var}(\psi)^e \vdash \psi$, adică, $\text{Var}(\varphi)^e \vdash \psi$.
Deoarece $\vdash \psi \rightarrow \neg\neg\psi$ ((41) din Propoziția 1.66), putem aplica (MP) pentru a obține $\text{Var}(\varphi)^e \vdash \neg\neg\psi = \neg\varphi$.



- $\varphi = \psi \rightarrow \chi$. Atunci $Var(\varphi) = Var(\psi) \cup Var(\chi)$, deci $Var(\psi)^e, Var(\chi)^e \subseteq Var(\varphi)^e$.

Dacă $e^+(\psi \rightarrow \chi) = 0$, atunci $e^+(\psi) = 1$ și $e^+(\chi) = 0$. Avem

$Var(\psi)^e \vdash \psi$ ipoteza de inducție pentru ψ

$Var(\chi)^e \vdash \neg\chi$ ipoteza de inducție pentru χ

$Var(\varphi)^e \vdash \{\psi, \neg\chi\}$ $Var(\psi)^e, Var(\chi)^e \subseteq Var(\varphi)^e$ și P. 1.54.(i)

$\{\psi, \neg\chi\} \vdash \neg(\psi \rightarrow \chi)$ (43) din Propoziția 1.66

$Var(\varphi)^e \vdash \neg(\psi \rightarrow \chi)$ Propoziția 1.54.(iv).

Dacă $e^+(\psi \rightarrow \chi) = 1$, atunci $e^+(\psi) = 0$ sau $e^+(\chi) = 1$.

În primul caz, obținem

$Var(\psi)^e \vdash \neg\psi$	ipoteza de inducție pentru ψ
$Var(\psi)^e \vdash \neg\psi \rightarrow (\psi \rightarrow \chi)$	(38) din P. 1.66 și P. 1.54.(ii)
$Var(\psi)^e \vdash \psi \rightarrow \chi$	(MP)
$Var(\varphi)^e \vdash \psi \rightarrow \chi$	$Var(\psi)^e \subseteq Var(\varphi)^e$ și P. 1.54.(i).

În al doilea caz, obținem

$Var(\chi)^e \vdash \chi$	ipoteza de inducție pentru χ
$Var(\chi)^e \vdash \chi \rightarrow (\psi \rightarrow \chi)$	(A1) și Propoziția 1.53.(i)
$Var(\chi)^e \vdash \psi \rightarrow \chi$	(MP)
$Var(\varphi)^e \vdash \psi \rightarrow \chi$	$Var(\chi)^e \subseteq Var(\varphi)^e$ și P. 1.54.(i). $\square$

Demonstrația propoziției anterioare ne dă o construcție **efectivă** a unei demonstrații a lui φ sau $\neg\varphi$ din premisele $Var(\varphi)^e$.



Teorema de completitudine

Teorema 1.71 (Teorema de completitudine)

Pentru orice formulă φ ,

$$\vdash \varphi \quad \text{dacă} \quad \models \varphi.$$

Dem.: " $\Rightarrow$ " Se aplică Teorema de corectitudine 1.69.

" $\Leftarrow$ " Fie φ o tautologie și $\text{Var}(\varphi) = \{x_1, \dots, x_n\}$. Demonstrăm prin inducție după k următoarea proprietate:

$$(*) \quad \text{pentru orice } k \leq n, \text{ pentru orice } e : V \rightarrow \{0, 1\}, \\ \{x_1^e, \dots, x_{n-k}^e\} \vdash \varphi.$$

Pentru $k = n$, $(*)$ ne dă $\vdash \varphi$.

$k = 0$. Fie $e : V \rightarrow \{0, 1\}$. Deoarece φ este tautologie, $e^+(\varphi) = 1$. Aplicând Propoziția 1.70, obținem că

$$\text{Var}(\varphi)^e = \{x_1^e, \dots, x_n^e\} \vdash \varphi.$$

Teorema de completitudine

$k \Rightarrow k + 1$. Presupunem că $(*)$ este adevărată pentru k și fie $e : V \rightarrow \{0, 1\}$. Trebuie să arătăm că $\{x_1^e, \dots, x_{n-k-1}^e\} \vdash \varphi$. Considerăm evaluarea $e' := e_{x_{n-k} \mapsto \neg e(x_{n-k})}$. Așadar, $e'(v) = e(v)$ pentru orice $v \neq x_{n-k}$ și

$$e'(x_{n-k}) = \begin{cases} 0 & \text{dacă } e(x_{n-k}) = 1 \\ 1 & \text{dacă } e(x_{n-k}) = 0. \end{cases}$$

Rezultă că $x_i^{e'} = x_i^e$ pentru orice $i \in \{1, \dots, n-k-1\}$ și

$$x_{n-k}^{e'} = \begin{cases} \neg x_{n-k} & \text{dacă } x_{n-k}^e = x_{n-k} \\ x_{n-k} & \text{dacă } x_{n-k}^e = \neg x_{n-k}. \end{cases}$$

Din $(*)$ pentru e și e' , obținem

$$\{x_1^e, \dots, x_{n-k-1}^e, x_{n-k}^e\} \vdash \varphi \text{ și } \{x_1^e, \dots, x_{n-k-1}^e, \neg x_{n-k}^e\} \vdash \varphi.$$

Aplicăm acum Propoziția 1.67 cu $\Gamma := \{x_1^e, \dots, x_{n-k-1}^e\}$ și $\psi := x_{n-k}$ pentru a conclud că $\{x_1^e, \dots, x_{n-k-1}^e\} \vdash \varphi$. □



Propoziția 1.72

Fie $\Gamma \cup \{\varphi, \psi\} \subseteq \text{Form}$. Presupunem că $\varphi \sim \psi$. Atunci

$$\Gamma \vdash \varphi \iff \Gamma \vdash \psi.$$

Dem.: Observăm că

$$\varphi \sim \psi \iff \models \varphi \rightarrow \psi \text{ și } \models \psi \rightarrow \varphi$$

Propoziția 1.16

$$\iff \vdash \varphi \rightarrow \psi \text{ și } \vdash \psi \rightarrow \varphi$$

Teorema de completitudine.

" $\Rightarrow$ " Presupunem că $\Gamma \vdash \varphi$. Deoarece $\vdash \varphi \rightarrow \psi$, rezultă din Propoziția 1.54.(ii) că $\Gamma \vdash \varphi \rightarrow \psi$. Aplicăm acum (MP) pentru a obține că $\Gamma \vdash \psi$.

" $\Leftarrow$ " Similar.





Fie Γ o mulțime de formule.

Definiția 1.73

- O evaluare $e : V \rightarrow \{0, 1\}$ este **model** al lui Γ dacă este model al fiecărei formule din Γ (adică $e \models \gamma$ pentru orice $\gamma \in \Gamma$).

Notăție: $e \models \Gamma$.

- Γ este **satisfiabilă** dacă are un model.
- Dacă Γ nu este satisfiabilă, spunem și că Γ este **nesatisfiabilă** sau **contradictorie**.

Notății: Mulțimea tuturor modelelor lui Γ se notează $Mod(\Gamma)$.

- $Mod(\Gamma) = \bigcap_{\varphi \in \Gamma} Mod(\varphi)$.



Mulțimi de formule - semantică

Fie Γ, Δ mulțimi de formule.

Definiția 1.74

O formulă φ este **consecință semantică** a lui Γ dacă $\text{Mod}(\Gamma) \subseteq \text{Mod}(\varphi)$. **Notăție:** $\Gamma \models \varphi$.

Dacă φ **nu** este consecință semantică a lui Γ , scriem $\Gamma \not\models \varphi$.

Notăm cu $\text{Cn}(\Gamma)$ mulțimea consecințelor semantice ale lui Γ .
Așadar,

$$\text{Cn}(\Gamma) = \{\varphi \in \text{Form} \mid \Gamma \models \varphi\}.$$

Definiția 1.75

- ▶ Δ este **consecință semantică** a lui Γ dacă $\text{Mod}(\Gamma) \subseteq \text{Mod}(\Delta)$.

Notăție: $\Gamma \models \Delta$.

- ▶ Γ și Δ sunt **(logic) echivalente** dacă $\text{Mod}(\Gamma) = \text{Mod}(\Delta)$.

Notăție: $\Gamma \sim \Delta$.



Următoarele rezultate colectează diverse proprietăți utile.

Observație

- ▶ $\psi \models \varphi$ ddacă $\{\psi\} \models \varphi$ ddacă $\{\psi\} \models \{\varphi\}$.
- ▶ $\psi \sim \varphi$ ddacă $\{\psi\} \sim \{\varphi\}$.

Propoziția 1.76

- ▶ $Mod(\emptyset) = Fun(V, \{0, 1\})$, adică orice evaluare $e : V \rightarrow \{0, 1\}$ este model al mulțimii vide. În particular, mulțimea vidă este satisfiabilă.
- ▶ $Cn(\emptyset)$ este mulțimea tuturor tautologiilor, adică φ este tautologie ddacă $\emptyset \models \varphi$.

Dem.: Exercițiu ușor.



Propoziția 1.77

Fie Γ o mulțime de formule. Următoarele afirmații sunt echivalente:

- (i) Γ este nesatisfiabilă.
- (ii) $\Gamma \models \varphi$ pentru orice formulă φ .
- (iii) $\Gamma \models \varphi$ pentru orice formulă nesatisfiabilă φ .
- (iv) $\Gamma \models \perp$.

Dem.: Exercițiu ușor.



Fie Γ o mulțime de formule și φ o formulă.

Notății

$\Gamma \not\vdash \varphi$: $\Leftrightarrow$ φ nu este Γ -teoremă

$\not\vdash \varphi$: $\Leftrightarrow$ φ nu este teoremă

$\Gamma \not\models \varphi$: $\Leftrightarrow$ φ nu este consecință semantică a lui Γ

$\not\models \varphi$: $\Leftrightarrow$ φ nu este tautologie.



Definiția 1.78

Fie Γ o mulțime de formule.

- ▶ Γ este **consistentă** dacă există o formulă φ astfel încât $\Gamma \not\vdash \varphi$.
- ▶ Γ este **inconsistentă** dacă nu este consistentă, adică, $\Gamma \vdash \varphi$ pentru orice formulă φ .

Observație

Fie Γ, Δ mulțimi de formule a.î. $\Gamma \subseteq \Delta$.

- ▶ Dacă Δ este consistentă, atunci și Γ este consistentă.
- ▶ Dacă Γ este inconsistentă, atunci și Δ este inconsistentă.



Propoziția 1.79

- (i) $\emptyset$ este consistentă.
- (ii) Mulțimea teoremelor este consistentă.

Dem.:

- (i) Dacă $\vdash \perp$, atunci, conform Teoremei de corectitudine 1.69, ar rezulta că $\models \perp$, o contradicție. Așadar $\nvdash \perp$, deci $\emptyset$ este consistentă.
- (ii) Aplicând Propoziția 1.54.(iv) pentru $\Gamma = \emptyset$, obținem că $Thm = Thm(Thm)$, adică, pentru orice φ ,
$$\vdash \varphi \text{ ddacă } Thm \vdash \varphi.$$

Din (i) rezultă că Thm este consistentă.





Propoziția 1.80

Pentru o mulțime de formule Γ sunt echivalente:

- (i) Γ este inconsistentă.*
- (ii) Pentru orice formulă ψ , $\Gamma \vdash \psi$ și $\Gamma \vdash \neg\psi$.*
- (iii) Există o formulă ψ a.î. $\Gamma \vdash \psi$ și $\Gamma \vdash \neg\psi$.*
- (iv) $\Gamma \vdash \perp$.*

Dem.: Exercițiu suplimentar.



Teorema de completitudine tare

Teorema 1.81 (Teorema de completitudine tare - versiunea 1)

Pentru orice mulțime de formule Γ ,

$$\Gamma \text{ este consistentă} \iff \Gamma \text{ este satisfiabilă.}$$

Teorema 1.82 (Teorema de completitudine tare - versiunea 2)

Pentru orice mulțime de formule Γ și orice formulă φ ,

$$\Gamma \vdash \varphi \iff \Gamma \models \varphi.$$

Observație

Se poate arăta că cele două versiuni sunt echivalente.



LOGICA DE ORDINUL ÎNTÂI

Definiția 2.1

Un limbaj $\mathcal{L}$ de ordinul I este format din:

- ▶ o mulțime numărabilă $V = \{v_n \mid n \in \mathbb{N}\}$ de variabile;
 - ▶ conectorii $\neg$ și $\rightarrow$;
 - ▶ paranteze: $(,)$;
 - ▶ simbolul de egalitate $=$;
 - ▶ cuantificatorul universal $\forall$;
 - ▶ o mulțime $\mathcal{R}$ de simboluri de relații;
 - ▶ o mulțime $\mathcal{F}$ de simboluri de funcții;
 - ▶ o mulțime $\mathcal{C}$ de simboluri de constante;
 - ▶ o funcție *aritate* $\text{ari} : \mathcal{F} \cup \mathcal{R} \rightarrow \mathbb{N}^*$.
- ▶ $\mathcal{L}$ este unic determinat de cvadruplul $\tau := (\mathcal{R}, \mathcal{F}, \mathcal{C}, \text{ari})$.
- ▶ τ se numește **signatura** lui $\mathcal{L}$ sau **vocabularul** lui $\mathcal{L}$ sau **alfabetul** lui $\mathcal{L}$ sau **tipul de similaritate** al lui $\mathcal{L}$

Fie $\mathcal{L}$ un limbaj de ordinul I.

- Mulțimea $\text{Sim}_{\mathcal{L}}$ a simbolurilor lui $\mathcal{L}$ este

$$\text{Sim}_{\mathcal{L}} := V \cup \{\neg, \rightarrow, (,), =, \forall\} \cup \mathcal{R} \cup \mathcal{F} \cup \mathcal{C}$$

- Elementele lui $\mathcal{R} \cup \mathcal{F} \cup \mathcal{C}$ se numesc simboluri ne-logice.
- Elementele lui $V \cup \{\neg, \rightarrow, (,), =, \forall\}$ se numesc simboluri logice.
- Notăm variabilele cu $x, y, z, v, \dots$, simbolurile de relații cu $P, Q, R, \dots$, simbolurile de funcții cu $f, g, h, \dots$ și simbolurile de constante cu $c, d, e, \dots$
- Pentru orice $m \in \mathbb{N}^*$ notăm:
 $\mathcal{F}_m$:= mulțimea simbolurilor de funcții de aritate m ;
 $\mathcal{R}_m$:= mulțimea simbolurilor de relații de aritate m .

Definiția 2.2

Mulțimea $\text{Expr}_{\mathcal{L}}$ a **expresiilor** lui $\mathcal{L}$ este mulțimea tuturor șirurilor finite de simboluri ale lui $\mathcal{L}$.

- ▶ Expresia vidă se notează λ .
- ▶ **Lungimea** unei expresii θ este numărul simbolurilor din θ .

Definiția 2.3

Fie $\theta = \theta_0\theta_1 \dots \theta_{k-1}$ o expresie a lui $\mathcal{L}$, unde $\theta_i \in \text{Sim}_{\mathcal{L}}$ pentru orice i .

- ▶ Dacă $0 \leq i \leq j \leq k - 1$, atunci expresia $\theta_i \dots \theta_j$ se numește (i, j) -**subexpresia** lui θ ;
- ▶ Spunem că o expresie ψ **apare** în θ dacă există $0 \leq i \leq j \leq k - 1$ a.î. ψ este (i, j) -subexpresia lui θ ;

Notăm cu $\text{Var}(\theta)$ mulțimea variabilelor care apar în θ .



Definiția 2.4

Termenii lui $\mathcal{L}$ sunt expresiile definite astfel:

- (T0) Orice variabilă este termen.
 - (T1) Orice simbol de constantă este termen.
 - (T2) Dacă $m \in \mathbb{N}^*$, $f \in \mathcal{F}_m$ și $t_1, \dots, t_m$ sunt termeni, atunci $ft_1 \dots t_m$ este termen.
 - (T3) Numai expresiile obținute aplicând regulile (T0), (T1), (T2) sunt termeni.
- ▶ Mulțimea termenilor se notează $\text{Term}_{\mathcal{L}}$.
 - ▶ Termeni: $t, s, t_1, t_2, s_1, s_2, \dots$

Definiția 2.5

Un termen t se numește **închis** dacă $\text{Var}(t) = \emptyset$.



Definiție alternativă

Mulțimea $Term_{\mathcal{L}}$ a termenilor lui $\mathcal{L}$ este intersecția tuturor mulțimilor de expresii Γ care satisfac următoarele proprietăți:

- ▶ orice variabilă este element al lui Γ ;
- ▶ orice simbol de constantă este element al lui Γ ;
- ▶ dacă $m \in \mathbb{N}^*$, $f \in \mathcal{F}_m$ și $t_1, \dots, t_m \in \Gamma$, atunci $ft_1 \dots t_m \in \Gamma$.



Citire unică (Unique readability)

Dacă t este un termen, atunci **exact** una din următoarele alternative are loc:

- ▶ $t = v$, unde $v \in V$;
- ▶ $t = c$, unde $c \in \mathcal{C}$;
- ▶ $t = ft_1 \dots t_m$, unde $m \in \mathbb{N}^*$, $f \in \mathcal{F}_m$ și $t_1, \dots, t_m$ sunt termeni ai lui $\mathcal{L}$.

Mai mult, scrierea lui t sub una din aceste forme este unică.

- ▶ demonstrații tehnice: [Monk, Mathematical Logic](#), Springer, 1976



Propoziția 2.6 (Principiul inducției pe termeni)

Fie Γ o mulțime de termeni care are următoarele proprietăți:

- ▶ *Γ conține variabilele și simbolurile de constante;*
- ▶ *dacă $m \in \mathbb{N}^*$, $f \in \mathcal{F}_m$ și $t_1, \dots, t_m \in \Gamma$, atunci $ft_1 \dots t_m \in \Gamma$.*

Atunci $\Gamma = \text{Term}_{\mathcal{L}}$.

Inducția pe termeni este folosită pentru a demonstra că toți termenii au o proprietate ***P***: definim Γ ca fiind mulțimea tuturor termenilor care satisfac ***P*** și aplicăm inducția pe termeni pentru a obține că $\Gamma = \text{Term}_{\mathcal{L}}$.



Propoziția 2.7 (Principiul recursiei pe termeni)

Fie D o mulțime și funcțiile

$$G_0 : V \cup \mathcal{C} \rightarrow D,$$

$$G_f : D^m \times \text{Term}_{\mathcal{L}}^m \rightarrow D \text{ pentru orice } m \in \mathbb{N}^*, f \in \mathcal{F}_m.$$

Atunci există o unică funcție

$$F : \text{Term}_{\mathcal{L}} \rightarrow D$$

care satisface următoarele proprietăți:

- (i) $F(t) = G_0(t)$ pentru orice $t \in V \cup \mathcal{C}$.*
- (ii) $F(ft_1 \dots t_m) = G_f(F(t_1), F(t_2), \dots, F(t_m), t_1, \dots, t_m)$ pentru orice $m \in \mathbb{N}^*, f \in \mathcal{F}_m$ și $t_1, \dots, t_m$ sunt termeni ai lui $\mathcal{L}$.*



Definiția 2.8

Formulele atomice ale lui $\mathcal{L}$ sunt expresiile de forma:

- ▶ $(s = t)$, unde s, t sunt termeni;
- ▶ $(Rt_1 \dots t_m)$, unde $m \in \mathbb{N}^*$, $R \in \mathcal{R}_m$ și $t_1, \dots, t_m$ sunt termeni.

Notăție

Mulțimea formulelor atomice se notează $\text{AtomicForm}_{\mathcal{L}}$.



Definiția 2.9

Formulele lui $\mathcal{L}$ sunt expresiile definite astfel:

- (F0) Orice formulă atomică este formulă.
- (F1) Dacă φ este formulă, atunci $(\neg\varphi)$ este formulă.
- (F2) Dacă φ și ψ sunt formule, atunci $(\varphi \rightarrow \psi)$ este formulă.
- (F3) Dacă φ este formulă, atunci $(\forall x\varphi)$ este formulă pentru orice variabilă x .
- (F4) Numai expresiile obținute aplicând regulile (F0), (F1), (F2), (F3) sunt formule.

Notății

- ▶ Mulțimea formulelor se notează $Form_{\mathcal{L}}$.
- ▶ Formule: $\varphi, \psi, \chi, \dots$



Definiție alternativă

Mulțimea $Form_{\mathcal{L}}$ a **formulelor** lui $\mathcal{L}$ este intersecția tuturor mulțimilor de expresii Γ care satisfac următoarele proprietăți:

- ▶ orice formulă atomică este element al lui Γ ;
- ▶ Γ este închisă la $\neg$, adică $\varphi \in \Gamma$ implică $(\neg\varphi) \in \Gamma$;
- ▶ Γ este închisă la $\rightarrow$, adică $\varphi, \psi \in \Gamma$ implică $(\varphi \rightarrow \psi) \in \Gamma$;
- ▶ Γ este închisă la $\forall x$ (pentru orice variabilă x), adică $\varphi \in \Gamma$ implică $(\forall x\varphi) \in \Gamma$ pentru orice variabilă x .



Citire unică (Unique readability)

Dacă φ este o formulă, atunci **exact** una din următoarele alternative are loc:

- ▶ $\varphi = (s = t)$, unde s, t sunt termeni;
- ▶ $\varphi = (Rt_1 \dots t_m)$, unde $m \in \mathbb{N}^*$, $R \in \mathcal{R}_m$ și $t_1, \dots, t_m$ sunt termeni;
- ▶ $\varphi = (\neg\psi)$, unde ψ este formulă;
- ▶ $\varphi = (\psi \rightarrow \chi)$, unde ψ, χ sunt formule;
- ▶ $\varphi = (\forall x\psi)$, unde x este variabilă și ψ este formulă.

Mai mult, scrierea lui φ sub una din aceste forme este unică.

- ▶ demonstrații tehnice: [Monk, Mathematical Logic](#), Springer, 1976



Propoziția 2.10 (Principiul inducției pe formule)

Fie Γ o mulțime de formule care are următoarele proprietăți:

- ▶ *Γ conține toate formulele atomice;*
- ▶ *Γ este închisă la $\neg, \rightarrow$ și $\forall x$ (pentru orice variabilă x).*

Atunci $\Gamma = \text{Form}_{\mathcal{L}}$.

Este folosită pentru a demonstra că toate formulele satisfac o proprietate **P**: definim Γ ca fiind mulțimea tuturor formulelor care satisfac **P** și aplicăm inducția pe formule pentru a obține că $\Gamma = \text{Form}_{\mathcal{L}}$.

Propoziția 2.11 (Principiul recursiei pe formule)

Fie D o mulțime și funcțiile

$$G_{\neg} : D \times \text{Form}_{\mathcal{L}} \rightarrow D, \quad G_{\rightarrow} : D^2 \times \text{Form}_{\mathcal{L}}^2 \rightarrow D, \\ G_0 : \text{AtomicForm}_{\mathcal{L}} \rightarrow D, \quad G_{\forall} : D \times V \times \text{Form}_{\mathcal{L}} \rightarrow D.$$

Atunci există o unică funcție

$$F : \text{Form}_{\mathcal{L}} \rightarrow D$$

care satisface următoarele proprietăți:

- (i) $F(\varphi) = G_0(\varphi)$ pentru orice formulă atomică φ .*
- (ii) $F((\neg\varphi)) = G_{\neg}(F(\varphi), \varphi)$ pentru orice formulă φ .*
- (iii) $F((\varphi \rightarrow \psi)) = G_{\rightarrow}(F(\varphi), F(\psi), \varphi, \psi)$ pentru orice formule φ, ψ .*
- (iv) $F((\forall x\varphi)) = G_{\forall}(F(\varphi), x, \varphi)$ pentru orice formulă φ și orice variabilă x .*



Conectori derivați

Conectorii $\vee$, $\wedge$, $\leftrightarrow$ și **cuantificatorul existențial** $\exists$ sunt introduși prin următoarele abrevieri:

$$\varphi \vee \psi \quad := \quad ((\neg\varphi) \rightarrow \psi)$$

$$\varphi \wedge \psi \quad := \quad \neg(\varphi \rightarrow (\neg\psi))$$

$$\varphi \leftrightarrow \psi \quad := \quad ((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$$

$$\exists x\varphi \quad := \quad (\neg\forall x(\neg\varphi)).$$



Notății

Fie $\varphi_1, \varphi_2, \dots, \varphi_n$ formule. Pentru $n \geq 3$, notăm

$$\varphi_1 \wedge \dots \wedge \varphi_n := (((\dots (\varphi_1 \wedge \varphi_2) \wedge \varphi_3) \wedge \dots \wedge \varphi_{n-1}) \wedge \varphi_n$$

$$\varphi_1 \vee \dots \vee \varphi_n := (((\dots (\varphi_1 \vee \varphi_2) \vee \varphi_3) \vee \dots \vee \varphi_{n-1}) \vee \varphi_n.$$

- ▶ $\varphi_1 \wedge \dots \wedge \varphi_n$ se mai scrie și $\bigwedge_{i=1}^n \varphi_i$ sau $\bigwedge_{i=1}^n \varphi_i$.
- ▶ $\varphi_1 \vee \dots \vee \varphi_n$ se mai scrie și $\bigvee_{i=1}^n \varphi_i$ sau $\bigvee_{i=1}^n \varphi_i$.



În practică, renunțăm la parantezele exterioare, le punem numai atunci când sunt necesare. Astfel, scriem $s = t$, $Rt_1 \dots t_m$, $\forall x\varphi$, $\neg\varphi$, $\varphi \rightarrow \psi$. Pe de altă parte, scriem $(\varphi \rightarrow \psi) \rightarrow \chi$.

Pentru a reduce din folosirea parantezelor, presupunem următoarele:

- ▶ Cuantificatorii $\forall$, $\exists$ au precedență mai mare decât ceilalți conectori. Așadar, $\forall x\varphi \rightarrow \psi$ este $(\forall x\varphi) \rightarrow \psi$ și nu $\forall x(\varphi \rightarrow \psi)$.
- ▶ $\neg$ are precedență mai mare decât $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$.
- ▶ $\wedge$, $\vee$ au precedență mai mare decât $\rightarrow$, $\leftrightarrow$.



- ▶ Scriem uneori $f(t_1, \dots, t_m)$ în loc de $ft_1 \dots t_m$ și $R(t_1, \dots, t_m)$ în loc de $Rt_1 \dots t_m$.
- ▶ Simbolurile de funcții sau relații de aritate 1 se numesc **unare**.
- ▶ Simbolurile de funcții sau relații de aritate 2 se numesc **binare**.
- ▶ Dacă f este un simbol de funcție binară scriem t_1ft_2 în loc de ft_1t_2 .
- ▶ Analog, dacă R este un simbol de relație binară, scriem t_1Rt_2 în loc de Rt_1t_2 .

Vom identifica un limbaj $\mathcal{L}$ cu mulțimea simbolurilor sale non-logice și vom scrie $\mathcal{L} = (\mathcal{R}, \mathcal{F}, \mathcal{C})$.

Definiția 2.12

O $\mathcal{L}$ -**structură** este un cvadruplu

$$\mathcal{A} = (A, \mathcal{F}^{\mathcal{A}}, \mathcal{R}^{\mathcal{A}}, \mathcal{C}^{\mathcal{A}})$$

unde

- ▶ A este o mulțime nevidă;
- ▶ $\mathcal{F}^{\mathcal{A}} = \{f^{\mathcal{A}} \mid f \in \mathcal{F}\}$ este o mulțime de operații pe A ; dacă f are aritatea m , atunci $f^{\mathcal{A}} : A^m \rightarrow A$;
- ▶ $\mathcal{R}^{\mathcal{A}} = \{R^{\mathcal{A}} \mid R \in \mathcal{R}\}$ este o mulțime de relații pe A ; dacă R are aritatea m , atunci $R^{\mathcal{A}} \subseteq A^m$;
- ▶ $\mathcal{C}^{\mathcal{A}} = \{c^{\mathcal{A}} \in A \mid c \in \mathcal{C}\}$.
- ▶ A se numește **universul** structurii $\mathcal{A}$. **Notăție:** $A = |\mathcal{A}|$
- ▶ $f^{\mathcal{A}}$ (respectiv $R^{\mathcal{A}}$, $c^{\mathcal{A}}$) se numește **denotația** sau **interpretarea** lui f (respectiv R , c) în $\mathcal{A}$.



Exemple - Limbajul egalității $\mathcal{L}_=$

$\mathcal{L}_= = (\mathcal{R}, \mathcal{F}, \mathcal{C})$, unde

$$\mathcal{R} = \mathcal{F} = \mathcal{C} = \emptyset.$$

$\mathcal{L}_=$ -structurile sunt mulțimile nevide.

Acest limbaj este potrivit doar pentru a exprima proprietăți ale egalității

Exemple de formule:

- egalitatea este simetrică:

$$\forall x \forall y (x = y \rightarrow y = x)$$

- universul are cel puțin trei elemente:

$$\exists x \exists y \exists z (\neg(x = y) \wedge \neg(y = z) \wedge \neg(z = x))$$



Exemple - Limbajul aritmeticii $\mathcal{L}_{ar}$

$\mathcal{L}_{ar} = (\mathcal{R}, \mathcal{F}, \mathcal{C})$, unde

- ▶ $\mathcal{R} = \{\dot{<}\}$; $\dot{<}$ este simbol de relație binar, adică are aritatea 2;
- ▶ $\mathcal{F} = \{\dot{+}, \dot{\times}, \dot{S}\}$; $\dot{+}, \dot{\times}$ sunt simboluri de operații binare și $\dot{S}$ este simbol de operație unar (adică are aritatea 1);
- ▶ $\mathcal{C} = \{\dot{0}\}$.

Scriem $\mathcal{L}_{ar} = (\dot{<}; \dot{+}, \dot{\times}, \dot{S}; \dot{0})$ sau $\mathcal{L}_{ar} = (\dot{<}, \dot{+}, \dot{\times}, \dot{S}, \dot{0})$.

Exemplul natural de $\mathcal{L}_{ar}$ -structură:

$$\mathcal{N} := (\mathbb{N}, <, +, \cdot, S, 0),$$

unde $S : \mathbb{N} \rightarrow \mathbb{N}$, $S(m) = m + 1$ este funcția succesor. Prin urmare,

$$\dot{<}^{\mathcal{N}} = <, \quad \dot{+}^{\mathcal{N}} = +, \quad \dot{\times}^{\mathcal{N}} = \cdot, \quad \dot{S}^{\mathcal{N}} = S, \quad \dot{0}^{\mathcal{N}} = 0.$$

- Alt exemplu de $\mathcal{L}_{ar}$ -structură: $\mathcal{A} = (\{0, 1\}, <, \vee, \wedge, \neg, 1)$.



Exemplu - Limbajul cu un simbol de relație binară

$\mathcal{L}_R = (\mathcal{R}, \mathcal{F}, \mathcal{C})$, unde

- ▶ $\mathcal{R} = \{R\}$; R simbol binar;
- ▶ $\mathcal{F} = \mathcal{C} = \emptyset$.

$\mathcal{L}_R$ -structurile sunt mulțimile nevide împreună cu o relație binară.

- ▶ Dacă suntem interesați de mulțimi parțial ordonate $(A, \leq)$, folosim simbolul $\leq$ în loc de R și notăm limbajul cu $\mathcal{L}_{\leq}$.
- ▶ Dacă suntem interesați de mulțimi strict ordonate $(A, <)$, folosim simbolul $<$ în loc de R și notăm limbajul cu $\mathcal{L}_{<}$.
- ▶ Dacă suntem interesați de grafuri $G = (V, E)$, folosim simbolul E în loc de R și notăm limbajul cu $\mathcal{L}_{Graf}$.
- ▶ Dacă suntem interesați de structuri $(A, \in)$, folosim simbolul $\in$ în loc de R și notăm limbajul cu $\mathcal{L}_{\in}$.



Exemple - Limbajul grupurilor $\mathcal{L}_{Gr}$

$\mathcal{L}_{Gr} = (\mathcal{R}, \mathcal{F}, \mathcal{C})$, unde

- ▶ $\mathcal{R} = \emptyset$;
- ▶ $\mathcal{F} = \{*, \dot{-}^1\}$; $*$ simbol binar, $\dot{-}^1$ simbol unar;
- ▶ $\mathcal{C} = \{\dot{e}\}$.

Scriem $\mathcal{L}_{Gr} = (\emptyset; *, \dot{-}^1; \dot{e})$ sau $\mathcal{L}_{Gr} = (*, \dot{-}^1, \dot{e})$.

Exemple naturale de $\mathcal{L}_{Gr}$ -structuri sunt grupurile: $\mathcal{G} = (G, \cdot, {}^{-1}, e)$.

Prin urmare, $*^{\mathcal{G}} = \cdot$, $\dot{-}^1{}^{\mathcal{G}} = {}^{-1}$, $\dot{e}^{\mathcal{G}} = e$.

Pentru a discuta despre grupuri abeliene (comutative), este tradițional să se folosească limbajul $\mathcal{L}_{AbGr} = (\mathcal{R}, \mathcal{F}, \mathcal{C})$, unde

- ▶ $\mathcal{R} = \emptyset$;
- ▶ $\mathcal{F} = \{\dot{+}, \dot{-}\}$; $\dot{+}$ simbol binar, $\dot{-}$ simbol unar;
- ▶ $\mathcal{C} = \{\dot{0}\}$.

Scriem $\mathcal{L}_{AbGr} = (\dot{+}, \dot{-}, \dot{0})$.



SEMANTICA



Fie $\mathcal{L}$ un limbaj de ordinul întâi și $\mathcal{A}$ o $\mathcal{L}$ -structură.

Definiția 2.13

O *interpretare* sau *evaluare* a (variabilelor) lui $\mathcal{L}$ în $\mathcal{A}$ este o funcție $e : V \rightarrow A$.

În continuare, $e : V \rightarrow A$ este o interpretare a lui $\mathcal{L}$ în $\mathcal{A}$.

Definiția 2.14 (Interpretarea termenilor)

Prin recursie pe termeni se definește *interpretarea* $t^{\mathcal{A}}(e) \in A$ a termenului t sub evaluarea e :

- ▶ dacă $t = x \in V$, atunci $t^{\mathcal{A}}(e) := e(x)$;
- ▶ dacă $t = c \in \mathcal{C}$, atunci $t^{\mathcal{A}}(e) := c^{\mathcal{A}}$;
- ▶ dacă $t = ft_1 \dots t_m$, atunci $t^{\mathcal{A}}(e) := f^{\mathcal{A}}(t_1^{\mathcal{A}}(e), \dots, t_m^{\mathcal{A}}(e))$.



Exemplu

Considerăm limbajul de ordinul I $\mathcal{L}_{ar} = (\dot{<}; \dot{+}, \dot{\times}, \dot{S}; \dot{0})$ și $\mathcal{L}_{ar}$ -structura $\mathcal{N} = (\mathbb{N}, <, +, \cdot, S, 0)$.

Fie $x, y \in V$ cu $x \neq y$ și

$$t = \dot{S}x \dot{\times} \dot{S}\dot{S}y = \dot{\times}(\dot{S}x, \dot{S}\dot{S}y).$$

Să se calculeze $t^{\mathcal{N}}(e)$, unde $e : V \rightarrow \mathbb{N}$ este o evaluare ce verifică $e(x) = 3$ și $e(y) = 7$.

Dem.: Pentru orice interpretare $e : V \rightarrow \mathbb{N}$, avem

$$\begin{aligned} t^{\mathcal{N}}(e) &= \dot{\times}^{\mathcal{N}}((\dot{S}x)^{\mathcal{N}}(e), (\dot{S}\dot{S}y)^{\mathcal{N}}(e)) = (\dot{S}x)^{\mathcal{N}}(e) \cdot (\dot{S}\dot{S}y)^{\mathcal{N}}(e) \\ &= \dot{S}^{\mathcal{N}}(x^{\mathcal{N}}(e)) \cdot \dot{S}^{\mathcal{N}}((\dot{S}y)^{\mathcal{N}}(e)) = S(e(x)) \cdot S(\dot{S}^{\mathcal{N}}(y^{\mathcal{N}}(e))) \\ &= S(e(x)) \cdot S(S(e(y))). \end{aligned}$$

Prin urmare, dacă $e(x) = 3$ și $e(y) = 7$, atunci

$$t^{\mathcal{N}}(e) = S(3) \cdot S(S(7)) = 4 \cdot 9 = 36.$$



Prin recursie pe formule se definește **interpretarea**

$$\varphi^{\mathcal{A}}(e) \in \{0, 1\}$$

a formulei φ sub evaluarea e .

$$(s = t)^{\mathcal{A}}(e) = \begin{cases} 1 & \text{dacă } s^{\mathcal{A}}(e) = t^{\mathcal{A}}(e) \\ 0 & \text{altfel.} \end{cases}$$

$$(Rt_1 \dots t_m)^{\mathcal{A}}(e) = \begin{cases} 1 & \text{dacă } (t_1^{\mathcal{A}}(e), \dots, t_m^{\mathcal{A}}(e)) \in R^{\mathcal{A}} \\ 0 & \text{altfel.} \end{cases}$$



Negația și implicația

- ▶ $(\neg\varphi)^{\mathcal{A}}(e) = 1 - \varphi^{\mathcal{A}}(e)$;
- ▶ $(\varphi \rightarrow \psi)^{\mathcal{A}}(e) = \varphi^{\mathcal{A}}(e) \rightarrow \psi^{\mathcal{A}}(e)$, unde,

$\rightarrow: \{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$,

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

Prin urmare,

- ▶ $(\neg\varphi)^{\mathcal{A}}(e) = 1 \iff \varphi^{\mathcal{A}}(e) = 0$.
- ▶ $(\varphi \rightarrow \psi)^{\mathcal{A}}(e) = 1 \iff \begin{aligned} &\varphi^{\mathcal{A}}(e) = 0 \text{ sau } \psi^{\mathcal{A}}(e) = 1 \\ &\iff \varphi^{\mathcal{A}}(e) \leq \psi^{\mathcal{A}}(e). \end{aligned}$



Notăție

Pentru orice variabilă $x \in V$ și orice $a \in A$, definim o nouă interpretare $e_{x \mapsto a} : V \rightarrow A$ prin

$$e_{x \mapsto a}(v) = \begin{cases} e(v) & \text{dacă } v \neq x \\ a & \text{dacă } v = x. \end{cases}$$

Interpretarea formulelor

$$(\forall x \varphi)^A(e) = \begin{cases} 1 & \text{dacă } \varphi^A(e_{x \mapsto a}) = 1 \text{ pentru orice } a \in A \\ 0 & \text{altfel.} \end{cases}$$

Fie $\mathcal{A}$ o $\mathcal{L}$ -structură și $e : V \rightarrow A$ o interpretare a lui $\mathcal{L}$ în $\mathcal{A}$.

Definiția 2.15

Fie φ o formulă. Spunem că:

- ▶ e **satisface** φ în $\mathcal{A}$ dacă $\varphi^{\mathcal{A}}(e) = 1$.
Notăție: $\mathcal{A} \models \varphi[e]$
- ▶ e **nu satisface** φ în $\mathcal{A}$ dacă $\varphi^{\mathcal{A}}(e) = 0$.
Notăție: $\mathcal{A} \not\models \varphi[e]$
- ▶ $(\mathcal{A}, e)$ este **model** al lui φ dacă $\mathcal{A} \models \varphi[e]$.

Definiția 2.16

Fie φ o formulă. Spunem că:

- ▶ φ este **satisfiabilă** dacă are un model.
- ▶ φ este **nesatisfiabilă** sau **contradictorie** dacă φ nu este satisfiabilă.



Corolar 2.17

Pentru orice formule φ, ψ și orice variabilă x ,

- (i) $\mathcal{A} \models (\neg\varphi)[e] \iff \mathcal{A} \not\models \varphi[e]$.
- (ii) $\mathcal{A} \models (\varphi \rightarrow \psi)[e] \iff \mathcal{A} \models \varphi[e] \text{ implică } \mathcal{A} \models \psi[e]$
 $\iff \mathcal{A} \not\models \varphi[e] \text{ sau } \mathcal{A} \models \psi[e]$.
- (iii) $\mathcal{A} \models (\forall x\varphi)[e] \iff \text{pentru orice } a \in A, \mathcal{A} \models \varphi[e_{x \mapsto a}]$.
- (iv) $\mathcal{A} \not\models (\forall x\varphi)[e] \iff \text{există } a \in A \text{ a.î. } \mathcal{A} \not\models \varphi[e_{x \mapsto a}]$.

Dem.: Exercițiu ușor.

Propoziția 2.18

Pentru orice formule φ, ψ și orice variabilă x ,

- (i) $(\varphi \vee \psi)^{\mathcal{A}}(e) = \varphi^{\mathcal{A}}(e) \vee \psi^{\mathcal{A}}(e)$.
- (ii) $(\varphi \wedge \psi)^{\mathcal{A}}(e) = \varphi^{\mathcal{A}}(e) \wedge \psi^{\mathcal{A}}(e)$.
- (iii) $(\varphi \leftrightarrow \psi)^{\mathcal{A}}(e) = \varphi^{\mathcal{A}}(e) \leftrightarrow \psi^{\mathcal{A}}(e)$.
- (iv) $(\exists x \varphi)^{\mathcal{A}}(e) = \begin{cases} 1 & \text{dacă există } a \in A \text{ a.î. } \varphi^{\mathcal{A}}(e_{x \mapsto a}) = 1 \\ 0 & \text{altfel.} \end{cases}$

$\vee, \wedge, \leftrightarrow: \{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$,

p	q	$p \vee q$	p	q	$p \wedge q$	p	q	$p \leftrightarrow q$
0	0	0	0	0	0	0	0	1
0	1	1	0	1	0	0	1	0
1	0	1	1	0	0	1	0	0
1	1	1	1	1	1	1	1	1



Propoziția 2.19

Pentru orice formule φ, ψ și orice variabilă x ,

- (i) $\mathcal{A} \models (\varphi \wedge \psi)[e] \iff \mathcal{A} \models \varphi[e] \text{ și } \mathcal{A} \models \psi[e].$
- (ii) $\mathcal{A} \models (\varphi \vee \psi)[e] \iff \mathcal{A} \models \varphi[e] \text{ sau } \mathcal{A} \models \psi[e].$
- (iii) $\mathcal{A} \models (\varphi \leftrightarrow \psi)[e] \iff \mathcal{A} \models \varphi[e] \text{ ddacă } \mathcal{A} \models \psi[e].$
- (iv) $\mathcal{A} \models (\exists x \varphi)[e] \iff \text{există } a \in A \text{ a.î. } \mathcal{A} \models \varphi[e_{x \mapsto a}].$
- (v) $\mathcal{A} \not\models (\exists x \varphi)[e] \iff \text{pentru orice } a \in A, \mathcal{A} \not\models \varphi[e_{x \mapsto a}].$

Dem.: Exercițiu ușor. Arătăm, de exemplu, (iv).

$$\begin{aligned} \mathcal{A} \models (\exists x \varphi)[e] &\iff \mathcal{A} \models (\neg \forall x \neg \varphi)[e] \iff \mathcal{A} \not\models (\forall x \neg \varphi)[e] \\ &\iff \text{există } a \in A \text{ a.î. } \mathcal{A} \not\models (\neg \varphi)[e_{x \mapsto a}] \\ &\iff \text{există } a \in A \text{ a.î. } \mathcal{A} \models \varphi[e_{x \mapsto a}]. \end{aligned}$$



Propoziția 2.20

Pentru orice $n \geq 2$ și orice formule $\varphi_1, \dots, \varphi_n$,

- ▶ $\mathcal{A} \models (\varphi_1 \wedge \dots \wedge \varphi_n)[e]$ ddacă $\mathcal{A} \models \varphi_i[e]$ pentru orice $i \in \{1, \dots, n\}$.
- ▶ $\mathcal{A} \models (\varphi_1 \vee \dots \vee \varphi_n)[e]$ ddacă există $i \in \{1, \dots, n\}$ a.î $\mathcal{A} \models \varphi_i[e]$.

Dem.: Exercițiu.

Fie φ formulă a lui $\mathcal{L}$.

Definiția 2.21

Spunem că φ este **adevărată** într-o $\mathcal{L}$ -structură $\mathcal{A}$ dacă pentru orice evaluare $e : V \rightarrow A$,

$$\mathcal{A} \models \varphi[e].$$

Spunem și că $\mathcal{A}$ **satisfacă** φ sau că $\mathcal{A}$ este un **model** al lui φ .

Notăție: $\mathcal{A} \models \varphi$

Definiția 2.22

Spunem că φ este formulă **universal adevărată** sau (**logic**) **validă** dacă pentru orice $\mathcal{L}$ -structură $\mathcal{A}$,

$$\mathcal{A} \models \varphi.$$

Notăție: $\models \varphi$

Fie φ, ψ formule ale lui $\mathcal{L}$.

Definiția 2.23

φ și ψ sunt **logic echivalente** dacă pentru orice $\mathcal{L}$ -structură $\mathcal{A}$ și orice evaluare $e : V \rightarrow A$,

$$\mathcal{A} \models \varphi[e] \iff \mathcal{A} \models \psi[e].$$

Notăție: $\varphi \models \psi$

Definiția 2.24

ψ este **consecință semantică** a lui φ dacă pentru orice $\mathcal{L}$ -structură $\mathcal{A}$ și orice evaluare $e : V \rightarrow A$,

$$\mathcal{A} \models \varphi[e] \implies \mathcal{A} \models \psi[e].$$

Notăție: $\varphi \models \psi$

Observație

- (i) $\varphi \models \psi$ ddacă $\models \varphi \rightarrow \psi$.
- (ii) $\varphi \models \psi$ ddacă $(\psi \models \varphi \text{ și } \varphi \models \psi)$ ddacă $\models \psi \leftrightarrow \varphi$.



Exemple

Considerăm limbajul de ordinul I $\mathcal{L}_{ar} = (\dot{<}; \dot{+}, \dot{\times}, \dot{S}; \dot{0})$ și

$$\varphi = x \dot{<} \dot{S}y \rightarrow (x \dot{<} y \vee x = y) = \dot{<}(x, \dot{S}y) \rightarrow (\dot{<}(x, y) \vee x = y).$$

Pentru orice $\mathcal{L}_{ar}$ -structură $\mathcal{A} = (A, \dot{<}^{\mathcal{A}}, \dot{+}^{\mathcal{A}}, \dot{\times}^{\mathcal{A}}, \dot{S}^{\mathcal{A}}, \dot{0}^{\mathcal{A}})$, și orice interpretare $e : V \rightarrow A$, avem

$$\begin{aligned} \mathcal{A} \models \varphi[e] &\iff \mathcal{A} \not\models (\dot{<}(x, \dot{S}y))[e] \text{ sau } \mathcal{A} \models (\dot{<}(x, y) \vee x = y)[e] \\ &\iff (e(x), \dot{S}^{\mathcal{A}}(e(y))) \notin \dot{<}^{\mathcal{A}} \text{ sau } \\ &\quad \mathcal{A} \models (\dot{<}(x, y))[e] \text{ sau } \mathcal{A} \models (x = y)[e] \\ &\iff (e(x), \dot{S}^{\mathcal{A}}(e(y))) \notin \dot{<}^{\mathcal{A}} \\ &\quad \text{sau } (e(x), e(y)) \in \dot{<}^{\mathcal{A}} \text{ sau } e(x) = e(y) \\ &\iff (\text{nu este adevărat că } e(x) \dot{<}^{\mathcal{A}} \dot{S}^{\mathcal{A}}(e(y))) \\ &\quad \text{sau } e(x) \dot{<}^{\mathcal{A}} e(y) \text{ sau } e(x) = e(y). \end{aligned}$$



Exemple

Fie $\mathcal{L}_{ar}$ -structura $\mathcal{N} = (\mathbb{N}, <, +, \cdot, S, 0)$. Să se arate că $\mathcal{N} \models \varphi$.

Dem.: Pentru orice interpretare $e : V \rightarrow \mathbb{N}$, avem

$$\begin{aligned}\mathcal{N} \models \varphi[e] &\iff \left(\text{nu este adevărat că } e(x) \dot{<}^{\mathcal{N}} S^{\mathcal{N}}(e(y)) \right) \\ &\quad \text{sau } e(x) \dot{<}^{\mathcal{N}} e(y) \text{ sau } e(x) = e(y) \\ &\iff \left(\text{nu este adevărat că } e(x) < S(e(y)) \right) \\ &\quad \text{sau } e(x) < e(y) \text{ sau } e(x) = e(y) \\ &\iff \left(\text{nu este adevărat că } e(x) < e(y) + 1 \right) \\ &\quad \text{sau } e(x) < e(y) \text{ sau } e(x) = e(y) \\ &\iff e(x) \geq e(y) + 1 \text{ sau } e(x) < e(y) \text{ sau } e(x) = e(y),\end{aligned}$$

ceea ce este adevărat. Prin urmare, $\mathcal{N} \models \varphi[e]$ pentru orice $e : V \rightarrow \mathbb{N}$, adică $\mathcal{N} \models \varphi$.



Exemple

Fie $\mathcal{L}_{ar}$ -structura $\mathcal{R} = (\mathbb{R}, <, +, \cdot, S, 0)$, unde

$S : \mathbb{R} \rightarrow \mathbb{R}$, $S(x) = x + 1$.

Să se arate că $\mathcal{R} \not\models \varphi$.

Dem.: Pentru orice interpretare $e : V \rightarrow \mathbb{R}$, avem

$$\begin{aligned}\mathcal{R} \models \varphi[e] &\iff \left(\text{nu este adevărat că } e(x) \dot{<}^{\mathcal{R}} S^{\mathcal{R}}(e(y)) \right) \\ &\quad \text{sau } e(x) \dot{<}^{\mathcal{R}} e(y) \text{ sau } e(x) = e(y) \\ &\iff \left(\text{nu este adevărat că } e(x) < S(e(y)) \right) \\ &\quad \text{sau } e(x) < e(y) \text{ sau } e(x) = e(y) \\ &\iff \left(\text{nu este adevărat că } e(x) < e(y) + 1 \right) \\ &\quad \text{sau } e(x) < e(y) \text{ sau } e(x) = e(y) \\ &\iff e(x) \geq e(y) + 1 \text{ sau } e(x) < e(y) \text{ sau } e(x) = e(y).\end{aligned}$$

Fie $e : V \rightarrow \mathbb{R}$ a.î. $e(x) = 1,5$ și $e(y) = 1,4$. Atunci $\mathcal{R} \not\models \varphi[e]$.

Prin urmare, $\mathcal{R} \not\models \varphi$.

Propoziția 2.25

Pentru orice termeni s, t, u ,

- (i) $\models t = t$;
- (ii) $\models s = t \rightarrow t = s$;
- (iii) $\models s = t \wedge t = u \rightarrow s = u$.

Dem.: Exercițiu ușor. Arătăm, de exemplu, (iii). Notăm, pentru simplitate,

$$\varphi := s = t \wedge t = u \rightarrow s = u.$$

Fie $\mathcal{A}$ o $\mathcal{L}$ -structură și $e : V \rightarrow A$. Trebuie să arătăm că $\mathcal{A} \models \varphi[e]$.
Obținem

$$\begin{aligned}\mathcal{A} \models \varphi[e] &\iff \mathcal{A} \models (s = t \wedge t = u)[e] \text{ implică } \mathcal{A} \models (s = u)[e] \\ &\iff (\mathcal{A} \models (s = t)[e] \text{ și } \mathcal{A} \models (t = u)[e]) \text{ implică } \mathcal{A} \models (s = u)[e] \\ &\iff (s^{\mathcal{A}}(e) = t^{\mathcal{A}}(e) \text{ și } t^{\mathcal{A}}(e) = u^{\mathcal{A}}(e)) \text{ implică } s^{\mathcal{A}}(e) = u^{\mathcal{A}}(e),\end{aligned}$$

ceea ce este evident.



Propoziția 2.26

Pentru orice formule φ, ψ și orice variabile x, y ,

$$\neg \exists x \varphi \models \forall x \neg \varphi \quad (51)$$

$$\neg \forall x \varphi \models \exists x \neg \varphi \quad (52)$$

$$\forall x (\varphi \wedge \psi) \models \forall x \varphi \wedge \forall x \psi \quad (53)$$

$$\exists x (\varphi \vee \psi) \models \exists x \varphi \vee \exists x \psi \quad (54)$$

Dem.: Fie $\mathcal{A}$ o $\mathcal{L}$ -structură și $e : V \rightarrow A$.

(51):

$$\begin{aligned} \mathcal{A} \models (\neg \exists x \varphi)[e] &\iff \mathcal{A} \not\models (\exists x \varphi)[e] \\ &\iff \text{pentru orice } a \in A, \mathcal{A} \not\models \varphi[e_{x \mapsto a}] \\ &\iff \text{pentru orice } a \in A, \mathcal{A} \models (\neg \varphi)[e_{x \mapsto a}] \\ &\iff \mathcal{A} \models (\forall x \neg \varphi)[e]. \end{aligned}$$



(53):

$$\begin{aligned}\mathcal{A} \models (\forall x(\varphi \wedge \psi))[e] &\iff \text{pentru orice } a \in A, \mathcal{A} \models (\varphi \wedge \psi)[e_{x \mapsto a}] \\ &\iff \text{pentru orice } a \in A, \mathcal{A} \models \varphi[e_{x \mapsto a}] \text{ și } \mathcal{A} \models \psi[e_{x \mapsto a}] \\ &\iff (\text{pentru orice } a \in A, \mathcal{A} \models \varphi[e_{x \mapsto a}]) \\ &\quad \text{și } (\text{pentru orice } a \in A, \mathcal{A} \models \psi[e_{x \mapsto a}]) \\ &\iff \mathcal{A} \models (\forall x \varphi)[e] \text{ și } \mathcal{A} \models (\forall x \psi)[e] \\ &\iff \mathcal{A} \models (\forall x \varphi \wedge \forall x \psi)[e].\end{aligned}$$

(52), (54): Exercițiu.





Propoziția 2.27

Pentru orice formule φ, ψ și orice variabile x, y ,

$$\forall x\varphi \vee \forall x\psi \models \forall x(\varphi \vee \psi) \quad (55)$$

$$\exists x(\varphi \wedge \psi) \models \exists x\varphi \wedge \exists x\psi \quad (56)$$

$$\forall x(\varphi \rightarrow \psi) \models \forall x\varphi \rightarrow \forall x\psi \quad (57)$$

$$\forall x(\varphi \rightarrow \psi) \models \exists x\varphi \rightarrow \exists x\psi \quad (58)$$

$$\forall x\varphi \models \exists x\varphi \quad (59)$$

$$\varphi \models \exists x\varphi \quad (60)$$

$$\forall x\varphi \models \varphi \quad (61)$$

Dem.: Exercițiu. Demonstrăm (57) ca exemplu. Fie $\mathcal{A}$ o $\mathcal{L}$ -structură și $e : V \rightarrow A$.

Avem că $\mathcal{A} \models (\forall x(\varphi \rightarrow \psi))[e] \iff$ pentru orice $a \in A$,

$\mathcal{A} \models (\varphi \rightarrow \psi)[e_{x \mapsto a}] \iff$ pentru orice $a \in A$,

$$\varphi^{\mathcal{A}}(e_{x \mapsto a}) \rightarrow \psi^{\mathcal{A}}(e_{x \mapsto a}) = 1 \iff$$

$$(*) \quad \text{pentru orice } a \in A, \varphi^{\mathcal{A}}(e_{x \mapsto a}) \leq \psi^{\mathcal{A}}(e_{x \mapsto a}).$$

Similar obținem că $\mathcal{A} \models (\forall x\varphi \rightarrow \forall x\psi)[e] \iff$

$$(**) \quad (\forall x\varphi)^{\mathcal{A}}(e) \leq (\forall x\psi)^{\mathcal{A}}(e).$$

Presupunem (*) și trebuie să demonstrăm (**).

Dacă $(\forall x\varphi)^{\mathcal{A}}(e) = 0$, (**) este evident.

Presupunem, așadar, că $(\forall x\varphi)^{\mathcal{A}}(e) = 1$, adică

$$(***) \quad \text{pentru orice } b \in A, \varphi^{\mathcal{A}}(e_{x \mapsto b}) = 1.$$

Ne rămâne de arătat că $(\forall x\psi)^{\mathcal{A}}(e) = 1$, adică

$$\text{pentru orice } c \in A, \psi^{\mathcal{A}}(e_{x \mapsto c}) = 1.$$

Fie $c \in A$. Din (*), avem că $\varphi^{\mathcal{A}}(e_{x \mapsto c}) \leq \psi^{\mathcal{A}}(e_{x \mapsto c})$, iar din (***),
că $\varphi^{\mathcal{A}}(e_{x \mapsto c}) = 1$. Rezultă că $\psi^{\mathcal{A}}(e_{x \mapsto c}) = 1$. □



Notății 2.28

Fie $\mathcal{A}$ o $\mathcal{L}$ -structură și $e : V \rightarrow A$. Pentru orice variabile distincte x, y și orice elemente $a, b \in A$, avem că

$$(e_{y \mapsto b})_{x \mapsto a} = (e_{x \mapsto a})_{y \mapsto b}.$$

Notăm valoarea comună $e_{x \mapsto a, y \mapsto b}$. Prin urmare,

$$e_{x \mapsto a, y \mapsto b} : V \rightarrow A, \quad e_{x \mapsto a, y \mapsto b}(v) = \begin{cases} e(v) & \text{dacă } v \neq x \text{ și } v \neq y \\ a & \text{dacă } v = x \\ b & \text{dacă } v = y. \end{cases}$$



Propoziția 2.29

Pentru orice formule φ, ψ și orice variabile distincte x, y ,

$$\forall x \forall y \varphi \models \forall y \forall x \varphi \quad (62)$$

$$\exists x \exists y \varphi \models \exists y \exists x \varphi \quad (63)$$

$$\exists y \forall x \varphi \models \forall x \exists y \varphi \quad (64)$$

Dem.: Exercițiu. Demonstrăm (64) ca exemplu. Fie $\mathcal{A}$ o $\mathcal{L}$ -structură și $e : V \rightarrow A$. Avem că

$$\begin{aligned} \mathcal{A} \models (\exists y \forall x \varphi)[e] &\iff \text{există } b \in A \text{ a.î. } \mathcal{A} \models (\forall x \varphi)[e_{y \mapsto b}] \\ &\iff \text{există } b \in A \text{ a.î. pentru orice } a \in A, \\ &\quad \mathcal{A} \models \varphi[(e_{y \mapsto b})_{x \mapsto a}]. \end{aligned}$$



Deci, $\mathcal{A} \models (\exists y \forall x \varphi)[e] \iff (*)$, unde

(*) există $b \in A$ a.î. pentru orice $a \in A$, $\mathcal{A} \models \varphi[e_{x \mapsto a, y \mapsto b}]$.

Obținem similar că $\mathcal{A} \models (\forall x \exists y \varphi)[e] \iff (**)$, unde

(**) pentru orice $a \in A$ există $c \in A$ a.î. $\mathcal{A} \models \varphi[e_{x \mapsto a, y \mapsto c}]$.

$(*) \Rightarrow (**)$ Fie b a.î. $(*)$ are loc. Pentru orice $a \in A$ luăm $c := b$.
Atunci $(**)$ are loc. □



Să se dea exemplu de limbaj $\mathcal{L}$ de ordinul 1 și de formulă φ a lui $\mathcal{L}$ astfel încât $\forall x \exists y \varphi \not\models \exists y \forall x \varphi$, unde x și y sunt variabile distincte.

Dem.: Considerăm $\mathcal{L}_{ar} = (\dot{<}, \dot{+}, \dot{\cdot}, \dot{S}, \dot{0})$, $\mathcal{L}_{ar}$ -structura $\mathcal{N} = (\mathbb{N}, <, +, \cdot, S, 0)$ și $e : V \rightarrow \mathbb{N}$ o evaluare arbitrară.
Fie

$$\varphi := x \dot{<} y.$$

Atunci

$$\begin{aligned} \mathcal{N} \models (\forall x \exists y \varphi)[e] &\iff \text{pentru orice } n \in \mathbb{N} \text{ există } m \in \mathbb{N} \text{ a.î.} \\ &\quad \mathcal{N} \models \varphi [e_{x \mapsto n, y \mapsto m}] \\ &\iff \text{pentru orice } n \in \mathbb{N} \text{ există } m \in \mathbb{N} \text{ a.î. } n < m, \end{aligned}$$

ceea ce este adevărat (se ia, de pildă, $m := n + 1$). Așadar,

$$\mathcal{N} \models (\forall x \exists y \varphi)[e].$$



Să se dea exemplu de limbaj $\mathcal{L}$ de ordinul I și de formulă φ a lui $\mathcal{L}$ astfel încât $\forall x \exists y \varphi \not\models \exists y \forall x \varphi$.

Dem.: (continuare) Pe de altă parte,

$$\begin{aligned} \mathcal{N} \models (\exists y \forall x \varphi)[e] &\iff \text{există } m \in \mathbb{N} \text{ a.î. pentru orice } n \in \mathbb{N} \\ &\quad \text{avem } \mathcal{N} \models \varphi[e_{x \mapsto n, y \mapsto m}] \\ &\iff \text{există } m \in \mathbb{N} \text{ a.î. pentru orice } n \in \mathbb{N} \\ &\quad \text{avem } n < m, \end{aligned}$$

ceea ce este fals. Așadar,

$$\mathcal{N} \not\models (\exists y \forall x \varphi)[e].$$



Fie Γ o mulțime de formule.

Definiția 2.30

Fie $\mathcal{A}$ o $\mathcal{L}$ -structură și $e : V \rightarrow A$ o interpretare a lui $\mathcal{L}$ în $\mathcal{A}$.
Spunem că $(\mathcal{A}, e)$ este **model** al lui Γ dacă $(\mathcal{A}, e)$ este model al fiecărei formule din Γ , adică

$$\mathcal{A} \models \gamma[e] \text{ pentru orice } \gamma \in \Gamma.$$

Definiția 2.31

- ▶ Γ este **satisfiabilă** dacă are un model.
- ▶ Γ este **nesatisfiabilă** sau **contradictorie** dacă Γ nu este satisfiabilă.



Definiția 2.32

Fie φ o formulă. Spunem φ este **consecință semantică** a lui Γ dacă pentru orice $\mathcal{L}$ -structură $\mathcal{A}$ și orice evaluare $e : V \rightarrow A$,

$$(\mathcal{A}, e) \text{ model al lui } \Gamma \implies (\mathcal{A}, e) \text{ model al lui } \varphi.$$

Notăție: $\Gamma \models \varphi$

Definiția 2.33

Fie Δ o mulțime de formule. Spunem că Δ este **consecință semantică** a lui Γ dacă

$$\Gamma \models \varphi \text{ pentru orice } \varphi \in \Delta.$$

Notăție: $\Gamma \models \Delta$



Lema 2.34

- (i) $\models \psi \iff \emptyset \models \psi$;
- (ii) Dacă $\Gamma \subseteq \Delta$ și $\Gamma \models \psi$, atunci $\Delta \models \psi$.
- (iii) Dacă $\Gamma \models \Delta$ și $\Delta \models \psi$, atunci $\Gamma \models \psi$.

Dem.: Exercițiu ușor.

Noțiunile de **tautologie** și **consecință semantică** din logica propozițională se pot aplica și unui limbaj de ordinul întâi. Intuitiv, o tautologie este o formulă "adevărată" numai pe baza interpretărilor conectorilor $\neg, \rightarrow$.

Definiția 2.35

O **$\mathcal{L}$ -evaluare de adevăr** este o funcție $F : \text{Form}_{\mathcal{L}} \rightarrow \{0, 1\}$ cu următoarele proprietăți: pentru orice formule φ, ψ ,

- ▶ $F(\neg\varphi) = \neg F(\varphi)$;
- ▶ $F(\varphi \rightarrow \psi) = F(\varphi) \rightarrow F(\psi)$.

Propoziția 2.36

Pentru orice $\mathcal{L}$ -structură $\mathcal{A}$ și orice evaluare $e : V \rightarrow A$, funcția

$$V_{e,\mathcal{A}} : \text{Form}_{\mathcal{L}} \rightarrow \{0, 1\}, \quad V_{e,\mathcal{A}}(\varphi) = \varphi^{\mathcal{A}}(e)$$

este o $\mathcal{L}$ -evaluare de adevăr.



Definiția 2.37

O formulă φ este **tautologie** dacă $F(\varphi) = 1$ pentru orice $\mathcal{L}$ -evaluare de adevăr F .

Notăție: $\models_{\text{taut}} \varphi$

Exemple de tautologii: $\varphi \rightarrow (\psi \rightarrow \varphi)$, $(\varphi \rightarrow \psi) \leftrightarrow (\neg\psi \rightarrow \neg\varphi)$

Propoziția 2.38

Orice tautologie este validă.

Dem.: Fie $\mathcal{A}$ o $\mathcal{L}$ -structură și $e : V \rightarrow A$ o evaluare. Deoarece φ este tautologie și $V_{e,\mathcal{A}}$ este $\mathcal{L}$ -evaluare de adevăr, rezultă că $\varphi^{\mathcal{A}}(e) = V_{e,\mathcal{A}}(\varphi) = 1$, adică $\mathcal{A} \models \varphi[e]$. □

Exemplu

$x = x$ este validă, dar nu este tautologie.



Tautologii

Fie φ, ψ formule.

Definiția 2.39

ψ este **consecință tautologică** a lui φ dacă pentru orice $\mathcal{L}$ -evaluare de adevăr F ,

$$F(\varphi) = 1 \implies F(\psi) = 1.$$

Notăție: $\varphi \models_{\text{taut}} \psi$

Definiția 2.40

Două formule φ și ψ sunt **tautologic echivalente** dacă $F(\varphi) = F(\psi)$ pentru orice $\mathcal{L}$ -evaluare de adevăr F .

Notăție: $\varphi \equiv_{\text{taut}} \psi$

Observație

- (i) $\varphi \models_{\text{taut}} \psi$ ddacă $\models_{\text{taut}} \varphi \rightarrow \psi$.
- (ii) $\varphi \equiv_{\text{taut}} \psi$ ddacă $(\psi \models_{\text{taut}} \varphi \text{ și } \varphi \models_{\text{taut}} \psi)$ ddacă $\models_{\text{taut}} \psi \leftrightarrow \varphi$.



Definiția 2.41

O formulă φ este **consecință tautologică** a unei mulțimi de formule Γ dacă pentru orice $\mathcal{L}$ -evaluare de adevăr F ,

$$F(\gamma) = 1 \text{ pentru orice } \gamma \in \Gamma \implies F(\varphi) = 1.$$

Notăție: $\Gamma \models_{\text{taut}} \varphi$

Propoziția 2.42

Dacă φ este consecință tautologică a lui Γ , atunci $\Gamma \models \varphi$.

Dem.: Fie $(\mathcal{A}, e)$ un model al lui Γ . Atunci $\gamma^{\mathcal{A}}(e) = 1$, deci $V_{e, \mathcal{A}}(\gamma) = 1$ pentru orice $\gamma \in \Gamma$. Deoarece φ este consecință tautologică a lui Γ , rezultă că $V_{e, \mathcal{A}}(\varphi) = 1$, deci $\varphi^{\mathcal{A}}(e) = 1$, adică $\mathcal{A} \models \varphi[e]$. □

Definiția 2.43

Fie $\varphi = \varphi_0\varphi_1 \dots \varphi_{n-1}$ o formulă a lui $\mathcal{L}$ și x o variabilă.

- ▶ Spunem că variabila x **apare legată pe poziția k** în φ dacă $x = \varphi_k$ și există $0 \leq i \leq k \leq j \leq n-1$ a.î. $\varphi_i \dots \varphi_j$ este de forma $\forall x \psi$ cu ψ formulă.
- ▶ Spunem că x **apare liberă pe poziția k** în φ dacă $x = \varphi_k$, dar x nu apare legată pe poziția k în φ .
- ▶ x este **variabilă legată** (bounded variable) a lui φ dacă există un k a.î. x apare legată pe poziția k în φ .
- ▶ x este **variabilă liberă** (free variable) a lui φ dacă există un k a.î. x apare liberă pe poziția k în φ .

Exemplu

Fie $\varphi = \forall x(x = y) \rightarrow x = z$. Variabile libere: x, y, z . Variabile legate: x .



Notăție: $FV(\varphi) :=$ mulțimea variabilelor libere ale lui φ .

Definiție alternativă

Mulțimea $FV(\varphi)$ a variabilelor libere ale unei formule φ poate fi definită și prin inducție pe formule:

$$FV(\varphi) = Var(\varphi), \quad \text{dacă } \varphi \text{ este formulă atomică;}$$

$$FV(\neg\varphi) = FV(\varphi);$$

$$FV(\varphi \rightarrow \psi) = FV(\varphi) \cup FV(\psi);$$

$$FV(\forall x\varphi) = FV(\varphi) \setminus \{x\}.$$

Notăție: $\varphi(x_1, \dots, x_n)$ dacă $FV(\varphi) \subseteq \{x_1, \dots, x_n\}$.



Propoziția 2.44

Pentru orice $\mathcal{L}$ -structură $\mathcal{A}$ și orice interpretări $e_1, e_2 : V \rightarrow A$,
pentru orice termen t ,

dacă $e_1(v) = e_2(v)$ pentru orice variabilă $v \in \text{Var}(t)$, atunci
$$t^{\mathcal{A}}(e_1) = t^{\mathcal{A}}(e_2).$$

Dem.: Exercițiu.



Propoziția 2.45

Pentru orice $\mathcal{L}$ -structură $\mathcal{A}$, orice interpretări $e_1, e_2 : V \rightarrow A$, pentru orice formulă φ ,

dacă $e_1(v) = e_2(v)$ pentru orice variabilă $v \in FV(\varphi)$, atunci

$$\mathcal{A} \models \varphi[e_1] \iff \mathcal{A} \models \varphi[e_2].$$

Dem.: Aplicăm inducția pe formule. Avem următoarele cazuri:

- $\varphi = t_1 = t_2$.

Atunci $Var(t_1) \subseteq FV(\varphi)$, $Var(t_2) \subseteq FV(\varphi)$, deci putem aplica Propoziția 2.44 pentru a obține că

$$t_1^{\mathcal{A}}(e_1) = t_1^{\mathcal{A}}(e_2) \quad \text{și} \quad t_2^{\mathcal{A}}(e_1) = t_2^{\mathcal{A}}(e_2).$$

Rezultă că

$$\begin{aligned} \mathcal{A} \models \varphi[e_1] &\iff t_1^{\mathcal{A}}(e_1) = t_2^{\mathcal{A}}(e_1) \iff t_1^{\mathcal{A}}(e_2) = t_2^{\mathcal{A}}(e_2) \\ &\iff \mathcal{A} \models \varphi[e_2]. \end{aligned}$$

- $\varphi = Rt_1 \dots t_m$.

Atunci $Var(t_i) \subseteq FV(\varphi)$ pentru orice $i = 1, \dots, m$ și aplicăm din nou Propoziția 2.44 pentru a obține că

$$t_i^A(e_1) = t_i^A(e_2) \text{ pentru orice } i = 1, \dots, m.$$

Rezultă că

$$\begin{aligned} \mathcal{A} \models \varphi[e_1] &\iff (t_1^A(e_1), \dots, t_m^A(e_1)) \in R^A \\ &\iff (t_1^A(e_2), \dots, t_m^A(e_2)) \in R^A \iff \mathcal{A} \models \varphi[e_2]. \end{aligned}$$

- $\varphi = \neg\psi$.

Deoarece $FV(\psi) = FV(\varphi)$, putem aplica ipoteza de inducție pentru a obține că

$$\mathcal{A} \models \psi[e_1] \iff \mathcal{A} \models \psi[e_2].$$

Rezultă că

$$\mathcal{A} \models \varphi[e_1] \iff \mathcal{A} \not\models \psi[e_1] \iff \mathcal{A} \not\models \psi[e_2] \iff \mathcal{A} \models \varphi[e_2].$$



- $\varphi = \psi \rightarrow \chi$.

Deoarece $FV(\psi), FV(\chi) \subseteq FV(\varphi)$, putem aplica ipoteza de inducție pentru a obține că

$$\mathcal{A} \models \psi[e_1] \iff \mathcal{A} \models \psi[e_2] \text{ și } \mathcal{A} \models \chi[e_1] \iff \mathcal{A} \models \chi[e_2].$$

Rezultă că

$$\begin{aligned} \mathcal{A} \models \varphi[e_1] &\iff \mathcal{A} \not\models \psi[e_1] \text{ sau } \mathcal{A} \models \chi[e_1] \\ &\iff \mathcal{A} \not\models \psi[e_2] \text{ sau } \mathcal{A} \models \chi[e_2] \\ &\iff \mathcal{A} \models \varphi[e_2]. \end{aligned}$$



Interpretarea formulelor

- $\varphi = \forall x\psi$ și

$$e_1(v) = e_2(v) \text{ pentru orice } v \in FV(\varphi) = FV(\psi) \setminus \{x\}.$$

Rezultă că pentru orice $a \in A$,

$$e_{1_{x \mapsto a}}(v) = e_{2_{x \mapsto a}}(v) \text{ pentru orice } v \in FV(\psi).$$

Prin urmare, putem aplica ipoteza de inducție pentru interpretările $e_{1_{x \mapsto a}}, e_{2_{x \mapsto a}}$ pentru a obține că

$$\text{pentru orice } a \in A, \mathcal{A} \models \psi[e_{1_{x \mapsto a}}] \iff \mathcal{A} \models \psi[e_{2_{x \mapsto a}}].$$

Rezultă că

$$\begin{aligned} \mathcal{A} \models \varphi[e_1] &\iff \text{pentru orice } a \in A, \mathcal{A} \models \psi[e_{1_{x \mapsto a}}] \\ &\iff \text{pentru orice } a \in A, \mathcal{A} \models \psi[e_{2_{x \mapsto a}}] \\ &\iff \mathcal{A} \models \varphi[e_2]. \end{aligned}$$





Propoziția 2.46

Pentru orice formulă φ și orice variabilă $x \notin FV(\varphi)$,

$$\varphi \models \exists x \varphi \quad (65)$$

$$\varphi \models \forall x \varphi \quad (66)$$

Dem.: Fie $\mathcal{A}$ o $\mathcal{L}$ -structură și $e : V \rightarrow A$.

(65):

$\mathcal{A} \models (\exists x \varphi)[e] \iff \text{există } a \in A \text{ a.î. } \mathcal{A} \models \varphi[e_{x \mapsto a}] \iff \text{există } a \in A \text{ a.î. } \mathcal{A} \models \varphi[e] \text{ (aplicând Propoziția 2.45)} \iff \mathcal{A} \models \varphi[e].$

(66): Exercițiu. □



Propoziția 2.47

Pentru orice formule φ, ψ și orice variabilă $x \notin FV(\varphi)$,

$$\forall x(\varphi \wedge \psi) \models \varphi \wedge \forall x\psi, \quad \forall x(\psi \wedge \varphi) \models \forall x\psi \wedge \varphi \quad (67)$$

$$\forall x(\varphi \vee \psi) \models \varphi \vee \forall x\psi, \quad \forall x(\psi \vee \varphi) \models \forall x\psi \vee \varphi \quad (68)$$

$$\exists x(\varphi \wedge \psi) \models \varphi \wedge \exists x\psi, \quad \exists x(\psi \wedge \varphi) \models \exists x\psi \wedge \varphi \quad (69)$$

$$\exists x(\varphi \vee \psi) \models \varphi \vee \exists x\psi, \quad \exists x(\psi \vee \varphi) \models \exists x\psi \vee \varphi \quad (70)$$

$$\forall x(\varphi \rightarrow \psi) \models \varphi \rightarrow \forall x\psi \quad (71)$$

$$\exists x(\varphi \rightarrow \psi) \models \varphi \rightarrow \exists x\psi \quad (72)$$

$$\forall x(\psi \rightarrow \varphi) \models \exists x\psi \rightarrow \varphi \quad (73)$$

$$\exists x(\psi \rightarrow \varphi) \models \forall x\psi \rightarrow \varphi \quad (74)$$

Dem.: Exercițiu.



Demonstrăm (67) ca exemplu. Fie $\mathcal{A}$ o $\mathcal{L}$ -structură și $e : V \rightarrow A$.

$$\begin{aligned} \mathcal{A} \models (\forall x(\varphi \wedge \psi))[e] &\iff \text{pentru orice } a \in A, \mathcal{A} \models (\varphi \wedge \psi)[e_{x \mapsto a}] \\ &\iff \text{pentru orice } a \in A, \mathcal{A} \models \varphi[e_{x \mapsto a}] \text{ și } \mathcal{A} \models \psi[e_{x \mapsto a}] \iff \text{pentru} \\ &\text{orice } a \in A, \mathcal{A} \models \varphi[e] \text{ și } \mathcal{A} \models \psi[e_{x \mapsto a}] \text{ (aplicând Propoziția 2.45)} \\ &\iff \mathcal{A} \models \varphi[e] \text{ și pentru orice } a \in A, \mathcal{A} \models \psi[e_{x \mapsto a}] \iff \mathcal{A} \models \varphi[e] \\ &\text{și } \mathcal{A} \models (\forall x\psi)[e] \iff \mathcal{A} \models (\varphi \wedge \forall x\psi)[e]. \quad \square \end{aligned}$$

Definiția 2.48

O formulă φ se numește **enunț** (sentence) dacă $FV(\varphi) = \emptyset$, adică φ nu are variabile libere.

Notăție: $Sent_{\mathcal{L}} :=$ mulțimea enunțurilor lui $\mathcal{L}$.

Propoziția 2.49

Fie φ un enunț. Pentru orice interpretări $e_1, e_2 : V \rightarrow A$,

$$\mathcal{A} \models \varphi[e_1] \iff \mathcal{A} \models \varphi[e_2]$$

Dem.: Este o consecință imediată a Propoziției 2.45 și a faptului că $FV(\varphi) = \emptyset$. □

Definiția 2.50

O $\mathcal{L}$ -structură $\mathcal{A}$ este un **model** al lui φ dacă $\mathcal{A} \models \varphi[e]$ pentru o (orice) evaluare $e : V \rightarrow A$. **Notăție:** $\mathcal{A} \models \varphi$



Propoziția 2.51

Pentru orice enunțuri φ, ψ ,

- (i) $\mathcal{A} \models \neg\varphi \iff \mathcal{A} \not\models \varphi$.
- (ii) $\mathcal{A} \models \varphi \rightarrow \psi \iff \mathcal{A} \models \varphi$ implică $\mathcal{A} \models \psi$
 $\iff \mathcal{A} \not\models \varphi$ sau $\mathcal{A} \models \psi$.
- (iii) $\mathcal{A} \models \varphi \wedge \psi \iff \mathcal{A} \models \varphi$ și $\mathcal{A} \models \psi$.
- (iv) $\mathcal{A} \models \varphi \vee \psi \iff \mathcal{A} \models \varphi$ sau $\mathcal{A} \models \psi$.
- (v) $\mathcal{A} \models \varphi \leftrightarrow \psi \iff \mathcal{A} \models \varphi$ dacă și numai dacă $\mathcal{A} \models \psi$.

Propoziția 2.52

Pentru orice $n \geq 2$ și orice enunțuri $\varphi_1, \dots, \varphi_n$,

- ▶ $\mathcal{A} \models \varphi_1 \wedge \dots \wedge \varphi_n$ dacă și numai dacă $\mathcal{A} \models \varphi_i$ pentru orice $i \in \{1, \dots, n\}$.
- ▶ $\mathcal{A} \models \varphi_1 \vee \dots \vee \varphi_n$ dacă există $i \in \{1, \dots, n\}$ a.î $\mathcal{A} \models \varphi_i$.



Fie φ un enunț al lui $\mathcal{L}$ și Γ o mulțime de enunțuri.

Definiția 2.53

O $\mathcal{L}$ -structură $\mathcal{A}$ este un *model* al lui Γ dacă $\mathcal{A} \models \gamma$ pentru orice $\gamma \in \Gamma$.

Notăție: $\mathcal{A} \models \Gamma$

Γ este *satisfiabilă* ddacă Γ are un model ddacă există o $\mathcal{L}$ -structură $\mathcal{A}$ a.î. $\mathcal{A} \models \Gamma$

$\Gamma \models \varphi$ ddacă pentru orice $\mathcal{L}$ -structură $\mathcal{A}$,

$$\mathcal{A} \models \Gamma \implies \mathcal{A} \models \varphi.$$



Fie x o variabilă a lui $\mathcal{L}$ și u termen al lui $\mathcal{L}$.

Definiția 2.54

Pentru orice termen t al lui $\mathcal{L}$, definim

$t_x(u) :=$ expresia obținută din t prin înlocuirea tuturor aparițiilor lui x cu u .

Propoziția 2.55

Pentru orice termen t al lui $\mathcal{L}$, $t_x(u)$ este termen al lui $\mathcal{L}$.



- ▶ Vrem să definim analog $\varphi_x(u)$ ca fiind expresia obținută din φ prin înlocuirea tuturor aparițiilor **libere** ale lui x cu u .
- ▶ De asemenea, vrem ca următoarele proprietăți naturale ale substituției să fie adevărate:

$$\models \forall x \varphi \rightarrow \varphi_x(u) \quad \text{și} \quad \models \varphi_x(u) \rightarrow \exists x \varphi.$$

Apar însă probleme.

Fie $\varphi := \exists y \neg(x = y)$ și $u := y$. Atunci $\varphi_x(u) = \exists y \neg(y = y)$.

- ▶ Pentru orice $\mathcal{L}$ -structură $\mathcal{A}$ cu $|A| \geq 2$, avem $\mathcal{A} \models \forall x \varphi$.
- ▶ $\varphi_x(u)$ nu este satisfiabilă.



Fie x o variabilă, u un termen și φ o formulă.

Definiția 2.56

Spunem că x este **liberă pentru u** în φ sau că u este **substituibil pentru x** în φ dacă pentru orice variabilă y care apare în u , nici o subformulă a lui φ de forma $\forall y\psi$ nu conține apariții libere ale lui x .

Observație

x este liberă pentru u în φ în oricare din următoarele situații:

- ▶ u nu conține variabile;
- ▶ φ nu conține variabile care apar în u ;
- ▶ nici o variabilă din u nu apare legată în φ ;
- ▶ x nu apare în φ ;
- ▶ φ nu conține apariții libere ale lui x .



Fie x o variabilă, u termen și φ o formulă a.î. x este liberă pentru u în φ .

Definiția 2.57

$\varphi_x(u) :=$ expresia obținută din φ prin înlocuirea tuturor aparițiilor **libere** ale lui x cu u .

Spunem că $\varphi_x(u)$ este o **substituție liberă**.

Propoziția 2.58

$\varphi_x(u)$ este formulă a lui $\mathcal{L}$.

Noțiunea de substituție liberă evită problemele menționate anterior și se comportă cum am aștepta.

Propoziția 2.59

Pentru orice termeni u_1 și u_2 și orice variabilă x ,

(i) pentru orice termen t ,

$$\models u_1 = u_2 \rightarrow t_x(u_1) = t_x(u_2).$$

(ii) pentru orice formulă φ a.î. x este liberă pentru u_1 și u_2 în φ ,

$$\models u_1 = u_2 \rightarrow (\varphi_x(u_1) \leftrightarrow \varphi_x(u_2)).$$

Propoziția 2.60

Fie φ o formulă și x o variabilă.

(i) Pentru orice termen u substituibil pentru x în φ ,

$$\models \forall x\varphi \rightarrow \varphi_x(u), \quad \models \varphi_x(u) \rightarrow \exists x\varphi.$$

(ii) $\models \forall x\varphi \rightarrow \varphi$, $\models \varphi \rightarrow \exists x\varphi$.

(iii) Pentru orice simbol de constantă c ,

$$\models \forall x\varphi \rightarrow \varphi_x(c), \quad \models \varphi_x(c) \rightarrow \exists x\varphi.$$



În general, dacă x și y sunt variabile, φ și $\varphi_x(y)$ nu sunt logic echivalente: fie $\mathcal{L}_{ar}$, $\mathcal{N}$ și $e : V \rightarrow \mathbb{N}$ a.î.

$e(x) = 3, e(y) = 5, e(z) = 4$. Atunci

- ▶ $\mathcal{N} \models (x < z)[e]$, deoarece $e(x) = 3 < 4 = e(z)$.
- ▶ Fie $\varphi := (x < z)_x(y) = y < z$. Avem că $\mathcal{N} \not\models \varphi[e]$, deoarece $e(y) = 5 > 4 = e(z)$.



Totuși, variabilele legate pot fi substituite, cu condiția să se evite conflicte.

Propoziția 2.61

Pentru orice formulă φ , variabile distincte x și y astfel încât y nu apare în φ ,

$$\exists x\varphi \models \exists y\varphi_x(y) \quad \text{și} \quad \forall x\varphi \models \forall y\varphi_x(y).$$

Folosim Propoziția 2.61 astfel: dacă $\varphi_x(u)$ nu este substituție liberă (adică x nu este liberă pentru u în φ), atunci înlocuim φ cu o formulă φ' logic echivalentă a.î. $\varphi'_x(u)$ este substituție liberă.



Definiția 2.62

Pentru orice formulă φ și orice variabile $y_1, \dots, y_k$, **varianta** $y_1, \dots, y_k$ -**liberă** φ' a lui φ este definită recursiv astfel:

- ▶ dacă φ este formulă atomică, atunci φ' este φ ;
- ▶ dacă $\varphi = \neg\psi$, atunci φ' este $\neg\psi'$;
- ▶ dacă $\varphi = \psi \rightarrow \chi$, atunci φ' este $\psi' \rightarrow \chi'$;
- ▶ dacă $\varphi = \forall z\psi$, atunci

$$\varphi' \text{ este } \begin{cases} \forall w\psi'_z(w) & \text{dacă } z \in \{y_1, \dots, y_k\} \\ \forall z\psi' & \text{altfel,} \end{cases}$$

unde w este prima variabilă din șirul $v_0, v_1, \dots$, care nu apare în ψ' și nu este printre $y_1, \dots, y_k$.



Definiția 2.63

φ' este **variantă** a lui φ dacă este varianta $y_1, \dots, y_k$ -liberă a lui φ pentru anumite variabile $y_1, \dots, y_k$.

Propoziția 2.64

- (i) Pentru orice formulă φ , dacă φ' este o variantă a lui φ , atunci $\varphi \models \varphi'$.
- (ii) Pentru orice formulă φ și orice termen t , dacă variabilele lui t se află printre $y_1, \dots, y_k$ și φ' este varianta $y_1, \dots, y_k$ -liberă a lui φ , atunci $\varphi'_x(t)$ este o substituție liberă.



Definiția 2.65

O formulă care nu conține cuantificatori se numește **liberă de cuantificatori** ("quantifier-free").

Definiția 2.66

O formulă φ este în **formă normală prenex** dacă

$$\varphi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \psi,$$

unde $n \in \mathbb{N}$, $Q_1, \dots, Q_n \in \{\forall, \exists\}$, $x_1, \dots, x_n$ sunt variabile și ψ este formulă liberă de cuantificatori. Formula ψ se numește **matricea** lui φ și $Q_1 x_1 Q_2 x_2 \dots Q_n x_n$ este **prefixul** lui φ .



Exemple de formule în formă normală prenex:

- ▶ Formulele libere de cuantificatori
- ▶ Formulele **universale**: $\varphi = \forall x_1 \forall x_2 \dots \forall x_n \psi$, unde $n \in \mathbb{N}$ și ψ este liberă de cuantificatori
- ▶ Formulele **existențiale**: $\varphi = \exists x_1 \exists x_2 \dots \exists x_n \psi$, unde $n \in \mathbb{N}$ și ψ este liberă de cuantificatori

Teorema 2.67 (Teorema de formă normală prenex)

Pentru orice formulă φ există o formulă φ^* în formă normală prenex a.î. $\varphi \models \varphi^*$ și $FV(\varphi) = FV(\varphi^*)$.

Dem.: Exercițiu suplimentar.



Fie $\mathcal{L}$ un limbaj de ordinul întâi care conține

- ▶ două simboluri de relații unare R, S și două simboluri de relații binare P, Q ;
- ▶ un simbol de funcție unară f și un simbol de funcție binară g ;
- ▶ două simboluri de constante c, d .

Fie x, y, z variabile distincte.



Exemplul 2.68

Să se găsească o formă normală prenex pentru

$$\varphi := \exists y(g(y, z) = c) \wedge \neg \exists x(f(x) = d)$$

Dem.: Avem

$$\varphi \quad \equiv \quad \exists y(g(y, z) = c) \wedge \forall x \neg(f(x) = d) \quad \text{aplicăm (51)}$$

$$\equiv \exists y(g(y, z) = c \wedge \forall x \neg(f(x) = d)) \quad \text{aplicăm (69)}$$

$$\equiv \exists y \forall x (g(y, z) = c \wedge \neg(f(x) = d)) \quad \text{aplicăm (67)}$$

Prin urmare,

$$\varphi^* = \exists y \forall x (g(y, z) = c \wedge \neg(f(x) = d))$$

este o formă normală prenex pentru φ .



Exemplul 2.69

Să se găsească o formă normală prenex pentru

$$\varphi := \neg \forall y (S(y) \rightarrow \exists z R(z)) \wedge \forall x (\forall y P(x, y) \rightarrow f(x) = d).$$

Dem.: Avem că

$$\begin{aligned} \varphi &\equiv \exists y \neg (S(y) \rightarrow \exists z R(z)) \wedge \forall x (\forall y P(x, y) \rightarrow f(x) = d) \text{ aplicăm (52)} \\ &\equiv \exists y \neg \exists z (S(y) \rightarrow R(z)) \wedge \forall x (\forall y P(x, y) \rightarrow f(x) = d) \text{ aplicăm (72)} \\ &\equiv \exists y \neg \exists z (S(y) \rightarrow R(z)) \wedge \forall x \exists y (P(x, y) \rightarrow f(x) = d) \text{ aplicăm (74)} \\ &\equiv \exists y \forall z \neg (S(y) \rightarrow R(z)) \wedge \forall x \exists y (P(x, y) \rightarrow f(x) = d) \text{ aplicăm (51)} \\ &\equiv \exists y \left(\forall z \neg (S(y) \rightarrow R(z)) \wedge \forall x \exists y (P(x, y) \rightarrow f(x) = d) \right) \text{ aplicăm (69)} \\ &\equiv \exists y \forall z \left(\neg (S(y) \rightarrow R(z)) \wedge \forall x \exists y (P(x, y) \rightarrow f(x) = d) \right) \text{ aplicăm (67)} \end{aligned}$$

Exemplul 2.70

Să se găsească o formă normală prenex pentru

$$\varphi := \neg \forall y (S(y) \rightarrow \exists z R(z)) \wedge \forall x (\forall y P(x, y) \rightarrow f(x) = d).$$

Dem.: (continuare)

$$\begin{aligned}\varphi &\equiv \exists y \forall z \left(\neg (S(y) \rightarrow R(z)) \wedge \forall x \exists y (P(x, y) \rightarrow f(x) = d) \right) \\ &\equiv \exists y \forall z \forall x \left(\neg (S(y) \rightarrow R(z)) \wedge \exists y (P(x, y) \rightarrow f(x) = d) \right) \text{ aplicăm (67)} \\ &\equiv \exists y \forall z \forall x \left(\neg (S(y) \rightarrow R(z)) \wedge \exists v (P(x, v) \rightarrow f(x) = d) \right) \\ &\quad \text{aplicăm Propoziția 2.61, unde } v \text{ este o variabilă nouă} \\ &\equiv \exists y \forall z \forall x \exists v \left(\neg (S(y) \rightarrow R(z)) \wedge (P(x, v) \rightarrow f(x) = d) \right) \text{ aplicăm (69)}\end{aligned}$$

Deci, $\varphi^* = \exists y \forall z \forall x \exists v (\neg (S(y) \rightarrow R(z)) \wedge (P(x, v) \rightarrow f(x) = d))$ este o formă normală prenex pentru φ .



Skolemizarea este o procedură prin care se elimină cuantificatorii existențiali din enunțuri de ordinul întâi în formă normală prenex, prin introducerea de noi simboluri de funcții/constante, numite **simboluri de funcții/constante Skolem**.

Observație

Orice enunț liber de cuantificatori este universal.

Fie $\mathcal{L}$ un limbaj de ordinul întâi și φ un enunț al lui $\mathcal{L}$ care este în formă normală prenex:

$$\varphi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \theta,$$

unde $n \in \mathbb{N}$, $Q_1, \dots, Q_n \in \{\forall, \exists\}$, $x_1, \dots, x_n$ sunt variabile distincte două câte două și θ este formulă liberă de cuantificatori.

Asociem lui φ un enunț universal φ^{Sk} într-un limbaj extins $\mathcal{L}^{Sk}(\varphi)$:
Dacă φ este universal, atunci $\varphi^{Sk} = \varphi$ și $\mathcal{L}^{Sk}(\varphi) = \mathcal{L}$.

Altfel, φ are una din formele:

- ▶ $\varphi = \exists x \psi$. Introducem un nou simbol de constantă c și considerăm $\varphi^1 = \psi_x(c)$, $\mathcal{L}^1 = \mathcal{L} \cup \{c\}$.
- ▶ $\varphi = \forall x_1 \dots \forall x_k \exists x \psi$ ($k \geq 1$). Introducem un nou simbol de funcție f de aritate k și considerăm $\varphi^1 = \forall x_1 \dots \forall x_k \psi_x(fx_1 \dots x_k)$, $\mathcal{L}^1 = \mathcal{L} \cup \{f\}$.

În ambele cazuri, φ^1 are cu un cuantificator existențial mai puțin decât φ .

Dacă φ^1 este enunț universal, atunci $\varphi^{Sk} = \varphi^1$. Dacă φ^1 nu este enunț universal, atunci formăm $\varphi^2, \varphi^3, \dots$, până ajungem la un enunț universal și acesta este φ^{Sk} .

φ^{Sk} este o **formă normală Skolem** a lui φ .



Exemple

- ▶ Fie θ o formulă liberă de cuantificatori a.î. $FV(\theta) = \{x\}$ și $\varphi = \exists x \theta$. Atunci $\varphi^1 = \theta_x(c)$, unde c este un nou simbol de constantă. Deoarece φ^1 este un enunț liber de cuantificatori, rezultă că $\varphi^{Sk} = \varphi^1 = \theta_x(c)$.
- ▶ Fie R un simbol de relație de aritate 3 și $\varphi = \exists x \forall y \forall z R(x, y, z)$. Atunci
$$\varphi^1 = \forall y \forall z (R(x, y, z))_x(c) = \forall y \forall z R(c, y, z),$$
unde c este un nou simbol de constantă. Deoarece φ^1 este un enunț universal, rezultă că $\varphi^{Sk} = \varphi^1 = \forall y \forall z R(c, y, z)$.
- ▶ Fie P un simbol de relație de aritate 2 și $\varphi = \forall y \exists z P(y, z)$. Atunci $\varphi^1 = \forall y (P(y, z))_z(f(y)) = \forall y P(y, f(y))$, unde f este un simbol nou de funcție unară. Deoarece φ^1 este un enunț universal, rezultă că $\varphi^{Sk} = \varphi^1 = \forall y P(y, f(y))$.



Exemplu

Fie $\mathcal{L}$ un limbaj care conține un simbol de relație binară R și un simbol de funcție unară f . Fie

$$\varphi := \forall y \exists z \forall u \exists v (R(y, z) \wedge f(u) = v).$$

$$\begin{aligned}\varphi^1 &= \forall y \forall u \exists v (R(y, z) \wedge f(u) = v)_z(g(y)) \\ &= \forall y \forall u \exists v (R(y, g(y)) \wedge f(u) = v),\end{aligned}$$

unde g este un nou simbol de funcție unară

$$\begin{aligned}\varphi^2 &= \forall y \forall u (R(y, g(y)) \wedge f(u) = v)_v(h(y, u)) \\ &= \forall y \forall u (R(y, g(y)) \wedge f(u) = h(y, u)),\end{aligned}$$

unde h este un nou simbol de funcție binară.

Deoarece φ^2 este un enunț universal, rezultă că

$$\varphi^{Sk} = \varphi^2 = \forall y \forall u (R(y, g(y)) \wedge f(u) = h(y, u)).$$



Teorema 2.71 (Teorema de formă normală Skolem)

Fie φ un enunț în formă normală prenex și φ^{Sk} o formă normală Skolem a sa.

- (i) $\models \varphi^{Sk} \rightarrow \varphi$, deci $\varphi^{Sk} \models \varphi$ în $\mathcal{L}^{Sk}(\varphi)$.*
- (ii) φ este satisfiabilă ddacă φ^{Sk} este satisfiabilă.*

Observație

În general, φ și φ^{Sk} nu sunt logic echivalente ca enunțuri în $\mathcal{L}^{Sk}(\varphi)$.



Notăție: Pentru orice mulțime de enunțuri Γ , notăm

$$\text{Mod}(\Gamma) := \text{clasa modelelor lui } \Gamma.$$

Notăm $\text{Mod}(\varphi_1, \dots, \varphi_n)$ în loc de $\text{Mod}(\{\varphi_1, \dots, \varphi_n\})$.

Lema 2.72

Pentru orice mulțimi de enunțuri Γ, Δ și orice enunț ψ ,

- (i) $\Gamma \models \psi \iff \text{Mod}(\Gamma) \subseteq \text{Mod}(\psi).$
- (ii) $\Gamma \subseteq \Delta \implies \text{Mod}(\Delta) \subseteq \text{Mod}(\Gamma).$
- (iii) Γ este satisfiabilă $\iff \text{Mod}(\Gamma) \neq \emptyset.$

Dem.: Exercițiu ușor.



Definiția 2.73

O $\mathcal{L}$ -*teorie* este o mulțime T de enunțuri ale lui $\mathcal{L}$ care este închisă la consecința semantică, adică:

$$\text{pentru orice enunț } \varphi, \quad T \models \varphi \implies \varphi \in T.$$

Definiția 2.74

Pentru orice mulțime de enunțuri Γ , *teoria generată de* Γ este mulțimea

$$\begin{aligned} Th(\Gamma) &:= \{\varphi \mid \varphi \text{ este enunț și } \Gamma \models \varphi\} \\ &= \{\varphi \mid \varphi \text{ este enunț și } Mod(\Gamma) \subseteq Mod(\varphi)\}. \end{aligned}$$



Propoziția 2.75

Fie Γ o mulțime de enunțuri.

- (i) $\text{Mod}(\Gamma) = \text{Mod}(\text{Th}(\Gamma))$.
- (ii) $\text{Th}(\Gamma)$ este cea mai mică teorie T a.î. $\Gamma \subseteq T$.

Dem.: Exercițiu.

- ▶ O teorie prezentată ca $\text{Th}(\Gamma)$ se numește **teorie axiomatică** sau teorie prezentată **axiomatic**. Γ se numește mulțime de **axiome** pentru $\text{Th}(\Gamma)$.
- ▶ Orice teorie poate fi prezentată axiomatice, dar suntem interesați de mulțimi de axiome care satisfac anumite condiții.



Definiția 2.76

O teorie T este **finit axiomatizabilă** dacă $T = Th(\Gamma)$ pentru o mulțime de enunțuri finită Γ .

Definiția 2.77

O clasă $\mathcal{K}$ de $\mathcal{L}$ -structuri este **axiomatizabilă** dacă $\mathcal{K} = Mod(\Gamma)$ pentru o mulțime de enunțuri Γ . Spunem și că Γ **axiomatizează** $\mathcal{K}$.

Definiția 2.78

O clasă $\mathcal{K}$ de $\mathcal{L}$ -structuri este **finit axiomatizabilă** dacă $\mathcal{K} = Mod(\Gamma)$ pentru o mulțime **finită** de enunțuri Γ .



Exemple - Teoria egalității

Pentru orice $n \geq 2$, notăm următorul enunț cu $\exists^{\geq n}$:

$$\exists x_1 \dots \exists x_n (\neg(x_1 = x_2) \wedge \neg(x_1 = x_3) \wedge \dots \wedge \neg(x_{n-1} = x_n)),$$

pe care îl scriem mai compact astfel:

$$\exists^{\geq n} = \exists x_1 \dots \exists x_n \left(\bigwedge_{1 \leq i < j \leq n} \neg(x_i = x_j) \right).$$

Propoziția 2.79

Pentru orice $\mathcal{L}$ -structură $\mathcal{A}$ și orice $n \geq 2$,

$$\mathcal{A} \models \exists^{\geq n} \iff \mathcal{A} \text{ are cel puțin } n \text{ elemente.}$$

Dem.: Exercițiu ușor.

Pentru uniformitate, notăm $\exists^{\geq 1} := \exists x(x = x)$.

Notății

Fie $n \geq 1$.

$$\blacktriangleright \exists^{\leq n} := \neg \exists^{\geq n+1}$$

$$\blacktriangleright \exists^{=n} := \exists^{\leq n} \wedge \exists^{\geq n}$$

Propoziția 2.80

Pentru orice $\mathcal{L}$ -structură $\mathcal{A}$ și orice $n \geq 1$,

$$\mathcal{A} \models \exists^{\leq n} \iff \mathcal{A} \text{ are cel mult } n \text{ elemente}$$

$$\mathcal{A} \models \exists^{=n} \iff \mathcal{A} \text{ are exact } n \text{ elemente.}$$

Dem.: Exercițiu ușor.

Propoziția 2.81

Fie $T := Th(\{\exists^{\geq n} \mid n \geq 1\})$. Atunci pentru orice $\mathcal{L}$ -structură $\mathcal{A}$,

$$\mathcal{A} \models T \iff \mathcal{A} \models \{\exists^{\geq n} \mid n \geq 1\} \iff \mathcal{A} \text{ este mulțime infinită.}$$

Dem.: Exercițiu ușor.



Exemple - Teoria grafurilor

Un **graf** este o pereche $G = (V, E)$ de mulțimi a.î. E este o mulțime de submulțimi cu 2 elemente ale lui V . Elementele lui V se numesc **vârfuri**, iar elementele lui E se numesc **muchii**.

- ▶ $\mathcal{L}_{Graf} = (\dot{E}, \emptyset, \emptyset) = (\dot{E})$
- ▶ $\mathcal{L}_{Graf}$ -structurile sunt $\mathcal{A} = (A, E)$, unde E este relație binară.

Fie $\Gamma := \{(IREFL), (SIM)\}$, unde

$$(IREFL) := \forall x \neg \dot{E}(x, x)$$

$$(SIM) := \forall x \forall y (\dot{E}(x, y) \rightarrow \dot{E}(y, x)).$$

Definiție

Teoria grafurilor este $T := Th(\Gamma)$.

- ▶ T este finit axiomatizabilă.
- ▶ modelele lui T sunt grafurile.
- ▶ Γ axiomatizează clasa grafurilor. Prin urmare, clasa grafurilor este finit axiomatizabilă.



Exemple - Teoria ordinii parțiale

- ▶ $\mathcal{L}_{\dot{\leq}} = (\dot{\leq}, \emptyset, \emptyset) = (\dot{\leq})$
- ▶ $\mathcal{L}_{\dot{\leq}}$ -structurile sunt $\mathcal{A} = (A, \leq)$, unde $\leq$ este relație binară.

Fie $\Gamma := \{(REFL), (ANTISIM), (TRANZ)\}$, unde

$$(REFL) \quad := \quad \forall x (x \dot{\leq} x)$$

$$(ANTISIM) \quad := \quad \forall x \forall y (x \dot{\leq} y \wedge y \dot{\leq} x \rightarrow x = y)$$

$$(TRANZ) \quad := \quad \forall x \forall y \forall z (x \dot{\leq} y \wedge y \dot{\leq} z \rightarrow x \dot{\leq} z)$$

Definiție

Teoria ordinii parțiale este $T := Th(\Gamma)$.

- ▶ T este finit axiomatizabilă.
- ▶ modelele lui T sunt mulțimile parțial ordonate.
- ▶ Γ axiomatizează clasa mulțimilor parțial ordonate. Prin urmare, clasa mulțimilor parțial ordonate este finit axiomatizabilă.



Exemple - Teoria ordinii totale

Fie $\Gamma := \{(ANTISIM), (TRANZ), (TOTAL)\}$, unde

$$(TOTAL) := \forall x \forall y (x \leq y \vee y \leq x)$$

Definiție

Teoria ordinii totale este $T := Th(\Gamma)$.

- ▶ T este finit axiomatizabilă.
- ▶ modelele lui T sunt mulțimile total ordonate.
- ▶ Γ axiomatizează clasa mulțimilor total ordonate. Prin urmare, clasa mulțimilor total ordonate este finit axiomatizabilă.



Exemple - Teoria ordinii stricte

- ▶ $\mathcal{L}_{\dot{<}} = (\dot{<}, \emptyset, \emptyset) = (\dot{<})$
- ▶ $\mathcal{L}_{\dot{<}}$ -structurile sunt $\mathcal{A} = (A, <)$, unde $<$ este relație binară.

Fie $\Gamma := \{(IREFL), (TRANZ)\}$, unde

$$(IREFL) := \forall x \neg (x \dot{<} x)$$

$$(TRANZ) := \forall x \forall y \forall z (x \dot{<} y \wedge y \dot{<} z \rightarrow x \dot{<} z)$$

Definiție

Teoria ordinii stricte este $T := Th(\Gamma)$.

- ▶ T este finit axiomatizabilă.
- ▶ modelele lui T sunt mulțimile strict ordonate.
- ▶ Γ axiomatizează clasa mulțimilor strict ordonate. Prin urmare, clasa mulțimilor strict ordonate este finit axiomatizabilă.



Exemple - Teoria ordinii dense

Fie $\Gamma := \{(IREFL), (TRANZ), (TOTAL), (DENS)\}$, unde

$$(TOTAL) := \forall x \forall y (x = y \vee x < y \vee y < x)$$

$$(DENS) := \forall x \forall y (x < y \rightarrow \exists z (x < z \wedge z < y)).$$

Definiție

Teoria ordinii dense este $T := Th(\Gamma)$.

- ▶ T este finit axiomatizabilă.
- ▶ modelele lui T sunt mulțimile dens ordonate.
- ▶ Γ axiomatizează clasa mulțimilor dens ordonate. Prin urmare, clasa mulțimilor dens ordonate este finit axiomatizabilă.



Exemple - Teoria relațiilor de echivalență

- ▶ $\mathcal{L}_{\equiv} = (\equiv, \emptyset, \emptyset) = (\equiv)$
- ▶ $\mathcal{L}_{\equiv}$ -structurile sunt $\mathcal{A} = (A, \equiv)$, unde $\equiv$ este relație binară.

Fie $\Gamma := \{(REFL), (SIM), (TRANZ)\}$, unde

$$(REFL) := \forall x(x \equiv x)$$

$$(SIM) := \forall x \forall y (x \equiv y \rightarrow y \equiv x)$$

$$(TRANZ) := \forall x \forall y \forall z (x \equiv y \wedge y \equiv z \rightarrow x \equiv z)$$

Definiție

Teoria relațiilor de echivalență este $T := Th(\Gamma)$.

- ▶ T este finit axiomatizabilă.
- ▶ Fie $\mathcal{K}$ clasa structurilor $(A, \equiv)$, unde $\equiv$ este relație de echivalență pe A . Avem că $\mathcal{K} = Mod(\Gamma)$, adică Γ axiomatizează $\mathcal{K}$. Prin urmare, $\mathcal{K}$ este finit axiomatizabilă.



Exemple - Teoria relațiilor de echivalență

- Dacă adăugăm axioma:

$$\forall x \exists y (\neg(x = y) \wedge x \dot{=} y \wedge \forall z (z \dot{=} x \rightarrow (z = x \vee z = y))),$$

obținem teoria relațiilor de echivalență cu proprietatea că orice clasă de echivalență are exact două elemente.



Teorema 2.82 (Teorema de compacitate)

O mulțime de enunțuri Γ este satisfiabilă dacă și numai dacă orice submulțime finită a sa este satisfiabilă.

- unul din rezultatele centrale ale logicii de ordinul întâi



Teorema de compacitate - aplicații

Fie $\mathcal{L}$ un limbaj de ordinul întâi.

Propoziția 2.83

Clasa $\mathcal{L}$ -structurilor finite nu este axiomatizabilă, adică nu există o mulțime de enunțuri Γ astfel încât

() pentru orice $\mathcal{L}$ -structură $\mathcal{A}$, $\mathcal{A} \models \Gamma \iff \mathcal{A}$ este finită.*

Dem.: Presupunem prin reducere la absurd că există $\Gamma \subseteq \text{Sen}_{\mathcal{L}}$ a.î. (*) are loc. Fie

$$\Delta := \Gamma \cup \{\exists^{\geq n} \mid n \geq 1\}.$$

Demonstrăm că Δ este satisfiabilă folosind Teorema de compacitate. Fie Δ_0 o submulțime finită a lui Δ . Atunci

$$\Delta_0 \subseteq \Gamma \cup \{\exists^{\geq n_1}, \dots, \exists^{\geq n_k}\} \quad \text{pentru un } k \in \mathbb{N}.$$

Fie $\mathcal{A}$ o $\mathcal{L}$ -structură finită a.î. $|\mathcal{A}| \geq \max\{n_1, \dots, n_k\}$. Atunci $\mathcal{A} \models \exists^{\geq n_i}$ pentru orice $i = 1, \dots, k$ și $\mathcal{A} \models \Gamma$ deoarece $\mathcal{A}$ este finită.



Teorema de compacitate - aplicații

Prin urmare, $\mathcal{A} \models \Gamma \cup \{\exists^{\geq n_1}, \dots, \exists^{\geq n_k}\}$, de unde rezultă că $\mathcal{A} \models \Delta_0$. Așadar, Δ_0 este satisfiabilă.

Aplicând Teorema de compacitate, rezultă că

$$\Delta = \Gamma \cup \{\exists^{\geq n} \mid n \geq 1\}.$$

are un model $\mathcal{B}$.

Deoarece $\mathcal{B} \models \Gamma$, $\mathcal{B}$ este finită.

Deoarece $\mathcal{B} \models \{\exists^{\geq n} \mid n \geq 1\}$, rezultă că $\mathcal{B}$ este infinită.

Am obținut o contradicție. □

Corolar 2.84

Clasa mulțimilor nevide finite nu este axiomatizabilă în $\mathcal{L}_=$.



Propoziția 2.85

Clasa $\mathcal{L}$ -structurilor infinite este axiomatizabilă, dar nu este finit axiomatizabilă.

Dem.: Notăm cu $\mathcal{K}_{Inf}$ clasa $\mathcal{L}$ -structurilor infinite.

Conform Propoziției 2.81, pentru orice $\mathcal{L}$ -structură $\mathcal{A}$,

$$\mathcal{A} \in \mathcal{K}_{Inf} \iff \mathcal{A} \text{ este infinită} \iff \mathcal{A} \models \{\exists^{\geq n} \mid n \geq 1\}.$$

Prin urmare,

$$\mathcal{K}_{Inf} = Mod(\{\exists^{\geq n} \mid n \geq 1\})$$

deci e axiomatizabilă.



Teorema de compacitate - aplicații

Presupunem că $\mathcal{K}_{Inf}$ este finit axiomatizabilă, deci există

$$\Gamma := \{\varphi_1, \dots, \varphi_n\} \subseteq \text{Sen}_{\mathcal{L}} \text{ a.î. } \mathcal{K}_{Inf} = \text{Mod}(\Gamma).$$

Fie $\varphi := \varphi_1 \wedge \dots \wedge \varphi_n$. Atunci $\mathcal{K}_{Inf} = \text{Mod}(\varphi)$.

Rezultă că pentru orice $\mathcal{L}$ -structură $\mathcal{A}$,

$$\mathcal{A} \text{ este finită} \iff \mathcal{A} \notin \mathcal{K}_{Inf} \iff \mathcal{A} \not\models \varphi \iff \mathcal{A} \models \neg\varphi.$$

Așadar, clasa $\mathcal{L}$ -structurilor finite este axiomatizabilă, ceea ce contrazice Propoziția 2.83. □.

Corolar 2.86

Clasa mulțimilor infinite nu este finit axiomatizabilă în $\mathcal{L}_{=}$.



Teorema de compacitate - aplicații

Propoziția 2.87

Fie Γ o mulțime de enunțuri ale lui $\mathcal{L}$ cu proprietatea

(*) pentru orice $m \in \mathbb{N}$, Γ are un model finit de cardinal $\geq m$.

Atunci Γ are un model infinit.

Dem.: Fie

$$\Delta := \Gamma \cup \{\exists^{\geq n} \mid n \geq 1\}.$$

Demonstrăm că Δ este satisfiabilă folosind Teorema de compacitate. Fie Δ_0 o submulțime finită a lui Δ . Atunci

$$\Delta_0 \subseteq \Gamma \cup \{\exists^{\geq n_1}, \dots, \exists^{\geq n_k}\} \text{ pentru un } k \in \mathbb{N}.$$

Fie $m := \max\{n_1, \dots, n_k\}$. Conform (*), Γ are un model finit $\mathcal{A}$ a.î. $|A| \geq m$. Atunci $\mathcal{A} \models \exists^{\geq n_i}$ pentru orice $i = 1, \dots, k$, deci $\mathcal{A} \models \Delta_0$.

Aplicând Teorema de compacitate, rezultă că Δ are un model $\mathcal{B}$. Prin urmare, $\mathcal{B}$ este un model infinit al lui Γ . □



Propoziția 2.88

Dacă un enunț φ este adevărat în orice $\mathcal{L}$ -structură infinită, atunci există $m \in \mathbb{N}$ cu proprietatea că φ este adevărat în orice $\mathcal{L}$ -structură finită de cardinal $\geq m$.

Dem.: Presupunem că nu e adevărat. Fie $\Gamma := \{\neg\varphi\}$. Atunci pentru orice $m \in \mathbb{N}$, Γ are un model finit de cardinal $\geq m$. Aplicând Propoziția 2.87, rezultă că Γ are un model infinit $\mathcal{A}$. Prin urmare, $\mathcal{A} \not\models \varphi$, ceea ce contrazice ipoteza. $\square$

Definiția 2.89

Fie $\mathcal{L} = (\mathcal{R}_{\mathcal{L}}, \mathcal{F}_{\mathcal{L}}, \mathcal{C}_{\mathcal{L}}; \text{ari}_{\mathcal{L}})$ și $\mathcal{L}^+ = (\mathcal{R}_{\mathcal{L}^+}, \mathcal{F}_{\mathcal{L}^+}, \mathcal{C}_{\mathcal{L}^+}; \text{ari}_{\mathcal{L}^+})$ două limbaje. Spunem că $\mathcal{L}^+$ este **extensie** a lui $\mathcal{L}$ sau că $\mathcal{L}$ este **sublimbaj** al lui $\mathcal{L}^+$ dacă

$$\mathcal{R}_{\mathcal{L}} \subseteq \mathcal{R}_{\mathcal{L}^+}, \quad \mathcal{F}_{\mathcal{L}} \subseteq \mathcal{F}_{\mathcal{L}^+}, \quad \mathcal{C}_{\mathcal{L}} \subseteq \mathcal{C}_{\mathcal{L}^+}$$

și $\text{ari}_{\mathcal{L}}$ este restricția lui $\text{ari}_{\mathcal{L}^+}$ la $\mathcal{R}_{\mathcal{L}} \cup \mathcal{F}_{\mathcal{L}}$.

Notăție: $\mathcal{L} \subseteq \mathcal{L}^+$

Definiția 2.90

Spunem că $\mathcal{L}^+$ este o **extensie prin constante** a lui $\mathcal{L}$ dacă

$$\mathcal{R}_{\mathcal{L}} = \mathcal{R}_{\mathcal{L}^+}, \quad \mathcal{F}_{\mathcal{L}} = \mathcal{F}_{\mathcal{L}^+}, \quad \mathcal{C}_{\mathcal{L}} \subseteq \mathcal{C}_{\mathcal{L}^+}.$$

Exemple

- $\mathcal{L}_= \subseteq \mathcal{L}$ pentru orice limbaj $\mathcal{L}$
- $\mathcal{L}_{<} = (\dot{<}) \subseteq (\dot{<}, \dot{+}, \dot{\times}, \dot{S}) \subseteq \mathcal{L}_{ar} = (\dot{<}, \dot{+}, \dot{\times}, \dot{S}, \dot{0})$

Dacă $\mathcal{L} \subseteq \mathcal{L}^+$, atunci orice termen (formulă) din $\mathcal{L}$ este termen (formulă) în $\mathcal{L}^+$.

Definiția 2.91

Fie $\mathcal{L} \subseteq \mathcal{L}^+$, $\mathcal{A}$ o $\mathcal{L}$ -structură și $\mathcal{A}^+$ o $\mathcal{L}^+$ -structură.

Spunem că $\mathcal{A}$ este $\mathcal{L}$ -*redușă* lui $\mathcal{A}^+$ sau că $\mathcal{A}^+$ este $\mathcal{L}^+$ -*extensia* lui $\mathcal{A}$ dacă

- ▶ $|\mathcal{A}| = |\mathcal{A}^+|$;
- ▶ pentru orice $R \in \mathcal{R}_{\mathcal{L}}$, $R^{\mathcal{A}} = R^{\mathcal{A}^+}$;
- ▶ pentru orice $f \in \mathcal{F}_{\mathcal{L}}$, $f^{\mathcal{A}} = f^{\mathcal{A}^+}$;
- ▶ pentru orice $c \in \mathcal{C}_{\mathcal{L}}$, $c^{\mathcal{A}} = c^{\mathcal{A}^+}$.

Notăție: $\mathcal{A} = \mathcal{A}^+ \upharpoonright \mathcal{L}$

Exemplu

$(\mathbb{N}, <, +, \cdot, S, 0)$ are redusele $(\mathbb{N}, +, \cdot)$, $(\mathbb{N}, S, 0)$, $(\mathbb{N}, <)$.

Fie $\mathcal{L} \subseteq \mathcal{L}^+$, $\mathcal{A}$ o $\mathcal{L}$ -structură, $A = |\mathcal{A}|$ și $\mathcal{A}^+$ o $\mathcal{L}^+$ -extensie a lui $\mathcal{A}$.

Propoziția 2.92

Pentru orice evaluare $e : V \rightarrow A$, termen t al lui $\mathcal{L}$ și formulă φ a lui $\mathcal{L}$,

$$t^{\mathcal{A}}(e) = t^{\mathcal{A}^+}(e) \quad \text{și} \quad \mathcal{A} \models \varphi[e] \iff \mathcal{A}^+ \models \varphi[e].$$

Corolar 2.93

Pentru orice enunț φ al lui $\mathcal{L}$, $\mathcal{A} \models \varphi \iff \mathcal{A}^+ \models \varphi$.

Propoziția 2.94

Pentru orice mulțime de enunțuri Γ a lui $\mathcal{L}$,

- (i) $\mathcal{A} \models \Gamma \iff \mathcal{A}^+ \models \Gamma$.
- (ii) $Th_{\mathcal{L}}(\Gamma) = Sen_{\mathcal{L}} \cap Th_{\mathcal{L}^+}(\Gamma)$.

Considerăm limbajul $\mathcal{L} = (\dot{+}, \dot{\times}, \dot{S}, \dot{0})$, unde $\dot{+}$, $\dot{\times}$ sunt simboluri de operații binare, $\dot{S}$ este simbol de operație unară și $\dot{0}$ este simbol de constantă.

Pentru orice $n \in \mathbb{N}$, definim prin inducție $\mathcal{L}$ -termenul $\Delta(n)$ astfel:

$$\Delta(0) = \dot{0}, \quad \Delta(n+1) = \dot{S}\Delta(n).$$

Fie $\mathcal{L}$ -structura $\mathcal{N} = (\mathbb{N}, +, \cdot, S, 0)$. Atunci $\Delta(n)^{\mathcal{N}} = n$ pentru orice $n \in \mathbb{N}$. Prin urmare, $\mathbb{N} = \{\Delta(n)^{\mathcal{N}} \mid n \in \mathbb{N}\}$.

Definiția 2.95

O $\mathcal{L}$ -structură $\mathcal{A}$ se numește **non-standard** dacă există $a \in A$ a.î. $a \neq \Delta(n)^{\mathcal{A}}$ pentru orice $n \in \mathbb{N}$. Un astfel de element a se numește **element non-standard**.



Teoria lui $\mathcal{N}$ se definește astfel:

$$Th(\mathcal{N}) := \{\varphi \in Sen_{\mathcal{L}} \mid \mathcal{N} \models \varphi\}.$$

Se poate demonstra ușor că $Th(\mathcal{N})$ este o teorie.

Teorema 2.96

Există un model non-standard al teoriei $Th(\mathcal{N})$.

Dem.: Fie c un simbol de constantă nou, $\mathcal{L}^+ = \mathcal{L} \cup \{c\}$ și

$$\Gamma = Th(\mathcal{N}) \cup \{\neg(\Delta(n) = c) \mid n \in \mathbb{N}\} \subseteq Sen_{\mathcal{L}^+}.$$

Demonstrăm că Γ este satisfiabilă folosind Teorema de compacitate. Fie Γ_0 o submulțime finită a lui Γ ,

$$\Gamma_0 \subseteq Th(\mathcal{N}) \cup \{\neg(\Delta(n_1) = c), \dots, \neg(\Delta(n_k) = c)\}.$$

Fie $n_0 > \max\{n_1, \dots, n_k\}$. Considerăm extensia $\mathcal{N}^+$ a lui $\mathcal{N}$ la $\mathcal{L}^+$ definită astfel: $c^{\mathcal{N}^+} := n_0$. Atunci $\mathcal{N}^+ \models \Gamma_0$.

Aplicând Teorema de compacitate, rezultă că Γ are un model

$$\mathcal{A}^+ = (A, +, \times, S, 0, c^{\mathcal{A}^+}).$$

Prin urmare,

$$\mathcal{A}^+ \models Th(\mathcal{N}) \text{ și, pentru orice } n \in \mathbb{N}, c^{\mathcal{A}^+} \neq \Delta(n)^{\mathcal{A}^+}.$$

Fie $\mathcal{L}$ -structura

$$\mathcal{A} = (A, +, \times, S, 0), \quad a := c^{\mathcal{A}^+} \in A.$$

Deoarece $\mathcal{A}^+ \models Th(\mathcal{N})$ și $Th(\mathcal{N}) \subseteq Sen_{\mathcal{L}}$, avem că $\mathcal{A} \models Th(\mathcal{N})$. Cum $\Delta(n)^{\mathcal{A}^+} = \Delta(n)^{\mathcal{A}}$, avem că $a \neq \Delta(n)^{\mathcal{A}}$ pentru orice $n \in \mathbb{N}$. Rezultă că a este element non-standard al lui $\mathcal{A}$.

Deci, $\mathcal{A}$ este model non-standard al teoriei $Th(\mathcal{N})$. □



Definiția 2.97

Fie A o mulțime nevidă. O relație de **bună ordonare** pe A este o relație de ordine totală $<$ pe A cu proprietatea că orice submulțime nevidă a lui A are minim.

Spunem că $(A, <)$ este mulțime **bine ordonată**.

Exemple

$(\mathbb{N}, <)$ este bine ordonată, dar $(\mathbb{Z}, <)$ nu este bine ordonată.



Propoziția 2.98

Clasa mulțimilor bine ordonate nu este axiomatizabilă în $\mathcal{L}_{<}$.

Dem.: Fie $\mathcal{K}$ clasa $\mathcal{L}_{<}$ -structurilor $\mathcal{A} = (A, <)$ a.î. $(A, <)$ este bine ordonată. Presupunem prin reducere la absurd că $\mathcal{K}$ este axiomatizabilă, deci că

există Γ o mulțime de enunțuri ale lui $\mathcal{L}_{<}$ a.î. $\mathcal{K} = \text{Mod}(\Gamma)$.

Fie $\mathcal{L}^+$ extensia lui $\mathcal{L}_{<}$ obținută prin adăugarea simbolurilor de constantă c_n , $n \in \mathbb{N}$. Fie

$$\Delta := \Gamma \cup \{c_{n+1} < c_n \mid n \in \mathbb{N}\} \subseteq \text{Sen}_{\mathcal{L}^+}.$$

Demonstrăm că Δ este satisfiabilă folosind Teorema de compacitate. Fie Δ_0 o submulțime finită a lui Δ . Atunci

$$\begin{aligned} \Delta_0 &\subseteq \Gamma \cup \{c_{n+1} < c_n \mid n \in I\}, \text{ unde } I \subseteq \mathbb{N} \text{ este finită} \\ &\subseteq \Gamma \cup \{c_{n+1} < c_n \mid n = 0, \dots, M\} \text{ pentru un } M \in \mathbb{N}. \end{aligned}$$



Aplicație a Teoremei de compacitate - mulțimi bine ordonate

Fie $(A, <)$ o mulțime infinită bine ordonată. Definim

$$\begin{aligned}a_{M+1} &:= \min A, \\a_M &:= \min A \setminus \{a_{M+1}\}, \\a_{M-1} &:= \min A \setminus \{a_{M+1}, a_M\}, \\&\vdots \\a_0 &:= \min A \setminus \{a_{M+1}, a_M, \dots, a_1\}.\end{aligned}$$

Atunci $a_{M+1} < a_M < \dots < a_0$.

Fie $\mathcal{A}^+$ extensia lui $\mathcal{A} = (A, <)$ la $\mathcal{L}^+$ obținută astfel:

$$c_0^{A^+} = a_0, \dots, c_{M+1}^{A^+} = a_{M+1}, \quad c_n^{A^+} \text{ arbitrar pentru } n > M + 1.$$

Atunci $\mathcal{A}^+ \models \Delta_0$.



Aplicație a Teoremei de compacitate - mulțimi bine ordonate

Aplicând Teorema de compacitate, rezultă că

$$\Delta = \Gamma \cup \{c_{n+1} \dot{<} c_n \mid n \in \mathbb{N}\}$$

are un model

$$\mathcal{B}^+ = (B, <, b_0, b_1, \dots, b_n, \dots),$$

deci $c_n^{\mathcal{B}^+} = b_n$ pentru orice $n \in \mathbb{N}$.

Fie $\mathcal{L}_{<}$ -structura

$$\mathcal{B} = (B, <).$$

Deoarece $\mathcal{B}^+ \models \Gamma$ și $\Gamma \subseteq \text{Sen}_{\mathcal{L}_{<}}$, avem că $\mathcal{B} \models \Gamma$. Prin urmare, $(B, <)$ este bine ordonată.

Deoarece $\mathcal{B}^+ \models \{c_{n+1} \dot{<} c_n \mid n \in \mathbb{N}\}$ rezultă că $b_{n+1} < b_n$ pentru orice $n \in \mathbb{N}$. Prin urmare,

$$S := \{b_n \mid n \in \mathbb{N}\} \quad \text{nu are minim.}$$

Dar S este o submulțime nevidă a lui B . Am obținut o contradicție.



Definiția 2.99

Mulțimea $Axm_{\mathcal{L}} \subseteq Form_{\mathcal{L}}$ a **axiomelor (logice)** ale lui $\mathcal{L}$ constă din:

- (i) toate tautologiile.
- (ii) formulele de forma

$$t = t, \quad s = t \rightarrow t = s, \quad s = t \wedge t = u \rightarrow s = u,$$

pentru orice termeni s, t, u .

- (iii) formulele de forma

$$t_1 = u_1 \wedge \dots \wedge t_m = u_m \rightarrow ft_1 \dots t_m = fu_1 \dots u_m,$$

$$t_1 = u_1 \wedge \dots \wedge t_m = u_m \rightarrow (Rt_1 \dots t_m \leftrightarrow Ru_1 \dots u_m),$$

pentru orice $m \geq 1$, $f \in \mathcal{F}_m$, $R \in \mathcal{R}_m$ și orice termeni t_i, u_i ($i = 1, \dots, m$).

- (iv) formulele de forma

$$\varphi_x(t) \rightarrow \exists x \varphi,$$

unde $\varphi_x(t)$ este o substituție liberă (**$\exists$ -axiomele**).



Definiția 2.100

Regulile de deducție (sau inferență) sunt următoarele: pentru orice formule φ, ψ ,

(i) *din φ și $\varphi \rightarrow \psi$ se inferă ψ (*modus ponens* sau (MP)):*

$$\frac{\varphi, \varphi \rightarrow \psi}{\psi}$$

(ii) *dacă $x \notin FV(\psi)$, atunci din $\varphi \rightarrow \psi$ se inferă $\exists x\varphi \rightarrow \psi$ (*$\exists$ -introducerea*):*

$$\frac{\varphi \rightarrow \psi}{\exists x\varphi \rightarrow \psi} \quad \text{dacă } x \notin FV(\psi).$$



Fie Γ o mulțime de formule ale lui $\mathcal{L}$.

Definiția 2.101

Γ -teoremele lui $\mathcal{L}$ sunt formulele definite astfel:

- ($\Gamma 0$) Orice axiomă logică este Γ -teoremă.
- ($\Gamma 1$) Orice formulă din Γ este Γ -teoremă.
- ($\Gamma 2$) Dacă φ și $\varphi \rightarrow \psi$ sunt Γ -teoreme, atunci ψ este Γ -teoremă.
- ($\Gamma 3$) Dacă $\varphi \rightarrow \psi$ este Γ -teoremă și $x \notin FV(\psi)$, atunci $\exists x \varphi \rightarrow \psi$ este Γ -teoremă.
- ($\Gamma 4$) Numai formulele obținute aplicând regulile ($\Gamma 0$) ($\Gamma 1$), ($\Gamma 2$) și ($\Gamma 3$) sunt Γ -teoreme.

Dacă φ este Γ -teoremă, atunci spunem și că φ este dedusă din ipotezele Γ .



Notății

$\Gamma \vdash_{\mathcal{L}} \varphi := \varphi$ este Γ -teoremă $\vdash_{\mathcal{L}} \varphi := \emptyset \vdash_{\mathcal{L}} \varphi$

Definiția 2.102

O formulă φ se numește **teoremă (logică)** a lui $\mathcal{L}$ dacă $\vdash_{\mathcal{L}} \varphi$.

Reformulând condițiile din definiția Γ -teoremelor folosind notația $\vdash$, obținem

Pentru orice mulțime de formule Γ și orice formule φ, ψ , au loc următoarele:

- (i) Dacă φ este axiomă, atunci $\Gamma \vdash_{\mathcal{L}} \varphi$;
- (ii) Dacă $\varphi \in \Gamma$, atunci $\Gamma \vdash_{\mathcal{L}} \varphi$;
- (iii) Dacă $\Gamma \vdash_{\mathcal{L}} \varphi$ și $\Gamma \vdash_{\mathcal{L}} \varphi \rightarrow \psi$, atunci $\Gamma \vdash_{\mathcal{L}} \psi$.
- (iv) Dacă $\Gamma \vdash_{\mathcal{L}} \varphi \rightarrow \psi$ și $x \notin FV(\psi)$, atunci $\Gamma \vdash_{\mathcal{L}} \exists x\varphi \rightarrow \psi$.



Definiția 2.103

O Γ -*demonstrație* (*demonstrație din ipotezele Γ*) a lui $\mathcal{L}$ este o secvență de formule $\theta_1, \dots, \theta_n$ astfel încât pentru fiecare $i \in \{1, \dots, n\}$, una din următoarele condiții este satisfăcută:

- (i) θ_i este axiomă;
- (ii) $\theta_i \in \Gamma$;
- (iii) există $k, j < i$ astfel încât $\theta_k = \theta_j \rightarrow \theta_i$;
- (iv) există $j < i$ astfel încât

$$\theta_j = \varphi \rightarrow \psi \text{ și } \theta_i = \exists x \varphi \rightarrow \psi, \text{ unde } x \notin FV(\psi).$$

O $\emptyset$ -*demonstrație* se va numi simplu *demonstrație*.



Definiția 2.104

Fie φ o formulă. O Γ -demonstrație a lui φ sau demonstrație a lui φ din ipotezele Γ este o Γ -demonstrație $\theta_1, \dots, \theta_n$ astfel încât $\theta_n = \varphi$.

Propoziția 2.105

Fie Γ o mulțime de formule. Pentru orice formulă φ ,

$\Gamma \vdash_{\mathcal{L}} \varphi$ ddacă există o Γ -demonstrație a lui φ .

Fie Γ o mulțime de formule.

Teorema 2.106 (Teorema Tautologiei (Post))

Fie $\psi, \varphi_1, \dots, \varphi_n$ astfel încât

- (i) ψ este consecință tautologică a mulțimii $\{\varphi_1, \dots, \varphi_n\}$.
- (ii) $\Gamma \vdash_{\mathcal{L}} \varphi_1, \Gamma \vdash_{\mathcal{L}} \varphi_2, \dots, \Gamma \vdash_{\mathcal{L}} \varphi_n$.

Atunci $\Gamma \vdash_{\mathcal{L}} \psi$.

Teorema 2.107 (Teorema Deducției)

Fie ψ o formulă și φ un **enunț**. Atunci

$$\Gamma \cup \{\varphi\} \vdash_{\mathcal{L}} \psi \quad \text{dacă} \quad \Gamma \vdash_{\mathcal{L}} \varphi \rightarrow \psi.$$

Propoziția 2.108

Pentru orice formulă φ și orice variabilă x ,

$$\Gamma \vdash \varphi \quad \Longleftrightarrow \quad \Gamma \vdash \forall x \varphi.$$



Definiția 2.109

Fie φ o formulă cu $FV(\varphi) = \{x_1, \dots, x_n\}$. *Închiderea universală* a lui φ este enunțul

$$\overline{\forall \varphi} := \forall x_1 \dots \forall x_n \varphi.$$

Notății 2.110

$$\overline{\forall \Gamma} := \{\overline{\forall \psi} \mid \psi \in \Gamma\}.$$

Propoziția 2.111

Pentru orice formulă φ ,

$$\Gamma \vdash \varphi \iff \Gamma \vdash \overline{\forall \varphi} \iff \overline{\forall \Gamma} \vdash \varphi \iff \overline{\forall \Gamma} \vdash \overline{\forall \varphi}.$$



Definiția 2.112

Fie Γ o mulțime de formule. Spunem că

- (i) Γ este **consistentă** dacă există o formulă φ astfel încât $\Gamma \not\vdash_{\mathcal{L}} \varphi$.
- (ii) Γ este **inconsistentă** dacă nu este consistentă, adică $\Gamma \vdash_{\mathcal{L}} \varphi$ pentru orice formulă φ .

Propoziția 2.113

Pentru orice mulțime de formule Γ , următoarele afirmații sunt echivalente:

- (i) Γ este inconsistentă.
- (ii) Pentru orice formulă ψ , $\Gamma \vdash \psi$ și $\Gamma \vdash \neg\psi$.
- (iii) Există o formulă ψ astfel încât $\Gamma \vdash \psi$ și $\Gamma \vdash \neg\psi$.



Teorema de completitudine

Teorema de completitudine - prima versiune

Fie Γ o mulțime de enunțuri.

Γ este consistentă $\iff \Gamma$ este satisfiabilă.

Teorema de completitudine - a doua versiune

Pentru orice mulțime de enunțuri Γ și orice enunț φ ,

$$\Gamma \vdash_{\mathcal{L}} \varphi \iff \Gamma \models_{\mathcal{L}} \varphi.$$

- ▶ Teorema de completitudine a fost demonstrată de Gödel în 1929 în teza sa de doctorat.
- ▶ Henkin a dat în teza sa de doctorat din 1947 o demonstrație simplificată.



APLICAȚIE LA SISTEME MULTIAGENT

Referință

Michael Wooldridge, [An Introduction to MultiAgent Systems](#),
Second Edition, John Wiley & Sons, 2009



Întrebarea **Ce este un agent?** nu are un răspuns definitiv. De-a lungul timpului au fost propuse numeroase răspunsuri.

Definiția 2.114

Un agent este un sistem (informatic) care este situat într-un anumit mediu și care este capabil de acțiune autonomă în acel mediu pentru a-și îndeplini obiectivele care i-au fost atribuite.

Agentul determină singur ce trebuie făcut pentru a satisface obiectivele pentru care a fost proiectat, în loc să primească permanent instrucțiuni de la utilizatorul sau proprietarul său.

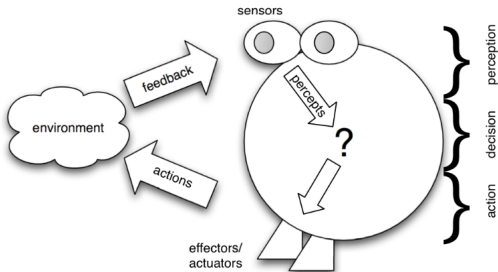


Figura 1: Un agent în mediul său

- ▶ Figura 1 oferă o vedere abstractă a unui agent în mediul său.
- ▶ Agentul primește **intrări senzoriale** din **mediu** și produce, ca ieșire, **acțiuni** care influențează acel mediu. Interacțiunea este de obicei una **continuă**.



- Notăm cu E mulțimea finită a **stărilor** mediului (*environment*):

$$E = \{e, e', e'', \dots\}$$

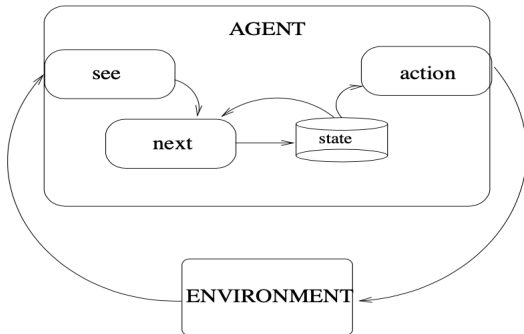
- Se presupune că un agent dispune de un repertoriu de posibile **acțiuni** disponibile:

$$Ac = \{\alpha, \alpha', \alpha'', \dots\}$$

- Acțiunile transformă starea mediului.
- Presupunem că mulțimea Ac conține o **acțiune specială** *null*, cu semnificația că nu va fi efectuată nicio acțiune.



Considerăm agenți bazați pe stări.



Acești agenți au o anumită **structură internă de date**, care este utilizată în mod tipic pentru a memora informații despre starea mediului și despre istoricul acestuia.



- ▶ I mulțimea stărilor interne ale agentului.
- ▶ Per este o mulțime nevidă de **percepte**.

Definiția 2.115

Funcția de **percepție** *see* este definită astfel:

$$see : E \rightarrow Per.$$

Funcția **see** descrie capacitatea agentului de a-și observa mediul. Rezultatul funcției **see** este un **percept** — o intrare perceptivă.

Exemplu: o cameră video sau un senzor în infraroșu al unui robot mobil.

Definiția 2.116

Funcția de **acțiune** *action* și funcția **next** sunt definite astfel:

$$action : I \rightarrow Ac, \quad next : I \times Per \rightarrow I.$$



Comportamentul unui agent bazat pe stări

- ▶ Agentul începe într-o stare internă inițială i_0 .
- ▶ Apoi observă starea mediului e și generează o percepție $see(e)$.
- ▶ Starea internă a agentului este apoi actualizată la $i_1 := next(i_0, see(e))$.
- ▶ Acțiunea selectată de agent este $\alpha := action(i_1)$.
- ▶ Agentul execută acțiunea α .
- ▶ Agentul intră într-un nou ciclu: percepe lumea prin intermediul lui see , își actualizează starea prin $next$ și alege o acțiune de executat prin $action$.



Agenți bazați pe logică

Abordarea **bazată pe logică** este abordarea clasică pentru construirea agenților.

- ▶ O **stare internă** a unui astfel de agent este o **bază de date** formată din formule ale **logicii de ordinul întâi**.
- ▶ Baza de date a agentului ar putea conține formule precum *Open(valve1)*, *Temperature(reactor6, 32)*, *Pressure(tank6, 28)*.
- ▶ Baza de date reprezintă **informațiile** pe care agentul le are despre mediul său.
- ▶ Baza de date a unui agent joacă un rol oarecum analog cu cel al **credinței** la oameni. Unele informații din baza de date pot fi incorecte — senzorii agentului pot fi defecti, raționamentul poate fi greșit, iar informațiile pot fi învechite.
- ▶ Astfel, faptul că *Open(valve1)* se află în bază nu înseamnă că *valva1* este deschisă; ea poate fi închisă.



Fie $\mathcal{L}$ un limbaj de ordinul întâi și $Form_{\mathcal{L}}$ mulțimea formulelor sale.

Presupunem că $\mathcal{L}$ conține:

- ▶ un simbol de relație unară Do ;
- ▶ un simbol de constantă c_{α} pentru fiecare acțiune $\alpha \in Ac$.
Pentru simplitate, scriem α în loc de c_{α} .
- ▶ O bază de date este o mulțime de formule a lui $\mathcal{L}$.
- ▶ Fie $\mathcal{D}$ mulțimea tuturor bazelor de date. Deci, $\mathcal{D} = 2^{Form_{\mathcal{L}}}$.
- ▶ Notăm cu $DB, DB_1, \dots$ elementele lui $\mathcal{D}$.
- ▶ O stare internă a unui agent este o bază de date. Deci, $I = \mathcal{D}$.



- ▶ Fixăm o mulțime $\Sigma \subseteq \text{Form}_{\mathcal{L}}$ de formule ale lui $\mathcal{L}$, ale cărei elemente se numesc **formule de deducție**.
- ▶ Folosim notația $DB \vdash_{\Sigma} \varphi$ pentru $DB \cup \Sigma \vdash \varphi$.
- ▶ Ideea este că
dacă $DB \vdash_{\Sigma} Do(\alpha)$, atunci α este acțiunea care urmează să fie executată de agent.
- ▶ Comportamentul agentului este determinat de **formulele sale de deducție (programul său)** și de **baza de date curentă**.

Definiția în pseudocod a funcției $action : \mathcal{D} \rightarrow Ac$ este prezentată în Figura 2.



1. function *action*($DB : \mathcal{D}$) returns an action Ac
2. begin
3. for each $\alpha \in Ac$ do
4. if $DB \vdash_{\Sigma} Do(\alpha)$ then
5. return α
6. end-if
7. end-for
8. for each $\alpha \in Ac$ do
9. if $DB \not\vdash_{\Sigma} \neg Do(\alpha)$ then
10. return α
11. end-if
12. end-for
13. return *null*
14. end function *action*

Figura 2: Funcția de selecție a acțiunilor agentului.



- ▶ În liniile 3–7, agentul ia pe rând fiecare dintre acțiunile sale posibile α și încearcă să demonstreze formula $Do(\alpha)$ pornind de la baza sa de date DB (transmisă ca parametru funcției), folosind mulțimea sa Σ de formule de deducție. Dacă agentul reușește să demonstreze $Do(\alpha)$, atunci α este returnată ca acțiunea care urmează să fie executată.
- ▶ Dacă agentul nu reușește să demonstreze $Do(\alpha)$ pentru nicio acțiune $\alpha \in Ac$, atunci încearcă să găsească o acțiune care este compatibilă cu formulele de deducție și cu baza de date, adică una care nu este explicit interzisă.
- ▶ În liniile 8–12, agentul încearcă să găsească o acțiune $\alpha \in Ac$ astfel încât $\neg Do(\alpha)$ să nu poată fi derivată din baza sa de date folosind formulele de deducție. Dacă reușește să găsească o astfel de acțiune, atunci aceasta este returnată ca acțiunea care urmează să fie executată.



- ▶ Dacă însă agentul nu reușește să găsească nicio acțiune care să fie măcar consistentă, atunci returnează acțiunea specială *null*, indicând că nu a fost selectată nicio acțiune.

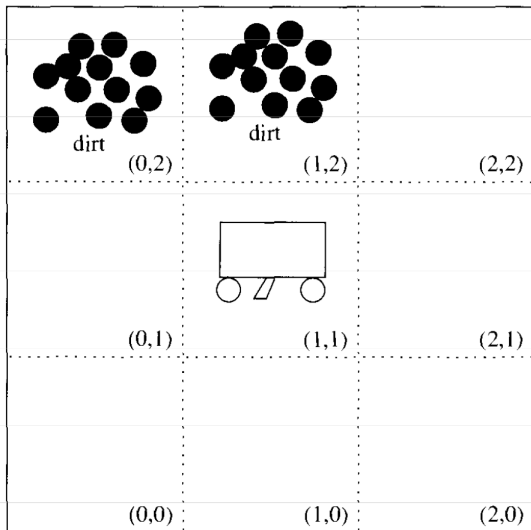


Exemplu: lumea aspiratorului (vacuum cleaning world)

- ▶ Avem un mic **agent robotic** care curăță o casă.
- ▶ Robotul este echipat cu un **senzor** care îi indică dacă se află deasupra vreunei **murdării** și cu un **aspirator** care poate fi folosit pentru a aspira **murdăria**.
- ▶ Robotul are întotdeauna o orientare bine determinată: **nord**, **sud**, **est** sau **vest**.
- ▶ Pe lângă faptul că poate aspira murdăria, agentul se poate deplasa **înainte cu un "pas"** sau se poate **roti cu 90 de grade spre dreapta**.
- ▶ Agentul se deplasează într-o cameră împărțită sub formă de grilă în mai multe pătrate de dimensiuni egale.
- ▶ Agentul nostru nu părăsește niciodată camera.
- ▶ Pentru simplitate, presupunem că încăperea este o grilă 3×3 , iar agentul începe întotdeauna în **pătratul (0, 0)**, **orientat spre nord**.



Exemplu: lumea aspiratorului (vacuum cleaning world)





Exemplu: lumea aspiratorului (vacuum cleaning world)

- ▶ Mulțimea *Per* a percepțiilor este definită astfel:

$$Per = \{murdar, nimic\},$$

unde **murdar** semnifică faptul că există murdărie sub el, iar **nimic** indică faptul că nu există nicio informație specială.

- ▶ Mulțimea *Ac* a acțiunilor este definită astfel:

$$Ac = \{\text{înainte}, \text{aspiră}, \text{rotește}\},$$

unde **înainte** înseamnă **mergi înainte**, **aspiră** înseamnă **aspiră murdăria**, iar **rotește** înseamnă **rotește-te cu 90 de grade spre dreapta**.

- ▶ Scopul agentului este de a parcurge continuu camera, căutând și îndepărtând murdăria.



Exemplu: *lumea aspiratorului* (vacuum cleaning world)

Folosim trei **predicate simple ale domeniului**:

$La(i, j)$ agentul se află la (i, j) ,

$Murdar(i, j)$ există murdărie la (i, j) ,

$Orientare(d)$ agentul este orientat în direcția d ,

unde $i, j \in \{0, 1, 2\}$ și $d \in \{\text{nord, sud, est, vest}\}$.

Prin urmare, limbajul de ordinul întâi $\mathcal{L}$ conține:

- ▶ două simboluri de relație binară **La** și **$Murdar$** ;
- ▶ un simbol de relație unară **$Orientare$** ;
- ▶ simbolurile de constantă **$nord, sud, est, vest$** ;
- ▶ simbolurile de constantă **$0, 1, 2$** .



Exemplu: *lumea aspiratorului* (*vacuum cleaning world*)

Funcția *next*

- ▶ analizează informațiile perceptuale obținute din mediu și baza de date curentă și generează o nouă bază de date care include aceste informații.
- ▶ elimină informațiile vechi sau irelevante și, de asemenea, încearcă să determine *noua* poziție și orientare a agentului.

Specificăm funcția *next* în mai multe părți.

Fie $old(DB)$ mulțimea informațiilor *vechi* dintr-o bază de date, pe care dorim ca funcția de actualizare *next* să le elimine.

$$old(DB) = DB \cap \Delta,$$

unde

$$\begin{aligned} \Delta = & \{La(i, j) \mid i, j \in \{0, 1, 2\}\} \cup \{Murdar(i, j) \mid i, j \in \{0, 1, 2\}\} \\ & \cup \{Orientare(d) \mid d \in \{\text{nord, sud, est, vest}\}\}. \end{aligned}$$



Exemplu: *lumea aspiratorului* (*vacuum cleaning world*)

- ▶ Avem nevoie de o funcție **new**, care oferă mulțimea noilor formule ce trebuie adăugate în baza de date:

$$new : \mathcal{D} \times Per \rightarrow \mathcal{D}.$$

- ▶ Aceasta trebuie să genereze formule de tipul:
 - ▶ $La(\dots)$, care descrie noua poziție a agentului;
 - ▶ $Orientare(\dots)$, care descrie orientarea agentului;
 - ▶ $Murdar(\dots)$, dacă a fost detectată murdărie în noua poziție.

Funcția **next** este definită astfel:

$$next : \mathcal{D} \times Per \rightarrow \mathcal{D}, \quad next(DB, p) = (DB - old(DB)) \cup new(DB, p)$$



Exemplu: *lumea aspiratorului* (*vacuum cleaning world*)

Formulele de deducție au forma generală

$\varphi \rightarrow \psi$, unde φ, ψ sunt formule ale lui $\mathcal{L}$

Curățare

$La(x, y) \wedge Murdar(x, y) \rightarrow Do(\text{aspiră})$ x, y sunt variabile

- ▶ Dacă agentul se află în poziția (x, y) și percepe murdărie, atunci acțiunea prescrisă va fi aspirarea murdăriei.
- ▶ Această acțiune are prioritate față de toate celelalte comportamente posibile ale agentului (cum ar fi navigarea).



Exemplu: *lumea aspiratorului* (vacuum cleaning world)

Traversarea

- ▶ Acțiunea de bază a agentului este parcurgerea mediului.
- ▶ Pentru simplitate, presupunem că robotul se va deplasa întotdeauna de la (0,0) la (0,1), apoi la (0,2) și după aceea la (1,2), (1,1) și așa mai departe. Odată ce agentul ajunge la (2,2), acesta trebuie să se întoarcă la (0,0).
- ▶ **Formulele de deducție** care tratează parcurgerea până la (0,2) sunt:

$$La(0,0) \wedge Orientare(nord) \wedge \neg Murdar(0,0) \rightarrow Do(\text{înainte})$$
$$La(0,1) \wedge Orientare(nord) \wedge \neg Murdar(0,1) \rightarrow Do(\text{înainte})$$
$$La(0,2) \wedge Orientare(nord) \wedge \neg Murdar(0,2) \rightarrow Do(\text{rotește})$$
$$La(0,2) \wedge Orientare(est) \rightarrow Do(\text{înainte})$$

- ▶ Formule similare pot fi generate cu ușurință pentru a conduce agentul până la (2,2) și înapoi la (0,0).