Programare Logică
Cursurile II, III, IV, V
Recapitulare Logică Propoziţională

Claudia MUREȘAN
cmuresan@fmi.unibuc.ro, c.muresan@yahoo.com

Universitatea din București
Facultatea de Matematică și Informatică
București

2019–2020, Semestrul II

Cuprinsul acestui curs

- **Logica matematică** este o ramură a matematicii care se ocupă cu exprimarea formalizată (i. e. formală, simbolică) a legilor gândirii și studierea acestora cu mijloace matematice.
- Ne propunem să studiem **logica clasică**, în două forme ale ei: **logica clasică a propozițiilor** și **logica clasică a predicatelor** sau a **propozițiilor cu variabile**. Vom face deosebirea dintre aceste două tipuri de logică clasică mai târziu. Ambele sunt **logici bivalente**, adică operează cu doar două **valori de adevăr**: **fals** și **adevărat**.
- Așadar, în **logica clasică**, toate enunțurile (propozițiile, afirmațiile) sunt presupuse a fi **adevărate** sau **false**. Aceasta nu este o condiție trivială, nici măcar dacă eliminăm din discuție enunțurile interogative și pe cele exclamative, după cum ne amintim din primul curs de logică matematică, din exemplul cu enunțul subiectiv, precum și cel cu **paradoxul mincinosului:** să se determine dacă următoarea afirmație este **adevărată** sau **falsă**: *Această afirmație este falsă.*

Logică matematică clasică. Calculul propozițional

- În acest curs vom recapitula **logica propozițională clasică** din cursurile de LOGICĂ MATEMATICĂ ŞI COMPUTAŢIONALĂ de anul trecut.
- Vom studia **sistemul formal al calculului propozițional clasic** sub trei aspecte fundamentale:
  1. **sintaxa**, care se ocupă de limbajul formal al calculului propozițional clasic, i. e. de cadrul formal, de exprimarea în simboluri a obiectelor matematice cu care vom lucra;
  2. **algebra**, care asociază o structură algebrică sistemului formal descris în partea de sintaxă și folosește această asociere pentru a transfera proprietățile algebrice ale acelei structuri în proprietăți logice, și invers;
  3. **semantica**, aceasta fiind partea în care, pe baza structurii algebrice atașate logicii, se calculează efectiv **valorile de adevăr** ale enunțurilor (**fals** sau **adevărat**).
- Există și alte aspecte sub care poate fi studiat un sistem logic, cum ar fi: aspectul **topologic**, cel **probabilist** etc., dar studierea lor depășeste cadrul și scopul acestui curs.
- Toate aceste aspecte sub care poate fi studiat un sistem logic sunt denumite *dimensiuni ale sistemului logic*.

Alfabetul sistemului formal al calculului propozițional clasic

Definiții și notații

Următoarele **simboluri** formează **alfabetul** sistemului formal al calculului propozițional clasic:

1. *variabilele propoziționale*, notate, de obicei, u, v, w etc., uneori cu indici, care formează o mulțime infinită, și de obicei considerată numărabilă; vom nota cu V mulțimea variabilelor propoziționale;
2. *conectorii logici primitivi*:
   $\neg$: *negația* (se citește: "non" sau "not");
   $\rightarrow$: *implicația* (se citește: "implică");
3. parantezele: (,), [, și].

Simbolurile enumerate mai sus se numesc *simboluri primitive* și sunt presupuse a fi două câte două distincte (de exemplu $\neg \notin V$ etc.).

La acestea se adaugă *conectorii logici derivați*, care se definesc pe baza conectorilor logici primitivi, și care vor fi prezentați mai jos.

Să notăm cu A *alfabetul sistemului formal al calculului propozițional clasic*, adică mulțimea simbolurilor primitive: $A = V \cup \{\neg, \rightarrow, (,), [,]\}$.

Cuvintele peste alfabetul simbolurilor primitive

Definiție

Șirurile (alăturările) finite și nevide de simboluri primitive se numesc *cuvinte*.

Notație

Așadar *mulțimea cuvintelor* calculului propozițional clasic este mulțimea A^+ a cuvintelor finite și nevide peste alfabetul A:
$A^+ = \{a_1 a_2 \dots a_n \mid n \in \mathbb{N}^*, a_1, a_2, \dots, a_n \in A\}$.

Exemplu

$u \to \neg\, v$, $\neg\,(u \to \neg\, v) \to w$, $\to u \to\to uv\neg\,)$ sunt **cuvinte**.

Observație

Intuiția ne determină să conferim "înțeles" simbolurilor primitive, și ne spune că primele două cuvinte din exemplul anterior "au sens", în timp ce al treilea "nu are sens". Dintre cuvintele peste alfabetul definit mai sus, le vom selecta pe cele care "au sens", și le vom numi *enunțuri*. Urmează definiția lor riguroasă:

Cuvintele care “au sens“: enunțurile

Definiție

Un *enunț* este un cuvânt φ care satisface una dintre condițiile următoare:

- (E_1) φ este o variabilă propozițională;
- (E_2) există un enunț ψ a. î. $\varphi = \neg\psi$;
- (E_3) există două enunțuri ψ și χ a. î. $\varphi = \psi \rightarrow \chi$.

Definiție

Variabilele propoziționale se numesc *enunțuri atomice* sau *enunțuri elementare*. Enunțurile care nu sunt variabile propoziționale, adică se află în cazul (E_2) sau (E_3) din definiția anterioară, se numesc *enunțuri compuse*.

Notație

Vom nota cu E mulțimea tuturor enunțurilor.

Paranteză – intersecția familiei vide de mulțimi

Fie I o mulțime arbitrară și $(A_i)_{i \in I}$ o familie de mulțimi indexată de I – a se vedea finalul primului curs.
Să transcriem definiția intersecției unei familii arbitrare de mulțimi pentru familia vidă, i.e. pentru cazul $I = \emptyset$:

$$\bigcap_{i \in \emptyset} A_i = \{x \mid (\forall\, i \in \emptyset)\,(x \in A_i)\} = \{x \mid (\forall\, i)\,(i \in \emptyset \Rightarrow x \in A_i)\}.$$

Proprietatea $i \in \emptyset$ este falsă pentru orice element i, așadar implicația $i \in \emptyset \Rightarrow x \in A_i$ este adevărată pentru orice i și orice x, deci proprietatea $(\forall\, i)\,(i \in \emptyset \Rightarrow x \in A_i)$ este adevărată pentru orice x. Sigur că nu există o mulțime care să conțină toate obiectele x. O astfel de mulțime ar conține, în particular, toate mulțimile, deci ar avea drept submulțime mulțimea tuturor mulțimilor, care nu există.
Intersecția familiei vide nu există decât raportat la o mulțime totală T: intersecția familiei vide de părți ale lui T se **definește** în următorul mod, și este egală cu mulțimea totală T:

$$\bigcap_{i \in \emptyset} A_i := \{x \in T \mid (\forall\, i \in \emptyset)\,(x \in A_i)\} = \{x \in T \mid (\forall\, i)\,(i \in \emptyset \Rightarrow x \in A_i)\} = T,$$

întrucât proprietatea $(\forall\, i)\,(i \in \emptyset \Rightarrow x \in A_i)$ este adevărată pentru orice x.

Mnemonic despre operatori de închidere și familii Moore

Notație (mulțimea părților unei mulțimi)

Pentru orice mulțime T, notăm cu $\mathcal{P}(T) = \{S \mid S \subseteq T\}$.

Fie T o mulțime.

Definiție

- Se numește *familie Moore de părți ale lui T* (sau *sistem de închidere pe mulțimea părților lui T*) o familie de părți ale lui T închisă la intersecții arbitrare, i. e. o familie de mulțimi $\mathcal{M} = (M_i)_{i \in I} \subseteq \mathcal{P}(T)$, cu I mulțime arbitrară, având proprietatea că, pentru orice $S \subseteq I$, $\bigcap_{s \in S} M_s \in \mathcal{M}$ (i. e., pentru orice $S \subseteq I$, există $i_S \in I$, astfel încât $\bigcap_{s \in S} M_s = M_{i_S}$). Familiile Moore se mai numesc *sisteme de închidere*.
- Se numește *operator de închidere pe $\mathcal{P}(T)$* o funcție $C : \mathcal{P}(T) \to \mathcal{P}(T)$, astfel încât, pentru orice $X, Y \in \mathcal{P}(T)$, au loc proprietățile:
    1. $C(C(X)) = C(X)$ (C este *idempotent*);
    2. $X \subseteq C(X)$ (C este *extensiv*);
    3. dacă $X \subseteq Y$, atunci $C(X) \subseteq C(Y)$ (C este *crescător*).

Mnemonic despre operatori de închidere și familii Moore

Remarcă

Orice familie Moore de părți ale lui T conține intersecția familiei vide de părți ale lui T, adică pe T.

Exemplu

- $id_{\mathcal{P}(T)}$ este un operator de închidere pe $\mathcal{P}(T)$.
- Funcția constantă $C : \mathcal{P}(T) \to \mathcal{P}(T)$, $(\forall X \in \mathcal{P}(T))\,(C(X) = T)$, este un operator de închidere pe $\mathcal{P}(T)$.
- $\mathcal{P}(T)$ este o familie Moore de părți ale lui T.
- $\{T\}$ este o familie Moore de părți ale lui T.
- $\emptyset$ nu este o familie Moore de părți ale lui T, pentru că nu îl conține pe T.

Așadar:

Remarcă

Orice familie Moore este nevidă.

Corespondența bijectivă între sistemele de închidere și operatorii de închidere pe o mulțime

Propoziție

Dacă $\mathcal{M}$ este o familie Moore de părți ale lui T, atunci, pentru orice $A \in \mathcal{P}(T)$, există o (unică – să ne amintim că minimul, dacă există, este unic) cea mai mică mulțime din $\mathcal{M}$ care include pe A (cea mai mică în sensul incluziunii), și aceasta este egală cu intersecția mulțimilor din $\mathcal{M}$ care includ pe A.

Propoziție ($\star$)

Fie I o mulțime nevidă și $\mathcal{M} = (M_i)_{i \in I}$ o familie Moore de părți ale lui T. Definim funcția $C_{\mathcal{M}} : \mathcal{P}(T) \to \mathcal{P}(T)$ astfel: pentru orice $X \in \mathcal{P}(T)$, $C_{\mathcal{M}}(X)$ este, prin definiție, cea mai mică mulțime din $\mathcal{M}$ care include pe X, adică

$$C_{\mathcal{M}}(X) := \bigcap_{\substack{M \in \mathcal{M}, \\ X \subseteq M}} M.$$

Atunci $C_{\mathcal{M}}$ este un operator de închidere pe $\mathcal{P}(T)$.

Corespondența bijectivă între sistemele de închidere și operatorii de închidere pe o mulțime

Propoziție $(\star\star)$

Fie $C : \mathcal{P}(T) \to \mathcal{P}(T)$ un operator de închidere pe $\mathcal{P}(T)$. Definim $\mathcal{M}_C = C(\mathcal{P}(T)) = \{C(X) \mid X \in \mathcal{P}(T)\} = \{X \in \mathcal{P}(T) \mid X = C(X)\} \subseteq \mathcal{P}(T)$ (familia mulțimilor închise din $\mathcal{P}(T)$ raportat la operatorul de închidere C). Atunci $\mathcal{M}_C$ este o familie Moore de părți ale lui T.

Propoziție

Aplicațiile din cele Propozițiile $(\star)$ și $(\star\star)$ sunt inverse una alteia, adică:

1. *pentru orice operator de închidere $C : \mathcal{P}(T) \to \mathcal{P}(T)$, $C_{\mathcal{M}_C} = C$;*
2. *pentru orice familie Moore $\mathcal{M}$ de părți ale lui T, $\mathcal{M}_{C_{\mathcal{M}}} = \mathcal{M}$.*

Așadar aceste aplicații sunt bijecții, deci mulțimea operatorilor de închidere pe $\mathcal{P}(T)$ și mulțimea familiilor Moore de părți ale lui T sunt în bijecție.

Remarcă

Conform definiției enunțurilor, **toate enunțurile** se obțin prin aplicarea regulilor (E_1), (E_2) și (E_3), așadar E este cea mai mică mulțime de cuvinte peste A care include pe V și este închisă la $\neg$ și $\rightarrow$, i. e. cea mai mică (în sensul incluziunii, adică în posetul $(\mathcal{P}(A^+), \subseteq)$) submulțime M a lui A^+ cu proprietățile:

1. $V \subseteq M$,
2. pentru orice $\varphi \in M$, rezultă că $\neg\varphi \in M$,
3. pentru orice $\varphi, \psi \in M$, rezultă că $\varphi \rightarrow \psi \in M$.

Remarcă (E e închiderea lui V în familia Moore a submulțimilor lui A^+ închise la $\neg$ și $\rightarrow$)

Fie $\mathcal{M} = \{M \subseteq A^+ \mid (\forall\,\psi, \chi \in M)\,(\neg\,\psi, \psi \rightarrow \chi \in M)\}$.
$\mathcal{M}$ este sistem de închidere pe $\mathcal{P}(A^+)$. Într–adevăr, $A^+ \in \mathcal{M}$, iar, dacă I este o mulțime nevidă și $(M_i)_{i \in I} \subseteq \mathcal{M}$, atunci, pentru orice $\psi, \chi \in \bigcap_{i \in I} M_i$, avem, pentru fiecare $i \in I$, $\psi, \chi \in M_i$, așadar $\neg\,\psi, \psi \rightarrow \chi \in M_i$, prin urmare $\neg\,\psi, \psi \rightarrow \chi \in \bigcap_{i \in I} M_i$, așadar $\bigcap_{i \in I} M_i \in \mathcal{M}$.
Dacă notăm cu $C_{\mathcal{M}} : \mathcal{P}(A^+) \rightarrow \mathcal{P}(A^+)$ operatorul de închidere asociat lui $\mathcal{M}$, atunci, conform remarcii anterioare, $E = C_{\mathcal{M}}(V)$.

Rolul parantezelor; parantezări corecte

Observație

În definiția de mai sus a enunțurilor, se subînțelege faptul că parantezele au rolul obișnuit: aici, parantezele pot încadra enunțuri, și se folosesc pentru a încadra enunțuri compuse în interiorul altor enunțuri compuse, indicând ordinea în care se aplică regulile (E_1), (E_2) și (E_3) pentru obținerea unui enunț compus (impropriu spus, ordinea "aplicării conectorilor logici primitivi" pentru obținerea acelui enunț).
O *parantezare corectă* a unui enunț este o dispunere a parantezelor în interiorul acelui enunț astfel încât fiecare pereche de paranteze să încadreze un (alt) enunț, și, desigur, astfel încât ordinea aplicării regulilor (E_1), (E_2) și (E_3) pentru obținerea enunțului respectiv să fie indicată corect de acea parantezare.
Adică, pentru orice enunțuri ψ și $\psi \rightarrow \chi$:

- $\neg\,(\psi)$ este o parantezare corectă a enunțului $\neg\,\psi$;
- $\psi \rightarrow (\chi)$, $(\psi) \rightarrow \chi$ și $(\psi) \rightarrow (\chi)$ sunt parantezări corecte ale enunțului $\psi \rightarrow \chi$.

Prioritățile conectorilor logici

Observație

Observăm că, în scrierea enunțurilor, dată de definiția de mai sus, conectorii logici primitivi apar scriși la fel ca niște operatori:

- $\neg$ apare scris la fel ca un operator unar;
- $\rightarrow$ apare scris la fel ca un operator binar.

Pentru a evita încărcarea scrierii cu prea multe paranteze, se face următoarea **convenție**: se acordă prioritate mai mare conectorului logic "unar" $\neg$ și prioritate mai mică celui "binar", $\rightarrow$.

Noțiunea de **prioritate** are aici semnificația obișnuită, de determinare a ordinii "aplicării conectorilor logici", corect spus de determinare a ordinii aplicării regulilor (E_1), (E_2) și (E_3) pentru construirea unui enunț.

Conectorul cu prioritate mai mare "se va aplica" primul, de fapt regula (E_2) se va aplica înaintea regulii (E_3), i. e., pentru orice enunțuri α și β, scrierea $\neg\alpha \rightarrow \beta$ va semnifica $(\neg\alpha) \rightarrow \beta$.

Egalitatea între enunțuri

Observație

Egalitatea între enunțuri care apare în scrierea regulilor (E_2) și (E_3) este egalitatea obișnuită între cuvinte peste un alfabet, între șiruri de simboluri, anume **literal identitatea**, adică egalitatea simbol cu simbol, i. e. egalitatea lungimilor și identitatea (coincidența) simbolurilor de pe aceeași poziție în fiecare cuvânt (ca la șiruri de caractere: identitatea literă cu literă, caracter cu caracter), desigur, modulo parantezarea aleasă.
Adică: două enunțuri **scrise numai cu simboluri primitive** (vom vedea ce sunt simbolurile derivate) sunt *egale* ddacă există câte o parantezare corectă pentru fiecare astfel încât, cu acele parantezări, enunțurile respective să fie literal identice (ca și cuvinte peste alfabetul prezentat mai sus, format din simbolurile primitive).

Remarcă

Dacă recitim observația anterioară, vom remarca faptul că orice enunț se află în **exact una** (i. e. **una și numai una**) dintre cele 3 situații prezentate de regulile (E_1), (E_2) și (E_3).

Remarcă

Noțiunea de enunț este definită recursiv: se pornește de la variabilele propoziționale și se aplică recursia dată de regulile (E_2) și (E_3).
Din faptul că enunțurile sunt șiruri **finite** de simboluri primitive și observația că, prin aplicarea oricăreia dintre regulile (E_1), (E_2) și (E_3), lungimea enunțului format până la momentul curent crește cu cel puțin câte o unitate, deducem faptul că orice enunț se obține prin aplicarea regulilor (E_1), (E_2) și (E_3) de un număr finit de ori, i. e. printr–un număr finit de aplicări ale regulilor (E_1), (E_2) și (E_3), i. e., pornind de la variabilele propoziționale (evident, de la un număr finit de variabile propoziționale), într–un număr finit de pași, fiecare pas constând în aplicarea unei reguli de recursie: (E_2) sau (E_3).

Pentru a putea defini riguros egalitatea de enunțuri, să definim:

Definiție (arborii binari asociați enunțurilor)

Orice enunț are un *arbore binar asociat*, definit **recursiv** astfel:

- pentru orice variabilă propozițională p, arborele binar asociat lui p este arborele cu un singur nod, etichetat cu p;
- pentru orice enunț ψ, arborele binar asociat enunțului $\neg\psi$ are rădăcina etichetată cu, conectorul logic $\neg$, și arborele binar asociat lui ψ ca unic subarbore;
- pentru orice enunțuri ψ și χ, arborele binar asociat enunțului $\psi \rightarrow \chi$ are rădăcina etichetată cu, conectorul logic $\rightarrow$, arborele binar asociat lui ψ ca subarbore stâng și arborele binar asociat lui χ ca subarbore drept.

Remarcă (unicitatea arborelui binar asociat unui enunț)

Cum un enunț se află în unul și numai unul dintre cazurile (E_1), (E_2) și (E_3), se demonstrează inductiv că un enunț are un unic arbore binar asociat: dacă S e mulțimea enunțurilor care au câte un unic arbore binar asociat, atunci, conform definiției anterioare:

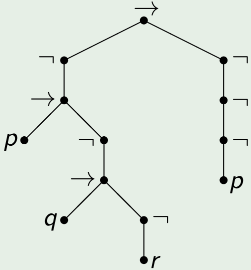
1. $V \subseteq S$,
2. pentru orice $\psi \in S$, rezultă $\neg\psi \in S$,
3. pentru orice $\psi, \chi \in S$, rezultă că $\psi \rightarrow \chi \in S$,

Remarcă (continuare)

prin urmare $S = E$, conform definiției mulțimii E a tuturor enunțurilor.

Exemplu

Arborele binar asociat enunțului $\neg(p \to \neg(q \to \neg r)) \to \neg\neg\neg p$, unde $p, q, r \in V$, este:



Definiție

Considerăm două enunțuri ca fiind *egale* (modulo parantezări corecte) ddacă au același arbore binar asociat, în care ordinea fiilor unui nod (deci diferența dintre subarborele stâng și subarborele drept) contează (adică inversând doi subarbori distincți ai unui nod obținem un arbore diferit).

Riguros:

Definiție (enunțurile, arborii binari asociați lor și parantezările corecte)

Mulțimea E a *enunțurilor* calculului propozițional clasic este cea mai mică submulțime $E \subseteq A^+$ a mulțimii cuvintelor (finite și nevide) peste alfabetul A al simbolurilor primitive având proprietățile:

1. $V \subseteq E$; oricărui $p \in V$ îi asociem arborele cu un singur nod, etichetat cu p;
2. dacă $\psi \in E$, atunci $\neg(\psi) \in E$;
   în plus, dacă $\psi \in V$ sau există $\alpha \in E$ astfel încât $\psi \in \{\neg\alpha, \neg(\alpha)\}$, atunci avem și $\neg\psi \in E$;
   enunțurilor $\neg\psi$ și $\neg(\psi)$ le asociem arborele binar având rădăcina etichetată cu $\neg$ și, ca unic subarbore al rădăcinii, arborele binar asociat lui ψ;
3. dacă $\psi, \chi \in E$, atunci $(\psi) \rightarrow (\chi) \in E$;
   în plus, dacă $\psi \in V$ sau există $\alpha \in E$ astfel încât $\psi \in \{\neg\alpha, \neg(\alpha)\}$, atunci avem și $\psi \rightarrow (\chi) \in E$;
   similar, dacă $\chi \in V$ sau există $\beta \in E$ astfel încât $\chi \in \{\neg\beta, \neg(\beta)\}$, atunci avem și $(\psi) \rightarrow \chi \in E$;
   iar, dacă există $\alpha, \beta \in E$ astfel încât $\psi \in V \cup \{\neg\alpha, \neg(\alpha)\}$ și $\chi \in V \cup \{\neg\beta, \neg(\beta)\}$, atunci avem și $\psi \rightarrow \chi \in E$;
   enunțurilor $\psi \rightarrow \chi$, $\psi \rightarrow (\chi)$, $(\psi) \rightarrow \chi$ și $(\psi) \rightarrow (\chi)$ le asociem arborele binar având rădăcina etichetată cu $\rightarrow$, arborele binar asociat lui ψ ca subarbore stâng al rădăcinii și arborele binar asociat lui χ ca subarbore drept al rădăcinii.

Mnemonic despre relații n–are, relații binare

Definiție

Fie $n \in \mathbb{N}^*$ și $A_1, A_2, \ldots, A_n$ mulțimi. Se numește *relație n–ară* între mulțimile $A_1, A_2, \ldots, A_n$ o submulțime a produsului cartezian $A_1 \times A_2 \times \ldots \times A_n$.

Observație

Pentru $n = 1$ în definiția anterioară se obține noțiunea de *relație unară* pe o mulțime: prin definiție, o *relație unară* pe o mulțime A este o submulțime a lui A.
Pentru $n = 2$ în definiția anterioară se obține noțiunea de *relație binară*.

Definiție

Fie A și B două mulțimi. Se numește *relație binară* între A și B o submulțime R a produsului direct $A \times B$.
Pentru fiecare $a \in A$ și fiecare $b \in B$, faptul că $(a, b) \in R$ se mai notează cu $a\,R\,b$ și se citește: *a este în relația R cu b*.

Exemplu

Pentru orice mulțimi A și B, produsul direct $A \times B$ este o relație binară între A și B (evident, cea mai mare în sensul incluziunii dintre toate relațiile binare între A și B).

- Fie A o mulțime.

Definiție

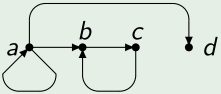
Se numește *relație binară pe A* o relație binară între A și A, i. e. o submulțime a produsului direct $A^2 = A \times A$.

Remarcă

Dacă A este finită și nevidă, iar R este o relație binară pe A, atunci perechea (A, R) este un graf orientat (cu mulțimea vârfurilor A și mulțimea arcelor R), așadar relația binară R poate fi reprezentată grafic chiar prin acest graf orientat.

Exemplu

Relația binară $R = \{(a,a), (a,b), (a,d), (b,c), (c,b)\}$ pe mulțimea cu exact 4 elemente $A = \{a,b,c,d\}$ poate fi reprezentată grafic astfel:



Tipuri de relații binare pe o mulțime

Definiție (tipuri de relații binare pe o mulțime)

Fie $R \subseteq A^2$ (i. e. R o relație binară pe A). R se zice:

- *reflexivă* ddacă, pentru orice $a \in A$, aRa;
- *ireflexivă* ddacă, pentru orice $a \in A$, $(a, a) \notin R$, i. e. nu există $a \in A$ cu aRa;
- *simetrică* ddacă, pentru orice $a, b \in A$, dacă aRb, atunci bRa;
- *antisimetrică* ddacă, pentru orice $a, b \in A$, dacă aRb și bRa, atunci $a = b$;
- *asimetrică* ddacă, pentru orice $a, b \in A$, dacă $(a, b) \in R$, atunci $(b, a) \notin R$;
- *tranzitivă* ddacă, pentru orice $a, b, c \in A$, dacă aRb și bRc, atunci aRc;
- *totală* ddacă, pentru orice $a, b \in A$ cu $a \neq b$, au loc aRb sau bRa;
- *completă* ddacă, pentru orice $a, b \in A$, au loc aRb sau bRa.

Tipuri de relații binare pe o mulțime

Definiție

Fie $R \subseteq A^2$ (i. e. R o relație binară pe A). R se numește:

- *(relație de) preordine* ddacă e reflexivă și tranzitivă;
- *(relație de) echivalență* ddacă e o preordine simetrică, i. e. o relație reflexivă, simetrică și tranzitivă;
- *(relație de) ordine (parțială)* ddacă e o preordine antisimetrică, i. e. o relație reflexivă, tranzitivă și antisimetrică;
- *(relație de) ordine totală* ddacă e simultan o relație de ordine și o relație totală;
- *(relație de) ordine strictă* ddacă e asimetrică și tranzitivă (echivalent, ddacă e ireflexivă și tranzitivă).

Remarcă

Întrucât orice relație de ordine este reflexivă, rezultă că o relație de ordine este totală ddacă este completă.

Partiție a unei mulțimi

Definiție

Fie A nevidă și $(A_i)_{i \in I}$ o familie nevidă (i. e. cu $I \neq \emptyset$) de submulțimi ale lui A. Familia $(A_i)_{i \in I}$ se numește *partiție* a lui A ddacă satisface următoarele condiții:

1. pentru orice $i \in I$, $A_i \neq \emptyset$
2. pentru orice $i, j \in I$, dacă $i \neq j$, atunci $A_i \cap A_j = \emptyset$ (i. e. mulțimile din familia $(A_i)_{i \in I}$ sunt două câte două disjuncte)
3. $\bigcup\limits_{i \in I} A_i = A$

Propoziție

Fie A nevidă și $(A_i)_{i \in I}$ o partiție a lui A. Atunci, pentru orice $x \in A$, există un unic $i_0 \in I$, a. î. $x \in A_{i_0}$.

Clase de echivalență, mulțime factor (mulțime cât)

Pentru cele ce urmează, vom considera mulțimea A nevidă, și o relație de echivalență $\sim$ pe A, i. e.:

- $\sim$ este o relație binară pe A: $\sim \subseteq A^2$
- $\sim$ este **reflexivă**: pentru orice $x \in A$, $x \sim x$
- $\sim$ este **simetrică**: pentru orice $x, y \in A$, dacă $x \sim y$, atunci $y \sim x$
- $\sim$ este **tranzitivă**: pentru orice $x, y, z \in A$, dacă $x \sim y$ și $y \sim z$, atunci $x \sim z$

Să observăm că, în definiția simetriei, putem interschimba x și y și continua seria de implicații, obținând implicație dublă, adică: $\sim$ este **simetrică** ddacă, pentru orice $x, y \in A$, are loc echivalența: $x \sim y$ ddacă $y \sim x$.

Definiție

Pentru fiecare $x \in A$, definim *clasa de echivalență a lui* x *raportat la* $\sim$ ca fiind următoarea submulțime a lui A, notată cu $\widehat{x}$ sau cu $x/_{\sim}$:
$\widehat{x} := x/_{\sim} := \{y \in A \mid x \sim y\}$.

Remarcă

Observăm că simetria lui $\sim$ ne asigură de faptul că: pentru orice $x \in A$, $\widehat{x} = \{y \in A \mid y \sim x\}$.

Proprietățile claselor de echivalență

Propoziție

Pentru orice $x, y \in A$:

- $x \sim y$ *ddacă* $y \sim x$ *ddacă* $x \in \widehat{y}$ *ddacă* $y \in \widehat{x}$ *ddacă* $\widehat{x} = \widehat{y}$;
- $(x, y) \notin \sim$ *ddacă* $(y, x) \notin \sim$ *ddacă* $x \notin \widehat{y}$ *ddacă* $y \notin \widehat{x}$ *ddacă* $\widehat{x} \cap \widehat{y} = \emptyset$.

Definiție

Fiecare $x \in A$ se numește *reprezentant al clasei* $\widehat{x}$.

Remarcă

- Pentru fiecare $x \in A$, orice $y \in \widehat{x}$ este reprezentant al clasei $\widehat{x}$.

Într–adevăr, conform propoziției precedente, pentru orice $x, y \in A$, are loc echivalența: $y \in \widehat{x}$ ddacă $\widehat{x} = \widehat{y}$, iar, conform definiției anterioare, orice y este reprezentant al clasei $\widehat{y}$, care este egală cu $\widehat{x}$ exact atunci când $y \in \widehat{x}$, deci orice $y \in \widehat{x}$ este reprezentant al clasei $\widehat{x}$.
Mai mult:

- pentru fiecare $x, y \in A$, y este reprezentant al clasei $\widehat{x}$ ddacă $y \in \widehat{x}$.

Definiție

Mulțimea claselor de echivalență ale lui $\sim$ se notează cu $A/_\sim$ și se numește *mulțimea factor a lui* A *prin* $\sim$ sau *mulțimea cât a lui* A *prin* $\sim$: $A/_\sim = \{\widehat{x} \mid x \in A\}$.
Funcția $x \mapsto \widehat{x}$ de la A la $A/_\sim$ e surjectivă și se numește *surjecția canonică*.

Observație

Denumirile de **mulțime factor** și **mulțime cât** se datorează faptului că mulțimea $A/_\sim$ din definiția anterioară se obține prin "împărțirea mulțimii A în clasele de echivalență ale lui $\sim$" (a se vedea următoarea propoziție).

Propoziție (clasele de echivalență formează o partiție)

Mulțimea factor $A/_\sim$ *este o partiție a lui* A.

Notație

Notăm cu:

- $\mathrm{Eq}(A)$ mulțimea relațiilor de echivalență pe A;
- $\mathrm{Part}(A)$ mulțimea partițiilor lui A.

Propoziție ($\mathrm{Eq}(A) \cong \mathrm{Part}(A)$)

Aplicația $\sim\ \mapsto A/_\sim$ *de la mulțimea* $\mathrm{Eq}(A)$ *a relațiilor de echivalență pe* A *la mulțimea* $\mathrm{Part}(A)$ *a partițiilor lui* A *este o bijecție.*

Egalitatea enunțurilor modulo parantezări corecte

Definiție (enunțurile, arborii binari asociați lor și parantezările corecte – continuare)

Considerăm relația binară $\Bumpeq$ pe E definită prin: oricare ar fi $\varphi, \psi \in E$, $\varphi \Bumpeq \psi$ ddacă φ și ψ au același arbore binar asociat, în care ordinea fiilor unui nod contează. Evident, $\Bumpeq$ este o relație de echivalență pe E.
Identificăm mulțimea factor $E/\Bumpeq$ cu E, identificând fiecare $\varphi \in E$ cu $\varphi/\Bumpeq$. Deci oricare două enunțuri φ, ψ astfel încât $\varphi \Bumpeq \psi$ vor fi considerate egale.

Exemplu

Dacă $p, q \in V$, atunci $(\neg\, p) \rightarrow (\neg\, (\neg\, q)) \Bumpeq \neg\, p \rightarrow \neg\,\neg\, q$, așadar considerăm $(\neg\, p) \rightarrow (\neg\, (\neg\, q)) = \neg\, p \rightarrow \neg\,\neg\, q$.

Conectorii logici derivați

Notație (abrevieri pentru enunțuri compuse)

Pentru orice enunțuri $\varphi, \psi \in E$, introducem notațiile (abrevierile):

$\varphi \vee \psi := \neg\,\varphi \rightarrow \psi$ (*disjuncția* dintre φ și ψ; se citește: φ “sau“ ψ)

$\varphi \wedge \psi := \neg\,(\varphi \rightarrow \neg\,\psi)$ (*conjuncția* dintre φ și ψ; se citește: φ “și“ ψ)

$\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$ (*echivalența logică* dintre φ și ψ; se citește: φ “echivalent cu“ ψ)

Definiție

Simbolurile $\vee$, $\wedge$ și $\leftrightarrow$ se numesc *conectorii logici derivați*.

Observație

Conectorii logici derivați se scriu ca niște operatori binari, și le vom acorda aceeași prioritate cu aceea a conectorului logic primitiv “binar“ $\rightarrow$.

Remarcă

În această prezentare a sistemului formal al logicii propoziționale clasice, am considerat negația și implicația ca și conectori logici primitivi, iar disjuncția, conjuncția și echivalența ca și conectori logici derivați, introduși prin notațiile de mai sus, pe baza celor primitivi.

Există prezentări ale sistemului formal al logicii propoziționale clasice care sunt echivalente cu cea din acest curs și care folosesc alți conectori logici primitivi.

Schemele (i. e. tipurile) de axiome

- Am definit limbajul cu care vom lucra. Acum vom defini, tot la acest nivel, formal, sintactic, noțiunea de “adevăr“ în logica pe care o construim. “Adevărurile sintactice“ vor fi “teoremele“ acestei logici, iar, pentru a le obține, vom da un set de axiome și vom defini o modalitate prin care, din adevăruri sintactice stabilite până la un moment dat, se deduc alte adevăruri sintactice. Acea modalitate se va numi **regula de deducție modus ponens**.

Definiție

O *axiomă* a sistemului formal al logicii propoziționale clasice este un enunț de oricare dintre următoarele trei forme, unde $\varphi, \psi, \chi \in E$ sunt enunțuri arbitrare:

$$\begin{array}{ll} (A_1) & \varphi \to (\psi \to \varphi) \\ (A_2) & (\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi)) \\ (A_3) & (\neg\,\varphi \to \neg\,\psi) \to (\psi \to \varphi) \end{array}$$

Fiecare dintre scrierile (A_1), (A_2) și (A_3) este o *schemă de axiome*, adică o regulă pentru generarea unui număr infinit de axiome.

Mulțimea axiomelor, i. e. a enunțurilor pornind de la care se deduc adevărurile sintactice

Definiție (continuare)

Axiomele logicii propoziționale clasice se obțin prin înlocuirea, în aceste scheme de axiome, a enunțurilor generice (arbitrare) φ, ψ, χ cu enunțuri precizate (date), adică axiomele sunt enunțuri de una dintre formele (A_1), (A_2) și (A_3), cu φ, ψ și χ enunțuri date.
Prin extensie, vom numi uneori schemele de axiome (A_1), (A_2) și (A_3), simplu, **axiome**.

Notație

Notăm cu Ax mulțimea axiomelor:

$$\begin{array}{rll} Ax = & \{\varphi \rightarrow (\psi \rightarrow \varphi), & \\ & (\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi)), & \\ & (\neg \varphi \rightarrow \neg \psi) \rightarrow (\psi \rightarrow \varphi) & \mid \varphi, \psi, \chi \in E\}. \end{array}$$

- Așa cum am anunțat, acum vom defini **deducția sintactică** (**inferența sintactică**), adică modalitatea prin care, din aceste axiome, se obțin toate **teoremele** (**adevărurile sintactice**) în acest sistem logic.

Remarcă

Se poate demonstra că schemele de axiome de mai sus sunt **independente**, i. e. niciuna dintre ele nu poate fi dedusă din celelalte două prin modalitatea de deducție sintactică pe care o vom defini.
O modalitate pentru a demonstra acest fapt este să se definească semantica logicii propoziționale clasice, ca în cele ce urmează, să se demonstreze Teorema de Completitudine Tare, care afirmă că deducția sintactică, coincide cu deducția semantică, apoi să se arate că, pentru fiecare două dintre aceste trei axiome, există cel puțin o evaluare care le atribuie valoarea 1 (reprezentând **adevărul**), dar celei de–a treia axiome îi atribuie valoarea 0 (reprezentând **falsul**).
Acest fapt arată și că, între aceste trei scheme de axiome, nu există două din care să se deducă sintactic **toate** adevărurile sintactice (teoremele) acestui sistem logic, care se deduc din toate cele trei scheme de axiome.
Se pot, însă, defini alte seturi echivalente de scheme de axiome (adică seturi de scheme de axiome din care se deduc sintactic aceleași teoreme), cu aceiași sau cu alți conectori logici primitivi.

Notație (scrierea regulilor de deducție)

Notația uzuală pentru reguli de deducție ale unui sistem logic, pe care o vom folosi în cele ce urmează, este aceasta: $\dfrac{\text{condiția } C_1}{\text{consecința } C_2}$, cu semnificația că: dacă este satisfăcută condiția C_1, atunci este satisfăcută consecința C_2.

Regula de deducție **modus ponens** și teoremele formale

Definiție (teoremele (formale), i. e. adevărurile sintactice)

Teoremele formale (numite și, simplu, *teoreme*, sau *adevăruri sintactice*) ale logicii propoziționale clasice sunt enunțurile definite prin următoarele trei reguli:

- (T_1) orice axiomă este o teoremă formală;
- (T_2) dacă $\varphi, \psi \in E$ sunt două enunțuri a. î. ψ și $\psi \rightarrow \varphi$ sunt teoreme formale, atunci φ este o teoremă formală;
- (T_3) orice teoremă formală a logicii propoziționale clasice poate fi obținută prin aplicarea regulilor (T_1) și (T_2) de un număr finit de ori.

Notații

Mulțimea tuturor teoremelor formale va fi notată cu T.
Faptul că un enunț φ este teoremă formală se notează: $\vdash \varphi$.

Definiție (regula de deducție modus ponens (MP))

Regula (T_2) se numește *regula de deducție modus ponens* (o vom abrevia "MP") și, cu notația stabilită mai sus, poate fi scrisă astfel în formă simbolică: $\dfrac{\vdash \psi, \ \vdash \psi \rightarrow \varphi}{\vdash \varphi}$, sau, echivalent: $\dfrac{\psi, \ \psi \rightarrow \varphi}{\varphi}$.

Definiția recursivă a mulțimii teoremelor formale

Remarcă

Regula (T_3), chiar fără precizarea finitudinii, spune că T este cea mai mică mulțime închisă la regulile (T_1) și (T_2), i. e. cea mai mică mulțime de enunțuri care include mulțimea axiomelor și e închisă la regula (MP), i. e. cea mai mică submulțime M a lui E cu proprietățile:

1. $M \supseteq Ax$,
2. pentru orice $\varphi, \psi \in E$, dacă $\psi, \psi \to \varphi \in M$, atunci $\varphi \in M$,

(cea mai mică în sensul incluziunii), pentru că regula (T_3) spune că nu se află în T niciun element care să nu se obțină prin aplicarea regulilor (T_1) și (T_2), adică niciun element care să nu fie nici axiomă, nici enunț obținut prin aplicarea succesivă a regulii **MP**, pornind de la axiome.

Demonstrațiile formale

Definiție

Fie φ un enunț. O *demonstrație formală pentru* φ este un șir finit și nevid de enunțuri $\varphi_1, \ldots, \varphi_n$, a. î. $n \in \mathbb{N}^*$, $\varphi_n = \varphi$ și, pentru fiecare $i \in \overline{1, n}$, este satisfăcută una dintre următoarele condiții:

1. φ_i este o axiomă;
2. există $k, j \in \overline{1, i-1}$ a. î. $\varphi_k = \varphi_j \rightarrow \varphi_i$.

n se numește *lungimea* demonstrației formale $\varphi_1, \ldots, \varphi_n$.

Remarcă

În scrierea definiției de mai sus, am folosit faptul că $\overline{1, 0} = \emptyset$ (pentru o scriere uniformă a definiției, fără a trata separat cazul $i = 1$). Având în vedere acest lucru, este clar că, într–o demonstrație formală $\varphi_1, \ldots, \varphi_n$, φ_1 este o axiomă.

Remarcă

Este imediat că, dacă $\varphi_1, \ldots, \varphi_n$ este o demonstrație formală, atunci, pentru orice $i \in \overline{1, n}$, $\varphi_1, \ldots, \varphi_i$ este o demonstrație formală.

Teoremele sunt enunțurile care admit demonstrații formale

Remarcă

Este ușor de observat că regulile (1) și (2) din definiția unei demonstrații formale exprimă exact regulile (T_1) și (T_2), respectiv, ceea ce arată imediat faptul că un enunț φ este o teoremă formală ddacă există o demonstrație formală pentru φ.

Remarcă

Desigur, o teoremă formală poate avea mai multe demonstrații formale și poate avea demonstrații formale de lungimi diferite.

Enunțurile deductibile din ipoteze

Definiție

Fie $\Sigma \subseteq E$ o mulțime de enunțuri. Enunțurile care *se deduc sintactic din ipotezele* Σ, numite și *consecințele sintactice ale lui* Σ, se definesc astfel:

- (CS_1) orice axiomă se deduce sintactic din ipotezele Σ;
- (CS_0) orice enunț $\varphi \in \Sigma$ se deduce sintactic din ipotezele Σ;
- (CS_2) dacă $\varphi, \psi \in E$ sunt două enunțuri a. î. ψ și $\psi \rightarrow \varphi$ se deduc sintactic din ipotezele Σ, atunci φ se deduce sintactic din ipotezele Σ;
- (CS_3) orice enunț care se deduce sintactic din ipotezele Σ se poate obține prin aplicarea regulilor (CS_1), (CS_0) și (CS_2) de un număr finit de ori.

Notație

Notăm faptul că un enunț φ se deduce sintactic din ipotezele $\Sigma \subseteq E$ prin: $\Sigma \vdash \varphi$.

Definiție (și regula de deducție din ipoteze este tot modus ponens)

Regula (CS_2) se numește tot *regula de deducție modus ponens* (o vom abrevia tot "MP") și, cu notația stabilită mai sus, poate fi scrisă astfel în formă simbolică:

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

Definiția recursivă a unei mulțimi de consecințe sintactice

Remarcă

Regula (CS_3), chiar fără precizarea finitudinii numărului de aplicări ale acestor reguli, spune că mulțimea consecințelor sintactice ale unei mulțimi Σ de enunțuri este cea mai mică mulțime închisă la regulile (CS_1), (CS_0) și (CS_2), adică cea mai mică mulțime de enunțuri care include mulțimea axiomelor și mulțimea Σ a ipotezelor și e închisă la regula (MP), i. e. cea mai mică submulțime M a lui E cu proprietățile:

1. $M \supseteq Ax$,
2. $M \supseteq \Sigma$,
3. pentru orice $\varphi, \psi \in E$, dacă $\psi, \psi \rightarrow \varphi \in M$, atunci $\varphi \in M$,

(cea mai mică în sensul incluziunii), pentru că (CS_3) spune că mulțimea consecințelor sintactice ale lui Σ nu conține alte elemente decât cele obținute din regulile (CS_1), (CS_0) și (CS_2).

Remarcă (deducția pornind de la axiome și de la ipotezele din Σ)

Definiția consecințelor sintactice ale lui Σ este exact definiția teoremelor formale în care mulțimea Ax se înlocuiește cu $Ax \cup \Sigma$.

Demonstrații formale din ipoteze

Definiție

Fie φ un enunț și Σ o mulțime de enunțuri. O *Σ–demonstrație formală pentru φ* este un șir finit și nevid de enunțuri $\varphi_1, \ldots, \varphi_n$, a. î. $n \in \mathbb{N}^*$, $\varphi_n = \varphi$ și, pentru fiecare $i \in \overline{1, n}$, este satisfăcută una dintre următoarele condiții:

1. φ_i este o axiomă;
2. $\varphi_i \in \Sigma$;
3. există $k, j \in \overline{1, i-1}$ a. î. $\varphi_k = \varphi_j \rightarrow \varphi_i$.

n se numește *lungimea* Σ–demonstrației formale $\varphi_1, \ldots, \varphi_n$.

Remarcă

Amintindu–ne că $\overline{1, 0} = \emptyset$, este clar că, într–o Σ–demonstrație formală $\varphi_1, \ldots, \varphi_n$, φ_1 este o axiomă sau un element al lui Σ.

Remarcă

Este imediat că, dacă Σ este o mulțime de enunțuri și $\varphi_1, \ldots, \varphi_n$ este o Σ–demonstrație formală, atunci, pentru orice $i \in \overline{1, n}$, $\varphi_1, \ldots, \varphi_i$ este o Σ–demonstrație formală.

Enunțurile deductibile din ipoteze sunt exact enunțurile care admit demonstrații formale din acele ipoteze

Remarcă

Este ușor de observat că regulile (1), (2) și (3) din definiția unei Σ–demonstrații formale exprimă exact regulile (CS_1), (CS_0) și (CS_2), respectiv, ceea ce arată imediat faptul că un enunț φ este o consecință sintactică a lui Σ ddacă există o Σ–demonstrație formală pentru φ.

Remarcă

Desigur, o consecință sintactică a lui Σ poate avea mai multe Σ–demonstrații formale și poate avea Σ–demonstrații formale de lungimi diferite.

Remarcă

Este imediat, direct din definițiile date, că, pentru orice enunț φ și orice mulțime de enunțuri Σ:

1. $\emptyset \vdash \varphi$ ddacă $\vdash \varphi$;
2. dacă $\vdash \varphi$, atunci $\Sigma \vdash \varphi$;
3. dacă $\varphi \in \Sigma$, atunci $\Sigma \vdash \varphi$.

Acesta este **sistemul Hilbert**

Întreaga prezentare de până acum a fost efectuată la **nivel sintactic**: am pornit de la un **alfabet** (o **mulțime de simboluri**), am definit un tip particular de **cuvinte peste acest alfabet**, numite **enunțuri**, apoi un tip particular de enunțuri, numite **teoreme formale**, și **deducția sintactică** (**inferența sintactică**), care indică modul în care, din teoreme formale, se obțin alte teoreme formale, cu generalizarea la **consecințe sintactice** ale unor mulțimi de enunțuri.

Definiție

Deducția pe baza axiomelor (A_1), (A_2), (A_3) și a regulii de deducție **MP** se numește *sistemul Hilbert* pentru calculul propozițional clasic.

Notație

Vom nota cu $\mathcal{L}$ acest sistem formal pentru logica propozițională clasică.

În tot restul acestui capitol – i.e. până la secțiunea despre rezoluția propozițională – ne vom referi la sistemul Hilbert pentru logica propozițională clasică.

Propoziție (o numim ad–hoc Propoziția $\star$)

Fie $\Sigma \subseteq E$, $\Delta \subseteq E$ și $\varphi \in E$. Atunci:

1. *dacă $\Sigma \subseteq \Delta$ și $\Sigma \vdash \varphi$, atunci $\Delta \vdash \varphi$;*
2. *dacă $\Sigma \vdash \varphi$, atunci există $\Gamma \subseteq \Sigma$ a. î. Γ este o mulțime finită și $\Gamma \vdash \varphi$;*
3. *dacă $\Sigma \vdash \psi$ pentru orice $\psi \in \Delta$ și $\Delta \vdash \varphi$, atunci $\Sigma \vdash \varphi$.*

Remarcă

Conform punctului (1), în (2) din Propoziția $\star$ avem chiar echivalență:
$\Sigma \vdash \varphi$ ddacă $(\exists \Gamma \subseteq \Sigma)\,(|\Gamma| < \aleph_0, \Gamma \vdash \varphi)$.

Propoziție (principiul identității și principiul terțului exclus)

Pentru orice $\varphi \in E$, următoarele enunțuri sunt teoreme formale:

1. **principiul identității** *(abreviat* **PI***):* $\vdash \varphi \rightarrow \varphi$;
2. **principiul terțului exclus** *(abreviat* **PTE***):* $\vdash \varphi \vee \neg \varphi$.

Teoremă (Teorema deducției – abreviată TD)

Pentru orice $\Sigma \subseteq E$ și orice $\varphi, \psi \in E$, are loc următoarea echivalență:

$$\Sigma \vdash \varphi \rightarrow \psi \quad \textit{ddacă} \quad \Sigma \cup \{\varphi\} \vdash \psi.$$

Limbajul acestei teorii matematice versus metalimbaj

Observație

Denumirea de **teoremă** pentru rezultatul anterior este o denumire din **metalimbaj**, pentru că acest rezultat este o proprietate a sistemului formal al logicii propoziționale clasice.

Denumirea de **teoremă formală** este din limbajul sistemului formal al logicii propoziționale clasice, ea denumește un tip de obiect cu care lucrează acest sistem formal.

Este importantă această distincție. Similaritatea celor două denumiri se datorează faptului că acest sistem formal (al logicii propoziționale clasice) este o formalizare a unor legi ale gândirii (în special a procedeelor gândirii care sunt, în mod curent, folosite în elaborarea raționamentelor matematice), și conține denumiri care îi sugerează întrebuințarea, destinația.

Aceleași considerații sunt valabile pentru **conectorii logici** din sistemul formal al calculului propozițional clasic: $\neg, \rightarrow, \vee, \wedge, \leftrightarrow$, versus **conectorii logici din metalimbaj**, folosiți în enunțurile referitoare la calculul propozițional clasic, scriși fie prin denumirile lor: "non" sau "not" sau "negație", "implică" sau "rezultă", "sau", "și", "echivalent" sau "ddacă", fie prin simbolurile consacrate, în cazul implicației și echivalenței: $\Rightarrow$ și $\Leftrightarrow$, respectiv.

La fel pentru alți termeni din acest sistem logic, precum și din sistemul formal al calculului cu predicate clasic, prezentat în ultimul curs.

Fie $n \in \mathbb{N}^*$ și $\varphi, \varphi_1, \ldots, \varphi_n \in E$, arbitrare, fixate.

Notație (pentru următoarele reguli de deducție)

Regula de deducție $\dfrac{\varphi_1, \ldots, \varphi_n}{\varphi}$ va semnifica faptul că φ se deduce printr–o demonstrație formală din ipotezele $\varphi_1, \ldots, \varphi_n$, adică: $\{\varphi_1, \ldots, \varphi_n\} \vdash \varphi$.

Remarcă (cu notația de tip $\dfrac{\text{ipoteze}}{\text{concluzie}}$)

Regula de deducție $\dfrac{\varphi_1, \ldots, \varphi_n}{\varphi}$ implică regula $\dfrac{\Sigma \vdash \varphi_1, \ldots, \Sigma \vdash \varphi_n}{\Sigma \vdash \varphi}$ pentru orice $\Sigma \subseteq E$, de unde, luând $\Sigma = \emptyset$, obținem și cazul particular: $\dfrac{\vdash \varphi_1, \ldots, \vdash \varphi_n}{\vdash \varphi}$.

Într–adevăr, dacă avem $\dfrac{\varphi_1, \ldots, \varphi_n}{\varphi}$, adică $\{\varphi_1, \ldots, \varphi_n\} \vdash \varphi$, și au loc $\Sigma \vdash \varphi_1, \ldots, \Sigma \vdash \varphi_n$, atunci, conform afirmației (3) din Propoziția $\star$, rezultă că $\Sigma \vdash \beta$.

Remarcă (modus ponens, cu scrierea de mai sus)

Conform afirmației (3) din Propoziția $\star$, pentru orice $\Sigma \subseteq E$, mulțimea consecințelor sintactice ale lui Σ, în particular mulțimea teoremelor formale, este inchisă la regula $\dfrac{\varphi, \varphi \rightarrow \psi}{\psi}$, pe care o numim tot **modus ponens**.

Notație (disjuncție și conjuncție logică între n enunțuri – abrevieri definite recursiv)

Fie $\gamma_1, \gamma_2, \dots, \gamma_n, \dots \in E$, arbitrare. Atunci, pentru orice $n \in \mathbb{N}^*$, avem următoarele scrieri (prioritățile: ca la operatori unari, deci aceeași prioritate ca $\neg$):

$$\bigvee_{i=1}^{n} \gamma_i := \begin{cases} \gamma_1, & n = 1, \\ (\bigvee_{i=1}^{n-1} \gamma_i) \vee \gamma_n, & n > 1, \end{cases} \quad \text{și} \quad \bigwedge_{i=1}^{n} \gamma_i := \begin{cases} \gamma_1, & n = 1, \\ (\bigwedge_{i=1}^{n-1} \gamma_i) \wedge \gamma_n, & n > 1. \end{cases}$$

Exercițiu (temă)

Să se arate că, pentru orice $\varphi, \psi, \chi \in E$:

$$\vdash ((\varphi \wedge \psi) \rightarrow \chi) \rightarrow (\varphi \rightarrow (\psi \rightarrow \chi)),$$

iar, folosind acest fapt, să se arate că, pentru orice $n \in \mathbb{N}^*$ și orice $\gamma_1, \gamma_2, \dots, \gamma_n, \varphi \in E$, au loc echivalențele:

$$\{\gamma_1, \gamma_2, \dots, \gamma_n\} \vdash \varphi \quad \text{ddacă} \quad \{\bigwedge_{i=1}^{n} \gamma_i\} \vdash \varphi \quad \text{ddacă} \quad \vdash (\bigwedge_{i=1}^{n} \gamma_i) \rightarrow \varphi.$$

A se vedea și Propoziția 7.2.53/pagina 160/cartea de G. Georgescu și A. Iorgulescu din bibliografia din breviarul cursului de logică matematică și computațională, precum și demonstrația Propoziției $\star$ de mai sus.

Conectorii logici derivați și axiomele

Notație

$E :=$ mulțimea enunțurilor.

Notație (conectorii logici derivați)

Pentru orice $\varphi, \psi \in E$:

- $\varphi \vee \psi := \neg\,\varphi \rightarrow \psi$
- $\varphi \wedge \psi := \neg\,(\varphi \rightarrow \neg\,\psi)$
- $\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$

((schemele de) axiome)

Pentru orice $\varphi, \psi, \chi \in E$:

$$
\begin{array}{ll}
(A_1) & \varphi \rightarrow (\psi \rightarrow \varphi) \\
(A_2) & (\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi)) \\
(A_3) & (\neg\,\varphi \rightarrow \neg\,\psi) \rightarrow (\psi \rightarrow \varphi)
\end{array}
$$

Notație

$Ax :=$ mulțimea axiomelor, i. e.: $Ax := \{\varphi \rightarrow (\psi \rightarrow \varphi), (\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi)), (\neg\,\varphi \rightarrow \neg\,\psi) \rightarrow (\psi \rightarrow \varphi) \mid \varphi, \psi, \chi \in E\}$.

Deducția sintactică: pornind de la axiome și, eventual, ipoteze, pe baza regulii de deducție **modus ponens** (MP)

Notație

$T :=$ mulțimea teoremelor formale (i. e. a adevărurilor sintactice).

Fie $\varphi, \psi, \chi \in E$, $\Sigma \subseteq E$ și $\Delta \subseteq E$, arbitrare.

Notație

- $\vdash \varphi$: φ este teoremă formală.
- $\Sigma \vdash \varphi$: φ este consecință sintactică a mulțimii Σ de ipoteze.

(regula de deducție **modus ponens** (MP))

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

Remarcă

- $Ax \subseteq T$
- $\vdash \varphi \iff \emptyset \vdash \varphi \implies \Sigma \vdash \varphi$
- $(\forall \sigma \in \Sigma) (\Sigma \vdash \sigma)$

Proprietăți sintactice în logica propozițională clasică

Propoziție ($\star$)

1. $\Sigma \subseteq \Delta$ *și* $\Sigma \vdash \varphi \Longrightarrow \Delta \vdash \varphi$
2. $\Sigma \vdash \varphi \Longleftrightarrow (\exists \Gamma \subseteq \Sigma)\,(|\Gamma| < \aleph_0, \Gamma \vdash \varphi)$
3. $(\forall \psi \in \Delta)\,(\Sigma \vdash \psi)$ *și* $\Delta \vdash \varphi \Longrightarrow \Sigma \vdash \varphi$

Propoziție

- **principiul identității** *(PI)*: $\vdash \varphi \rightarrow \varphi$
- **principiul terțului exclus** *(PTE)*: $\vdash \varphi \vee \neg \varphi$

Teoremă (Teorema deducției (TD))

$$\Sigma \vdash \varphi \rightarrow \psi \quad \Longleftrightarrow \quad \Sigma \cup \{\varphi\} \vdash \psi$$

Ordine versus ordine strictă

Remarcă

Dacă A e o mulțime nevidă, atunci nu există nicio relație binară pe A care să fie și reflexivă, și ireflexivă, prin urmare nu există nicio relație binară pe A care să fie și relație de ordine, și relație de ordine strictă.

Exercițiu

Fie A o mulțime, O mulțimea relațiilor de ordine pe A și S mulțimea relațiilor de ordine strictă pe A.
Să se demonstreze că aplicațiile $\varphi : O \to S$ și $\psi : S \to O$, definite prin:

- pentru orice $\leq \in O$,
  $\varphi(\leq) = \leq \setminus \{(a, a) \mid a \in A\} = \{(x, y) \in A^2 \mid x \leq y$ și $x \neq y\}$ (de obicei notată $\varphi(\leq) = <$ și numită *relația de ordine strictă asociată lui* $\leq$),
- pentru orice $< \in S$,
  $\psi(<) = < \cup \{(a, a) \mid a \in A\} = \{(x, y) \in A^2 \mid x < y$ sau $x = y\}$,

sunt:

- corect definite, i. e. într-adevăr $\varphi(O) \subseteq S$ și $\psi(S) \subseteq O$;
- inverse una alteia, deci sunt bijecții între O și S.

Mulțimi ordonate și elemente distinse ale acestora

Definiție

- O mulțime A înzestrată cu o relație de ordine $\leq \subseteq A^2$ se notează $(A, \leq)$ și se numește *mulțime (parțial) ordonată* sau *poset* (de la englezescul "partially ordered set").
- Dacă, în plus, $\leq$ este o relație de ordine totală, atunci $(A, \leq)$ se numește *mulțime total ordonată* sau *mulțime liniar ordonată* sau *lanț*.

- Fie $(A, \leq)$ un poset și $X \subseteq A$.

Definiție

Un element $a \in A$ se numește:

- *minorant pentru* X ddacă, pentru orice $x \in X$, $a \leq x$
- *majorant pentru* X ddacă, pentru orice $x \in X$, $x \leq a$

Remarcă

X poate avea mai mulți minoranți (majoranți), și poate să nu aibă niciun minorant (majorant).

Definiție

- Un minorant al lui X care aparține lui X (i. e. un element $m \in X$ cu $m \leq x$ pentru orice $x \in X$) se numește *minim al lui X* sau *prim element al lui X* sau *cel mai mic element al lui X* și se notează cu $\min(X)$ sau $\min(X, \leq)$.
- Un majorant al lui X care aparține lui X (i. e. un element $M \in X$ cu $x \leq M$ pentru orice $x \in X$) se numește *maxim al lui X* sau *ultim element al lui X* sau *cel mai mare element al lui X* și se notează cu $\max(X)$ sau $\max(X, \leq)$.

Remarcă

Minimul nu există întotdeauna. Dar antisimetria lui $\leq$ implică faptul că minimul (dacă există) este unic (ceea ce justifică notația de mai sus pentru minim, care indică faptul că minimul lui X este unic determinat de X (și $\leq$)).
La fel pentru maxim.

Definiție

Un poset cu minim și maxim se numește *poset mărginit*. (Minimul și maximul trebuie să fie ale întregului poset, deci trebuie luat $X := A$ în definiția anterioară.)

Remarcă

O mulțime care are minim sau maxim are cel puțin un element (pentru că minimul unei mulțimi aparține acelei mulțimi și la fel și maximul), deci nu poate fi vidă.

Alte elemente distinse într–un poset

Definiție

Infimumul lui X este cel mai mare minorant al lui X, adică maximul mulțimii minoranților lui X, și se notează cu $\inf(X)$ sau $\inf(X, \leq)$.
Supremumul lui X este cel mai mic majorant al lui X, adică minimul mulțimii majoranților lui X, și se notează cu $\sup(X)$ sau $\sup(X, \leq)$.

Remarcă

Infimumul nu există întotdeauna, nici măcar atunci când mulțimea minoranților este nevidă.
Dar, fiind maximul unei mulțimi, infimumul (dacă există) este unic (ceea ce îi justifică denumirea cu articol hotărât și notația, fiecare dintre acestea indicând faptul că infimumul este unic determinat de X (și $\leq$)).
La fel pentru supremum.

Definiția unei algebre Boole

O **algebră Boole** este o **latice distributivă mărginită complementată**, i. e. o structură $(B, \vee, \wedge, \leq, \bar{\cdot}, 0, 1)$ compusă din:

- o mulțime B,
- o relație de ordine parțială $\leq$ pe B,
- două operații binare $\vee$ și $\wedge$ pe B, notate infixat,
- două constante $0, 1 \in B$,
- o operație unară $\bar{\cdot}$ pe B,

iar aceste componente au proprietățile:

- $(B, \vee, \wedge, \leq)$ este o **latice**, i. e.:
  - oricare ar fi $x, y \in B$, există $\sup\{x, y\}$ și $\inf\{x, y\}$ în posetul $(B, \leq)$;
  - $\vee$ și $\wedge$ sunt **idempotente**, **comutative** și **asociative**, i. e.: pentru orice $x, y, z \in B$, au loc: $x \vee x = x$, $x \vee y = y \vee x$, $x \vee (y \vee z) = (x \vee y) \vee z$, și la fel pentru $\wedge$;
  - $\vee$ și $\wedge$ verifică **absorbția**: pentru orice $x, y \in B$, $x \vee (x \wedge y) = x$ și $x \wedge (x \vee y) = x$;
  - pentru orice $x, y \in B$, $x \leq y$ ddacă $x \vee y = y$ ddacă $x \wedge y = x$;
  - pentru orice $x, y \in B$, $x \vee y = \sup\{x, y\}$;
  - pentru orice $x, y \in B$, $x \wedge y = \inf\{x, y\}$;

- laticea $(B, \vee, \wedge, \leq)$ este **distributivă**, i. e.:
  - $\vee$ este **distributivă** față de $\wedge$, i. e.: pentru orice $x, y, z \in B$, $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$;
  - $\wedge$ este **distributivă** față de $\vee$, i. e.: pentru orice $x, y, z \in B$, $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$;
- $(B, \vee, \wedge, \leq, 0, 1)$ este o **latice mărginită**, i. e., în plus:
  - 0 este **minimul** posetului $(B, \leq)$;
  - 1 este **maximul** posetului $(B, \leq)$;
- laticea mărginită $(B, \vee, \wedge, \leq, 0, 1)$ este **complementată** și satisface **unicitatea complementului**, datorită **distributivității**, iar $\bar{\cdot}$ este operația de **complementare**:
  - pentru orice $x \in B$, $\overline{x}$ este **unicul complement** al lui x, adică **unicul** element $\overline{x} \in B$ care satisface: $\begin{cases} x \vee \overline{x} = 1 \\ \text{și} \\ x \wedge \overline{x} = 0. \end{cases}$

În plus, în orice algebră Boole $(B, \vee, \wedge, \leq, \bar{\cdot}, 0, 1)$, se definesc următoarele **operații binare derivate**:

- **implicația (booleană)**, $\rightarrow$: pentru orice $x, y \in B$, $x \rightarrow y := \overline{x} \vee y$;
- **echivalența (booleană)**, $\leftrightarrow$: pentru orice $x, y \in B$, $x \leftrightarrow y := (x \rightarrow y) \wedge (y \rightarrow x)$.

Proprietăți ale unei algebre Boole

- Fie $\mathcal{B} := (B, \vee, \wedge, \leq, \bar{\cdot}, 0, 1)$ o algebră Boole.

Propoziție (autodualitatea complementării)

Pentru orice $x \in B$, $\overline{\overline{x}} = x$.

Propoziție (legile lui de Morgan)

Pentru orice $x, y \in B$:

1. $\overline{x \vee y} = \overline{x} \wedge \overline{y}$
2. $\overline{x \wedge y} = \overline{x} \vee \overline{y}$

Propoziție

Pentru orice $x, y \in B$, au loc următoarele echivalențe:

1. $x = y$ *ddacă* $\overline{x} = \overline{y}$
2. $x \leq y$ *ddacă* $\overline{y} \leq \overline{x}$
3. $x \leq y$ *ddacă* $x \to y = 1$
4. $x = y$ *ddacă* $x \leftrightarrow y = 1$

Algebra Lindenbaum–Tarski a Logicii Propoziționale Clasice

- Această secțiune a cursului expune construcția unei algebre Boole care este asociată în mod canonic sistemului formal $\mathcal{L}$.
- Prin această asociere, proprietățile sintactice ale lui $\mathcal{L}$ se reflectă în proprietăți booleene, și invers.

- Pe tot parcursul acestei secțiuni, $\Sigma \subseteq E$ va fi o mulțime arbitrară fixată de enunțuri ale lui $\mathcal{L}$.
- Σ va reprezenta o mulțime de ipoteze, ceea ce este adesea numită o *teorie* a lui $\mathcal{L}$.

Definiție (o relație de echivalență pe mulțimea enunțurilor)

Definim o relație binară $\sim_\Sigma$ pe mulțimea E a enunțurilor lui $\mathcal{L}$, astfel: pentru orice $\varphi, \psi \in E$,

$$\varphi \sim_\Sigma \psi \text{ ddacă } \Sigma \vdash \varphi \leftrightarrow \psi.$$

Lemă

$\sim_\Sigma$ este o relație de echivalență pe E.

Notație

Să notăm, pentru fiecare $\varphi \in E$, cu $\hat{\varphi}^\Sigma := \{\psi \in E \mid \varphi \sim_\Sigma \psi\}$ clasa de echivalență a lui φ raportat la relația de echivalență $\sim_\Sigma$, și să considerăm mulțimea factor $E/_{\sim_\Sigma} = \{\hat{\varphi}^\Sigma \mid \varphi \in E\}$.

Definiție (o relație de ordine pe mulțimea factor)

Pe mulțimea factor $E/_{\sim_\Sigma}$, definim relația binară $\leq_\Sigma$, prin: oricare ar fi $\varphi, \psi \in E$, $\hat{\varphi}^\Sigma \leq_\Sigma \hat{\psi}^\Sigma$ ddacă $\Sigma \vdash \varphi \to \psi$.

Lemă

$\leq_\Sigma$ este bine definită (i. e. nu depinde de reprezentanții claselor de echivalență ale lui $\sim_\Sigma$) și este o relație de ordine parțială pe $E/_{\sim_\Sigma}$.

Propoziție

Posetul $(E/_{\sim_\Sigma}, \leq_\Sigma)$ este o latice distributivă, în care, pentru orice $\varphi, \psi \in E$, $\inf\{\hat{\varphi}^\Sigma, \hat{\psi}^\Sigma\} = \widehat{\varphi \wedge \psi}^\Sigma$ *și* $\sup\{\hat{\varphi}^\Sigma, \hat{\psi}^\Sigma\} = \widehat{\varphi \vee \psi}^\Sigma$.

Vom nota, pentru orice $\varphi, \psi \in E$, $\hat{\varphi}^\Sigma \wedge_\Sigma \hat{\psi}^\Sigma := \widehat{\varphi \wedge \psi}^\Sigma$ și $\hat{\varphi}^\Sigma \vee_\Sigma \hat{\psi}^\Sigma := \widehat{\varphi \vee \psi}^\Sigma$.

Remarcă

Pentru orice $\varphi, \psi \in E$, $\Sigma \vdash (\varphi \wedge \neg\varphi) \rightarrow \psi$ și $\Sigma \vdash \psi \rightarrow (\varphi \vee \neg\varphi)$, așadar, pentru orice $\varphi, \psi \in E$, $\widehat{\varphi \wedge \neg\varphi}^\Sigma \leq_\Sigma \hat{\psi}^\Sigma \leq_\Sigma \widehat{\varphi \vee \neg\varphi}^\Sigma$. Aceasta înseamnă că, indiferent cine este $\varphi \in E$, $\widehat{\varphi \wedge \neg\varphi}^\Sigma$ și $\widehat{\varphi \vee \neg\varphi}^\Sigma$ sunt, respectiv, primul element și ultimul element al laticii $(E/_{\sim_\Sigma}, \vee_\Sigma, \wedge_\Sigma, \leq_\Sigma)$. Vom nota $0_\Sigma := \widehat{\varphi \wedge \neg\varphi}^\Sigma$ și $1_\Sigma := \widehat{\varphi \vee \neg\varphi}^\Sigma$, pentru un $\varphi \in E$, arbitrar. Unicitatea minimului și a maximului unui poset arată că această definiție nu depinde de alegerea lui $\varphi \in E$, adică 0_Σ și 1_Σ sunt bine definite.

Așadar:

Propoziție

$(E/_{\sim_\Sigma}, \vee_\Sigma, \wedge_\Sigma, \leq_\Sigma, 0_\Sigma, 1_\Sigma)$ *este o latice distributivă mărginită.*

Definiție

Pentru orice $\varphi \in E$, definim: $\overline{\hat{\varphi}^{\Sigma}}^{\Sigma} := \widehat{\neg\varphi}^{\Sigma}$.

Remarcă

Definiția de mai sus pentru operația unară $\bar{\cdot}^{\Sigma} : E/_{\sim_\Sigma} \rightarrow E/_{\sim_\Sigma}$ este corectă, pentru că nu depinde de reprezentanții claselor din $E/_{\sim_\Sigma}$.

Propoziție

$(E/_{\sim_\Sigma}, \vee_\Sigma, \wedge_\Sigma, \leq_\Sigma, \bar{\cdot}^{\Sigma}, 0_\Sigma, 1_\Sigma)$ *este o algebră Boole.*

Definiție

Algebra Boole $(E/_{\sim_\Sigma}, \vee_\Sigma, \wedge_\Sigma, \leq_\Sigma, \bar{\cdot}^{\Sigma}, 0_\Sigma, 1_\Sigma)$ se numește *algebra Lindenbaum–Tarski a lui* Σ *asociată sistemului formal* $\mathcal{L}$.

Remarcă (surjecția canonică transformă conectorii logici în operații booleene: nu e morfism boolean, pentru că E nu e algebră Boole)

Dacă notăm cu $p_\Sigma : E \to E/_{\sim_\Sigma}$ surjecția canonică ($p_\Sigma(\varphi) := \hat{\varphi}^\Sigma$ pentru orice $\varphi \in E$), atunci, oricare ar fi $\varphi, \psi \in E$, au loc următoarele identități (unde $\to_\Sigma$ și $\leftrightarrow_\Sigma$ sunt, respectiv, implicația și echivalența booleană în algebra Boole $(E/_{\sim_\Sigma}, \vee_\Sigma, \wedge_\Sigma, \leq_\Sigma, \bar{\cdot}^\Sigma, 0_\Sigma, 1_\Sigma)$):

- (a) $p_\Sigma(\varphi \vee \psi) = p_\Sigma(\varphi) \vee_\Sigma p_\Sigma(\psi)$;
- (b) $p_\Sigma(\varphi \wedge \psi) = p_\Sigma(\varphi) \wedge_\Sigma p_\Sigma(\psi)$;
- (c) $p_\Sigma(\neg\,\varphi) = \overline{p_\Sigma(\varphi)}^\Sigma$;
- (d) $p_\Sigma(\varphi \to \psi) = p_\Sigma(\varphi) \to_\Sigma p_\Sigma(\psi)$;
- (e) $p_\Sigma(\varphi \leftrightarrow \psi) = p_\Sigma(\varphi) \leftrightarrow_\Sigma p_\Sigma(\psi)$.

Într–adevăr, identitățile (a), (b) și (c) sunt chiar definițiile operațiilor algebrei Boole $E/_{\sim_\Sigma}$. Definiția implicației booleene, (a) și (c) arată că $p_\Sigma(\varphi) \to_\Sigma p_\Sigma(\psi) = \overline{p_\Sigma(\varphi)}^\Sigma \vee_\Sigma p_\Sigma(\psi) = p_\Sigma(\neg\,\varphi \vee \psi)$, ceea ce arată că (d) este echivalent cu faptul că $\Sigma \vdash (\varphi \to \psi) \leftrightarrow (\neg\,\varphi \vee \psi)$, care rezultă din (15), (16) și prima remarcă din această secțiune. (e) se obține, direct, din (b) și (d).
Cele cinci identități de mai sus arată cum conectorii logici sunt convertiți în operații booleene.

Enunțurile deductibile din Σ sunt în 1_Σ

Lemă

Pentru orice $\varphi \in E$, $\Sigma \vdash \varphi$ *ddacă* $\hat{\varphi}^\Sigma = 1_\Sigma$.

Remarcă

Lema anterioară ne oferă o metodă algebrică pentru a verifica dacă un enunț este o consecință sintactică a lui Σ.

Notă

- În mod tipic, pentru a folosi lema anterioară în cadrul unei **demonstrații algebrice**, prin calcul boolean, pentru o deducție formală din ipoteze: $\Sigma \vdash \varphi$, cu $\Sigma \subseteq E$ și $\varphi \in E$, se folosește faptul că, pentru orice ipoteză $\sigma \in \Sigma$, are loc $\Sigma \vdash \sigma$, așadar $\hat{\sigma}^\Sigma = 1_\Sigma$.

Algebra Lindenbaum-Tarski a Logicii Propoziționale Clasice

Notație

În cazul în care $\Sigma = \emptyset$:

- relația de echivalență $\sim_\emptyset$ se notează, simplu, $\sim$, și are următoarea definiție: pentru orice $\varphi, \psi \in E$,

$$\varphi \sim \psi \text{ ddacă } \vdash \varphi \leftrightarrow \psi;$$

- clasele de echivalență ale lui $\sim$, $\hat{\varphi}^\emptyset$ ($\varphi \in E$), se notează $\hat{\varphi}$;
- relația de ordine $\leq_\emptyset$ se notează $\leq$;
- operațiile $\vee_\emptyset$, $\wedge_\emptyset$, $\bar{\cdot}^\emptyset$, $0_\emptyset$ și $1_\emptyset$ se notează, respectiv, $\vee$, $\wedge$, $\bar{\cdot}$, 0 și 1.

Definiție

$\sim$ se numește *echivalența logică* sau *echivalența semantică* între enunțuri. Algebra Boole $(E/\sim, \vee, \wedge, \leq, \bar{\cdot}, 0, 1)$ se numește *algebra Lindenbaum-Tarski asociată sistemului formal $\mathcal{L}$*.

Lema anterioară devine, în acest caz, o caracterizare a teoremelor formale:

Lemă

Pentru orice $\varphi \in E$, $\vdash \varphi$ ddacă $\hat{\varphi} = 1$.

Interpretări (evaluări, semantici) pentru logica $\mathcal{L}$

Definiție (o interpretare e o funcție de la mulțimea variabilelor propoziționale la algebra Boole standard: $\mathcal{L}_2 = \{0, 1\}$, cu $0 < 1$, i. e. o funcție care dă valori de adevăr variabilelor propoziționale)

O *interpretare* (*evaluare*, *semantică*) a lui $\mathcal{L}$ este o funcție oarecare $h : V \to \mathcal{L}_2$.

Propoziție (există o unică funcție care prelungește o interpretare la mulțimea tuturor enunțurilor și transformă conectorii logici primitivi în operații Booleene, așadar calculează valorile de adevăr ale tuturor enunțurilor pornind de la cele ale variabilelor propoziționale)

Pentru orice interpretare $h : V \to \mathcal{L}_2$, există o unică funcție $\tilde{h} : E \to \mathcal{L}_2$ care satisface următoarele proprietăți:

- (a) $\tilde{h}(u) = h(u)$, *pentru orice* $u \in V$;
- (b) $\tilde{h}(\neg \varphi) = \overline{\tilde{h}(\varphi)}$, *pentru orice* $\varphi \in E$;
- (c) $\tilde{h}(\varphi \to \psi) = \tilde{h}(\varphi) \to \tilde{h}(\psi)$, *pentru orice* $\varphi, \psi \in E$.

Definiție

Funcția $\tilde{h} : E \to \mathcal{L}_2$ din propoziția anterioară se numește tot *interpretare*.

Observație

Condiția (*a*) din propoziția anterioară spune că $\tilde{h}\mid_V = h$, adică funcția $\tilde{h}$ prelungește pe h la E.
În condițiile (*b*) și (*c*), în membrii stângi, în argumentele lui $\tilde{h}$, $\neg$ și $\rightarrow$ sunt conectorii logici primitivi, pe când, în membrii drepți, $\bar{\cdot}$ și $\rightarrow$ sunt operațiile de complementare și, respectiv, implicație ale algebrei Boole $\mathcal{L}_2$. Așadar, putem spune că funcția $\tilde{h}$ transformă conectorii logici în operații booleene în algebra Boole standard.
Vom păstra notația $\tilde{h}$ pentru această unică funcție depinzând de interpretarea h.

Corolar (prelungirea unei interpretări la mulțimea enunțurilor transformă toți conectorii logici în operații booleene)

Pentru orice interpretare h și orice $\varphi, \psi \in E$, au loc:

- (*d*) $\tilde{h}(\varphi \vee \psi) = \tilde{h}(\varphi) \vee \tilde{h}(\psi)$
- (*e*) $\tilde{h}(\varphi \wedge \psi) = \tilde{h}(\varphi) \wedge \tilde{h}(\psi)$
- (*f*) $\tilde{h}(\varphi \leftrightarrow \psi) = \tilde{h}(\varphi) \leftrightarrow \tilde{h}(\psi)$

Satisfacere și mulțimi satisfiabile

Fie h o interpretare, φ un enunț și Σ o mulțime de enunțuri.

Definiție (enunțuri adevărate pentru anumite valori de adevăr atribuite variabilelor propoziționale din scrierea lor)

Spunem că φ este *adevărat în interpretarea h* sau că *h satisface φ* ddacă $\tilde{h}(\varphi) = 1$.
φ se zice *fals în interpretarea h* ddacă $\tilde{h}(\varphi) = 0$.
Spunem că *h satisface Σ* sau că h este un *model pentru Σ* ddacă h satisface toate elementele lui Σ.
Spunem că Σ *admite un model* sau că mulțimea Σ este *satisfiabilă* ddacă există un model pentru Σ.
Spunem că φ *admite un model* sau că φ este *satisfiabil* ddacă $\{\varphi\}$ este satisfiabilă.

Notație

Faptul h satisface enunțul φ se notează cu: $h \vDash \varphi$.
Faptul că h satisface mulțimea Σ de enunțuri se notează cu: $h \vDash \Sigma$.

Remarcă

Dacă h este model pentru Σ, atunci h este model pentru orice submulțime a lui Σ.

Vom vedea că orice interpretare satisface mulțimea T a teoremelor formale. Deocamdată, să observăm că:

Remarcă

Orice interpretare satisface $\emptyset$.
Într–adevăr, pentru orice interpretare h, avem:

$$h \vDash \emptyset \iff (\forall \varphi \in \emptyset)\,(\tilde{h}(\varphi) = 1) \iff \overbrace{(\forall \varphi)\,(\underbrace{\underbrace{\varphi \in \emptyset}_{\text{fals pentru orice } \varphi} \implies \tilde{h}(\varphi) = 1)}_{\text{adevărat pentru orice } \varphi}}^{\text{adevărat}}.$$

Definiție (adevărurile semantice și deducția semantică)

Enunțul φ se zice *universal adevărat* ddacă φ este adevărat în orice interpretare.
Enunțurile universal adevărate se mai numesc *adevărurile semantice* sau *tautologiile* lui $\mathcal{L}$.
Spunem că φ *se deduce semantic din* Σ sau că φ *este o consecință semantică a lui* Σ ddacă φ este adevărat în orice interpretare care satisface pe Σ.

Notație

Faptul că φ este universal adevărat se notează cu: $\vDash \varphi$.
Faptul că φ se deduce semantic din Σ se notează cu: $\Sigma \vDash \varphi$.

Le fel cum adevărurile sintactice (i. e. teoremele formale) sunt exact enunțurile deductibile sintactic din $\emptyset$:

Remarcă (adevărurile semantice sunt exact enunțurile deductibile semantic din $\emptyset$)

$$\vDash \varphi \text{ ddacă } \emptyset \vDash \varphi.$$

Într–adevăr, conform definițiilor de mai sus, remarcii anterioare și faptului că o implicație cu antecedentul adevărat este adevărată ddacă și concluzia implicației este adevărată:

$$\emptyset \vDash \varphi \Leftrightarrow (\forall\, h : V \to \mathcal{L}_2)\,(\underbrace{h \vDash \emptyset}_{\text{adevărat}} \Rightarrow h \vDash \varphi) \Leftrightarrow (\forall\, h : V \to \mathcal{L}_2)\,(h \vDash \varphi) \Leftrightarrow \vDash \varphi.$$

Observație (valorile de adevăr atribuite enunțurilor de o interpretare se calculează pe baza valorilor de adevăr atribuite variabilelor propoziționale de acea interpretare, folosind proprietatea interpretării de a transforma conectorii logici în operații booleene)

Valoarea unei interpretări într–un anumit enunț, uneori numită *interpretarea acelui enunț*, este valoarea de adevăr 0 sau 1 care se obține atunci când se atribuie, prin acea interpretare, valori de adevăr din $\mathcal{L}_2$ tuturor variabilelor propoziționale care apar în acel enunț. Un enunț universal adevărat, i. e. un adevăr semantic, o tautologie, este un enunț a cărui valoare de adevăr este 1 pentru orice valori de adevăr atribuite variabilelor propoziționale care apar în acel enunț.

În cele ce urmează, vom vedea două rezultate deosebit de importante privind sistemul formal $\mathcal{L}$: **Teorema de completitudine** și o generalizare a ei, **Teorema de completitudine tare**, numită și **Teorema de completitudine extinsă**. **Teorema de completitudine a lui** $\mathcal{L}$ afirmă că adevărurile sintactice ale lui $\mathcal{L}$ coincid cu adevărurile semantice ale lui $\mathcal{L}$, i. e. teoremele formale ale lui $\mathcal{L}$ sunt exact enunțurile universal adevărate, tautologiile lui $\mathcal{L}$. **Teorema de completitudine tare pentru** $\mathcal{L}$ afirmă că, în $\mathcal{L}$, consecințele sintactice ale unei mulțimi Σ de enunțuri coincid cu consecințele semantice ale lui Σ, i. e. enunțurile care se deduc sintactic din Σ sunt exact enunțurile care se deduc semantic din Σ.

Teoremă (Teorema de completitudine tare (extinsă) pentru $\mathcal{L}$ (TCT))

Pentru orice enunț φ și orice mulțime de enunțuri Σ:

$$\Sigma \vdash \varphi \quad \textit{ddacă} \quad \Sigma \vDash \varphi.$$

Teoremă (Teorema de completitudine pentru $\mathcal{L}$ (TC))

Pentru orice enunț φ,

$$\vdash \varphi \quad \textit{ddacă} \quad \vDash \varphi.$$

Demonstrație: Se aplică **Teorema de completitudine tare** pentru $\Sigma = \emptyset$.

Notă

Uneori,

- implicația $\vdash \varphi \Rightarrow \vDash \varphi$ este numită *corectitudinea lui $\mathcal{L}$*,
- iar implicația $\vdash \varphi \Leftarrow \vDash \varphi$ este numită *completitudinea lui $\mathcal{L}$*.

Dar, cel mai adesea, **echivalența** din teorema anterioară este numită *completitudinea lui $\mathcal{L}$*.

Corolar (noncontradicția lui $\mathcal{L}$ (principiul noncontradicției))

Niciun enunț φ nu satisface și $\vdash \varphi$, și $\vdash \neg \varphi$.

Propoziție

Algebra Lindenbaum–Tarski a logicii propoziționale clasice, $E/_\sim$, este netrivială.

Propoziție

Pentru orice interpretare $h : V \to \mathcal{L}_2$ și orice $\Sigma \subseteq E$ a. î. $h \vDash \Sigma$, există un unic morfism boolean $\dddot{h}^{\Sigma} : E/_{\sim_\Sigma} \to \mathcal{L}_2$ care face următoarea diagramă comutativă, anume morfismul boolean definit prin: pentru orice $\varphi \in E$, $\dddot{h}^{\Sigma}(\hat{\varphi}^{\Sigma}) := \tilde{h}(\varphi)$:

$$\begin{tikzcd} V \arrow[r, hook] \arrow[dr, "h"'] & E \arrow[r, "p_\Sigma"] \arrow[d, "\tilde{h}"] & E/_{\sim_\Sigma} \arrow[dl, "\dddot{h}^{\Sigma}"] \\ & \mathcal{L}_2 & \end{tikzcd}$$

Corolar (din propoziția precedentă pentru $\Sigma = \emptyset$)

Pentru orice interpretare $h : V \to \mathcal{L}_2$, există un unic morfism boolean $\dddot{h} : E/_\sim \to \mathcal{L}_2$ care face următoarea diagramă comutativă, anume morfismul boolean definit prin: pentru orice $\varphi \in E$, $\dddot{h}(\hat{\varphi}) := \tilde{h}(\varphi)$:

$$\begin{tikzcd} V \arrow[r, hook] \arrow[dr, "h"'] & E \arrow[r, "p"] \arrow[d, "\tilde{h}"] & E/_\sim \arrow[dl, "\dddot{h}"] \\ & \mathcal{L}_2 & \end{tikzcd}$$

Definiție

O mulțime Σ de enunțuri se numește *sistem deductiv* ddacă este închisă la deducții, i. e., pentru orice $\varphi \in E$, are loc:

$$\Sigma \vdash \varphi \quad \Rightarrow \quad \varphi \in \Sigma, \qquad \text{adică:}$$

$$\{\varphi \in E \mid \Sigma \vdash \varphi\} \subseteq \Sigma.$$

Exemplu

Mulțimea E a tuturor enunțurilor este sistem deductiv.

Propoziție

Mulțimea teoremelor formale este cel mai mic sistem deductiv (în raport cu incluziunea, i.e. în posetul $(\mathcal{P}(E), \subseteq)$).

Spunem că o mulțime Σ de enunțuri este *închisă la* **modus ponens** ddacă, pentru orice enunțuri φ, ψ, dacă $\psi, \psi \to \varphi \in \Sigma$, atunci $\varphi \in \Sigma$.

Propoziție (caracterizare pentru sistemele deductive)

Pentru orice $\Sigma \subseteq E$, sunt echivalente:

1. Σ *este sistem deductiv;*
2. Σ *include mulțimea axiomelor și este închisă la* **modus ponens***;*
3. Σ *include mulțimea teoremelor formale și este închisă la* **modus ponens***.*

Mulțimea sistemelor deductive este familie Moore

Propoziție

Intersecția oricărei familii de sisteme deductive este sistem deductiv, i. e. mulțimea sistemelor deductive este un **sistem de închidere**.

Notație

Notăm cu $D : \mathcal{P}(E) \to \mathcal{P}(E)$ operatorul de închidere asociat sistemului de închidere format din sistemele deductive.

Corolar

Pentru orice mulțime Σ de enunțuri, $D(\Sigma)$ este cel mai mic sistem deductiv care include pe Σ, anume intersecția tuturor sistemelor deductive care includ pe Σ.

Definiție

Pentru orice $\Sigma \subseteq E$, $D(\Sigma)$ se numește *sistemul deductiv generat de* Σ.

Remarcă

Conform definiției unui operator de închidere, pentru orice $\Sigma, \Delta \in \mathcal{P}(E)$:

1. $\Sigma \subseteq D(\Sigma)$ (D este **extensiv**);
2. $\Sigma \subseteq \Delta$ implică $D(\Sigma) \subseteq D(\Delta)$ (D este **crescător**);
3. $D(D(\Sigma)) = D(\Sigma)$ (D este **idempotent**).

Propoziție (sistemul deductiv generat de o mulțime Σ de enunțuri este mulțimea consecințelor sintactice ale lui Σ)

Pentru orice mulțime Σ de enunțuri:

$$D(\Sigma) = \{\varphi \in E \mid \Sigma \vdash \varphi\}.$$

Corolar

Operatorul de închidere D este finitar, *adică, oricare ar fi $\Sigma \subseteq E$, are loc:*

$$D(\Sigma) = \bigcup \{D(\Gamma) \mid \Gamma \subseteq \Sigma, |\Gamma| < \aleph_0\}.$$

Mulțimi consistente (i. e. necontradictorii)

Definiție

Fie Σ o mulțime de enunțuri.

- Σ se zice *inconsistentă* ddacă $\Sigma \vdash \varphi$ pentru orice $\varphi \in E$ (i. e. ddacă orice enunț este consecință sintactică a lui Σ);
- Σ se zice *consistentă* ddacă Σ nu este inconsistentă.

Exemplu

Mulțimea E a tuturor enunțurilor este inconsistentă.

Remarcă

Orice submulțime a unei mulțimi consistente este consistentă.
Prin urmare, orice mulțime care include o mulțime inconsistentă este inconsistentă.

Remarcă

Mulțimea T a teoremelor formale este consistentă.
Într–adevăr, conform unei propoziții de mai sus, T este sistem deductiv, deci este egală cu mulțimea enunțurilor φ cu $T \vdash \varphi$, iar $T \subsetneq E$, conform **principiului noncontradicției**.

Exemplu (consecință a celor două remarci precedente)

$\emptyset$ și Ax sunt mulțimi consistente.

Propoziție

Sistemul deductiv generat de o mulțime consistentă este o mulțime consistentă.

Propoziție (mulțimile consistente sunt mulțimile de enunțuri din care nu se deduc contradicții)

Pentru orice $\Sigma \subseteq E$*, sunt echivalente:*

1. Σ *este inconsistentă;*
2. *există* $\varphi \in E$*, astfel încât* $\Sigma \vdash \varphi \wedge \neg \varphi$*;*
3. *există* $\varphi \in E$*, astfel încât* $\Sigma \vdash \varphi$ *și* $\Sigma \vdash \neg \varphi$*;*
4. *există* $\varphi \in E$*, astfel încât* $\Sigma \vdash \neg (\varphi \rightarrow \varphi)$*;*
5. *pentru orice* $\varphi \in E$*,* $\Sigma \vdash \neg (\varphi \rightarrow \varphi)$*.*

Corolar

Fie $\Sigma \subseteq E$ *și* $\varphi \in E$*. Atunci:*

1. $\Sigma \cup \{\varphi\}$ *este inconsistentă ddacă* $\Sigma \vdash \neg \varphi$*;*
2. $\Sigma \cup \{\neg \varphi\}$ *este inconsistentă ddacă* $\Sigma \vdash \varphi$*.*

Propoziție

Pentru orice $\Sigma \subseteq E$*, are loc echivalența:* Σ *e consistentă ddacă algebra Boole* $E/_{\sim_\Sigma}$ *e netrivială.*

Definiție (mulțimi consistente maximale)

Un element maximal al mulțimii mulțimilor consistente raportat la incluziune se numește *mulțime consistentă maximală*.

Propoziție

Orice mulțime consistentă este inclusă într–o mulțime consistentă maximală.

Corolar

Există mulțimi consistente maximale.

Demonstrație: Aplicăm propoziția anterioară, de exemplu, pentru mulțimea consistentă $\emptyset$.

Propoziție

Dacă Σ este o mulțime consistentă maximală, atunci:

1. *Σ este sistem deductiv;*
2. *pentru orice $\varphi, \psi \in E$, avem: $\varphi \vee \psi \in \Sigma$ ddacă $\begin{cases} \varphi \in \Sigma \text{ sau} \\ \psi \in \Sigma; \end{cases}$*
3. *oricare ar fi $\varphi \in E$, are loc: $\varphi \in \Sigma$ ddacă $\neg\varphi \notin \Sigma$;*
4. *pentru orice $\varphi, \psi \in E$: $\varphi \rightarrow \psi \in \Sigma$ ddacă $\neg\varphi \vee \psi \in \Sigma$ ddacă $\begin{cases} \neg\varphi \in \Sigma \text{ sau} \\ \psi \in \Sigma. \end{cases}$*

Mulțimile consistente sunt exact mulțimile satisfiabile

Remarcă

T (și, așadar, orice submulțime a lui T) admite ca model orice interpretare. Într–adevăr, pentru orice $\varphi \in T$ și orice interpretare h, avem $\vdash \varphi$, așadar $\vDash \varphi$ conform (TC), prin urmare $\tilde{h}(\varphi) = 1$.

Propoziție

O mulțime de enunțuri e satisfiabilă ddacă e consistentă.

Notă

A se vedea, în cartea de G. Georgescu și A. Iorgulescu din bibliografia din breviarul cursului de logică matematică și computațională, cum se obține **Teorema de completitudine tare** din **Teorema de completitudine**, folosind faptul că orice mulțime consistentă e satisfiabilă.

Rezoluția în calculul propozițional clasic

Pentru această secțiune a cursului, precum și pentru rezoluția în calculul cu predicate, care stă la baza implementării limbajului Prolog, se poate consulta cartea următoare:

G. Metakides, A. Nerode, *Principles of Logic and Logic Programming*

- traducere de A. Florea, B. Boldur: *Principii de Logică și Programare Logică*, Editura Tehnică, București, 1998.

Definiție (FNC și FND)

- Un *literal* este o variabilă propozițională sau negația unei variabile propoziționale:

$$p \text{ sau } \neg\, p, \text{ cu } p \in V.$$

- O *clauză* este o disjuncție de literali.

Orice clauză se identifică cu mulțimea literalilor care o compun.

- Un enunț $\varphi(\in E)$ *este în formă normală conjunctivă* (sau *este o formă normală conjunctivă*) (*FNC*) ddacă φ este o conjuncție de clauze, i. e. o conjuncție de disjuncții de literali.

Orice FNC se identifică cu mulțimea clauzelor care o compun.

- Un enunț $\varphi(\in E)$ *este în formă normală disjunctivă* (sau *este o formă normală disjunctivă*) (*FND*) ddacă φ este o disjuncție de conjuncții de literali.

Observație

Întrucât toate enunțurile au lungime finită (i. e. sunt șiruri finite de simboluri primitive), conjuncțiile și disjuncțiile la care face referire definiția de mai sus sunt finite.

Relația de **echivalență semantică**: $\sim = \sim_{\emptyset} \in \mathrm{Eq}(E)$: relația de echivalență pe E care servește la construirea algebrei Lindenbaum–Tarski a logicii propoziționale clasice: pentru orice $\varphi, \psi \in E$,

$$\varphi \sim \psi \text{ ddacă } \vdash \varphi \leftrightarrow \psi.$$

Folosind definiția lui $\sim$, **Teorema de completitudine**, definiția adevărurilor semantice, compatibilitatea oricărei interpretări cu conectorii logici și o proprietate valabilă în orice algebră Boole, obținem:

Remarcă (două enunțuri sunt echivalente semantic ddacă au aceeași valoare în orice interpretare)

Oricare ar fi $\varphi, \psi \in E$, au loc următoarele echivalențe:
$\varphi \sim \psi \Leftrightarrow \vdash \varphi \leftrightarrow \psi \Leftrightarrow \vDash \varphi \leftrightarrow \psi \Leftrightarrow (\forall\, h : V \to L_2)\,(\tilde{h}(\varphi \leftrightarrow \psi) = 1) \Leftrightarrow (\forall\, h : V \to L_2)\,(\tilde{h}(\varphi) \leftrightarrow \tilde{h}(\psi) = 1) \Leftrightarrow (\forall\, h : V \to L_2)\,(\tilde{h}(\varphi) = \tilde{h}(\psi))$.

Așadar:

Remarcă

Dacă $\varphi, \psi \in E$ astfel încât $\varphi \sim \psi$, atunci: φ e satisfiabil ddacă ψ e satisfiabil.

Punerea unui enunț în FNC (sau FND)

Propoziție (existența FNC și FND pentru orice enunț)

Oricare ar fi $\varphi \in E$*, există o FNC* $\psi \in E$ *și o FND* $\chi \in E$ *(care nu sunt unice), astfel încât* $\varphi \sim \psi \sim \chi$*.*

Remarcă

Oricare ar fi $\varphi \in E$, putem determina o FNC (sau FND) $\psi \in E$ cu $\varphi \sim \gamma$, folosind un tabel semantic pentru φ, sau folosind următoarele proprietăți imediate, valabile pentru orice $\alpha, \beta, \gamma \in E$:

- **înlocuirea implicațiilor și echivalențelor:**

$$\alpha \to \beta \sim \neg\,\alpha \vee \beta \text{ și } \alpha \leftrightarrow \beta \sim (\neg\,\alpha \vee \beta) \wedge (\neg\,\beta \vee \alpha)$$

- **idempotența lui** $\vee$ **și** $\wedge$**:**

$$\alpha \vee \alpha \sim \alpha \wedge \alpha \sim \alpha$$

- **comutativitatea lui** $\vee$ **și** $\wedge$**:**

$$\alpha \vee \beta \sim \beta \vee \alpha \text{ și } \alpha \wedge \beta \sim \beta \wedge \alpha$$

Remarcă (continuare)

- **asociativitatea lui $\vee$ și $\wedge$:**

$$(\alpha \vee \beta) \vee \gamma \sim \alpha \vee (\beta \vee \gamma) \text{ și } (\alpha \wedge \beta) \wedge \gamma \sim \alpha \wedge (\beta \wedge \gamma)$$

- **principiul dublei negații:**

$$\neg\neg\alpha \sim \alpha$$

- **legile lui de Morgan:**

$$\neg(\alpha \vee \beta) \sim \neg\alpha \wedge \neg\beta \text{ și } \neg(\alpha \wedge \beta) \sim \neg\alpha \vee \neg\beta$$

- **absorbția:**

$$\alpha \vee (\alpha \wedge \beta) \sim \alpha \wedge (\alpha \vee \beta) \sim \alpha$$

- **legile de distributivitate:**

$$\alpha \vee (\beta \wedge \gamma) \sim (\alpha \vee \beta) \wedge (\alpha \vee \gamma) \text{ și } \alpha \wedge (\beta \vee \gamma) \sim (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$

- **proprietățile:**

$$\alpha \vee (\beta \wedge \neg\beta) \sim \alpha \vee (\beta \wedge \neg\beta \wedge \gamma) \sim \alpha \wedge (\beta \vee \neg\beta) \sim \alpha \wedge (\beta \vee \neg\beta \vee \gamma) \sim \alpha$$

Observație (echivalența semantică ($\sim$) versus egalitatea de enunțuri)

Fie $\varphi \in E$. Atunci $\varphi \sim \neg\neg\varphi$, dar $\varphi \neq \neg\neg\varphi$. De exemplu, enunțurile *"Plouă."* și *"Nu e adevărat că nu plouă."* sunt echivalente semantic (adică au același sens, același înțeles), dar nu coincid (sunt fraze diferite, nici nu sunt compuse din același număr de cuvinte).

- Următoarea notație este definită, recursiv, pe întreaga mulțime E:

Notație (mulțimea variabilelor propoziționale care apar într–un enunț φ se notează $V(\varphi)$)

Pentru orice $p \in V$ și orice $\varphi, \psi \in E$, notăm:

1. $V(p) = \{p\}$
2. $V(\neg\varphi) = V(\varphi)$
3. $V(\varphi \rightarrow \psi) = V(\varphi) \cup V(\psi)$

- Amintesc că, într–un tabel semantic pentru un enunț φ, ne interesează variabilele propoziționale care apar în φ, adică elementele lui $V(\varphi)$.

A se vedea, la seminar, metoda prin care un enunț φ poate fi pus în FNC folosind un tabel semantic pentru φ.

Definiție (formă clauzală)

Fie $\varphi \in E$ și $M \subseteq E$, astfel încât M este finită.

- O *formă clauzală* pentru φ este o FNC (i. e. o mulțime de clauze) ψ cu $\psi \sim \varphi$.
- O *formă clauzală* pentru M este o reuniune de forme clauzale pentru elementele lui M.

Remarcă

Orice mulțime finită de enunțuri poate fi pusă într–o formă clauzală.

Propoziție

O mulțime de enunțuri e satisfiabilă ddacă orice submulțime finită a sa este satisfiabilă.

Schița demonstrației: Fie $\Gamma \subseteq E$.
"$\Rightarrow$:" Orice model pentru Γ este model pentru toate submuțimile finite ale lui Γ.
"$\Leftarrow$:" Fie $\varphi \in \Gamma$.
Ca în demonstrația propoziției următoare sau direct din faptul că Γ e satisfiabilă ddacă e consistentă și o caracterizare a mulțimilor consistente, Γ e satisfiabilă ddacă $\Gamma \setminus \{\varphi\} \nvdash \neg\varphi$ ddacă, pentru orice submulțime finită $\Lambda \subseteq \Gamma \setminus \{\varphi\}$, $\Lambda \nvdash \neg\varphi$ conform Propoziției $\star$, (2).

Deducție semantică versus satisfiabilitate

Propoziție (conform căreia tehnica rezoluției de mai jos poate fi folosită pentru a demonstra teoreme formale sau deducții formale din ipoteze)

Fie $n \in \mathbb{N}^*$, $\varphi_1, \varphi_2, \ldots, \varphi_n, \psi \in E$ și $\Gamma \subseteq E$.

(a) Atunci următoarele afirmații sunt echivalente:

1. $\{\varphi_1, \ldots, \varphi_n\} \vDash \psi$
2. $\{\varphi_1, \ldots, \varphi_n, \neg\,\psi\}$ *nu e satisfiabilă*
3. $\varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\,\psi$ *nu e satisfiabil*

(b) În particular (pentru $n = 0$), următoarele afirmații sunt echivalente:

1. $\vDash \psi$
2. $\neg\,\psi$ *nu e satisfiabil*

(c) Mai mult, următoarele afirmații sunt echivalente:

1. $\Gamma \vDash \psi$
2. $\Gamma \cup \{\neg\,\psi\}$ *nu e satisfiabilă*
3. $\neg\,\psi$ *nu e satisfiabil, sau există $k \in \mathbb{N}^*$ și $\psi_1, \ldots, \psi_k \in \Gamma$ astfel încât $\psi_1 \wedge \ldots \wedge \psi_k \wedge \neg\,\psi$ nu e satisfiabil.*

Demonstrație: Folosim definiția deducției semantice și a adevărurilor semantice, proprietatea interpretărilor de a transforma conectorii logici în operații booleene în $\mathcal{L}_2$, precum și faptul că $\mathcal{L}_2 = \{0, 1\}$, cu $0 \neq 1$.

(a) $\{\varphi_1, \dots, \varphi_n\} \vDash \psi$ ddacă, pentru orice $h : V \to \mathcal{L}_2$, $\tilde{h}(\varphi_1) = \ldots = \tilde{h}(\varphi_n) = 1 \Rightarrow \tilde{h}(\psi) = 1$ ddacă nu există $h : V \to \mathcal{L}_2$ cu $\tilde{h}(\varphi_1) = \ldots = \tilde{h}(\varphi_n) = \tilde{h}(\neg\, \psi) = 1$ ddacă $\{\varphi_1, \dots, \varphi_n, \neg\, \psi\}$ nu e satisfiabilă ddacă nu există $h : V \to \mathcal{L}_2$ cu $\tilde{h}(\varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\, \psi) = 1$ ddacă $\varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\, \psi$ nu e satisfiabil.

(b) $\vDash \psi$ ddacă, pentru orice $h : V \to \mathcal{L}_2$, $\tilde{h}(\psi) = 1$ ddacă nu există $h : V \to \mathcal{L}_2$ cu $\tilde{h}(\neg\, \psi) = 1$ ddacă $\neg\, \psi$ nu e satisfiabil.

(c) $\Gamma \vDash \psi$ ddacă, pentru orice $h : V \to \mathcal{L}_2$, $h \vDash \Gamma \Rightarrow \tilde{h}(\psi) = 1$ ddacă nu există $h : V \to \mathcal{L}_2$ cu $h \vDash \Gamma \cup \{\neg\, \psi\}$ ddacă $\Gamma \cup \{\neg\, \psi\}$ nu e satisfiabilă ddacă $\neg\, \psi$ nu e satisfiabil, sau există $k \in \mathbb{N}^*$ și $\psi_1, \dots, \psi_k \in \Gamma$ astfel încât $\{\psi_1, \dots, \psi_k, \neg\, \psi\}$ nu e satisfiabilă, conform lemei anterioare, ddacă $\neg\, \psi$ nu e satisfiabil, sau există $k \in \mathbb{N}^*$ și $\psi_1, \dots, \psi_k \in \Gamma$ astfel încât $\psi_1 \wedge \ldots \wedge \psi_k \wedge \neg\, \psi$ nu e satisfiabil, conform punctului (a).

Observație

Dacă enunțurile $\psi, \dots, \psi_k, \neg\, \psi$ sunt în FNC (o FNC pentru $\neg\, \psi$ se obține imediat dintr–o FND pentru ψ), atunci enunțul $\psi_1 \wedge \ldots \wedge \psi_k \wedge \neg\, \psi$ este în FNC.

Problema satisfiabilității

Remarcă

Formele clauzale pentru o mulțime finită de enunțuri $\{\varphi_1, \ldots, \varphi_n\}$ sunt exact formele clauzale pentru enunțul $\varphi_1 \wedge \ldots \wedge \varphi_n$.

Problemă

Fiind dat un enunț φ în FNC, să se determine dacă φ e satisfiabil.

- O soluție computațională pentru problema de mai sus este **algoritmul Davis–Putnam**, bazat pe **rezoluție**.
- **Rezoluția propozițională** poate fi privită ca o regulă de deducție pentru calculul propozițional clasic.
- Utilizând **rezoluția**, se poate construi un demonstrator automat **corect și complet** pentru calculul propozițional clasic, fără alte teoreme formale și reguli de deducție, pentru că **regula rezoluției este echivalentă cu schemele de axiome** (A_1)**,** (A_2)**,** (A_3) **plus regula MP**.
- Limbajul de programare logică PROLOG este fundamentat pe rezoluția pentru calculul cu predicate clasic (care înglobează rezoluția propozițională).

Clauze și mulțimi de clauze

Definiție (și mnemonic)

- O *clauză* este o mulțime finită de literali ($\{L_1, \dots, L_n\}$, cu $n \in \mathbb{N}$ și $L_1, \dots, L_n \in V \cup \{\neg p \mid p \in V\}$).
- **Clauza vidă** (i. e. clauza fără literali, clauza fără elemente) se notează cu $\square$ (pentru a o deosebi de **mulțimea vidă de clauze**, $\emptyset$, în cele ce urmează).
- O clauză C se zice *trivială* ddacă există $p \in V$ cu $p, \neg p \in C$.
- Orice clauză nevidă $C = \{L_1, \dots, L_n\}$ (cu $n \in \mathbb{N}^*$ și $L_1, \dots, L_n \in V \cup \{\neg p \mid p \in V\}$) se identifică cu enunțul în FND $\varphi = L_1 \vee \dots \vee L_n$. Clauza C se zice *satisfiabilă* ddacă enunțul φ e satisfiabil.
- Clauza vidă ($\square$) e considerată **nesatisfiabilă** (**justificare:** $\square$ se identifică cu $\bigvee\limits_{i \in \emptyset} L_i$; pentru orice $h : V \to L_2$, $\tilde{h}(\bigvee\limits_{i \in \emptyset} L_i) = \bigvee\limits_{i \in \emptyset} \tilde{h}(L_i) = 0 \neq 1$ în $\mathcal{L}_2$).
- Orice mulțime finită de clauze $M = \{C_1, \dots, C_k\}$ (cu $k \in \mathbb{N}$ și $C_1, \dots, C_k$ clauze) se identifică cu $C_1 \wedge \dots \wedge C_k$, deci cu o FNC.
- O mulțime finită de clauze se zice *satisfiabilă* ddacă toate clauzele din componența ei sunt satisfăcute de o aceeași interpretare (au un același model, au un model comun).

Satisfiabilitate pentru (mulțimi de) clauze

Remarcă

Sunt imediate următoarele proprietăți:

- o mulțime finită de clauze este satisfiabilă ddacă FNC asociată ei e satisfiabilă;
- $\emptyset$ (mulțimea vidă de clauze) este satisfiabilă, chiar e satisfăcută de orice interpretare;
- orice clauză trivială e satisfiabilă, chiar e satisfăcută de orice interpretare;
- orice mulțime finită de clauze triviale este satisfiabilă, chiar e satisfăcută de orice interpretare.

(Rezoluția (regulă de deducție pentru logica propozițională clasică))

Pentru orice clauze C, D, dacă $p \in V$ astfel încât $p, \neg\, p \notin C$ și $p, \neg\, p \notin D$, atunci:

$$\frac{C \cup \{p\}, D \cup \{\neg\, p\}}{C \cup D}, \text{ adică } \frac{(C \vee p) \wedge (D \vee \neg\, p)}{C \vee D}$$

Propoziție

Fie C și D clauze, iar S, T și U mulțimi finite de clauze. Atunci:

1. *dacă C e satisfiabilă și $C \subseteq D$, atunci D e satisfiabilă; prin urmare, dacă D e nesatisfiabilă și $C \subseteq D$, atunci C e nesatisfiabilă;*
2. *$C \cup D$ e satisfiabilă ddacă C e satisfiabilă sau D e satisfiabilă;*
3. *dacă $p \in V \setminus V(C)$, atunci $C \cup \{p\}$ și $C \cup \{\neg\, p\}$ sunt satisfiabile;*
4. *dacă S e nesatisfiabilă și $S \subseteq T$, atunci T e nesatisfiabilă; prin urmare, dacă T e satisfiabilă și $S \subseteq T$, atunci S e satisfiabilă;*
5. *dacă U e satisfiabilă și există $p \in V \setminus V(U)$, $G \in S$ și $H \in T$ astfel încât $p \in G \setminus H$ și $\neg\, p \in H \setminus G$, atunci $U \cup S$ și $U \cup T$ sunt satisfiabile;*
6. *dacă $p \in V$ astfel încât $p, \neg\, p \notin C$ și $p, \neg\, p \notin D$, iar mulțimea de clauze $\{C \cup \{p\}, D \cup \{\neg\, p\}\}$ e satisfiabilă, atunci $C \cup D$ e satisfiabilă (**regula rezoluției**).*

Definiție (derivări prin rezoluție)

Fie o mulțime finită de clauze $\{D_1, \ldots, D_k\}$ și φ enunțul în FNC corespunzător acestei mulțimi de clauze:

$$\varphi = D_1 \wedge \ldots \wedge D_k.$$

Dacă $i, j \in \overline{1, k}$ a. î. $i \neq j$ și există $p \in V$ cu $p \in D_i$ și $\neg\, p \in D_j$, atunci mulțimea de clauze $R := \{(D_i \setminus \{p\}) \cup (D_j \setminus \{\neg\, p\})\} \cup \{D_t \mid t \in \overline{1, k} \setminus \{i, j\}\}$ sau o FNC corespunzătoare ei, de exemplu enunțul

$$((D_i \setminus \{p\}) \vee (D_j \setminus \{\neg\, p\})) \wedge \bigwedge_{t \in \overline{1,k} \setminus \{i,j\}} D_t,$$

se numește *rezolvent* al enunțului φ sau al mulțimii de clauze $\{D_1, \ldots, D_k\}$. Deducția

$$\frac{D_1, \ldots, D_k}{R}$$

se numește *derivare prin rezoluție* a lui φ sau a mulțimii $\{D_1, \ldots, D_k\}$. Vom numi orice succesiune de derivări prin rezoluție tot *derivare prin rezoluție*. O succesiune de derivări prin rezoluție care începe cu o FNC/mulțime de clauze μ și se termină cu o FNC/mulțime de clauze ν se numește *derivare prin rezoluție a lui ν din μ*.

Notă

Rezoluția este o metodă de verificare a satisfiabilității pentru mulțimi (finite) de enunțuri în formă clauzală.

Notă

Aplicarea regulii rezoluției simultan pentru două variabile diferite este **greșită**.

Remarcă

- Dacă într–o derivare prin rezoluție a unei mulțimi finite M de enunțuri în **formă clauzală** apare $\square$, atunci M nu e satisfiabilă.
- În schimb, o derivare prin rezoluție a lui M în care nu apare $\square$ **nu arată** că M ar fi satisfiabilă.
- Pentru a arăta că o mulțime finită M de enunțuri (în formă clauzală) este satisfiabilă, putem găsi un model pentru M sau putem aplica **algoritmul Davis–Putnam**, care este echivalent cu obținerea tuturor derivărilor posibile prin rezoluție ale formei clauzale a lui M.

Algoritmul Davis–Putnam (abreviat *DP*)

INPUT: mulțime finită și nevidă S de clauze netriviale;

$S_1 := S$; $i := 1$;

PASUL 1: luăm o $v_i \in V(S_i)$;
$T_i^0 := \{C \in S_i \mid \neg\, v_i \in C\}$;
$T_i^1 := \{C \in S_i \mid v_i \in C\}$;
$T_i := T_i^0 \cup T_i^1$;

PASUL 2: dacă $T_i^0 \neq \emptyset$ și $T_i^1 \neq \emptyset$,
atunci $U_i := \{(C_0 \setminus \{\neg\, v_i\}) \cup (C_1 \setminus \{v_i\}) \mid C_0 \in T_i^0, C_1 \in T_i^1\}$;
altfel $U_i := \emptyset$;

PASUL 3: $S_{i+1} := (S_i \setminus T_i) \cup U_i$;
$S_{i+1} := S_{i+1} \setminus \{C \in S_{i+1} \mid (\exists\, p \in V)\,(p, \neg\, p \in C)\}$
(eliminăm din S_{i+1} clauzele triviale);

PASUL 4: dacă $S_{i+1} = \emptyset$,
atunci OUTPUT: S *e satisfiabilă*;
altfel, dacă $\square \in S_{i+1}$,
atunci OUTPUT: S *nu e satisfiabilă*;
altfel $i := i + 1$ și mergi la PASUL 1.

Propoziție (terminarea algoritmului DP)

Algoritmul DP se termină după cel mult $|V(S)|$ execuții ale pașilor $1-4$, cu $S_{i+1}=\emptyset$ sau $\square \in S_{i+1}$.

Demonstrație: Cu notațiile din algoritmul DP, are loc, pentru fiecare i:

$$V(S_{i+1}) \subsetneq V(S_i),$$

așadar algoritmul DP se termină după cel mult $|V(S)|$ iterații.

Propoziție

Fie S o mulțime finită și nevidă de clauze.
Dacă S e satisfiabilă, atunci orice rezolvent al lui S e satisfiabil.

Demonstrație: Fie φ enunțul în FNC corespunzător lui S și $\rho \in E$ un rezolvent al lui φ. Atunci, pentru un enunț γ în FNC, o variabilă propozițională p și două clauze C și D cu $p \notin V(C) \cup V(D)$, avem:

$$\varphi = (C \vee p) \wedge (D \vee \neg p) \wedge \gamma \quad \text{și} \quad \rho = (C \vee D) \wedge \gamma,$$

așadar, pentru orice interpretare h, avem:
$\tilde{h}(\varphi) = (\tilde{h}(C) \vee h(p)) \wedge (\tilde{h}(D) \vee \overline{h(p)}) \wedge \tilde{h}(\gamma) =$
$((\tilde{h}(C) \wedge \tilde{h}(D)) \vee (\tilde{h}(C) \wedge \overline{h(p)}) \vee (\tilde{h}(D) \wedge h(p)) \vee (h(p) \wedge \overline{h(p)})) \wedge \tilde{h}(\gamma) =$
$((\tilde{h}(C) \wedge \tilde{h}(D)) \vee (\tilde{h}(C) \wedge \overline{h(p)}) \vee (\tilde{h}(D) \wedge h(p))) \wedge \tilde{h}(\gamma) \leq ((\tilde{h}(C) \vee \tilde{h}(D)) \wedge \tilde{h}(\gamma) = \tilde{\rho},$

prin urmare orice interpretare care satisface pe φ satisface și pe ρ; în particular, dacă φ e satisfiabil, atunci ρ e satisfiabil.

Corolar

Fie S o mulțime finită și nevidă de clauze.
Dacă S e satisfiabilă, atunci algoritmul DP, aplicat lui S, se termină cu $S_{i+1} = \emptyset$.

Teoremă

Fie S o mulțime finită și nevidă de clauze. Atunci următoarele afirmații sunt echivalente:

1. *S nu e satisfiabilă;*
2. *există o derivare prin rezoluție a lui $\square$ (sau a unei mulțimi de clauze conținând pe $\square$) din S.*

Demonstrație: ②⇒①: A se observa că $\square$ se poate obține prin rezoluție numai din două clauze de forma $\{p\}, \{\neg p\}$, cu $p \in V$, și orice mulțime formată din două astfel de clauze e nesatisfiabilă.
$\square$ nu e satisfiabilă, așadar, conform propoziției anterioare, dacă există o derivare prin rezoluție a lui $\square$ din S, atunci S e nesatisfiabilă.
①⇒②: **Schița demonstrației,** după articolul:

J. Gallier, The Completeness of Propositional Resolution: a Simple and Constructive Proof, *Logical Methods in Computer Science* **2**(5:3) (2006), 1–7.

Presupunem că S e nesatisfiabilă.
Pentru orice clauză D, notăm $c(D) := |D| - 1$ (numărul de literali din compoziția lui D minus o unitate). Pentru orice mulțime finită și nevidă de clauze $M = \{D_1, \ldots, D_k\}$, notăm $c(M) := c(D_1) + \ldots + c(D_k)$.
Procedăm prin inducție după $c(S)$.
Pasul de verificare: Dacă $c(S) = 0$, atunci $c(D) = 0$ pentru fiecare clauză D a lui S, așadar fiecare clauză a lui S este un literal. Cum S e nesatisfiabilă, există $p \in V$ a. î. p și $\neg p$ sunt clauze ale lui S, prin urmare din S se poate deriva prin rezoluție o mulțime de clauze care conține pe $\square$.
Pasul de inducție: Presupunem că $c(S) > 0$ și că orice mulțime M de clauze cu $c(M) < c(S)$ are o derivare prin rezoluție care se termină cu o mulțime conținând pe $\square$.
Cum $c(S) > 0$, există o clauză D a lui S cu $c(D) > 0$, prin urmare $D = C \vee L$ pentru un literal L și o clauză nevidă C în care nu apare variabila propozițională din L, așa că avem $c(C) < c(D)$.
Fie φ enunțul în FNC corespunzător lui S, care e nesatisfiabil conform ipotezei acestei implicații. Atunci, pentru un enunț γ în FNC, avem:

$$\varphi \sim (C \vee L) \wedge \gamma \sim (C \wedge \gamma) \vee (L \wedge \gamma),$$

prin urmare enunțurile $C \wedge \gamma$ și $L \wedge \gamma$ sunt nesatisfiabile. Observăm că au loc: $c(C \wedge \gamma) < c(\varphi)$ și $c(L \wedge \gamma) < c(\varphi)$, așadar, conform ipotezei de induție, fiecare dintre enunțurile $C \wedge \gamma$ și $L \wedge \gamma$ admite o derivare prin rezoluție în care apare $\square$.

Să considerăm o astfel de derivare pentru $C \wedge \gamma$ și să înlocuim pe C cu $C \vee L = D$, transformând enunțul $C \wedge \gamma$ în $(C \vee L) \wedge \gamma \sim \varphi$, și modificând corespunzător această derivare, astfel că, la finalul ei:

- fie apare tot $\square$, caz în care am obținut deja o derivare a lui φ care conduce la $\square$,
- fie apare clauza L în locul lui $\square$, caz în care procedăm astfel:

ținând seama de faptul că $\varphi \sim D \wedge \gamma \sim D \wedge \gamma \wedge \gamma$, la fiecare pas al derivării prin rezoluție a lui φ obținute prin modificarea de mai sus, la mulțimea curentă de clauze adăugăm o mulțime de clauze corespunzând lui γ, astfel că la finalul acestei noi derivări vom avea o mulțime de clauze corespunzătoare enunțului $L \wedge \gamma$, din care, conform celor de mai sus, se poate deriva prin rezoluție o mulțime de clauze conținând pe $\square$. Punând "cap la cap" aceste derivări, obținem o derivare prin rezoluție din φ (echivalent, din S) a unei mulțimi de clauze conținând pe $\square$.

Corolar (Teorema Davis–Putnam)

Algoritmul DP este corect și complet.

Notație

Dacă $\Gamma \subseteq E$ și $\varphi \in E$, atunci notăm cu $\Gamma \vdash_R \varphi$ faptul că există o derivare prin rezoluție a lui $\square$ dintr–o formă clauzală a lui $\Gamma \cup \{\neg \varphi\}$.

Rezoluția propozițională $\Longleftrightarrow$ sistemul Hilbert

Corolar

Pentru orice $\Gamma \subseteq E$ *și orice* $\varphi \in E$, *următoarele afirmații sunt echivalente:*

1. $\Gamma \vDash \varphi$;
2. $\Gamma \vdash_R \varphi$.

Remarcă

Conform corolarului anterior și (TCT), **regula rezoluției** este corectă și completă pentru calculul propozițional clasic, adică deducția pe baza **regulii rezoluției** este echivalentă cu deducția pe baza axiomelor (A_1), (A_2), (A_3) și a regulii de deducție **MP**, i.e. cu **sistemul Hilbert**.

Așadar folosind **regula rezoluției** putem realiza o prezentare echivalentă pentru logica propozițională clasică.