

Capitolul 4

Teorii logice și sisteme deductive

Această **parte a logicii formale**, alcătuită din *teoriile logice și sistemele deductive (de demonstrație, inferențiale)* este una dintre cele mai răspândite modalități de exprimare exactă a noilor descoperiri utilizată de către comunitatea științifică. Trecând de la *un studiu naiv al realității la un studiu semi-fundamentat*, **este absolut necesar ca obiectele, conceptele, relațiile manipulate să admită definiții precise, ca anumite proprietăți să fie demonstrate într-un cadru clar specificat, ca totul să poată fi făcut constructiv.** Într-un limbaj mai mult sau mai puțin apropiat de limbajul natural, **orice știință folosește afirmații reprezentate prin formule, un concept de adevăr asociat acestora, un mediu de natură sintactică, organizat, pentru demonstrarea adevărului unei formule.** În logică, *sistemele deductive oferă mediul de demonstrare „mecanică” iar teoriile logice posibilitatea defnirii adevărului la un nivel global.* În plus, între aceste *meta-concepte* există o legătură clar subliniată. Ele sunt utilizate, așa după cum tocmai am amintit, atât în procesul de formalizare a (conținutului) altor științe dar și în studiile logice de bază, ceea ce *poate genera din nou dificultăți de înțelegere provenite din dualitatea limbaj de bază – metalimbaj.*

Să trecem întâi în revistă conceptele de sistem deductiv și teorie logică precum și legătura dintre ele, la un *nivel informal*. După cum deja cunoaștem, indiferent de tipul de logică (**LP**, **LP1**, **LP1₊**, **LP2**,

etc.), din punct de vedere *sintactic* se pornește cu un *alfabet* și se construiesc *formule* peste acel alfabet. Sunt apoi identificate *subclase* „importante”, cum ar fi *clasa formulelor Horn* sau *clasa formulelor aflate în FNSC*. O asemenea subclasă de formule, dar și de obiecte mai complexe (pe care le vom numi *metaformule*), poate fi *descrisă finitar* cu ajutorul unui *sistem deductiv*. Un sistem deductiv se bazează concret pe o definiție constructivă, care apelează la noțiunile de *axiomă* și de *regulă de inferență*. Axiomele sunt formulele plasate inițial (prin **Baza** definiției) în subclasa corespunzătoare, iar regulile de inferență (de deducție, de demonstrație) reprezintă modalitățile prin care se obțin formule noi (numite și *teoreme*) din formule vechi (**Pasul inductiv**). *Cu ajutorul acestora se definește în mod formal, la nivel global, conceptul de raționament (demonstrație)*. Un exemplu imediat de sistem deductiv este cel bazat pe rezoluție. Astfel, în **LP** pornim cu o mulțime de clauze F (formulă aflată în **FNC**, generatoare de axiome) și putem găsi $\text{Res}^*(F)$ utilizând rezoluția într-un pas ca (unică *schemă* de) regulă de inferență (orice element din $\text{Res}^*(F)$ este „demonstrabil prin rezoluție” pornind cu „axiomele” din F). O *teorie logică* este o (sub)clasă de formule *închisă la consecință semantică*. Cu alte cuvinte, o mulțime T de formule este teorie logică dacă pentru fiecare submulțime $S \subseteq T$ și fiecare (altă) formulă G care este consecință semantică din S , avem și $G \in T$. Exemple imediate de teorii logice sunt constituite din clasele formulelor valide (din **LP**, **LP1**, **LP1_≡**, etc.). *Cu ajutorul acestora se definește formal, la nivel global, conceptul de „adevăr”* (elemente lui T de mai înainte fiind receptate ca niște formule „adevărate”). *Legătura* dintre teoriile logice și sistemele de demonstrație se exprimă prin

teoreme de corectitudine și completitudine, adică teoreme de tipul: *Tot ceea ce este „adevărat” este demonstrabil (completitudine) și tot ceea ce este demonstrabil este „adevărat” (corectitudine)*. Mai precis, se poate porni cu o teorie logică și se poate încerca „axiomatizarea” ei (adică găsirea unui sistem deductiv prin care se „generează”, sintactic, aceeași clasă de formule), cu scopul ca mulțimea teoremelor să coincidă cu mulțimea formulelor „adevărate” (cele care formează teoria). Se poate și invers, adică putem pleca cu un sistem deductiv pentru care putem afla „imediat” clasa teoremelor (clasa formulelor generate prin raționamente specifice regulilor sale de inferență) și punându-se apoi problema ca această clasă să formeze o teorie logică (sau măcar o „parte” a unei asemenea teorii), constituită din formule „adevărate” („valide”, „satisfiabile”, „nesatisfiabile”, etc.).

În realitate lucrurile nu stau chiar așa de simplu cum au fost prezentate mai sus (din cauza lipsei de spațiu tratarea noastră fiind departe de a fi exhaustivă). Astfel, subiectul general al teoriilor logice este foarte vast, vorbindu-se de *teorii degenerate*, de *teorii inconsistente*, de *teorii recursive și/sau recursiv enumerabile*, etc. Mai mult, orice teorie logică (în sens clasic) trebuie să conțină toate formulele valide (sau doar acele formule valide care au o formă sintactică specificată), pentru că *orice formulă validă este consecință semantică din orice altă clasă de formule*. Dacă o teorie logică conține o contradicție, atunci acea teorie coincide cu întreaga clasă precizată de formule (**LP**, **LP1**, **LP1₌**, etc.), lucru care rezultă direct din definiții. Nici în cazul sistemelor deductive nu dispunem de un context ușor de manipulat. Utilizarea unui *număr finit* de (*scheme de*) axiome și reguli

de inferență pentru „prezentarea” unor teorii logice poate fi imposibilă, existând și un număr impresionant de *tipuri generale* de asemenea sisteme. De asemenea, forma în care sunt exprimate teoremele de corectitudine și completitudine poate depinde în mod esențial de sistemul deductiv sau de teoria aleasă (ba uneori chiar și de alfabetul peste care este construită mulțimea de formule). Să reținem faptul important că sistemele deductive pot fi utilizate „pentru ele însele”, ca mecanisme formale (simple *alternative pentru definițiile constructive*). Cu ajutorul lor se pot astfel defini sintactic mulțimi care nu au întotdeauna o caracterizare semantică legată de o noțiune precisă de „adevăr”. Paragrafele următoare, în care detaliem câteva dintre aspectele menționate, adoptă în mare linia sugerată de [CAZ1], concepția generală fiind însă originală. De altfel, definițiile *noțiunilor de sistem de demonstrație și teorie logică* diferă ușor față de cele întâlnite în literatura de specialitate, datorită atât bagajului insuficient de cunoștințe presupus a fi fost acumulat anterior parcurgerii acestui material de către cititor, cât și datorită nevoii autorului de sistematizare și generalizare a conținutului.

§1. Sisteme deductive

Vom nota cu numele generic **FORM** clasa de *metaformule* în care ne vom plasa. Termenul ales se explică prin faptul că o *metaformulă* va denota nu numai un element din **LP**, **LP1**, **LP1₌**, sau dintr-o subclasă fixată a acestora, ci și o listă de asemenea elemente, sau, ieșind din sfera logicii, obiecte cu mult mai complicate decât

formulele „clasice”. Mulțimea **FORM** va fi precizată în mod explicit doar atunci când va fi necesar, singura cerință permanentă fiind ca aceasta să fie definită constructiv.

Definiția 4.1 (sistem deductiv). Se numește *sistem deductiv (de demonstrație, inferențial, axiomatic)* în **FORM** un cuplu $SD = \langle A, R \rangle$ unde $A \subseteq \mathbf{FORM}$ este o mulțime de **axiome** iar $R \subseteq \mathbf{FORM}^+ \times \mathcal{C}$ o mulțime de **reguli de inferență (de deducție, de demonstrație)**. ■

În cele de mai sus, \mathbf{FORM}^+ denotă mulțimea relațiilor de oricâte argumente (cel puțin unul) peste **FORM**, iar \mathcal{C} reprezintă o mulțime de *condiții de aplicabilitate*. Fiecare regulă de inferență $r \in R$, are astfel aspectul $r = \langle \langle G_1, G_2, \dots, G_n, G \rangle, c \rangle$, unde $n \in \mathbf{N}$, $G_1, G_2, \dots, G_n \in \mathbf{FORM}$ și $c \in \mathcal{C}$. G_1, G_2, \dots, G_n sunt *ipotezele (premizele) regulii*, G reprezintă *concluzia (consecința)* iar c desemnează *cazurile (modalitățile) în care regula poate fi aplicată*. Vom scrie chiar $r = \langle \langle \{G_1, G_2, \dots, G_n\}, G \rangle, c \rangle$ deoarece *ordinea ipotezelor nu este esențială*. Mulțimea \mathcal{C} nu a fost specificată formal (putem spune totuși că elementele sale sunt **metapredicate**) din cauza generalității ei și pentru a nu complica inutil expunerea. Similar cu situația rezoluției, *regulile vor fi folosite pentru a construi demonstrații în pași succesivi, la un pas aplicându-se o regulă*. Există însă posibilitatea ca înafara restricțiilor sintactice „locale”, date de forma formulelor implicate (ceea ce face ca regulile să fie, de obicei, *scheme*

de reguli) să se interzică aplicarea regulii (schemei) pe considerente semantice „globale” (forma demonstrației, apariția în demonstrație a unei formule nedorite la acel pas, păstrarea completitudinii unei teorii, etc.). Astfel încât dacă **C** este atașată unei reguli *r* (*atenție, C poate lipsi, mai exact ea poate fi „condiția adevărată indiferent de context”*) înseamnă că în **orice** demonstrație, *r* va putea fi aplicată la un moment dat **doar dacă** **C** este adevărată la momentul respectiv. O regulă $r = \langle \langle \{G_1, G_2, \dots, G_n\}, G \rangle, c \rangle$, va fi scrisă și ca:

$$\frac{G_1, G_2, \dots, G_n}{G}, c$$

În cazul în care $n = 0$ și **C** lipsește, *r* poate fi identificată ca fiind o axiomă, după cum rezultă din definiția care urmează. Câteodată, alături de **C**, sunt explicitate separat și restricțiile sintactice locale asupra (formei) metaformulelor.

Definiția 4.2 (demonstrație). Fie $SD = \langle A, R \rangle$ un sistem deductiv în **FORM**. Se numește *demonstrație (pentru F_m , pornind cu F_1) în SD* o listă de metaformule $D = F_1, F_2, \dots, F_m$ astfel încât pentru fiecare $i \in [m]$, fie $F_i \in A$, fie F_i este obținut din $F_{j_1}, F_{j_2}, \dots, F_{j_k}$ folosind o regulă $r = \langle \langle \{F_{j_1}, F_{j_2}, \dots, F_{j_k}\}, F_i \rangle, c \rangle \in R$, unde $j_1, j_2, \dots, j_k < i$. ■

Prin urmare, fiecare element al listei *D* este fie o axiomă, fie este concluzia unei reguli de inferență ale cărei ipoteze sunt elemente anterioare din listă. Analogia cu cele deja fixate în **Capitolul 2** despre demonstrațiile prin rezoluție (**Definiția 2.10** este un caz particular al

definiției precedente) este clară și aceasta poate fi continuată prin definirea *numărului de pași ai unei demonstrații, reprezentarea unei demonstrații printr-un arbore*, etc. O demonstrație se va mai numi și **deducție (sintactică)** sau chiar **raționament**. După cum am precizat, **putem defini D de mai sus, constructiv, ca fiind un arbore cu rădăcina F_m , în care frunzele (Baza) sunt axiome și fiecare nod nou (concluzie) se obține din noduri vechi (ipoteze), în timpul aplicării Pasului inductiv**, folosindu-se câte o regulă de inferență posibil de a fi aplicată (conform **C**). Este posibil ca pentru anumite sisteme (a se vedea sistemul **SD0**, §3 din acest capitol), care nu au axiome sau pentru care condițiile **C** o impun, arborele să aibă o definiție directă mai simplă, sau, alternativ, *noțiunea de consecință sintactică să fie definită cu ajutorul arborelui (necoincizând cu el)*.

Definiția 4.3 (teoreme). Fie $SD = \langle A, R \rangle$ un sistem deductiv în **FORM**. **Mulțimea teoremelor lui SD** este mulțimea metaformulelor care admit demonstrații în SD , adică:

$$Th(SD) = \{F \in \mathbf{FORM} \mid \text{există o demonstrație } D \text{ pentru } F \text{ în } SD\}. \blacksquare$$

Este imediat faptul că și $Th(SD)$ admite o definiție constructivă (similar cu Teorema 2.11, din cazul rezoluției, se arată că $Th(SD)$ coincide cu mulțimea dată mai jos):

Baza. $A \subseteq Th(SD)$.

Pas constructiv. Dacă $r = \langle \langle \{G_1, G_2, \dots, G_n\}, G \rangle, c \rangle \hat{I} R$ și $G_1, G_2, \dots, G_n \hat{I} Th(SD)$ atunci $G \hat{I} Th(SD)$.

Faptul că există o demonstrație pentru F în SD va mai fi notat prin $\bullet_{SD} F$ (renunțând și la indice în cazul în care nu există confuzii). Din motive tehnice, legate în general de demonstrațiile teoremelor de corectitudine și completitudine, este posibil ca în anumite situații să lucrăm, pe lângă axiome, cu o mulțime suplimentară de metaformule, notată I , și să vorbim despre **demonstrații folosind I** (notat $I \bullet_{SD} F$, în cazul în care este vorba despre o formulă F ; se mai spune că **F este consecință sintactică din I** sau **teoremă în ipotezele I**). Practic, *deși din punctul de vedere al noțiunii de „adevăr”, sensul semantic al celor două mulțimi (A și I) nu coincide întotdeauna*, folosirea mulțimii de ipoteze suplimentare I nu înseamnă altceva decât să lucrăm exact ca mai înainte, dar în sistemul $SD' = \langle A', R \rangle$, unde $A' = A \cup I$. Teoremele vor apare astfel ca fiind consecințe sintactice din mulțimea vidă de ipoteze.

Pentru că exemplele care urmează sunt suficient de individualizate, deși au substrat comun, le vom numerota în același mod ca pe definiții (de altfel, unele vor fi reluate ulterior).

Exemplul 4.1 (metoda rezoluției). Metoda rezoluției, atât în cazul **LP** cât și în cazul **LP1**, poate fi privită ca reprezentând un sistem de demonstrație. Vom trata doar cazul **LP** și trebuie spus că vom construi

*sisteme de demonstrație dedicate, câte unul pentru fiecare formulă F aflată în FNC și reprezentată ca o mulțime finită de clauze (conform teoremei de compactitate, F poate fi și infinită). Sistemele dedicate pot fi caracterizate de faptul că nu generează, în general, clasa formulelor valide (nici măcar o submulțime a acesteia), de aceea teoremele de corectitudine și completitudine au și ele o formă „nestandard” (vom reveni la acest lucru în §2 din acest capitol, după tratarea unor aspecte formale legate de teoriile logice (conform **Exemplului 4.2**, reluat). Pentru a justifica această afirmație, trebuie să specificăm pe rând:*

- **Clasa metaformulelor: $FORM = LP$. Formulele care ne interesează sunt însă doar clauzele din LP.** Notăm mulțimea tuturor clauzelor din LP cu \mathcal{C} .
- **Mulțimea de axiome: $A_F = \emptyset$.** Am putea considera drept axiome chiar mulțimea F. Nu facem acest lucru din considerente conceptuale, încă netransparente. Diferența la nivel sintactic dintre axiome și ipotezele suplimentare nu există deocamdată, iar în ceea ce privește demonstrațiile vom lucra (practic) cu sisteme de tipul SD' .
- **Mulțimea de reguli de inferență:**

$$R_F = \{ \langle \{C_1, C_2\}, Res(C_1, C_2), true \rangle \mid C_1, C_2 \in \mathcal{C} \text{ și } C_1 \neq C_2 \text{ și există măcar un literal } L \text{ astfel încât } L \in C_1 \text{ și } \bar{L} \in C_2 \}.$$

În acest moment, sistemul $SD_F = \langle A_F, R_F \rangle$ este pe deplin precizat (true denotă *condiția mereu adevărată*, adică regulile, dacă respectă cerințele

sintactice fixate și anume faptul că sunt clauze din **LP**, pot fi aplicate fără restricții pe parcursul demonstrațiilor), exceptând faptul deja amintit că vom pune (renunțăm la indici și la faptul că ar trebui folosită notația SD' în loc de SD)

$$Th(SD) = \{ G \in Cl \mid F \bullet_{SD} G \},$$


adică vom considera demonstrațiile plecând de la mulțimea de axiome $A' = A \cup F$ (considerăm $I = F$). Este imediat faptul că pentru fiecare asemenea SD , avem $Th(SD) = Res^*(F)$. ■

Sistemele anterioare („de rezoluție”) au caracteristic și faptul că pot fi **finit specificate**, adică atât mulțimea de axiome (și/sau mulțimea de ipoteze suplimentare) cât și mulțimea de reguli de inferență sunt fie finite (cazul lui F), fie reprezintă un număr finit de **scheme** (cazul regulilor, lucru evident din specificarea mulțimii respective). Clasa regulilor poate fi însă descrisă finitar și astfel:

Pentru fiecare $C_1, C_2 \in Cl$, dacă $C_1 \neq C_2$ și există măcar un literal L astfel încât $L \in C_1$ și $\bar{L} \in C_2$, atunci:

$$r: \frac{C_1, C_2}{Res(C_1, C_2)}.$$

În acest mod, spunem că pentru descrierea regulilor *am folosit o singură schemă*. ■

 **Exercițiul 4.1.** Arătați că metoda rezoluției de bază pentru **LP1** poate fi privită ca sistem de demonstrație.

Exemplul 4.2 (sistemul SD3). Este un sistem deductiv standard, finit specificat, care generează, după cum vom vedea (**Teorema 4.1**), întreaga clasă (și numai pe aceasta) a formulelor valide din **LP1** (sistemul a fost introdus pentru prima dată de către A. Church în 1954).

- **Axiome** (A_{SD3}). Condițiile sintactice sunt: $F, G, H \in \mathbf{LP1}$, $x \in X$, $t \in T$, oarecare. Suplimentar, în 4., x trebuie să nu apară liber în F iar în 5., substituția $s = [x/t]$ trebuie să fie permisă pentru F (reamintim, t nu conține nume de variabile care să apară legate în F):
 1. $F \rightarrow (G \rightarrow F)$.
 2. $(F \rightarrow (G \rightarrow H)) \rightarrow ((F \rightarrow G) \rightarrow (F \rightarrow H))$.
 3. $(\neg F \rightarrow \neg G) \rightarrow ((\neg F \rightarrow G) \rightarrow F)$.
 4. $(\forall x)(F \rightarrow G) \rightarrow (F \rightarrow (\forall x)G)$.
 5. $(\forall x)F \rightarrow (F)[x/t]$.

Să remarcăm faptul că **LP1** trebuie considerată ca fiind construită peste alfabetul care conține drept conectori doar pe \neg și \rightarrow , iar unicul cuantificator acceptat este \forall . Dacă dorim să utilizăm și ceilalți conectori (sau cuantori), putem face acest lucru doar utilizându-i ca notații (de exemplu, $A \vee B$ va reprezenta $\neg A \rightarrow B$, etc.).

- **Reguli de inferență** (R_{SD3}). Există doar restricții de natură sintactică (lipsind condițiile de aplicabilitate): $F, G \in \mathbf{LP1}$, $x \in X$ sunt oarecare, dar în 2., x trebuie să nu apară liber în F . Prima schemă de regulă este deja amintită, și anume

modus ponens (pe scurt, **(MP)**) iar a doua este așa-numita **regulă a generalizării (RG)**.

$$1I. \frac{F \rightarrow G, F}{G} .$$

$$2I. \frac{F}{(\forall x)F} .$$

Să arătăm, de exemplu, că în **SD3** se poate genera teorema $T = (A \rightarrow A)$ (în cele ce urmează vom mai renunța pe parcurs la unele paranteze, dacă înțelegerea nu este afectată, deși formal acest lucru nu este admis). Astfel, folosim întâi instanța axiomei (schemei) 1., obținută dacă luăm $F = A$ și $G = (A \rightarrow A)$. Primul element al listei care reprezintă demonstrația va fi: $E_1 = A \rightarrow ((A \rightarrow A) \rightarrow A)$. Folosim acum axioma 2., punând $F = A$, $G = (A \rightarrow A)$ și $H = A$. Obținem:

$$E_2 = (A \rightarrow ((A \rightarrow A) \rightarrow A)) \rightarrow ((A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A)).$$

Aplicăm acum **(MP)** pentru $E_2 = F \rightarrow G$ și $E_1 = F$ (se observă imediat că $G = (A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A)$) și găsim:

$$E_3 = (A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A).$$

Punem acum $F = A$ și $G = A$, în axioma 1., rezultând:

$$E_4 = A \rightarrow (A \rightarrow A).$$

În sfârșit, putem folosi **(MP)** pentru E_3 și E_4 (luând $F = A \rightarrow (A \rightarrow A)$ și $G = (A \rightarrow A)$), pentru a obține ceea ce doream, adică:

$$E_5 = T = (A \rightarrow A).$$

Prin urmare, am găsit în **SD3** demonstrația $D : E_1, E_2, E_3, E_4, E_5 = T$ și putem spune că $(A \rightarrow A) \in Th(\mathbf{SD3})$ sau că T este consecință sintactică din mulțimea vidă de formule suplimentare. În plus, aceasta este

evident o formulă validă. Cum și T este de fapt o schemă, rezultă și că $(\neg A \rightarrow \neg A)$ este teoremă, etc.

Profităm de exemplul în curs pentru a schița **linia generală de demonstrare a teoremelor de corectitudine** (pentru cazul standard al sistemelor care generează toate formulele valide).

I. Se arată că axiomele sunt formule valide. Conform Capitolului 2, vom folosi tabelele de adevăr pentru prima axiomă:


F	G	$G \rightarrow F$	$F \rightarrow (G \rightarrow F)$
0	0	1	1
0	1	0	1
1	0	1	1
1	1	1	1

II. Se arată că regulile de inferență sunt corecte (termenul corespunzător în engleză este *sound*, adică „sănătos”). Acest lucru înseamnă: *presupunem că ipotezele G_1, G_2, \dots, G_k sunt formule valide și arătăm că și concluzia G este formulă validă* (atenție, nu este nimic contratoriu în ceea ce spunem, chiar dacă nu este adevărat că formulele folosite în axiome sunt în toate cazurile formule valide).

Pentru aceasta este suficient să arătăm că $G_1 \cup G_2 \cup \dots \cup G_k \rightarrow G$ este validă. De exemplu, (MP) este corectă, deoarece avem:

F	G	$F \rightarrow G$	$(F \rightarrow G) \wedge F$	$((F \rightarrow G) \wedge F) \rightarrow G$
0	0	1	0	1
0	1	1	0	1
1	0	0	0	1
1	1	1	1	1

Concluzia că teoremele sunt formule valide rezultă imediat din definiția constructivă a lui $Th(\mathbf{SD3})$ (formal, este vorba de o demonstrație prin inducție structurală, de altfel foarte simplă). ■

 **Exercițiul 4.2.** Arătați că axiomele 2. – 5. sunt formule valide și că **(RG)** este o regulă corectă.

Despre sistemele deductive se pot spune multe alte lucruri, la nivel general. Astfel, o **proprietate gobală**, deseori cerută, este cea de **consistență**. Astfel, o mulțime de metaformule J este consistentă într-un sistem deductiv, dacă nu există nici o metaformulă F astfel încât să avem atât $J \bullet F$ cât și $J - F$ ($J - F$ notează faptul că nu este adevărat că $J \bullet F$). Prin extensie, un sistem deductiv este consistent (**necontradictoriu**) dacă nu există nici o metaformulă F astfel încât să avem atât $\bullet F$ cât și $- F$. O altă proprietate importantă este cea a **minimalității** (**independenței**). Astfel, în anumite situații este

important ca un sistem să conțină *cât mai puține axiome și reguli de inferență*, deși acest lucru s-ar putea să conducă la existența unor demonstrații mai lungi și mai alambicate. Înafara sensului strict de minimalitate (lucru care depinde și de alfabetul peste care este construită **FORM**), dintr-un sistem dat se pot elimina acele axiome care sunt consecințe semantice din altele (în cazul considerării unei noțiuni suport de adevăr) precum și așa- numitele *reguli de inferență derivate*. Astfel, considerând orice prefix al oricărei demonstrații (privită textual) D dintr-un sistem SD , acesta poate fi considerat ca o nouă regulă de inferență („derivată” din cele inițiale): *concluzia noii reguli este ultima formulă din demonstrația respectivă, iar ipotezele sunt reprezentate de restul formulelor care apar*. Eliminările de reguli derivate se „aprobă” doar dacă se menține *echivalența* din sistemul inițial și cel rezultat (după efectuarea eliminărilor). *Două sisteme SD și SD_I sunt echivalente dacă pentru fiecare mulțime de metaformule J și fiecare metaformulă F avem: $J \bullet_{SD} F$ dacă și numai dacă $J \bullet_{SD_I} F$* . Nu vom detalia nici acest subiect. Vom reveni totuși asupra câtorva aspecte suplimentare atât în §2. și §3. din acest capitol. Este absolut necesar să tratăm mai întâi câteva aspecte formale legate de teoriile logice.

§2. Teorii logice

Sistemele deductive sunt folosite **în principal** (în mod standard) ca element ajutător în construirea sau manipularea eficientă a unei *teorii logice*. Există mai multe accepțiuni ale ultimului termen chiar printre logicieni. **Vom accepta definiția care urmează din**

considerente legate de programarea logică. Astfel, există numeroase *lumi* (părți ale realității, colecție de cunoștințe, etc.) care sunt cunoscute pur și simplu, dar fiind foarte complexe este greu de spus dacă anumite cunoștințe nou asimilate fac parte din aceeași lume, sau chiar dacă anumite cunoștințe vechi nu sunt cumva contradictorii. Dorind să ne menținem în cadrul general folosit până în prezent, introducerea unei noțiuni de *adevăr* (deocamdată, în sens clasic, adică binar, etc.) în legătură cu o metaformulă este acum obligatorie. Să presupunem că orice clasă de metaformule **FORM** are atașată și o *clasă de structuri admisibile de adevăr*, notată *Str*, o structură fiind o funcție $S : \mathbf{FORM} \rightarrow \mathbf{B}$. Dacă **FORM** admite o definiție structurală, așa cum de altfel am și convenit de la bun început (prin **Baza** se plasează în **FORM** metaformulele „atomice”; cu ajutorul unor operatori, cum ar fi conectorii, cuantorii, etc., se introduc metaformule noi folosindu-se metaformule vechi, prin **Pasul constructiv**), atunci admitem și că fiecare *S* este **unica extensie homomorfă** a unei structuri definite inițial pe mulțimea metaformulelor atomice. În acest mod, se pot păstra toate definițiile (conceptele) semantice folosite până în prezent, fără modificări de esență (inclusiv conceptul de *consecință semantică*).

Definiția 4.4 (teorii logice). Se numește *teorie (logică)* orice subclasă *TE* a lui **FORM** închisă la consecință semantică. ■

În modul sugerat trebuie înțeleasă reprezentarea prin metaformule a unei baze de cunoștințe. Din păcate, după cum deja

cunoaştem, nu există metode semantice efective (algoritmice) convenabile pentru a testa dacă o mulţime dată de metaformule este sau nu închisă la consecinţă semantică (sau dacă o anumită metaformulă este satisfiabilă sau validă). Alternativa este de a folosi metode sintactice, care au avantajul că pot fi uşor *automatizate*. În cazul de faţă, se pune problema *axiomatizării teoriilor logice, cu ajutorul sistemelor de demonstraţie*. Acest lucru înseamnă că având dată o teorie $TE \subseteq \mathbf{FORM}$, trebuie să găsim o submulţime $A' = A \cup I \cap TE$, de „axiome” şi/sau ipoteze suplimentare, precum şi o mulţime de reguli de inferenţă R (adică un sistem de demonstraţie $SD' = \langle A', R \rangle$) astfel încât $TE = Th(SD')$. În acest caz, se impune de obicei ca A' să fie măcar o mulţime satisfiabilă (există măcar o structură S astfel încât pentru fiecare $F \in A'$ avem $S(F) = 1$), sau chiar ca ea să fie alcătuită numai din metaformule valide (după cum am mai observat, dacă A' conţine o contradicţie, orice metaformulă este consecinţă semantică din A'). Forma generală a lui A' se explică tocmai prin aceea că am presupus că A conţine formulele *valide* iar I pe cele *satisfiabile*. Mai general, să presupunem că pornim cu o mulţime de metaformule $A' \subseteq \mathbf{FORM}$, de cunoştinţe primare, unanim acceptate ca fiind „adevărate”, adică despre care ştim (nu ne interesează deocamdată prin ce metodă am aflat acest lucru) că reprezintă formule valide/satisfiabile în contextul descris mai sus. În concluzie, pentru a axiomatiza teoria noastră, trebuie să mai găsim şi o mulţime de reguli de inferenţă R astfel încât să avem $Cs(A') = Th(SD')$ (am notat cu $Cs(A')$ mulţimea

tuturor consecințelor semantice din A' , în raport cu noțiunea de adevăr adoptată, și cu SD' sistemul compus din A' și R). După cum am mai amintit, putem lua în considerare și situația inversă, în care avem dat un sistem $SD' = \langle A', R \rangle$ și **dorim să vedem dacă $Th(SD')$ este într-adevăr o teorie logică, sau, mai mult, dacă $Cs(A') = Th(SD')$.**

Definiția 4.5. Un sistem de demonstrație $SD' = \langle A', R \rangle$ se numește **corect și complet** pentru o teorie TE dacă $TE = Th(SD') = Cs(A')$ și $A' \subseteq TE$. O teorie TE este **axiomatizabilă** dacă există un sistem deductiv $SD' = \langle A', R \rangle$ corect și complet pentru ea, adică satisfăcând condițiile anterioare. Dacă SD' este finit specificabil (axiomatizabil), atunci teoria corespunzătoare se numește **finit axiomatizabilă**. ■

În cele de mai sus, dacă I este mulțimea vidă atunci TE este alcătuită doar din metaformule valide. În cazul teoriilor „reale”, I cuprinde în general cunoștințele primare ale lumii respective, iar A axiomele „logice” (de genul celor „puse” în **SD3**). Din păcate, în această ultimă situație, A conține în marea majoritate a cazurilor și axiomele egalității, astfel încât aceste teorii sunt în general nedecidabile (a se vedea **SAT1** pentru **LPI**₌). Alte tipuri de teorii se obțin prin „translarea” proprietăților sistemului deductiv atașat, dacă acesta există (de exemplu, putem vorbi de teorii consistente, teorii decidabile sau semidecidabile, etc.). Importante sunt teoriile *nedegenerate*. O teorie logică se numește **degenerată** dacă coincide cu mulțimea vidă sau

coincide cu întreaga clasă de metaformule în care se lucrează, **FORM** (de exemplu, dacă teoria conține o contradicție ea este degenerată; teoriile inconsistente sunt de asemenea degenerate). Nici axiomatizările „banale” nu sunt interesante (*este clar că, la modul general, toate teoriile sunt axiomatizabile, dacă luăm $A' = TE$ și $R = \emptyset$*). Înainte de a prezenta câteva exemple, să tragem concluzia necesară asupra formei generale a unei teoreme de corectitudine și completitudine în noul context.

TEOREMĂ DE CORECTITUDINE ȘI COMPLETITUDINE. Fie o clasă de metaformule **FORM**, o clasă de structuri admisibile Str pentru **FORM**, un sistem deductiv $SD' = \langle A', R \rangle$ în **FORM**, unde $A' = A \cup I$ (A fiind alcătuită din formule valide și I din formule satisfiabile) și o teorie logică $TE \subseteq \mathbf{FORM}$, astfel încât $TE = Cs(A')$. Atunci $Th(SD') = Cs(A')$. ■

Observație. A demonstra corectitudinea înseamnă a arăta că $Th(SD') \subseteq Cs(A')$ iar completitudinea, că $Th(SD') \supseteq Cs(A')$. Teorema se mai poate enunța și sub una din formele echivalente, destul de des întâlnite:

- Teoria TE admite un sistem deductiv corect și complet.
- În condițiile precizate, avem, pentru fiecare metaformulă $F \in \mathbf{FORM}$: $I \bullet_{SD} F$ dacă și numai dacă $I' \vdash F$.
- Teoria TE este (eventual, finit) axiomatizabilă.

În cazul în care este vorba de o teorie formată doar din formule valide (atunci va lipsi I), teorema capătă forma simplificată:

- Pentru fiecare $F \in \mathbf{FORM}$, avem: $\bullet_{SD} F$ dacă și numai dacă $\vdash F$. ■

În cele de mai sus am folosit notația $\vdash_{SD} F$ pentru faptul că $F \in Th(SD)$, unde $SD = \langle A, R \rangle$, sau, în momentul în care SD este implicit sau lisește, $\vdash F$ poate nota doar faptul că F este o formulă validă. *Din punct de vedere practic, teoriile finit axiomatizabile sunt cele mai utile.*

Nu vom intra în detalii nici în privința numeroaselor rezultate (importante și interesante) care pot fi demonstrate în acest moment. Se poate consulta [CAZ1] pentru completări, iar noi ne limităm la a prezenta fără demonstrație (de fapt, metateorema este sigur adevărată în cazul **LP1** și sistemului **SD0**):

Teorema deducției (pe scurt, TD). Pentru fiecare $A, B \in \mathbf{FORM}$ și fiecare $I \subseteq \mathbf{FORM}$, dacă $I, A \bullet B$ atunci $I \bullet A \rightarrow B$. ■

Exemplul 4.2 (reluat). Avem $\mathbf{FORM} = \mathbf{LP1}$, Str este clasa fixată a structurilor (conform **Capitolului 3**), $TE = Val(\mathbf{LP1})$ (clasa formulelor valide din **LP1**), $SD = \mathbf{SD3}$, $I \subseteq \mathbf{LP1}$ o mulțime oarecare de formule închise, satisfiabile și $F \in \mathbf{LP1}$ o formulă oarecare. Atunci:

Teorema 4.1 (teorema de completitudine a lui K. Gödel, 1930). $\vdash_{SD3} F$ dacă și numai dacă $\models F$.

Demonstrație. Corectitudinea ($\vdash_{SD3} F$ implică $\models F$), a fost deja demonstrată. **Completitudinea** ($\models F$ implică $\vdash_{SD3} F$) este mult prea laborioasă pentru a o putea reda în lucrarea de față. De fapt ([CAZ1]), ea se demonstrează indirect, folosind sistemul echivalent **SD0**, al deducției naturale (conform **Exemplului 4.4**). Faptul că \vdash este satisfiabilă nu este esențial, deoarece dacă \vdash este nesatisfiabilă atunci (meta)teorema este imediat adevărată (deși...neinteresantă în acest caz). (**q. e. d.**)

Prin urmare, teoria $Val(LP1)$ este chiar finit axiomatizabilă și putem testa validitatea unei formule încercând să-i găsim o demonstrație. $Val(LP1)$ nu este decidabilă, dar este semidecidabilă (putem „enumera” toate demonstrațiile posibile și alege apoi pe cea dorită). Teoria este și consistentă (necontradictorie). Mai mult, axiomele lui **SD3** sunt independente și R nu conține reguli derivate. Cu toate aceste calități, am văzut că nici demonstrațiile în **SD3** nu sunt chiar simple, ca să nu vorbim de găsirea unor algoritmi eficienți de „enumerare” a acestora. ■

Exemplul 4.1 (reluat). În această situație nestandard, avem de-a face cu **FORM** = **LP1** (de fapt, cu **FORM** = $\mathcal{C}h$). Apoi, fiecărei mulțimi (eventual finite) de clauze din **LP1**, notată F (fiecare clauză fiind la

rândul ei o mulțime finită de literali), îi putem asocia mulțimea $\text{Res}^*(F)$ care desigur *nu este o teorie logică în acest caz*. De altfel, nici F nu este (întotdeauna) o mulțime satisfiabilă (de ipoteze suplimentare). În plus, nici nu ne interesează „adevărul” ci „neadevărul”. Metoda rezoluției este de altfel o exemplificare perfectă a unei situații despre care am amintit deja: sistemul deductiv (aici, SD_F) este folosit ca definiție constructivă, pentru generarea unei mulțimi ($\text{Res}^*(F)$), care nu este, și nici nu vrem să fie, o teorie logică. Putem admite că această mulțime are totuși o caracterizare semantică (în sensul că o submulțime a sa, F , este sau nu satisfiabilă). Teorema de corectitudine și completitudine există dar are o formă mai specială, apropiată însă de forma clasică (în sensul că se caracterizează „neadevărul” în mod sintactic). ■

Exemplul 4.3 (teoria grupurilor). O altă clasă generală de exemple, situată undeva între cele prezentate anterior (standard, nestandard), este cel al așa-numitor *teorii (matematice) formale* ([CAZ1]). Și aici se începe cu un sistem deductiv „de bază”, conținând, de obicei, ca axiome, axiomele logice, adică cele ale oricărui sistem corect și complet pentru $Val(\mathbf{LP1})$, cum ar fi $SD3$, la care se adaugă axiomele egalității, împreună cu niște „axiome” specifice (acestea din urmă constituind de fapt mulțimea suplimentară de ipoteze). Mulțimea teoremelor pentru sistemul amintit (de tip SD') nu va coincide însă cu $Val(\mathbf{LP1})$, nici cu o subclasă a sa, ci cu o clasă de formule valide „în sens restrâns” (care formează o teorie logică „în sens restrâns”). Termenul „în sens restrâns” se referă la faptul că universurile pentru

structurile semantice admise pot fi doar anumite mulțimi (de exemplu, cum ar fi cele care constituie o structură de grup împreună cu o operație dată). Structurile admise conțin în plus și simboluri cu interpretări standard, cum ar fi egalitatea, totul fiind similar cu ceea ce am făcut în cazul „tregerii” de la **LP1** la **LP1₌**. Cadrul standard prezetat anterior se păstrează, inclusiv enunțul teoremei de corectitudine și completitudine (*care, de fapt, este imediat adevărată, în general admițându-se prin convenție faptul că „teoria” în cauză este tocmai mulțimea toremelor din sistemul deductiv ales*). Sensul noțiunilor de formulă „satisfiabilă”, „validă”, etc., este relativ la clasa restrânsă de structuri considerată.

Vom exemplifica cu **TG, teoria formală grupurilor**. Ne plasăm în **FORM** = **LP1**_{=, •, 1, ~}, simbolurile specificate în mulțimea din indice reprezentând egalitatea (simbol predicativ binar), operația de grup (simbol funcțional binar), elementul neutru (constantă funcțională), respectiv operația de simetrizare (simbol funcțional unar). Deja putem spune care este clasa *Str*, a structurilor admisibile $S = \langle U_S, I_S \rangle$: U_S va fi orice mulțime D dotată cu o lege de compoziție internă \circ astfel încât $\langle D, \circ \rangle$ formează grup; în oricare asemenea S , $=$ va fi interpretat ca egalitatea pe D , \bullet ca legea \circ , 1 ca elementul neutru al legii \circ , iar dacă un anumit term t va avea interpretarea d în D atunci termul $\sim t$ va avea ca interpretare simetricul lui d față de legea \bullet . Sistemul SD_{TG} va fi dat astfel de:

Axiome (A). Mulțimea lor este reuniunea celor două mulțimi descrise mai jos. Să punctăm faptul că de multe ori *axiomele egalității* sunt

incluse printre axiomele logice și că uneori sunt utilizate variante ale acestora (în funcție de scopul pentru care sunt introduse). Nu toate axiomele egalității amintite sunt folosite în mod direct în **TG** (asemenea axiome particulare există și printre axiomele grupului).

- **Axiomele logice:** Sunt cele ale sistemului **SD3** (1. – 5.).

- **Axiomele egalității:**

$$6. (\forall x)(x = x).$$

$$7. (\forall x)(\forall y)(x = y \rightarrow y = x).$$

$$8. (\forall x)(\forall y)(\forall z)(x = y \wedge y = z \rightarrow x = z).$$

$$9. (\forall *) (x_1 = y_1 \wedge x_2 = y_2 \wedge \dots \wedge x_n = y_n \rightarrow$$

$$f(x_1, x_2, \dots, x_n) = f(y_1, y_2, \dots, y_n)),$$

pentru fiecare $n \in \mathbf{N}^*$ și fiecare $f \in F_n$.

$$10. (\forall *) (x_1 = y_1 \wedge x_2 = y_2 \wedge \dots \wedge x_n = y_n \rightarrow$$

$$P(x_1, x_2, \dots, x_n) = P(y_1, y_2, \dots, y_n)),$$

pentru fiecare $n \in \mathbf{N}^*$ și fiecare $P \in P_n$.

Axiomele grupului (sunt de fapt *I*, ipotezele suplimentare; câteodată axiomele egalității sunt plasate tot aici):

$$11. (\forall x)(\forall y)(\sim x = \sim y).$$

$$12. (\forall x)(\forall y)(\forall z)(x = y \rightarrow x \bullet z = y \bullet z \wedge z \bullet x = z \bullet y).$$

$$13. (\forall x)(\forall y)(\forall z)(x \bullet (y \bullet z) = (x \bullet y) \bullet z).$$

$$14. (\forall x)(1 \bullet x = x).$$

$$15. (\forall x)((\sim x) \bullet x = 1).$$

Reguli de inferență: (**MP**) și (**RG**) de la **SD3**.

Vom reveni cu un exemplu de demonstrație în teoria grupurilor după ce vom introduce sistemul **SD0**, *echivalent* cu **SD3** (**Exercițiul 4.5**). Datorită acestui ultim fapt, vom putea folosi și *deducția naturală* în cazul teoriei grupurilor. ■

În finalul acestui capitol prezentăm alte câteva (tipuri de) sisteme deductive, având o largă utilizare atât din punct de vedere teoretic cât și practic.

§3. Clasificarea sistemelor deductive

Începem cu o trecere în revistă a unor posibilități de clasificare ([CAZ1]) a sistemelor deductive, din care se pot trage concluzii utile cu privire la calitățile și defectele unor asemenea sisteme. *Clasificările prezentate (care nu sunt singurele acceptate) se referă în principal la sistemele standard* (în sensul descris anterior). Sistemele deductive se pot astfel împărți:

- **În funcție de conectivele logice alese.** Există **sisteme boolean complete** sau **boolean incomplete**. Știm că în interpretarea conectorilor logici prin structuri aceștia devin funcții booleene. Termenul de (in)completitudine se referă la faptul că mulțimea interpretărilor conectorilor aleși în alfabetul de bază peste care este construit sistemul formează (sau nu) o mulțime completă de funcții (în sensul **Capitolului 2**). Sistemul **SD3** este boolean complet, în timp ce un sistem care, de exemplu, folosește drept

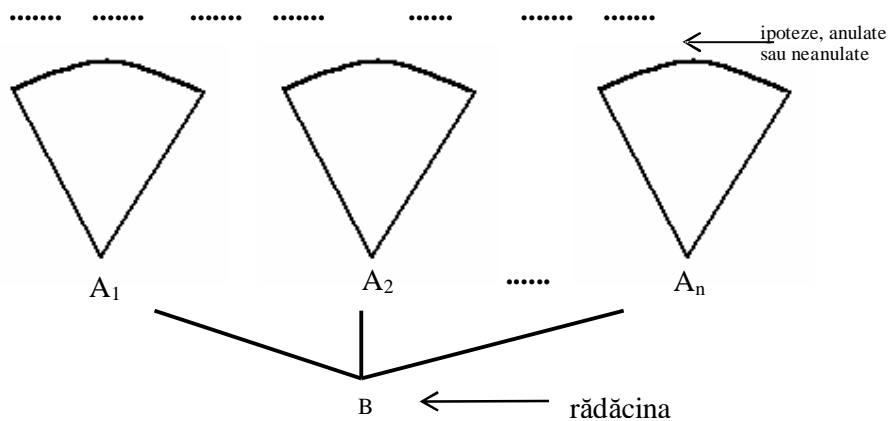
conector doar \rightarrow (rezultând așa-numitul *calcul implicațional*), va fi boolean incomplet.

- **În funcție de relația avută cu o anumită teorie logică.** Sistemele pot fi corecte sau nu, complete sau nu pentru o teorie dată. Toate sistemele implicate într-o teoremă de tip Gödel sunt corecte și complete, conform definiției noastre. După cum am mai precizat, completitudinea este mai greu de atins (demonstrat) și de aceea este de multe ori doar *adoptată prin convenție*. Corectitudinea este de obicei impusă, deși poate avea și niște forme mai deosebite (vezi sistemele nestandard, cum ar fi rezoluția).
- **În funcție de importanța acordată axiomelor sau regulilor de inferență.** Din acest punct de vedere, se poate acorda o atenție deosebită regulilor (adică modului de raționament, de obținere de cunoștințe noi) în dauna axiomelor (cunoștințelor primare). Acest tip de sisteme se numesc **Gentzen-Jaskowski**. Un asemenea sistem va fi **SD0 (deducția naturală)**, care este echivalent cu **SD3** și va fi prezentat în acest capitol. În cazul în care balanța este inversată (există „mult mai multe” axiome decât reguli de inferență, ca în cazul **SD3**), sistemele sunt cunoscute sub numele de **sisteme Hilbert**.
- **După clasa FORM aleasă.** De exemplu (pentru logica clasică), putem avea **sisteme propoziționale** sau **sisteme predicative**.

Să considerăm un sistem de tip Gentzen, foarte cunoscut în literatura de specialitate (introdus pentru prima oară de G. Gentzen și S. Jaskowski în 1934).

Exemplu (deducția naturală, sistemul SD0). Clasa **FORM** este **LP1**. Alfabetul conține în acest caz doar conectorii \neg , \wedge și cuantificatorul \forall . După cum am precizat, într-un asemenea sistem regulile de inferență sunt „mai importante” decât axiomele, sistemul **SD0 neavând chiar nici o axiomă**. Pentru a simplifica înțelegerea, **vom defini direct o demonstrație în sensul de deducției naturale ca fiind un anumit arbore** (vezi mai jos), fără a folosi definiția generală. Un **arbore de deducție naturală** are pe nivelul 0 (în frunze) formule oarecare (ipoteze ale unor reguli de inferență din sistem, inclusiv elemente din eventuala mulțime suplimentară I), iar nivelele următoare se obțin constructiv, conform definiției generale (rădăcina fiind „rezultatul final”). **Caracteristic acestui sistem este faptul că acele condiții c de aplicabilitate ale regulilor, dacă există, sunt de tipul „se anulează ipoteza F”** (aici termenul ipoteză nu se referă la ipotezele regulii respective, ci la **toate** formulele **F** prezente în frunzele arborelui curent). Pentru ca anularea să nu fie efectivă (având drept consecință ștergerea unui nod din graf, ceea ce conduce întotdeauna la anumite complicații tehnice), vom adopta soluția de a *marca ipotezele anulate* (cu cifre, de exemplu). Dacă se dorește, pentru evitarea confuziilor, mărcile pot fi diferite, în cazul în care regulile aplicate sunt diferite și/sau dacă sunt aplicate la momente diferite. Tot pentru evitarea

confuziilor, aceeași marcă se va asocia și nodului care constituie concluzia regulii care, aplicată, a cauzat anularea. Ipotezele anulate modifică însă clasa de demonstrații acceptate într-un asemenea sistem: avem $I \bullet_{SD0} G$ (G este consecință sintactică în $SD0$ utilizând mulțimea suplimentară de formule I ; sau, există o demonstrație pentru G în $SD0$ utilizând I ; sau, există o deducție naturală pentru G în $SD0$) dacă există un arbore de deducție naturală având rădăcina G și **cu toate ipotezele neanulate aparținând lui I** . Grafic:



În acest mod, vom avea desigur $\bullet_{SD0} G$ doar dacă va exista un arbore de deducție naturală cu rădăcina G , având **toate** ipotezele anulate.

Pentru a prezenta concret sistemul, rămâne să dăm **regulile de inferență din care este alcătuit**, care vor primi și un nume înafara numărului de secvență. Vom avea câte o (schemă de) regulă pentru fiecare A , $B \in \mathbf{LP1}$, fiecare $x \in X$ și fiecare $t \in T$. În 5., este necesar ca

substituția $[x/t]$ să fie permisă pentru A , iar în 6., ca x să nu apară liber în nici o ipoteză neanulată. Schemele 3. și 4. au variante datorită necesității de a se „prinde” comutativitatea conjuncției la nivel sintactic (ne vom referi la ele ca 3', respectiv 4'). Deoarece substituția $[x/x]$ este permisă pentru orice formulă, regula 5. are și forma particulară $\langle\langle\{(\forall x)A\}, A\rangle, \text{true}\rangle$ (care va fi notată 5'). Să remarcăm și faptul că regula 6. nu are nevoie de nici o restricție sintactică în momentul în care se lucrează cu formule închise. Mnemonicele provin de la următoarele cuvinte: **E** – eliminare; **I** – introducere; **N** – negație; **C** – conjuncție.

$$1. (\mathbf{EN}) \quad \frac{B, \top B}{A}, \quad \text{c: se anulează ipoteza } \dot{A}.$$

$$2. (\mathbf{IN}) \quad \frac{B, \top B}{\top A}, \quad \text{c: se anulează ipoteza } A.$$

$$3. (\mathbf{EC}) \quad \frac{A \wedge B}{A} \quad \text{și} \quad \frac{A \wedge B}{B}.$$

$$4. (\mathbf{IC}) \quad \frac{A, B}{A \wedge B} \quad \text{și} \quad \frac{A, B}{B \wedge A}.$$

$$5. (\mathbf{E''}) \quad \frac{(\forall x)A}{A[x/t]}.$$

$$6. (\mathbf{I''}) \quad \frac{A}{(\forall x)A}.$$

■

Demonstrația teoremelor următoare poate fi găsită în [CAZ1].

Teorema 4.2. Sistemul **SD0** este corect și complet pentru $\text{Val}(\text{LP1})$. ■

Teorema 4.3. Sistemele **SD0** și **SD3** sunt echivalente, adică pentru fiecare mulțime de formule închise $J \subseteq \text{LP1}$ și fiecare formulă $F \in \text{LP1}$, avem: $J \bullet_{\text{SD0}} F$ dacă și numai dacă $J \bullet_{\text{SD3}} F$. ■

În plus, putem spune că **SD0** este un *sistem predicativ (de tip Gentzen, standard), finit specificat și boolean complet*. Dacă introducem \vee , \rightarrow , \leftrightarrow și \exists în alfabetul de bază, putem folosi și următoarele *reguli derivate*:

7. **(ED)** $\frac{A \vee B, \top A}{B}$ și $\frac{A \vee B, \top B}{A}$.
8. **(ID)** $\frac{A}{A \vee B}$ și $\frac{A}{B \vee A}$.
9. **(EI)** $\frac{A, A \rightarrow B}{B}$.
10. **(II)** $\frac{B}{A \rightarrow B}$, $\text{c: se anulează ipoteza } A$.
11. **(EE)** $\frac{A \leftrightarrow B}{A \rightarrow B}$ și $\frac{A \leftrightarrow B}{B \rightarrow A}$.
12. **(IE)** $\frac{A \rightarrow B, B \rightarrow A}{A \leftrightarrow B}$.
13. **(ES)** $\frac{(\exists x)A, B}{B}$, $\text{c: se anulează ipoteza } A \text{ din subarborele}$
având rădăcina acest B.

$$14. (\text{IS}) \quad \frac{A[x/t]}{(\exists x)A}.$$

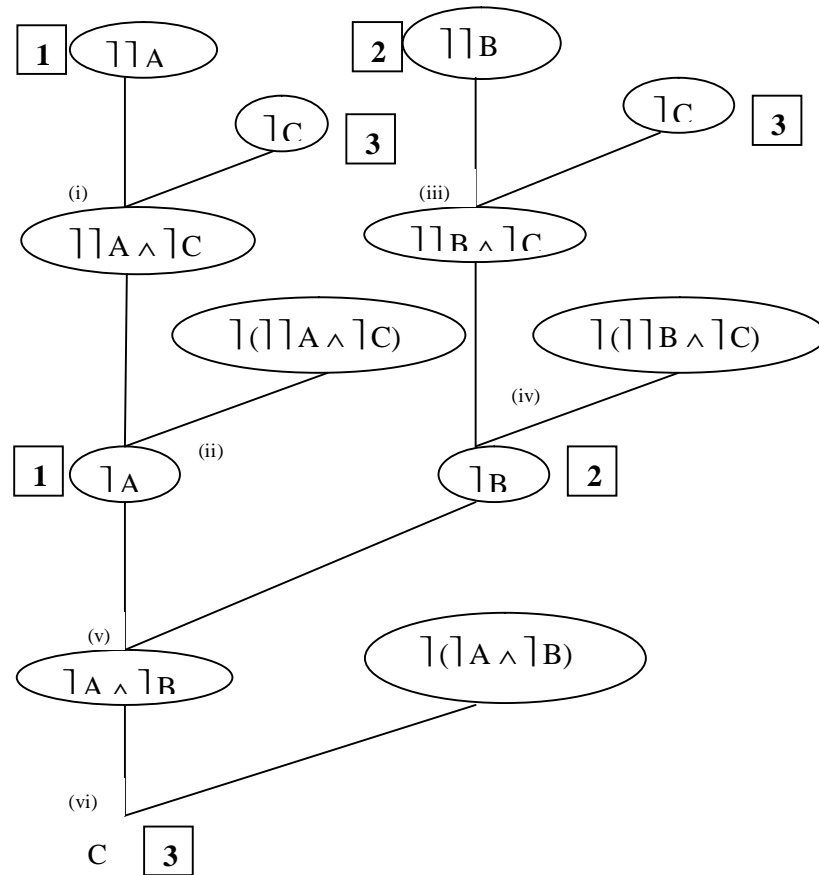
$$15. (\text{DN}) \quad \frac{A}{\neg \neg A} \text{ și } \frac{\neg \neg A}{A}.$$

Analog cu precizările deja făcute pentru regulile de bază, și schemele de mai sus sunt valabile pentru fiecare $A, B \in \mathbf{LP1}$, fiecare $x \in X$ și fiecare $t \in T$. În 13., condiția sintactică este dată de faptul că x nu trebuie să aibă apariții libere în ipotezele neanulate, diferite de A și prezente în subarboarele având rădăcina **exact acel** B pentru care se aplică regula respectivă. De asemenea, în 14., condiția sintactică este ca substituția $[x/t]$ să fie permisă pentru A . Mnemonicul **E** de pe a doua poziție (din 11. și 12.) nu mai provine din cuvântul „eliminare”, ci de la **echivalență**; mnemonicul **I**, de pe a doua poziție din 9., 10., provine de la **implicație**, iar **D** – de la **disjuncție** (**D** de pe prima poziție în 15. provine de la **dublă**). Reamintim că în regulile având variante cea de a doua schemă va fi referită prin același număr (nume), urmat de un apostrof. Extinderea alfabetului și folosirea regulilor derivate, pot simplifica mult unele demonstrații, care sunt – poate chiar mai mult decât la sistemul **SD3** – suficient de sofisticate pentru un începător.

Exemplu. Să se arate că avem

$$\neg(\neg \neg A \wedge \neg C), \neg(\neg \neg B \wedge \neg C), \neg(\neg A \wedge \neg B) \bullet C, \text{ în } \mathbf{SD0}.$$


Într-adevăr, putem construi arborele de deducție:



■

Observație. În arborele de mai sus, cifrele denotă, așa cum am mai precizat, o marcă aplicată ipotezelor care trebuie anulate în momentul aplicării unei anumite reguli. Pașii de aplicare i-am notat distinct, prin ((i) – (vi)), ca de altfel și mărcile. De exemplu, la primul pas de deducție (notat (i)), se folosesc drept ipoteze formulele oarecare $\neg\neg A$ și $\neg C$, pentru a aplica o instanță a regulii (IC) (în definiția generală cele

două formule sunt denotate prin A , respectiv B). Această regulă nu are atașată nici o condiție de aplicare. Apoi, la pasul (ii), se continuă construcția demonstrației (arborelui) prin aplicarea regulii (**IN**), care are ca ipoteze formulele $\neg A \wedge \neg C$ și $\neg(\neg A \wedge \neg C)$ (în descrierea generală ele sunt notate B , respectiv $\neg B$) și drept concluzie formula $\neg A$, având și condiția de aplicare $C = \text{se anulează ipoteza } A$. *Marca 1 a fost aplicată, conform celor stabilite, nodului concluzie al aplicării regulii și ... lui $\neg A$, în loc de A !* Acest lucru nu este permis, tehnic vorbind (de fapt, pentru a fi foarte exacti, suntem în culpă și cu utilizarea strictă a parantezelor; soluția aici este simplă, introducându-se de la bun început și variante „cu paranteze” pentru toate regulile). Pentru a aplica corect teoria, ne putem folosi de instanțe ale regulii derivate (**DN**) (deducem $\neg A$ din A , înainte de efectuarea primului pas (i) și astfel putem anula în (ii) ceea ce trebuie, adică pe A). Pe parcursul construcției arborelui se mai întâlnesc asemenea cazuri și devine evidentă necesitatea de a avea la dispoziție o „bibliotecă” de reguli derivate, pentru a lucra simplu cu un asemenea tip de deducție naturală. *Acestea trebuie însă demonstrate în prealabil*, lucru care nu este chiar ușor (totuși, se pare că este mai ușor de conceput un „mecanism automat de raționament” pentru **SD0** decât pentru **SD3**). O alternativă, bazată de fapt pe aceeași idee (utilizarea unor reguli derivate), este să folosim formulele din I sub o formă echivalentă, de exemplu $I = \{A \rightarrow C, B \rightarrow C, A \vee B\}$. ■

 **Exercițiul 4.3** ([CAZ1]). Fie $A \in C$, $B \in C$, $A \dot{\vee} B$ și C formule oarecare din **LP1** și $I \subseteq \mathbf{LP1}$. Să presupunem că $I \bullet_{SD0} A \in C$, $B \in C$, $A \dot{\vee} B$ (adică fiecare dintre formulele din membrul drept este consecință sintactică din I în **SD0**). Arătați că $I \bullet_{SD0} C$ (aceasta se numește în logică **metoda de demonstrație prin disjuncția cazurilor**).

Exemplu (calculul cu secvențe, sistemul SD1). Pornim inițial cu **LP1**, construit peste un alfabet care conține toți conectorii și cuantificatorii cunoscuți (desigur că unii dintre ei pot fi adoptați prin notație, dar îi vom folosi fără restricție): $\neg, \wedge, \rightarrow, \leftrightarrow, \forall, \exists$. Se numește **secvență** orice formulă care are forma: $A_1 \wedge A_2 \wedge \dots \wedge A_m \rightarrow B_1 \vee B_2 \vee \dots \vee B_n$, unde $n, m \in \mathbf{N}$, $A_1, A_2, \dots, A_m, B_1, B_2, \dots, B_n \in \mathbf{LP1}$ (m, n pot fi și egali cu 0, dar nu simultan). Prin urmare, vom lucra practic cu clauze, dar notația pe care o vom adopta ne conduce la ideea că *secvențele sunt de mai degrabă metaformule (alt tip de obiect, oricum mai complex) decât formulele cu care am fost familiarizați în capitolele anterioare*. Astfel, vom scrie o secvență sub forma (desigur că în cele ce urmează \Rightarrow nu are sensul de metaimplicație):

$$A_1, A_2, \dots, A_m \Rightarrow B_1, B_2, \dots, B_n.$$

Mai mult, vom considera cei doi membri ai relației de mai sus ca fiind mulțimi (atunci când ordinea elementelor va fi esențială, vom specifica explicit acest lucru). Prin urmare, **o secvență va fi de forma $U \dot{\vee} V$ (U și V pot fi și mulțimea vidă, dar nu simultan)** și vom putea scrie $U' = U, A$ în loc de $U' = U \cup \{A\}$, în ideea că, din anumite motive,

elementul A din U' trebuie pus în evidență. Vom extinde notația la submulțimi oarecare, adică vom putea scrie (de exemplu) V, W în loc de $V \cup W$ și V, A, B în loc de $V \cup \{A\} \cup \{B\}$. Astfel, punem $\mathbf{FORM} = \{U \mid U \text{ este secvență în } \mathbf{LP1}\}$. Sistemul $\mathbf{SD1}$, după cum vom vedea, deși atribuit în principal lui Gentzen (1934) și având o singură schemă de axiome (este drept, foarte generală), se apropie (în privința modalității de utilizare) mai mult de un sistem de tip Hilbert. Sistemele deductive bazate pe secvențe au și o răspândită utilizare în situații nestandard (legate numai de definirea constructivă a unor mulțimi „în mod axiomatic”, fără referire la „adevăr”). Mai concret, $\mathbf{SD1}$ este un *sistem predicativ finit specificat și boolean complet*, având:

Axiome. Pentru fiecare $U, V \in \mathbf{FORM}$ și pentru fiecare $A \in \mathbf{LP1}$:

$$U, A \Rightarrow V, A.$$

Reguli de inferență. Schemele de mai jos (care, din nou, sunt numerotate dar au atașat și un nume mnemonic care nu mai necesită explicații - exceptând poate (**RT**) care înseamnă regula tăieturii) sunt valabile pentru fiecare $U, V \in \mathbf{FORM}$, fiecare $A, B \in \mathbf{LP1}$, fiecare $x \in X$ și fiecare $t \in T$. În regula 5., substituția $[x/t]$ trebuie să fie permisă pentru A , iar în 6., x nu trebuie să apară liber în nici o formulă din U sau V . În momentul în care vor exista mai multe premize într-o regulă, vom folosi pentru separarea lor „;”. **Atenție la faptul că regulile 5. și 7. au o infinitate de premize (de exemplu, $U, (A)[x/t] \Rightarrow V$ din (\vdash) trebuie înțeles ca reprezentând $U, (A)[x/t_1] \Rightarrow V$; $U, (A)[x/t_2] \Rightarrow V$; ... , adică se iau în considerare toate elementele t din**

T , pentru care $[x/t]$ este permisă pentru A ; rolul lui A în **(RT)** este similar, instanțele unei scheme referindu-se la celelalte elemente având statutul de a fi „oarecare”.

1. (\rightarrow \vdash)
$$\frac{U \Rightarrow V, A}{U, \neg A \Rightarrow V}.$$
2. (\rightarrow \vdash)
$$\frac{U, A, B \Rightarrow V}{U, A \wedge B \Rightarrow V}.$$
3. (\vdash \rightarrow)
$$\frac{U, A \Rightarrow V}{U \Rightarrow V, \neg A}.$$
4. (\vdash \rightarrow)
$$\frac{U \Rightarrow V, A; U \Rightarrow V, B}{U \Rightarrow V, A \wedge B}.$$
5. (\forall \vdash)
$$\frac{U, (A)[x/t] \Rightarrow V}{U, (\forall x)A \Rightarrow V}.$$
6. (\vdash \forall)
$$\frac{U \Rightarrow V, A}{U \Rightarrow V, (\forall x)A}.$$
7. **(RT)**
$$\frac{U, A \Rightarrow V; U \Rightarrow V, A}{U \Rightarrow V}.$$

Conform cadrului general fixat, o metaformulă (aici, o secvență) $U \Rightarrow V$ este teoremă în **SD1** dacă există un arbore (care reprezintă o demonstrație a lui $U \Rightarrow V$) având în rădăcină pe $U \Rightarrow V$ și axiome în frunze (în acest sens însă, sistemul doar generează constructiv mulțimea teoremelor). Dacă vrem să legăm **SD1** de *Val* (**LP1**) și să enunțăm o teoremă de corectitudine și completitudine, trebuie să interpretăm relația de consecință sintactică $/ \bullet F$ ($/ \hat{=} \text{LP1}$, $F \hat{=} \text{LP1}$) prin: există $G_1, G_2, \dots, G_n \hat{=} /$ astfel încât

$G_1, G_2, \dots, G_n \vdash F \hat{=} Th(SD1)$. În particular, $\bullet_{SD1} F$ în seamă că **metaformula** $\vdash F$ (adică $\emptyset \vdash F$) este **teoremă în SD1**. Deși se poate enunța o teoremă de corectitudine și completitudine mai generală (oricum *sistemul este corect*; demonstrarea completitudinii este mai dificilă datorită regulilor cu un număr infinit de premise; pe de altă parte, se poate arăta că *putem renunța la (RT)*, fără a se reduce *puterea deductivă* a sistemului, demonstrațiile fiind în schimb mai lungi), ne limităm la a aminti că avem:

Teorema 4.4. Fie orice $F \in LP1$. Atunci:

‘ F dacă și numai dacă $\bullet_{SD1} F$. ■

Sistemul **SD1** are avantajul, față de **SD3** și **SD0**, că putem deriva imediat un „procedeu automat” ([CAZ1]) de construire a unei demonstrații (arbore). Să presupunem astfel că avem de demonstrat secvența $U \Rightarrow V$ în **SD1**. Alegem un nod, care este frunză (inițial, acesta nu va putea fi decât rădăcina, adică nodul care conține $U \Rightarrow V$), și anume secvența care se află plasată în el, o formulă din secvență și operația logică principală a formulei (conectorul sau cuantorul care furnizează o primă împărțire a formulei în subformule). Se aplică acum una dintre regulile 1. – 7., obținându-se unul sau (maxim) două noi noduri, apoi se reia **alegerea** unei frunze. Procedul se oprește atunci când frunzele conțin doar axiome (acestea se pot numi *noduri terminale*; ideea este să alegem spre aplicare acele reguli care generează noi noduri, care sunt fie terminale, fie conțin secvențe având

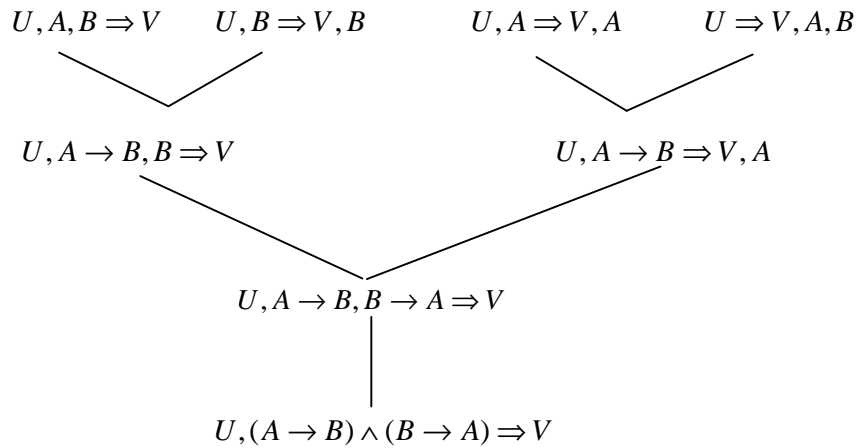
mai puține operații logice). Desigur că ceea ce am descris mai sus nu este încă un algoritm, care să poată prelucra întreaga clasă de secvențe (nu este nici măcar un semialgoritm), datorită numărului infinit de premise din 5. și 7. Vom ilustra totuși procedeul printr-un exemplu, imediat după ce dăm și câteva reguli de inferență derivate (presupunem că $U, V \in \mathbf{FORM}$, $A, B \in \mathbf{LP1}$, $x \in X$, $t \in T$ sunt elemente oarecare; că în 14., x nu apare liber în nici o formulă din U sau V ; că în 15. nu mai avem o infinitate de premise ca în 5., valoarea concretă a lui t furnizând doar o instanță a regulii în ansamblu; totuși, în 15., substituția $[x/t]$ trebuie să fie permisă pentru A).

Reguli de inferență derivate pentru **SD1**:

8. ($\dot{\cup}$ \otimes)
$$\frac{U, A \Rightarrow V; U, B \Rightarrow V}{U, A \vee B \Rightarrow V}.$$
9. (\otimes $\dot{\cup}$)
$$\frac{U \Rightarrow V, A, B}{U \Rightarrow V, A \vee B}.$$
10. (\otimes $\dot{\mathbf{P}}$)
$$\frac{U \Rightarrow V, A; U, B \Rightarrow V}{U, A \rightarrow B \Rightarrow V}.$$
11. ($\dot{\mathbf{P}}$ \otimes)
$$\frac{U, A \Rightarrow V, B}{U \Rightarrow V, A \rightarrow B}.$$
12. (\ll $\dot{\mathbf{P}}$)
$$\frac{U, A, B \Rightarrow V; U \Rightarrow V, A, B}{U, A \leftrightarrow B \Rightarrow V}.$$
13. ($\dot{\mathbf{P}}$ \ll)
$$\frac{U, A \Rightarrow V, B; U, B \Rightarrow V, A}{U \Rightarrow V, A \leftrightarrow B}.$$
14. ($\$$ $\dot{\mathbf{P}}$)
$$\frac{U, A \Rightarrow V}{U, (\exists x) A \Rightarrow V}.$$

$$15. (\text{P } \$) \quad \frac{U \Rightarrow V, A[x/t]}{U \Rightarrow (\exists x)A, V}.$$


Exemplu. Regula (« P ») se poate obține pe baza deducției (metaformulele $U, B \Rightarrow V, B$ și $U, A \Rightarrow V, A$ sunt axiome):



 **Exercițiul 4.4** ([CAZ1]). Arătați că formula

$$(A \dot{\cup} C) \dot{\cup} (A \dot{\cap} B) \dot{\cup} (C \dot{\cap} D) \dot{\cap} (B \dot{\cup} D) \in \mathbf{LP1}$$

este demonstrabilă în **SD1**.

 **Exercițiul 4.5** ([CAZ1]). Se consideră teoria grupurilor **TG** și sistemul deductiv asociat, SD_{TG} . Fie $u, v \in T$, termi oarecare. Să se arate că $u = v \dot{\cap} v = u$ este teoremă în **TG**, folosind deducția naturală.

Încheiem acest paragraf precizând faptul că sistemul **SD2** (pe care nu îl vom trata detaliat), atribuit lui A. Schwichtenberg (1977), este un sistem asemănător cu **SD1**, dar *se referă la formulele LP1 în care negația poate fi plasată doar imediat în fața unei formule atomice*. Pentru acesta, se poate demonstra mai ușor posibilitatea eliminării unei reguli de tipul (**RT**).

§4. Recapitulare și Index

Subiectul capitolului curent este vast și de un deosebit interes, atât pentru filozofi, matematicieni și logicieni, cât și pentru cercetătorii din alte domenii științifice, inclusiv informaticienii. Prin conceptul de *teorie logică*, noțiunea de adevăr este tratată la nivel global, ca reflectare a unei părți *coerente* a realității. O *bază inițială de cunoștințe*, alcătuită din afirmații presupuse a fi *adevărate* (într-un sens bine precizat), și prezentată sintactic ca o mulțime de (*meta*)*formule*, poate fi ulterior completată cu noi afirmații (cunoștințe), despre care nu se poate ști aprioric că „reflectă aceeași realitate”. Utilizând un procedeu standard (obținerea de *consecințe semantice*), se pot afla chiar **toate** afirmațiile „adevărate” în contextul respectiv. Din punct de vedere algoritmic, problema aflării tuturor consecințelor semantice dintr-o mulțime dată de formule și chiar *problema de decizie* mai simplă *Este formula F o consecință semantică din mulțimea G ?*, sunt de cele mai multe ori *nedecidabile* (sau cel mult *semidecidabile*). Chiar în cazul unor probleme de acest tip *decidabile*, rezultatele privind complexitatea (algoritmilor care le rezolvă) sunt de obicei descurajatoare (algoritmii

fiind, în cazurile nebanale, exponențiali ca timp de execuție, urmează că problema este în fapt *netratabilă*). *Abordările sintactice* ale rezolvării problemelor au cel puțin două avantaje deloc de neglijat. Astfel, se pot selecta din start subclase de formule *interesante, cu o formă convenabilă*, pentru care rezolvarea problemei este „mai simplă”. Apoi, algoritmi generali bazați pe sintaxă (deși *nu neapărat mai eficienți*), sunt mai flexibili, mai ușor de transformat și adaptat, mai ușor de extins și de a fi aplicați și în alte situații/contexte. **Sistemele deductive** formalizează cele de mai sus, apelând la concepte cum ar fi *axiomă, regulă de inferență, teoremă, consecință sintactică*. Legătura dintre cele două noțiuni generale se stabilește în mod concret prin **teoreme de corectitudine și completitudine**. Chiar dacă uneori suntem nevoiți să renunțăm la demonstrarea completitudinii sau chiar la atașarea unui concept de „adevăr” formulelor, *sistemele de demonstrație pot fi importante prin ele însele*. Astfel, înafara exemplelor considerate, standard sau nestandard (SD0, SD1, SD3, etc.), mai poate fi amintit și așa-numitul *calcul cu tabele* ([OHL]), folositor înafara contextului teoriilor logice.

A Teoriile logice și sistemele de demonstrație constituie prin urmare cadrul formal prin care pot fi studiate într-un mod foarte precis părți ale realității prin prisma oricărei științe (vezi de exemplu **teoriile matematice formale**), conceptele menționate în primele capitole nemaifiind ambigue. Pentru logicieni subiectul este inepuizabil și foarte atrăgător, obținându-se și astăzi noi rezultate surprinzătoare. Informaticii îi revine sarcina de „ține pasul” cu noile descoperiri în

domeniu, de a adapta algoritmi existenți și conceptele legate de programarea logică, urmărindu-se eficientizarea acestora și apropierea de idealurile *Inteligenței artificiale*.

Atenționăm asupra faptului că termenii pe care i-am selectat în **Indexul** care urmează este posibil să nu fie chiar cei mai importanți, iar paginile indicate s-ar putea să nu fie chiar primele în care apare cuvântul respectiv (aceasta datorită generalității subiectului):

teorie logică, 191

sistem deductiv (de demonstrație), 191

axiomă, 191

regulă de inferență (reguli corecte sau sound), 191

demonstrație (consecință sintactică; deducție; raționament), 191

metaformulă, 193

sistem deductiv standard, 200

regulă de inferență derivată, 204

axiomatizarea unei teorii logice, 207

teorie logică nedegenerată, consistentă, finit axiomatizabilă, 207

teoremă de corectitudine și completitudine, 208

sistem deductiv corect și complet, 208

sistem deductiv necontradictoriu (consistent), 210

sistem deductiv independent și minimal, 210

sistem deductiv boolean complet, 214-215

sistem deductiv propozițional sau predicativ, 214-215

sistem deductiv de tip Hilbert, 215

sistem deductiv de tip Gentzen, 215

deducție naturală, 215

sisteme deductive echivalente, 218

sistem deductiv finit specificat, 219

calculul cu secvențe, 223

§5. Exerciții

1. Să se demonstreze în sistemul **SD3** următoarele formule ([CAZ1]):
 - (i) $(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C))$.
 - (ii) $(A \rightarrow (B \rightarrow C)) \rightarrow (B \rightarrow (A \rightarrow C))$.
 - (iii) $\bigvee A \rightarrow A$.
2. Rezolvați **Exercițiul 4.1**.
3. Rezolvați **Exercițiul 4.2**.
4. Rezolvați **Exercițiul 4.3**.
5. Rezolvați **Exercițiul 4.4**.
6. Rezolvați **Exercițiul 4.5**.
7. Fie sistemul deductiv **SIN1** ([CAZ1]), dat prin axiomele și regulile de mai jos.

Axiome. Pentru fiecare $A, B, C, D \in \mathbf{LP}$, avem:

Ax1. $A \rightarrow (B \rightarrow A)$

Ax2. $(C \rightarrow (A \rightarrow B)) \rightarrow ((C \rightarrow A) \rightarrow (C \rightarrow B))$

Ax3. $A \wedge B \rightarrow A$

Ax4. $A \wedge B \rightarrow B$

$$\text{Ax5. } A \rightarrow (B \rightarrow A \wedge B)$$

$$\text{Ax6. } A \rightarrow A \vee B$$

$$\text{Ax7. } B \rightarrow A \vee B$$

$$\text{Ax8. } (A \rightarrow D) \rightarrow ((B \rightarrow D) \rightarrow (A \vee B \rightarrow D))$$

$$\text{Ax9. } (A \leftrightarrow B) \rightarrow (A \rightarrow B)$$

$$\text{Ax10. } (A \leftrightarrow B) \rightarrow (B \rightarrow A)$$

$$\text{Ax11. } (A \rightarrow B) \rightarrow ((B \rightarrow A) \rightarrow (A \leftrightarrow B))$$

$$\text{Ax12. } (A \rightarrow \neg A) \rightarrow \neg A$$

$$\text{Ax13. } \neg A \rightarrow (A \rightarrow B)$$

Reguli de inferență. Acestea sunt **(MP)** și **regula substituției, (RS)**. Schema este valabilă pentru fiecare $F, G_1, G_2, \dots, G_n \in \mathbf{LP}$. În cele de mai jos, $F(A_1, A_2, \dots, A_n)$ denotă faptul că F este construită peste (exact) mulțimea de variabile propoziționale $\{A_1, A_2, \dots, A_n\}$, iar $F(G_1, G_2, \dots, G_n)$ denotă o formulă F în care fiecare apariție a lui A_1 se înlocuiește cu G_1 , fiecare apariție a lui A_2 , cu G_2 , ș. a. m. d., fiecare apariție a lui A_n , se înlocuiește cu G_n , *simultan*:

$$(\text{RS}) \quad \frac{F(A_1, A_2, \dots, A_n)}{F(G_1, G_2, \dots, G_n)}$$

Acesta este un sistem deductiv propozițional standard, de tip Hilbert, construit peste **LP** care are toți conectorii standard. Sistemul este boolean complet, dar **incomplet** din punctul de vedere al $\mathcal{Val}(\mathbf{LP})$. Astfel, **se poate arăta că formule valide cunoscute, cum ar fi $\neg \neg A \rightarrow A$ (legea negării negației) sau**

$A \vee \neg A$ (*legea terțiului exclus*) nu sunt teoreme în **SIN1**. Ca observație suplimentară, să spunem că dacă păstrăm doar $Ax1. - Ax.11$. (împreună cu **(MP)** și **(RS)**), sistemul mai poartă numele de *calculul propozițional pozitiv* (datorat lui Hilbert). Acesta este desigur tot incomplet față de Val (**LP**), dar are avantajul că în el putem demonstra *tot ceea ce avem nevoie, în ipoteza că nu vrem să folosim negația*.

Arătați că:

- (i) Sistemul este corect față de Val (**LP**).
 - (ii) $(A \rightarrow B) \rightarrow ((A \rightarrow \neg B) \rightarrow \neg A)$, *legea reducerii la absurd*, este teoremă în **SIN1**.
8. Dați o axiomatizare a teoriei formale a grupurilor, fără a folosi simbolurile care desemnează elementul neutru și operația de simetrizare.