Capitolul 4

Teorii logice şi sisteme deductive

Această **parte a logicii formale**, alcătuită din *teoriile logice şi sistemele deductive (de demonstraţie, inferenţiale)* este una dintre cele mai răspândite modalităţi de exprimare exactă a noilor descoperiri utilizată de către comunitatea ştiinţifică. Trecând de la *un studiu naiv al realităţii la un studiu semi-fundamentat,* **este absolut necesar ca obiectele, conceptele, relaţiile manipulate să admită definiţii precise, ca anumite proprietăţi să fie demonstrate într-un cadru clar specificat, ca totul să poată fi făcut constructiv**. Într-un limbaj mai mult sau mai puţin apropiat de limbajul natural, **orice ştiinţă foloseşte afirmaţii 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 definirii adevărului la un nivel global***. În plus, între aceste *meta-concepte* există o legătură clar subliniată. Ele sunt utilizate, aşa după cum tocmai am amintit, atât în procesul de formalizare a (conţinutului) altor ştiinţe dar şi în studiile logice de bază, ceea ce *poate genera din nou dificultăţi de înţelegere provenite din dualitatea limbaj de bază – metalimbaj*.

Să trecem întâi în revistă conceptele de sistem deductiv şi teorie logică precum şi legătura dintre ele, la un ***nivel informal***. După cum deja cunoaştem, indiferent de tipul de logică (**LP**, **LP1**, **LP1**$_{=}$, **LP2**, etc.), din punct de vedere ***sintactic*** se porneşte cu un ***alfabet*** şi 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 şi 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 definiţie constructivă, care apelează la noţiunile de ***axiomă*** şi de ***regulă de inferenţă***. Axiomele sunt formulele plasate iniţial (prin **Baza** definiţiei) în subclasa corespunzătoare, iar regulile de inferenţă (de deducţie, de demonstraţie) reprezintă modalităţile prin care se obţin formule noi (numite şi ***teoreme***) din formule vechi (**Pasul inductiv**). *Cu ajutorul acestora se defineşte în mod formal,* **la nivel global**, *conceptul de **raţionament (demonstraţie)***. Un exemplu imediat de sistem deductiv este cel bazat pe rezoluţie. Astfel, în **LP** pornim cu o mulţime de clauze F (formulă aflată în **FNC**, generatoare de axiome) şi putem găsi Res*(F) utilizând rezoluţia într-un pas ca (unică *schemă* de) regulă de inferenţă (orice element din Res*(F) este „demonstrabil prin rezoluţie” pornind cu „axiomele” din F). O ***teorie logică*** este o (sub)clasă de formule ***închisă la consecinţă semantică***. Cu alte cuvinte, o mulţime T de formule este teorie logică dacă pentru fiecare submulţime $S \subseteq T$ şi fiecare (altă) formulă G care este consecinţă semantică din S, avem şi $G \in T$. Exemple imediate de teorii logice sunt constituite din clasele formulelor valide (din **LP**, **LP1**, **LP1**$_=$, etc.). *Cu ajutorul acestora se defineşte formal,* **la nivel global**, *conceptul de „**adevăr**”* (elemente lui T de mai înainte fiind receptate ca nişte formule „adevărate”). ***Legătura*** dintre teoriile logice şi sistemele de demonstraţie se exprimă prin

teoreme de corectitudine şi completitudine, adică teoreme de tipul: *Tot ceea ce este „adevărat” este demonstrabil (completitudine) şi tot ceea ce este demonstrabil este „adevărat” (corectitudine)*. Mai precis, se poate porni cu o teorie logică şi se poate încerca „axiomatizarea” ei (adică găsirea unui sistem deductiv prin care se „generează”, sintactic, aceeaşi clasă de formule), cu scopul ca mulţimea teoremelor să coincidă cu mulţimea formulelor „adevărate” (cele care formează teoria). Se poate şi invers, adică putem pleca cu un sistem deductiv pentru care putem afla „imediat” clasa teoremelor (clasa formulelor generate prin raţionamente specifice regulilor sale de inferenţă) şi 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 aşa de simplu cum au fost prezentate mai sus (din cauza lipsei de spaţiu 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 şi/sau recursiv enemerabile*, etc. Mai mult, orice teorie logică (în sens clasic) trebuie să conţină toate formulele valide (sau doar acele formule valide care au o formă sintactică specificată), pentru că *orice formulă validă este consecinţă semantică din orice altă clasă de formule*. Dacă o teorie logică conţine o contradicţie, atunci acea teorie coincide cu întreaga clasă precizată de formule (**LP**, **LP1**, $\mathbf{LP1}_{=}$, etc.), lucru care rezultă direct din definiţii. Nici în cazul sistemelor deductive nu dispunem de un context uşor de manipulat. Utilizarea unui *număr finit* de (*scheme de*) axiome şi reguli de inferenţă pentru „prezentarea” unor teorii logice poate fi imposibilă, existând şi un număr impresionant de *tipuri generale* de asemenea sisteme. De asemenea, forma în care sunt exprimate teoremele de corectitudine şi completitudine poate depinde în mod esenţial de sistemul deductiv sau de teoria aleasă (ba uneori chiar şi de alfabetul peste care este construită mulţimea de formule). Să reţinem faptul important că sistemele deductive pot fi utilizate „pentru ele însele”, ca mecanisme formale (simple *alternative pentru definiţiile constructive*). Cu ajutorul lor se pot astfel defini sintactic mulţimi care nu au întotdeauna o caracterizare semantică legată de o noţiune precisă de „adevăr”. Paragrafele următoare, în care detaliem câteva dintre aspectele menţionate, adoptă în mare linia sugerată de [CAZ1], *concepţia generală fiind însă originală.* De altfel, definiţiile *noţiunilor de sistem de demonstraţie şi teorie logică diferă uşor faţă de cele întâlnite în literatura de specialitate, datorită atât bagajului insuficient de cunoştinţe presupus a fi fost acumulat anterior parcurgerii acestui material de către cititor, cât şi datorită nevoii autorului de sistematizare şi generalizare a conţinutului*.

§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 şi o listă de asemenea elemente, sau, ieşind din sfera logicii, obiecte cu mult mai complicate decât* *formulele „clasice”*. Mulţimea **FORM** va fi precizată în mod explicit doar atunci când va fi necesar, singura cerinţă permanentă fiind ca aceasta să fie definită constructiv.

Definiţia 4.1 (sistem deductiv). Se numeşte *sistem deductiv* **(de demonstraţie, inferenţial, axiomatic)** în **FORM** un cuplu $\mathcal{SD} = <\mathcal{A}, \mathcal{R}>$ unde $\mathcal{A} \subseteq$ **FORM** este o mulţime de **axiome** iar $\mathcal{R} \subseteq \mathbf{FORM}^+ \times \mathcal{C}$ o mulţime de **reguli de inferenţă (de deducţie, de demonstraţie).** ■

În cele de mai sus, $\mathbf{FORM}^+$ denotă mulţimea relaţiilor de oricâte argumente (cel puţin unul) peste **FORM**, iar $\mathcal{C}$ reprezintă o mulţime de *condiţii de aplicabilitate*. Fiecare regulă de inferenţă $\mathsf{r} \in \mathcal{R}$, are astfel aspectul $\mathsf{r} = < < G_1, G_2, \dots , G_n, G>, \mathsf{c}>$, unde $n \in \mathbf{N}$, $G_1, G_2, \dots , G_n \in$ **FORM** şi $\mathsf{c} \in \mathcal{C}$.. $G_1, G_2, \dots , G_n$ sunt *ipotezele (premizele) regulii*, G reprezintă *concluzia (consecinţa)* iar c desemnează *cazurile (modalităţile) în care regula poate fi aplicată*. Vom scrie chiar $\mathsf{r} = < < \{G_1, G_2, \dots , G_n\}, G>, \mathsf{c}>$ deoarece *ordinea ipotezelor nu este esenţială*. Mulţimea $\mathcal{C}$ nu a fost specificată formal (putem spune totuşi că elementele sale sunt **metapredicate**) din cauza generalităţii ei şi pentru a nu complica inutil expunerea. Similar cu situaţia rezoluţiei, *regulile vor fi folosite pentru a construi demonstraţii în paşi succesivi, la un pas aplicându-se o regulă*. Există însă posibilitatea ca înafara restricţiilor 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 demonstraţiei, apariţia în demonstraţie a unei formule nedorite la acel pas, păstrarea completitudinii unei teorii, etc.). Astfel încât dacă c este ataşată unei reguli r (*atenţie,* c *poate lipsi, mai exact ea poate fi „condiţia adevărată indiferent de context”*) înseamnă că în **orice** demonstraţie, r va putea fi aplicată la un moment dat **doar dacă** c este adevărată la momentul respectiv. O regulă r = < < $\{G_1, G_2, \ldots, G_n\}$, G>, c>, va fi scrisă şi ca:

$$\frac{G_1, G_2, \ \ldots \ , G_n}{G}, \mathsf{c}$$

În cazul în care n = 0 şi c lipseşte, r poate fi identificată ca fiind o axiomă, după cum rezultă din definiţia care urmează. Câteodată, alături de c, sunt explicitate separat şi restricţiile sintactice locale asupra (formei) metaformulelor.

Definiţia 4.2 (demonstraţie). Fie $\mathcal{SD}$ = <$\mathcal{A}$, $\mathcal{R}$> un sistem deductiv în **FORM**. Se numeşte *demonstraţie* **(pentru F_m, pornind cu F_1) în $\mathcal{SD}$** o listă de metaformule $\mathcal{D}$ = $F_1, F_2, \ldots, F_m$ astfel încât pentru fiecare $i \in [m]$, fie $F_i \in \mathcal{A}$, fie F_i este obţinut din $F_{j_1}, F_{j_2}, \ldots, F_{j_k}$ folosind o regulă r = < < $\{F_{j_1}, F_{j_2}, \ldots, F_{j_k}\}$, F_i>, c> $\in \mathcal{R}$, unde $j_1, j_2, \ldots, j_k < i$. ■

Prin urmare, fiecare element al listei $\mathcal{D}$ este fie o axiomă, fie este concluzia unei reguli de inferenţă ale cărei ipoteze sunt elemente anterioare din listă. Analogia cu cele deja fixate în **Capitolul 2** despre demonstraţiile prin rezoluţie (**Definiţia 2.10** este un caz particular al definiţiei precedente) este clară şi aceasta poate fi continuată prin definirea *numărului de paşi ai unei demonstraţii, reprezentarea unei demonstraţii printr-un arbore*, etc. O demonstraţie se va mai numi şi **deducţie (sintactică)** sau chiar **raţionament**. După cum am precizat, **putem defini** $\mathcal{D}$ de mai sus, constructiv, **ca fiind un arbore cu rădăcina** $\mathbf{F_m}$, în care frunzele (**Baza**) sunt axiome şi fiecare nod nou (concluzie) se obţine din noduri vechi (ipoteze), în timpul aplicării **Pasului inductiv**, folosindu-se câte o regulă de inferenţă 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 condiţiile c o impun, arborele să aibă o definiţie directă mai simplă, sau, alternativ, *noţiunea de consecinţă sintactică să fie definită cu ajutorul arborelui (necoincizând cu el).*

Definiţia 4.3 (teoreme). Fie $\mathcal{SD} = <\mathcal{A}, \mathcal{R}>$ un sistem deductiv în **FORM**. **Mulţimea teoremelor lui** $\mathcal{SD}$ este mulţimea metaformulelor care admit demonstraţii în $\mathcal{SD}$, adică:

$\mathcal{T}h(\mathcal{SD}) = \{F \in$ **FORM** | există o demonstraţie $\mathcal{D}$ pentru F în $\mathcal{SD}\}$. ■

*Este imediat faptul că şi $\mathcal{T}h(\mathcal{SD})$ admite o definiţie constructivă (similar cu **Teorema 2.11**, din cazul rezoluţiei, se arată că $\mathcal{T}h(\mathcal{SD})$ coincide cu mulţimea dată mai jos):*

Baza. $\mathcal{A} \subseteq \mathcal{T}h(\mathcal{SD})$.

Pas constructiv. *Dacă* $r = < < \{G_1, G_2, \dots, G_n\}, G>, c> \in \mathcal{R}$ *şi* $G_1, G_2, \dots, G_n \in \mathcal{Th}(\mathcal{SD})$ *atunci* $G \in \mathcal{Th}(\mathcal{SD})$.

Faptul că există o demonstraţie pentru F în $\mathcal{SD}$ va mai fi notat prin $\square_{\mathcal{SD}}$ F (renunţând şi la indice în cazul în care nu există confuzii). Din motive tehnice, legate în general de demonstraţiile teoremelor de corectitudine şi completitudine, este posibil ca în anumite situaţii să lucrăm, pe lângă axiome, cu o mulţime suplimentară de metaformule, notată *I*, şi să vorbim despre **demonstraţii folosind *I*** (notat $I \square_{\mathcal{SD}}$ F, în cazul în care este vorba despre o formulă F; se mai spune că **F este consecinţă sintactică din *I*** sau **teoremă în ipotezele *I***). Practic, *deşi din punctul de vedere al noţiunii de „adevăr", sensul semantic al celor două mulţimi*($\mathcal{A}$ şi *I*) *nu coincide întotdeauna*, folosirea mulţimii de ipoteze suplimentare *I* nu înseamnă altceva decât să lucrăm exact ca mai înainte, dar în sistemul $\mathcal{SD}' = <\mathcal{A}', \mathcal{R}>$, unde $\mathcal{A}' = \mathcal{A} \cup I$. Teoremele vor apare astfel ca fiind consecinţe sintactice din mulţimea vidă de ipoteze.

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

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

sisteme de demonstraţie dedicate, **câte unul pentru fiecare formulă F aflată în FNC** şi reprezentată ca o mulţime finită de clauze (conform teoremei de compactitate, F poate fi şi infinită). Sistemele dedicate pot fi caracterizate de faptul că nu generează, în general, clasa formulelor valide (nici măcar o submulţime a acesteia), de aceea teoremele de corectitudine şi completitudine au şi 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 afirmaţie, trebuie să specificăm pe rând:

- **Clasa metaformulelor**: **FORM** = **LP**. **Formulele care ne interesează sunt însă doar clauzele din LP**. Notăm mulţimea tuturor clauzelor din **LP** cu $\mathcal{Cl}$.
- **Mulţimea de axiome**: $\mathcal{A}_F = \emptyset$. Am putea considera drept axiome chiar mulţimea F. Nu facem acest lucru din considerente conceptuale, încă netransparente. Diferenţa la nivel sintactic dintre axiome şi ipotezele suplimentare nu există deocamdată, iar în ceea ce priveşte demonstraţiile vom lucra (practic) cu sisteme de tipul $\mathcal{SD}'$.
- **Mulţimea de reguli de inferenţă**:
  $\mathcal{R}_F = \{<<\{C_1, C_2\}, Res(C_1, C_2)>, \text{true}> \mid C_1, C_2 \in \mathcal{Cl}$ şi $C_1 \neq C_2$ şi există măcar un literal L astfel încât $L \in C_1$ şi $\overline{L} \in C_2\}$.

În acest moment, sistemul $\mathcal{SD}_F = <\mathcal{A}_F, \mathcal{R}_F>$ este pe deplin precizat (true denotă *condiţia mereu adevărată*, adică regulile, dacă respectă cerinţele sintactice fixate şi anume faptul că sunt clauze din **LP**, pot fi aplicate fără restricţii pe parcursul demonstraţiilor), exceptând faptul deja amintit că vom pune (renunţăm la indici şi la faptul că ar trebui folosită notaţia $\mathcal{SD}$' în loc de $\mathcal{SD}$)

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

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

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

Pentru fiecare C_1, $C_2 \in \mathcal{Cl}$, *dacă* $C_1 \neq C_2$ *şi există măcar un literal* L *astfel încât* L $\in C_1$ *şi* $\overline{\mathrm{L}} \in C_2$, *atunci:*

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

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

Exerciţiul 4.1. *Arătaţi că metoda rezoluţiei de bază pentru* **LP1** *poate fi privită ca sistem de demonstraţie.*

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

- **Axiome** ($\mathcal{A}_{\mathbf{SD3}}$). Condiţiile sintactice sunt: F, G, H $\in$ **LP1**, x $\in \mathcal{X}$, t $\in \mathcal{T}$, oarecare. Suplimentar, în 4., x trebuie să nu apară liber în F iar în 5., substituţia s = [x/t] trebuie să fie permisă pentru F (reamintim, t nu conţine nume de variabile care să apară legate în F):
    1. $F \to (G \to F)$.
    2. $(F \to (G \to H)) \to ((F \to G) \to (F \to H))$.
    3. $(\rceil F \to \rceil G) \to ((\rceil F \to G) \to F)$.
    4. $(\forall x)(F \to G) \to (F \to (\forall x)G)$.
    5. $(\forall x)F \to (F)[x/t]$.

  Să remarcăm faptul că **LP1** trebuie considerată ca fiind construită peste alfabetul care conţine drept conectori doar pe $\rceil$ şi $\to$, iar unicul cuantificator acceptat este $\forall$. Dacă dorim să utilizăm şi ceilalţi conectori (sau cuantori), putem face acest lucru doar utilizându-i ca notaţii (de exemplu, $A \vee B$ va reprezenta $\rceil A \to B$, etc.).
- **Reguli de inferenţă ($\mathcal{R}_{\mathbf{SD3}}$).** Există doar restricţii de natură sintactică (lipsind condiţiile de aplicabilitate): F, G $\in$ **LP1,** x $\in \mathcal{X}$ sunt oarecare, dar în 2., x trebuie să nu apară liber în F. Prima schemă de regulă este deja amintită, şi anume

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

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

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

Să arătăm, de exemplu, că în **SD3** se poate genera teorema T = (A → A) (în cele ce urmează vom mai renunţa pe parcurs la unele paranteze, dacă înţelegerea nu este afectată, deşi formal acest lucru nu este admis). Astfel, folosim întâi instanţa axiomei (schemei) 1., obţinută dacă luăm F = A şi G = (A → A). Primul element al listei care reprezintă demonstraţia va fi: E_1 = A → ((A → A) → A). Folosim acum axioma 2., punând F = A, G = (A → A) şi H = A. Obţinem:

E_2 = (A → ((A → A) → A)) → ((A → (A → A)) → (A → A)).

Aplicăm acum **(MP)** pentru E_2 = F → G şi E_1 = F (se observă imediat că G = (A → (A → A)) → (A → A)) şi găsim:

E_3 = (A → (A → A)) → (A → A).

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

E_4 = A → (A → A).

În sfârşit, putem folosi **(MP)** pentru E_3 şi E_4 (luând F = A → (A → A) şi G = (A → A)), pentru a obţine ceea ce doream, adică:

E_5 = T = (A → A).

Prin urmare, am găsit în **SD3** demonstraţia $\mathcal{D}$: E_1, E_2, E_3, E_4, E_5 = T şi putem spune că (A → A) ∈ $\mathcal{Th}$(**SD3**) sau că T este consecinţă sintactică din mulţimea vidă de formule suplimentare. În plus, aceasta este evident o formulă validă. Cum şi T este de fapt o schemă, rezultă şi că ($\rceil A \rightarrow \rceil A$) este teoremă, etc.

Profităm de exemplul în curs pentru a schiţa **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 inferenţă sunt *corecte* (termenul corespunzător în engleză este ***sound, adică „sănătos”***). Acest lucru înseamnă: *presupunem că ipotezele G_1, G_2, ... G_k sunt formule valide şi arătăm că şi concluzia G este formulă validă* (atenţie, nu este nimic contractoriu 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 $G_1 \wedge G_2 \wedge ... \wedge G_k \rightarrow G$ este validă*. De exemplu, **(MP)** este corectă, deoarece avem:

F	G	$F \to G$	$(F \to G) \wedge F$	$((F \to G) \wedge F) \to G$
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 definiţia constructivă a lui $\mathcal{Th}$(**SD3**) (formal, este vorba de o demonstraţie prin inducţie structurală, de altfel foarte simplă). ■

Exerciţiul 4.2. *Arătaţi că axiomele 2. – 5. sunt formule valide şi 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 **consistenţă**. Astfel, o mulţime de metaformule $\mathcal{J}$ este consistentă într-un sistem deductiv, dacă nu există nici o metaformulă F astfel încât să avem atât $\mathcal{J}$ □ F cât şi $\mathcal{J}$ □ F ($\mathcal{J}$ □ F notează faptul că nu este adevărat că $\mathcal{J}$ □ F). Prin extensie, un sistem deductiv este consistent (**necontradictoriu**) dacă nu există nici o metaformulă F astfel încât să avem atât □ F cât şi □ F. O altă proprietate importantă este cea a **minimalităţii** (**independenţei**). Astfel, în anumite situaţii este important ca un sistem să conţină *cât mai puţine axiome şi reguli de* *inferenţă*, deşi acest lucru s-ar putea să conducă la existenţa unor demonstraţii mai lungi şi mai alambicate. Înafara sensului strict de minimalitate (lucru care depinde şi de alfabetul peste care este construită **FORM**), dintr-un sistem dat se pot elimina acele axiome care sunt consecinţe semantice din altele (în cazul considerării unei noţiuni suport de adevăr) precum şi aşa- numitele ***reguli de inferenţă derivate***. Astfel, considerând orice prefix al oricărei demonstraţii (privită textual) $\mathcal{D}$ dintr-un sistem $\mathcal{SD}$, acesta poate fi considerat ca o nouă regulă de inferenţă („derivată” din cele iniţiale): *concluzia noii reguli este ultima formulă din demonstraţia respectivă, iar ipotezele sunt reprezentate de restul formulelor care apar*. Eliminările de reguli derivate se „aprobă” doar dacă se menţine *echivalenţa* din sistemul iniţial şi cel rezultat (după efectuarea eliminărilor). *Două sisteme $\mathcal{SD}$ şi $\mathcal{SD}1$ sunt* ***echivalente*** *dacă pentru fiecare mulţime de metaformule $\mathcal{J}$ şi fiecare metaformulă F avem: $\mathcal{J} \square_{\mathcal{SD}} F$ dacă şi numai dacă $\mathcal{J} \square_{\mathcal{SD}1} F$*. Nu vom detalia nici acest subiect. Vom reveni totuşi asupra câtorva aspecte suplimentare atât în **§2.** şi **§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 accepţiuni ale ultimului termen chiar printre logicieni. **Vom accepta definiţia care urmează din considerente legate de programarea logică**. Astfel, există numeroase *lumi* (părţi ale realităţii, colecţie de cunoştinţe, etc.) care sunt cunoscute pur şi simplu, dar fiind foarte complexe este greu de spus dacă anumite cunoştinţe nou asimilate fac parte din aceeaşi lume, sau chiar dacă anumite cunoştinţe vechi nu sunt cumva contradictorii. Dorind să ne menţinem în cadrul general folosit până în prezent, introducerea unei noţiuni 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 ataşată şi o ***clasă de structuri admisibile de adevăr,*** notată $\mathcal{S}tr$, o structură fiind o funcţie $\mathcal{S}$: **FORM** $\rightarrow$ **B**. Dacă **FORM** admite o definiţie structurală, aşa cum de altfel am şi 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 şi că fiecare $\mathcal{S}$ **este unica extensie homomorfă** a unei structuri definite iniţial pe mulţimea metaformulelor atomice. În acest mod, se pot păstra toate definiţiile (conceptele) semantice folosite până în prezent, fără modificări de esenţă (inclusiv conceptul de *consecinţă semantică*).

Definiţia 4.4 (teorii logice). Se numeşte *teorie (logică)* orice subclasă $\mathcal{TE}$ a lui **FORM** închisă la consecinţă semantică. ■

În modul sugerat trebuie înţeleasă reprezentarea prin metaformule a unei baze de cunoştinţe. 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 $\mathcal{TE} \subseteq$ **FORM**, trebuie **să găsim o submulţime** $\mathcal{A}' = \mathcal{A} \cup \mathcal{I} \subseteq \mathcal{TE}$**, de „axiome" şi/sau ipoteze suplimentare, precum şi o mulţime de reguli de inferenţă** $\mathcal{R}$ **(adică un sistem de demonstraţie** $\mathcal{SD}' = <\mathcal{A}', \mathcal{R}>$**) astfel încât** $\mathcal{TE} = \mathcal{Th}(\mathcal{SD}')$. În acest caz, se impune de obicei ca $\mathcal{A}'$ să fie măcar o mulţime satisfiabilă (există măcar o structură $\mathcal{S}$ astfel încât pentru fiecare F $\in \mathcal{A}'$ avem $\mathcal{S}$(F) = 1), sau chiar ca ea să fie alcătuită numai din metaformule valide (după cum am mai observat, *dacă* $\mathcal{A}'$ *conţine o contradicţie, orice metaformulă este consecinţă semantică din* $\mathcal{A}'$). Forma generală a lui $\mathcal{A}'$ se explică tocmai prin aceea că *am presupus că* $\mathcal{A}$ *conţine formulele* ***valide*** *iar* $\mathcal{I}$ *pe cele* ***satisfiabile***). Mai general, să presupunem că pornim cu o mulţime de metaformule $\mathcal{A}' \subseteq$ **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** $\mathcal{R}$ **astfel încât să avem** $\mathcal{Cs}(\mathcal{A}') = \mathcal{Th}(\mathcal{SD}')$ (am notat cu $\mathcal{Cs}(\mathcal{A}')$ mulţimea tuturor consecinţelor semantice din $\mathcal{A}'$, în raport cu noţiunea de adevăr adoptată, şi cu $\mathcal{SD}'$ sistemul compus din $\mathcal{A}'$ şi $\mathcal{R}$). După cum am mai amintit, putem lua în considerare şi situaţia inversă, în care avem dat un sistem $\mathcal{SD}' = <\mathcal{A}', \mathcal{R}>$ şi **dorim să vedem dacă $\mathcal{Th}(\mathcal{SD}')$ este într-adevăr o teorie logică, sau, mai mult, dacă $\mathcal{Cs}(\mathcal{A}') = \mathcal{Th}(\mathcal{SD}')$**.

Definiţia 4.5. Un sistem de demonstraţie $\mathcal{SD}' = <\mathcal{A}', \mathcal{R}>$ se numeşte **corect şi complet** pentru o teorie $\mathcal{TE}$ dacă $\mathcal{TE} = \mathcal{Th}(\mathcal{SD}') = \mathcal{Cs}(\mathcal{A}')$ şi $\mathcal{A}' \subseteq \mathcal{TE}$. O teorie $\mathcal{TE}$ este **axiomatizabilă** dacă există un sistem deductiv $\mathcal{SD}' = <\mathcal{A}', \mathcal{R}>$ corect şi complet pentru ea, adică satisfăcând condiţiile anterioare. Dacă $\mathcal{SD}'$ este finit specificabil (axiomatizabil), atunci teoria corespunzătoare se numeşte **finit axiomatizabilă**. ■

În cele de mai sus, dacă I este mulţimea vidă atunci $\mathcal{TE}$ este alcătuită doar din metaformule valide. În cazul teoriilor „reale”, I cuprinde în general cunoştinţele primare ale lumii respective, iar $\mathcal{A}$ axiomele „logice” (de genul celor „puse” în **SD3**). Din păcate, în această ultimă situaţie, $\mathcal{A}$ conţine în marea majoritate a cazurilor şi axiomele egalităţii, astfel încât aceste teorii sunt în general nedecidabile (a se vedea **SAT1** pentru **LP1$_=$**). Alte tipuri de teorii se obţin prin „translarea” proprietăţilor sistemului deductiv ataşat, dacă acesta există (de exemplu, putem vorbi de teorii consistente, teorii decidabile sau semidecidabile, etc.). Importante sunt teoriile *nedegenerate*. O teorie logică se numeşte **degenerată** dacă coincide cu mulţimea vidă sau coincide cu întreaga clasa de metaformule în care se lucrează, **FORM** (de exemplu, dacă teoria conţine o contradicţie 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* $\mathcal{A}'$ = $\mathcal{TE}$ *şi* $\mathcal{R}$ = ∅). Înainte de a prezenta câteva exemple, să tragem concluzia necesară asupra formei generale a unei teoreme de corectitudine şi completitudine în noul context.

TEOREMĂ DE CORECTITUDINE ŞI COMPLETITUDINE. Fie o clasă de metaformule **FORM**, o clasă de structuri admisibile $\mathcal{Str}$ pentru **FORM**, un sistem deductiv $\mathcal{SD}'$ = <$\mathcal{A}'$, $\mathcal{R}$> în **FORM**, unde $\mathcal{A}' = \mathcal{A} \cup I$ ($\mathcal{A}$ fiind alcătuită din formule valide şi I din formule satisfiabile) şi o teorie logică $\mathcal{TE} \subseteq$ **FORM**, astfel încât $\mathcal{TE} = \mathcal{C}s(\mathcal{A}')$. Atunci $\mathcal{T}h(\mathcal{SD}') = \mathcal{C}s(\mathcal{A}')$. ■

Observaţie. A demonstra corectitudinea înseamnă a arăta că $\mathcal{T}h(\mathcal{SD}') \subseteq \mathcal{C}s(\mathcal{A}')$ iar completitudinea, că $\mathcal{T}h(\mathcal{SD}') \supseteq \mathcal{C}s(\mathcal{A}')$. Teorema se mai poate enunţa şi sub una din formele echivalente, destul de des întâlnite:

- Teoria $\mathcal{TE}$ admite un sistem deductiv corect şi complet.
- În condiţiile precizate, avem, pentru fiecare metaformulă F∈ **FORM**: I □$_{\mathcal{SD}}$F dacă şi numai dacă I □ F.
- Teoria $\mathcal{TE}$ 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$\in$ **FORM**, avem: $\square_{\mathcal{SD}}$F dacă şi numai dacă $\square$F.■

În cele de mai sus am folosit notaţia $\square_{\mathcal{SD}}$F pentru faptul că F$\in\mathcal{Th}(\mathcal{SD})$, unde $\mathcal{SD}$ = <$\mathcal{A}$, $\mathcal{R}$>, sau, în momentul în care $\mathcal{SD}$ este implicit sau liseşte, $\square$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 privinţa numeroaselor rezultate (importante şi interesante) care pot fi demonstrate în acest moment. Se poate consulta [CAZ1] pentru completări, iar noi ne limităm la a prezenta fără demonstraţie (de fapt, metateorema este sigur adevărată în cazul **LP1** şi sistemului **SD0**):

Teorema deducţiei (pe scurt, TD). Pentru fiecare A, B $\in$**FORM** şi fiecare I $\subseteq$ **FORM**, dacă I, A $\square$ B atunci I $\square$ A $\rightarrow$ B. ■

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

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

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

Prin urmare, teoria $\mathcal{V}al$(**LP1**) *este chiar finit axiomatizabilă* şi putem testa validitatea unei formule încercând să-i găsim o demonstraţie. $\mathcal{V}al$(**LP1**) nu este decidabilă, dar este semidecidabilă (putem „enumera” toate demonstraţiile posibile şi alege apoi pe cea dorită). *Teoria este şi consistentă (necontradictorie)*. Mai mult, *axiomele lui* **SD3** *sunt independente şi* $\mathcal{R}$ *nu conţine reguli derivate*. Cu toate aceste calităţi, am văzut că nici demonstraţiile în **SD3** nu sunt chiar simple, ca să nu vorbim de găsirea unor algoritmi eficienţi de „enumerare” a acestora. ■

Exemplul 4.1 (reluat). În această situaţie nestandard, avem de-a face cu **FORM** = **LP1** (de fapt, cu **FORM** = $\mathcal{C}l$). Apoi, fiecărei mulţimi (eventual finite) de clauze din **LP1**, notată F (fiecare clauză fiind la rândul ei o mulţime finită de literali), îi putem asocia mulţimea Res*(F) care desigur *nu este o teorie logică în acest caz*. De altfel, nici *F nu este (întotdeauna) o mulţime satisfiabilă (de ipoteze suplimentare)*. În plus, nici nu ne interesează „adevărul” ci „neadevărul”. Metoda rezoluţiei este de altfel o exemplificare perfectă a unei situaţii despre care am amintit deja: sistemul deductiv (aici, $\mathcal{SD}_F$) este folosit ca definiţie constructivă, pentru generarea unei mulţimi (Res*(F)), care nu este, şi nici nu vrem să fie, o teorie logică. Putem admite că această mulţime are totuşi o caracterizare semantică (în sensul că o submulţime a sa, F, este sau nu satisfiabilă). Teorema de corectitudine şi 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 aşa-numitor *teorii (matematice) formale* ([CAZ1]). Şi aici se începe cu un sistem deductiv „de bază”, conţinând, de obicei, ca axiome, axiomele logice, adică cele ale oricărui sistem corect şi complet pentru $\mathcal{V}al$(**LP1**), cum ar fi **SD3**, la care se adaugă axiomele egalităţii, împreună cu nişte „axiome” specifice (acestea din urmă constituind de fapt mulţimea suplimentară de ipoteze). Mulţimea teoremelor pentru sistemul amintit (de tip $\mathcal{SD}'$) nu va coincide însă cu $\mathcal{V}al$(**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 mulţimi (de exemplu, cum ar fi cele care constituie o structură de grup împreună cu o operaţie dată). Structurile admise conţin în plus şi simboluri cu interpretări standard, cum ar fi egalitatea, totul fiind similar cu ceea ce am făcut în cazul „trecerii" de la **LP1** la **LP1**$_=$. Cadrul standard prezetat anterior se păstrează, inclusiv enunţul teoremei de corectitudine şi completitudine (*care, de fapt, este imediat adevărată, în general admiţându-se prin convenţie faptul că „teoria" în cauză este tocmai mulţimea toremelor din sistemul deductiv ales*). Sensul noţiunilor 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**$_{\{=,\ \bullet,\ 1,\ \sim\}}$, simbolurile specificate în mulţimea din indice reprezentând egalitatea (simbol predicativ binar), operaţia de grup (simbol funcţional binar), elementul neutru (constantă funcţională), respectiv operaţia de simetrizare (simbol funcţional unar). Deja putem spune care este clasa $\mathcal{S}tr$, a structurilor admisibile $\mathcal{S} = <\mathcal{U}_{\mathcal{S}}, I_{\mathcal{S}}>$: $\mathcal{U}_{\mathcal{S}}$ va fi orice mulţime D dotată cu o lege de compoziţie internă $\circ$ astfel încât $<D, \circ>$ formează grup; în oricare asemenea $\mathcal{S}$, $=$ 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 faţă de legea $\bullet$. Sistemul $\mathcal{SD}_{\mathbf{TG}}$ va fi dat astfel de:

Axiome ($\mathcal{A}$). Mulţimea lor este reuniunea celor două mulţimi descrise mai jos. Să punctăm faptul că de multe ori *axiomele egalităţii* sunt incluse printre axiomele logice şi că uneori sunt utilizate variante ale acestora (în funcţie de scopul pentru care sunt introduse). Nu toate axiomele egalităţii amintite sunt folosite în mod direct în **TG** (asemenea axiome particulare există şi printre axiomele grupului).

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

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. $(\forall_*)(x_1 = y_1 \wedge x_2 = y_2 \wedge ... \wedge x_n = y_n \rightarrow$
$f(x_1, x_2, ... , x_n) = f(y_1, y_2, ... , y_n))$,
pentru fiecare $n \in$ **N*** şi fiecare $f \in \mathcal{F}_n$.

10. $(\forall_*)(x_1 = y_1 \wedge x_2 = y_2 \wedge ... \wedge x_n = y_n \rightarrow$
$P(x_1, x_2, ... , x_n) = P(y_1, y_2, ... , y_n))$,
pentru fiecare $n \in$ **N*** şi fiecare $P \in \mathcal{P}_n$.

Axiomele grupului (sunt de fapt ***I*, ipotezele suplimentare;** câteodată axiomele egalităţii 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. $(\forall x)(\mathbf{1} \bullet x = x)$.

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

Reguli de inferenţă: (**MP**) şi (**RG**) de la **SD3**.

Vom reveni cu un exemplu de demonstraţie în teoria grupurilor după ce vom introduce sistemul **SD0**, *echivalent* cu **SD3** (**Exerciţiul 4.5**). Datorită acestui ultim fapt, vom putea folosi şi *deducţia 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 şi practic.

§3. Clasificarea sistemelor deductive

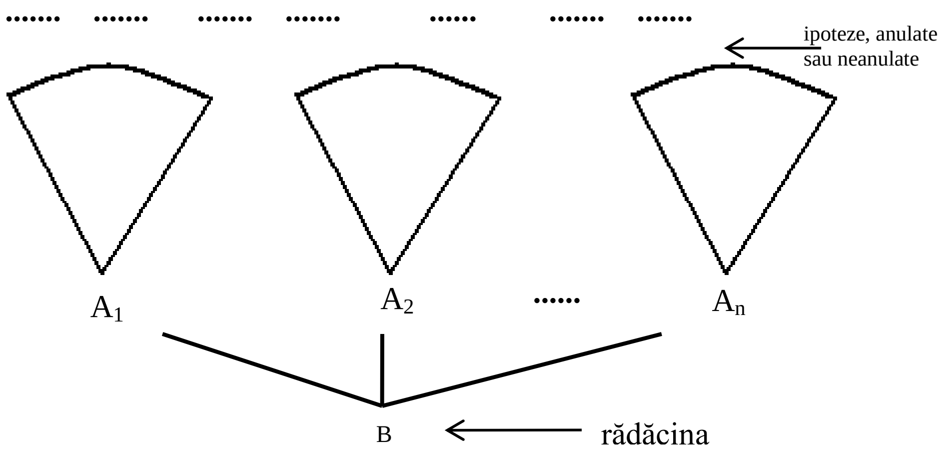
Începem cu o trecere în revistă a unor posibilităţi de clasificare ([CAZ1]) a sistemelor deductive, din care se pot trage concluzii utile cu privire la calităţile şi defectele unor asemenea sisteme. *Clasificările prezentate (care nu sunt singurele acceptate) se referă în principal la sistemele standard*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

- **În funcţie de *conectivele logice* alese**. Există **sisteme boolean complete** sau **boolean incomplete**. Ştim că în interpretarea conectorilor logici prin structuri aceştia devin funcţii booleene. Termenul de (in)completitudine se referă la faptul că mulţimea interpretărilor conectorilor aleşi în alfabetul de bază peste care este construit sistemul formează (sau nu) o mulţime completă de funcţii (în sensul **Capitolului 2**). Sistemul **SD3** este boolean complet, în timp ce un sistem care, de exemplu, foloseşte drept conector doar $\rightarrow$ (rezultând aşa-numitul *calcul implicaţional*), va fi boolean incomplet.

- **În funcţie de *relaţia* 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 şi complete, conform definiţiei noastre. După cum am mai precizat, completitudinea este mai greu de atins (demonstrat) şi de aceea este de multe ori doar *adoptată prin convenţie*. Corectitudinea este de obicei impusă, deşi poate avea şi nişte forme mai deosebite (vezi sistemele nestandard, cum ar fi rezoluţia).
- **În funcţie de *importanţa* acordată axiomelor sau regulilor de inferenţă**. Din acest punct de vedere, se poate acorda o atenţie deosebită regulilor (adică modului de raţionament, de obţinere de cunoştinţe noi) în dauna axiomelor (cunoştinţelor primare). Acest tip de sisteme se numesc **Gentzen-Jaskowski**. Un asemenea sistem va fi **SD0** (**deducţia naturală**), care este echivalent cu **SD3** şi va fi prezentat în acest capitol. În cazul în care balanţa este inversată (există „mult mai multe” axiome decât reguli de inferenţă, ca în cazul **SD3**), sistemele sunt cunoscute sub numele de **sisteme Hilbert**.
- **După *clasa* FORM aleasă**. De exemplu (pentru logica clasică), putem avea **sisteme propoziţionale** sau **sisteme predicative**.

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

Exemplu (deducţia naturală, sistemul SD0). Clasa **FORM** este **LP1**. Alfabetul conţine în acest caz doar conectorii $\rceil$, $\wedge$ şi cuantificatorul $\forall$. După cum am precizat, într-un asemenea sistem regulile de inferenţă sunt „mai importante” decât axiomele, sistemul **SD0 neavând chiar nici o axiomă**. Pentru a simplifica înţelegerea, **vom defini** ***direct*** **o** ***demonstraţie în sensul de deducţiei naturale*** **ca fiind** ***un anumit arbore*** (vezi mai jos), fără a folosi definiţia generală. Un ***arbore de deducţie naturală*** are pe nivelul 0 (în frunze) formule oarecare (ipoteze ale unor reguli de inferenţă din sistem, inclusiv elemente din eventuala mulţime suplimentară *I*), iar nivelele următoare se obţin constructiv, conform definiţiei generale (rădăcina fiind „rezultatul final”). **Caracteristic acestui sistem este faptul că acele condiţii 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 consecinţă ştergerea unui nod din graf, ceea ce conduce întotdeauna la anumite complicaţii tehnice), vom adopta soluţia de a *marca ipotezele anulate* (cu cifre, de exemplu). Dacă se doreşte, pentru evitarea confuziilor, mărcile pot fi diferite, în cazul în care regulile aplicate sunt diferite şi/sau dacă sunt aplicate la momente diferite. Tot pentru evitarea confuziilor, aceeaşi marcă se va asocia şi nodului care constituie concluzia regulii care, aplicată, a cauzat anularea. Ipotezele anulate modifică însă clasa de demonstraţii acceptate într-un asemenea sistem: avem $I \vdash_{\mathbf{SD0}}$ G (*G este consecinţă sintactică în **SD0** utilizând mulţimea suplimentară de formule I; sau, există o demonstraţie pentru G în **SD0** utilizând I; sau, există o deducţie naturală pentru G în **SD0***) dacă există un arbore de deducţie naturală având rădăcina G şi **cu toate ipotezele neanulate aparţinând lui *I*.** Grafic:



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

Pentru a prezenta concret sistemul, rămâne să dăm **regulile de inferenţă din care este alcătuit**, care vor primi şi un nume înafara numărului de secvenţă. Vom avea câte o (schemă de) regulă pentru fiecare A, B ∈ **LP1**, fiecare x ∈ $\mathcal{X}$ şi fiecare t ∈ $\mathcal{T}$. *În 5., este necesar ca* *substituţia [x/t] să fie permisă pentru A, iar în 6., ca x să nu apară liber în nici o ipoteză neanulată*. Schemele 3. şi 4. au variante datorită necesităţii de a se „prinde” comutativitatea conjuncţiei la nivel sintactic (ne vom referi la ele ca 3’., respectiv 4’.). Deoarece substituţia [x/x] este permisă pentru orice formulă, regula 5. are şi forma particulară <<{(∀x)A}, A>, true> (care va fi notată 5’.). Să remarcăm şi faptul că regula 6. nu are nevoie de nici o restricţie sintactică în momentul în care se lucrează cu formule închise. Mnemonicele provin de la următoarele cuvinte: **E** – eliminare; **I** – introducere; **N** – negaţie; **C** – conjuncţie.

1. (**EN**) $\dfrac{B, \neg B}{A}$, c: *se anulează ipoteza* $\neg A$.

2. (**IN**) $\dfrac{B, \neg B}{\neg A}$, c: *se anulează ipoteza* A.

3. (**EC**) $\dfrac{A \wedge B}{A}$ şi $\dfrac{A \wedge B}{B}$.

4. (**IC**) $\dfrac{A, B}{A \wedge B}$ şi $\dfrac{A, B}{B \wedge A}$.

5. (**E∀**) $\dfrac{(\forall x)A}{A[x/t]}$.

6. (**I∀**) $\dfrac{A}{(\forall x)A}$.

■

Demonstraţia teoremelor următoare poate fi găsită în [CAZ1].

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

Teorema 4.3. Sistemele **SD0** şi **SD3** sunt echivalente, adică pentru fiecare mulţime de formule închise $\mathcal{J} \subseteq$ **LP1** şi fiecare formulă $F \in$ **LP1**, avem: $I \vdash_{\mathbf{SD0}} F$ dacă şi numai dacă $I \vdash_{\mathbf{SD3}} F$. ■

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

7. (**ED**) $\dfrac{A \vee B, \rceil A}{B}$ şi $\dfrac{A \vee B, \rceil B}{A}$.

8. (**ID**) $\dfrac{A}{A \vee B}$ şi $\dfrac{A}{B \vee A}$.

9. (**EI**) $\dfrac{A, A \rightarrow B}{B}$.

10. (**II**) $\dfrac{B}{A \rightarrow B}$, C: *se anulează ipoteza A*.

11. (**EE**) $\dfrac{A \Leftrightarrow B}{A \rightarrow B}$ şi $\dfrac{A \Leftrightarrow B}{B \rightarrow A}$.

12. (**IE**) $\dfrac{A \rightarrow B, B \rightarrow A}{A \Leftrightarrow B}$.

13. (**E∃**) $\dfrac{(\exists x)A, B}{B}$, C: *se anulează ipoteza A din subarborele având rădăcina* ***acest*** *B*.

14. (**I**∃) $\dfrac{A[x/t]}{(\exists x)A}$.

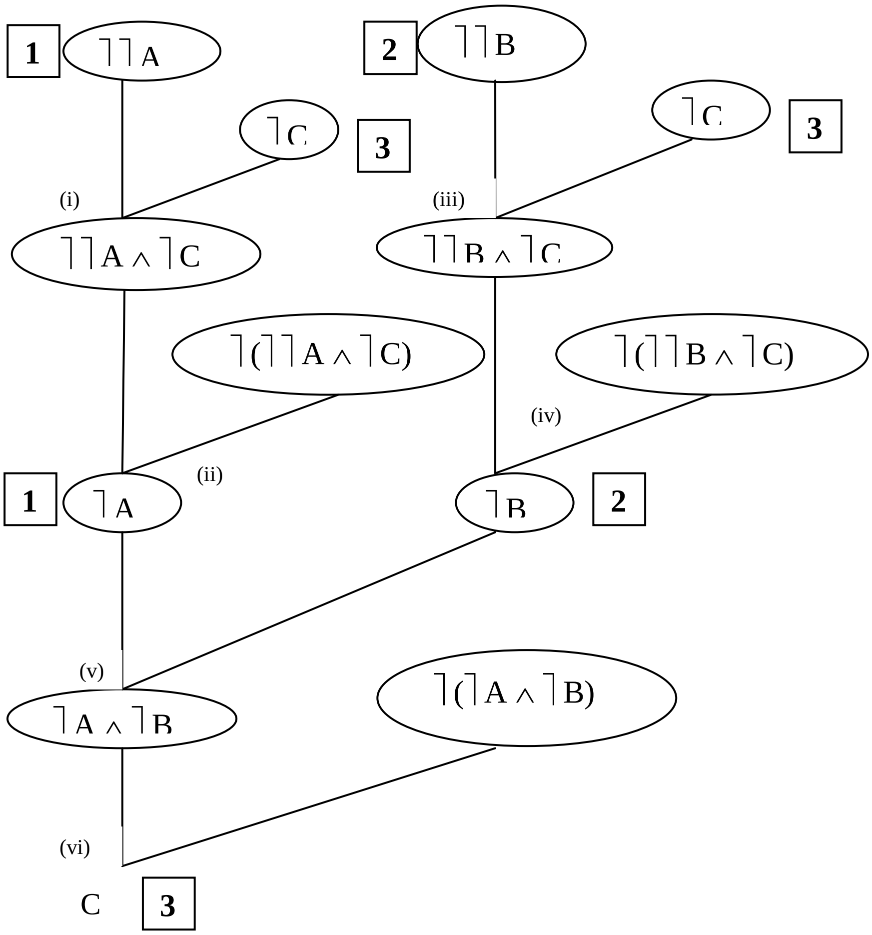
15. (**DN**) $\dfrac{A}{\neg\neg A}$ şi $\dfrac{\neg\neg A}{A}$.

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

Exemplu. Să se arate că avem

$\neg(\neg\neg A \wedge \neg C), \neg(\neg\neg B \wedge \neg C), \neg(\neg A \wedge \neg B)$ □ C, în **SD0**.

Într-adevăr, putem construi arborele de deducţie:



■

Observaţie. În arborele de mai sus, cifrele denotă, aşa cum am mai precizat, o marcă aplicată ipotezelor care trebuie anulate în momentul aplicării unei anumite reguli. Paşii de aplicare i-am notat distinct, prin ((i) – (vi)), ca de altfel şi mărcile. De exemplu, la primul pas de deducţie (notat (i)), se folosesc drept ipoteze formulele oarecare ⌉⌉A şi ⌉C, pentru a aplica *o instanţă* a regulii (**IC**) (în definiţia generală cele două formule sunt denotate prin A, respectiv B). Această regulă nu are ataşată nici o condiţie de aplicare. Apoi, la pasul (ii), se continuă construcţia demonstraţiei (arborelui) prin aplicarea regulii (**IN**), care are ca ipoteze formulele $\rceil\rceil A \wedge \rceil C$ şi $\rceil(\rceil\rceil A \wedge \rceil C)$ (în descrierea generală ele sunt notate B, respectiv $\rceil B$) şi drept concluzie formula $\rceil A$, având şi condiţia de aplicare c = *se anulează ipoteza A. Marca* ***1*** *a fost aplicată, conform celor stabilite, nodului concluzie al aplicării regulii şi ... lui* $\rceil\rceil A$*, în loc de A!* Acest lucru nu este permis, tehnic vorbind (de fapt, pentru a fi foarte exacţi, suntem în culpă şi cu utilizarea strictă a parantezelor; soluţia aici este simplă, introducându-se de la bun început şi variante „cu paranteze" pentru toate regulile). Pentru a aplica corect teoria, ne putem folosi de instanţe ale regulii derivate (**DN**) (deducem $\rceil\rceil A$ din A, înainte de efectuarea primului pas (i) şi astfel putem anula în (ii) ceea ce trebuie, adică pe A). Pe parcursul construcţiei arborelui se mai întâlnesc asemenea cazuri şi devine evidentă necesitatea de a avea la dispoziţie o „bibliotecă" de reguli derivate, pentru a lucra simplu cu un asemenea tip de deducţie naturală. *Acestea trebuie însă demonstrate în prealabil*, lucru care nu este chiar uşor (totuşi, se pare că este mai uşor de conceput un „mecanism automat de raţionament" pentru **SD0** decât pentru **SD3**). O alternativă, bazată de fapt pe aceeaşi idee (utilizarea unor reguli derivate), este să folosim formulele din I sub o formă echivalentă, de exemplu $I = \{A \rightarrow C, B \rightarrow C, A \vee B\}$. ■

Exerciţiul 4.3 *(*[CAZ1]*). Fie $A \to C$, $B \to C$, $A \vee B$ şi C formule oarecare din* **LP1** *şi $I \subseteq$* **LP1**. *Să presupunem că $I \;\square_{SD0}\; A \to C$, $B \to C$, $A \vee B$ (adică fiecare dintre formulele din membrul drept este consecinţă sintactică din I în **SD0**). Arătaţi că $I \;\square_{SD0}\; C$ (aceasta se numeşte în logică **metoda de demonstraţie prin disjuncţia cazurilor**).*

Exemplu (calculul cu secvenţe, sistemul SD1). Pornim iniţial cu **LP1**, construit peste un alfabet care conţine toţi conectorii şi cuantificatorii cunoscuţi (desigur că unii dintre ei pot fi adoptaţi prin notaţie, dar îi vom folosi fără restricţie): $\rceil$, $\wedge$, $\to$, $\leftrightarrow$, $\forall$, $\exists$. Se numeşte **secvenţă** orice formulă care are forma: $A_1 \wedge A_2 \wedge \ldots \wedge A_m \to B_1 \vee B_2 \vee \ldots \vee B_n$, unde $n, m \in$ **N**, $A_1, A_2, \ldots, A_m, B_1, B_2, \ldots, B_n \in$ **LP1** (m, n pot fi şi egali cu 0, dar nu simultan). Prin urmare, vom lucra practic cu clauze, dar notaţia pe care o vom adopta ne conduce la ideea că *secvenţele sunt de mai degrabă metaformule (alt tip de obiect, oricum mai complex) decât formulele cu care am fost familiarizaţi în capitolele anterioare*. Astfel, vom scrie o secvenţă sub forma (desigur că în cele ce urmează $\Rightarrow$ nu are sensul de metaimplicaţie):

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

Mai mult, vom considera cei doi membri ai relaţiei de mai sus ca fiind mulţimi (atunci când ordinea elementelor va fi esenţială, vom specifica explicit acest lucru). Prin urmare, **o secvenţă va fi de forma *U* $\Rightarrow$ *V* (*U* şi *V* pot fi şi mulţimea vidă, dar nu simultan)** şi vom putea scrie *U'* = *U*, A în loc de *U'* = *U* ∪ {A}, în ideea că, din anumite motive, elementul A din *U'* trebuie pus în evidenţă. Vom extinde notaţia la submulţimi oarecare, adică vom putea scrie (de exemplu) *V*, *W* în loc de *V* ∪ *W* şi *V*, A, B în loc de *V* ∪ {A} ∪ {B}. Astfel, punem **FORM** = {*U* | *U* este secvenţă în **LP1**}. Sistemul **SD1**, după cum vom vedea, deşi atribuit în principal lui Gentzen (1934) şi având o singură schemă de axiome (este drept, foarte generală), se apropie (în privinţa modalităţii de utilizare) mai mult de un sistem de tip Hilbert. Sistemele deductive bazate pe secvenţe au şi o răspândită utilizare în situaţii nestandard (legate numai de definirea constructivă a unor mulţimi „în mod axiomatic”, fără referire la „adevăr”). Mai concret, **SD1** este un *sistem predicativ finit specificat şi boolean complet*, având:

Axiome. Pentru fiecare *U*, *V* ∈ **FORM** şi pentru fiecare A ∈ **LP1**:

$$U, \mathrm{A} \Rightarrow V, \mathrm{A}.$$

Reguli de inferenţă. Schemele de mai jos (care, din nou, sunt numerotate dar au ataşat şi un nume mnemonic care nu mai necesită explicaţii - exceptând poate (**RT**) care înseamnă regula tăieturii) sunt valabile pentru fiecare *U*, *V* ∈ **FORM**, fiecare A, B ∈ **LP1**, fiecare x ∈ $\mathcal{X}$ şi fiecare t ∈ $\mathcal{T}$. În regula 5., substituţia [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 „;”. Atenţie la faptul că regulile 5. şi 7. au o infinitate de premize (de exemplu**,** *U*, (A)[x/t] ⇒ *V* din (∀⇒) trebuie înţeles ca reprezentând *U*, $(\mathrm{A})[x/t_1]$ ⇒ *V***;** *U*, $(\mathrm{A})[x/t_2]$ ⇒ *V***;** ... , adică se iau în considerare *toate* elementele t din $\mathcal{T}$, pentru care [x/t] este permisă pentru A; rolul lui A în **(RT)** este similar, instanţele unei scheme referindu-se la celelalte elemente având statutul de a fi „oarecare”.

1. ($\rceil \Rightarrow$) $\dfrac{U \Rightarrow V, \mathrm{A}}{U, \rceil\mathrm{A} \Rightarrow V}$.

2. ($\wedge \Rightarrow$) $\dfrac{U, \mathrm{A}, \mathrm{B} \Rightarrow V}{U, \mathrm{A} \wedge \mathrm{B} \Rightarrow V}$.

3. ($\Rightarrow \rceil$) $\dfrac{U, \mathrm{A} \Rightarrow V}{U \Rightarrow V, \rceil\mathrm{A}}$.

4. ($\Rightarrow \wedge$) $\dfrac{U \Rightarrow V, \mathrm{A}; U \Rightarrow V, B}{U \Rightarrow V, \mathrm{A} \wedge \mathrm{B}}$.

5. ($\forall \Rightarrow$) $\dfrac{U, (\mathrm{A})[\mathrm{x/t}] \Rightarrow V}{U, (\forall x)\mathrm{A} \Rightarrow V}$.

6. ($\Rightarrow \forall$) $\dfrac{U \Rightarrow V, A}{U \Rightarrow V, (\forall x)\mathrm{A}}$.

7. (**RT**) $\dfrac{U, \mathrm{A} \Rightarrow V; U \Rightarrow V, A}{U \Rightarrow V}$.

Conform cadrului general fixat, o metaformulă (aici, o secvenţă) $U \Rightarrow V$ este teoremă în **SD1** dacă există un arbore (care *reprezintă o demonstraţie* a lui $U \Rightarrow V$) având în rădăcină pe $U \Rightarrow V$ şi axiome în frunze (*în acest sens însă, sistemul doar generează constructiv mulţimea teoremelor*). Dacă vrem să legăm **SD1** de $\mathcal{Val}$ (**LP1**) şi să enunţăm o teoremă de corectitudine şi completitudine, trebuie să **interpretăm relaţia de consecinţă sintactică** $\boldsymbol{I}$ □ **F (**$\boldsymbol{I} \subseteq$ **LP1, F** $\in$ **LP1) prin: există** $\mathbf{G_1, G_2, \ldots, G_n} \in \boldsymbol{I}$ **astfel încât** $\mathbf{G_1, G_2, \ldots, G_n \Rightarrow F} \in \mathcal{Th}$**(SD1). În particular,** □$_{\mathbf{SD1}}$ **F în seamnă că**

metaformula $\Rightarrow$ F (adică Ø $\Rightarrow$ F) este teoremă în SD1. Deşi se poate enunţa o teoremă de corectitudine şi completitudine mai generală (oricum *sistemul este corect*; demonstrarea completitudinii este mai dificilă datorită regulilor cu un număr infinit de premize; pe de altă parte, se poate arăta că *putem renunţa la* **(RT)**, fără a se reduce *puterea deductivă* a sistemului, demonstraţiile fiind în schimb mai lungi), ne limităm la a aminti că avem:

Teorema 4.4. Fie orice F $\in$ **LP1**. Atunci:
□F dacă şi numai dacă □$_{\mathbf{SD1}}$ F. ■

Sistemul **SD1** are avantajul, faţă de **SD3** şi **SD0**, că putem deriva imediat un „procedeu automat” ([CAZ1]) de construire a unei demonstraţii (arbore). Să presupunem astfel că avem de demonstrat secvenţa $U \Rightarrow V$ în **SD1**. Alegem un nod, care este frunză (iniţial, acesta nu va putea fi decât rădăcina, adică nodul care conţine $U \Rightarrow V$), şi anume secvenţa care se află plasată în el, o formulă din secvenţă şi operaţia logică principală a formulei (conectorul sau cuantorul care furnizează o primă împărţire a formulei în subformule). Se aplică acum una dintre regulile 1. – 7., obţinându-se unul sau (maxim) două noi noduri, apoi se reia alegerea unei frunze. Procedeul se opreşte atunci când frunzele conţin 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 conţin secvenţe având mai puţine operaţii logice). Desigur că ceea ce am descris mai sus nu este încă un algoritm, care să poată prelucra întrega clasă de secvenţe (nu este nici măcar un semialgoritm), datorită numărului infinit de premize din 5. şi 7. Vom ilustra totuşi procedeul printr-un exemplu, imediat după ce dăm şi câteva reguli de inferenţă derivate (presupunem că U, $V \in$ **FORM**, A, B $\in$ **LP1**, x $\in \mathcal{X}$, t $\in \mathcal{T}$ 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 premize ca în 5., valoarea concretă a lui t furnizând doar o instanţă a regulii în ansamblu*; totuşi, în 15., substituţia [x/t] trebuie să fie permisă pentru A).

Reguli de inferenţă derivate pentru **SD1**:

8. $(\vee\rightarrow)$ $\qquad \dfrac{U, \mathrm{A} \Rightarrow V; U, \mathrm{B} \Rightarrow V}{U, \mathrm{A} \vee \mathrm{B} \Rightarrow V}$.

9. $(\rightarrow\vee)$ $\qquad \dfrac{U \Rightarrow V, \mathrm{A}, \mathrm{B}}{U \Rightarrow V, \mathrm{A} \vee \mathrm{B}}$.

10. $(\rightarrow\Rightarrow)$ $\qquad \dfrac{U \Rightarrow V, \mathrm{A}; U, \mathrm{B} \Rightarrow V}{U, \mathrm{A} \rightarrow \mathrm{B} \Rightarrow V}$.

11. $(\Rightarrow\rightarrow)$ $\qquad \dfrac{U, \mathrm{A} \Rightarrow V, \mathrm{B}}{U \Rightarrow V, \mathrm{A} \rightarrow \mathrm{B}}$.

12. $(\leftrightarrow\Rightarrow)$ $\qquad \dfrac{U, \mathrm{A}, \mathrm{B} \Rightarrow V; U \Rightarrow V, \mathrm{A}, \mathrm{B}}{U, \mathrm{A} \leftrightarrow \mathrm{B} \Rightarrow V}$.

13. $(\Rightarrow\leftrightarrow)$ $\qquad \dfrac{U, \mathrm{A} \Rightarrow V, \mathrm{B}; U, \mathrm{B} \Rightarrow V, \mathrm{A}}{U \Rightarrow V, \mathrm{A} \leftrightarrow \mathrm{B}}$.

14. $(\exists\Rightarrow)$ $\qquad \dfrac{U, A \Rightarrow V}{U, (\exists \mathrm{x}) A \Rightarrow V}$.

15. $(\Rightarrow\exists)$ $\qquad \dfrac{U \Rightarrow V, \mathrm{A}[\mathrm{x}/\mathrm{t}]}{U \Rightarrow (\exists \mathrm{x})\mathrm{A}, V}$.

Exemplu. Regula (↔⇒) se poate obţine pe baza deducţiei (metaformulele U, B ⇒ V, B şi U, A ⇒ V, A sunt axiome):

$$\dfrac{\dfrac{\dfrac{U, A, B \Rightarrow V \qquad U, B \Rightarrow V, B}{U, A \to B, B \Rightarrow V} \qquad \dfrac{U, A \Rightarrow V, A \qquad U \Rightarrow V, A, B}{U, A \to B \Rightarrow V, A}}{U, A \to B, B \to A \Rightarrow V}}{U, (A \to B) \wedge (B \to A) \Rightarrow V}$$

■

Exerciţiul 4. 4 *(*[CAZ1]*). Arătaţi că formula*

$(A \vee C) \wedge (A \to B) \wedge (C \to D) \to (B \vee D) \in$ **LP1**

este demonstrabilă în ***SD1****.*

Exerciţiul 4.5 *(*[CAZ1]*). Se consideră teoria grupurilor* ***TG*** *şi sistemul deductiv asociat,* $\mathcal{SD}_{TG}$*. Fie u, v* $\in \mathcal{T}$ *termi oarecare. Să se arate că u = v → v = u este teoremă în* ***TG****, folosind deducţia 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 negaţia poate fi plasată* ***doar imediat în faţa*** *unei formule atomice*. Pentru acesta, se poate demonstra mai uşor posibilitatea eliminării unei reguli de tipul **(RT)**.

§4. Recapitulare şi Index

Subiectul capitolului curent este vast şi de un deosebit interes, atât pentru filozofi, matematicieni şi logicieni, cât şi pentru cercetătorii din alte domenii ştiinţifice, inclusiv informaticienii. Prin conceptul de ***teorie logică***, noţiunea de adevăr este tratată la nivel global, ca reflectare a unei părţi *coerente* a realităţii. O *bază iniţială de cunoştinţe*, alcătuită din afirmaţii presupuse a fi *adevărate* (într-un sens bine precizat), şi prezentată sintactic ca o mulţime de ***(meta)formule***, poate fi ulterior completată cu noi afirmaţii (cunoştinţe), despre care nu se poate şti aprioric că „reflectă aceeaşi realitate”. Utilizând un procedeu standard (obţinerea de ***consecinţe semantice***), se pot afla chiar **toate** afirmaţiile „adevărate” în contextul respectiv. Din punct de vedere algoritmic, problema aflării tuturor consecinţelor semantice dintr-o mulţime dată de formule şi chiar *problema de decizie* mai simplă *Este formula F o consecinţă semantică din mulţimea* $\mathcal{G}$?, 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, exponenţiali ca timp de execuţie, urmează că problema este în fapt *netratabilă*). *Abordările sintactice* ale rezolvării problemelor au cel puţin 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, algoritmii generali bazaţi pe sintaxă (deşi *nu neapărat mai eficienţi*), sunt mai flexibili, mai uşor de tranformat şi adaptat, mai uşor de extins şi de a fi aplicaţi şi în alte situaţii/contexte. ***Sistemele deductive*** formalizează cele de mai sus, apelând la concepte cum ar fi ***axiomă***, ***regulă de inferenţă***, ***teoremă, consecinţă sintactică***. Legătura dintre cele două noţiuni generale se stabileşte în mod concret prin ***teoreme de corectitudine şi completitudine***. Chiar dacă uneori suntem nevoiţi să renunţăm la demonstrarea completitudinii sau chiar la ataşarea unui concept de „adevăr" formulelor, *sistemele de demonstraţie pot fi importante prin ele însele*. Astfel, înafara exemplelor considerate, standard sau nestandard (SD0, SD1, SD3, etc.), mai poate fi amintit şi aşa-numitul *calcul cu tabele* ([OHL]), folositor înafara contextului teoriilor logice.

Teoriile logice şi sistemele de demonstraţie constituie prin urmare cadrul formal prin care pot fi studiate într-un mod foarte precis părţi ale realităţii prin prisma oricărei ştiinţe (vezi de exemplu ***teoriile matematice formale***), conceptele menţionate în primele capitole nemaifiind ambigue. Pentru logicieni subiectul este inepuizabil şi foarte atrăgător, obţinându-se şi astăzi noi rezultate surprinzătoare. Informaticii îi revine sarcina de „ţine pasul" cu noile descoperiri în domeniu, de a adapta algoritmii existenţi şi conceptele legate de programarea logică, urmărindu-se eficientizarea acesteia şi apropierea de idealurile *Inteligenţei artificiale*.

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

teorie logică, *191*
sistem deductiv (de demonstraţie), *191*
axiomă, *191*
regulă de inferenţă (reguli corecte sau sound), 191
demonstraţie (consecinţă sintactică; deducţie; raţionament), *191*
metaformulă, *193*
sistem deductiv standard, 200
regulă de inferenţă derivată, *204*
axiomatizarea unei teorii logice, 207
teorie logică nedegenerată, consistentă, finit axiomatizabilă, *207*
teoremă de corectitudine şi completitudine, *208*
sistem deductiv corect şi complet, *208*
sistem deductiv necontradictoriu (consistent), 210
sistem deductiv independent şi minimal, *210*
sistem deductiv boolean complet, 214-215
sistem deductiv propoziţional sau predicativ, *214-215*
sistem deductiv de tip Hilbert, 215
sistem deductiv de tip Gentzen, 215
deducţie naturală, *215*
sisteme deductive echivalente, 218

sistem deductiv finit specificat, 219

calculul cu secvenţe, 223

§5. Exerciţii

1. Să se demonstreze în sistemul **SD3** următoarele formule ([CAZ1]):

(i) $(A \to B) \to ((B \to C) \to (A \to C))$.

(ii) $(A \to (B \to C)) \to (B \to (A \to C))$.

(iii) $\rceil\rceil A \to A$.

2. Rezolvaţi **Exerciţiul 4.1**.
3. Rezolvaţi **Exerciţiul 4.2**.
4. Rezolvaţi **Exerciţiul 4.3**.
5. Rezolvaţi **Exerciţiul 4.4**.
6. Rezolvaţi **Exerciţiul 4.5**.
7. Fie sistemul deductiv **SIN1** ([CAZ1]), dat prin axiomele şi regulile de mai jos.

Axiome. Pentru fiecare A, B, C, D $\in$ **LP**, avem:

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

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

Ax3. $A \wedge B \to A$

Ax4. $A \wedge B \to B$

Ax5. $A \to (B \to A \wedge B)$

Ax6. $A \to A \vee B$

Ax7. $B \to A \vee B$

Ax8. $(A \rightarrow D) \rightarrow ((B \rightarrow D) \rightarrow (A \vee B \rightarrow D))$

Ax9. $(A \leftrightarrow B) \rightarrow (A \rightarrow B)$

Ax10. $(A \leftrightarrow B) \rightarrow (B \rightarrow A)$

Ax11. $(A \rightarrow B) \rightarrow ((B \rightarrow A) \rightarrow (A \leftrightarrow B))$

Ax12. $(A \rightarrow \urcorner A) \rightarrow \urcorner A$

Ax13. $\urcorner A \rightarrow (A \rightarrow B)$

Reguli de inferenţă. Acestea sunt (**MP**) şi **regula substituţiei**, (**RS**). Schema este valabilă pentru fiecare F, G_1, G_2, ... , $G_n \in$ **LP**. În cele de mai jos, $F(A_1, A_2, ... , A_n)$ denotă faptul că F este construită peste (exact) mulţimea de variabile propoziţionale $\{A_1, A_2, ... , A_n\}$, iar $F(G_1, G_2, ... , G_n)$ denotă o formulă F în care fiecare apariţie a lui A_1 se în locuieşte cu G_1, fiecare apariţie a lui A_2, cu G_2, ş. a. m. d. , fiecare apariţie a lui A_n, se înlocuieşte cu G_n, *simultan*:

$$(\mathbf{RS}) \qquad \frac{F(A_1, A_2, ..., A_n)}{F(G_1, G_2, ..., G_n)}$$

Acesta este un sistem deductiv propoziţional standard, de tip Hilbert, construit peste **LP** care are toţi conectorii standard. Sistemul este boolean complet, dar **incomplet** din punctul de vedere al $\mathcal{Val}$(**LP**). Astfel, **se poate arăta că formule valide cunoscute, cum ar fi $\urcorner\urcorner A \rightarrow A$ (*legea negării negaţiei*) sau $A \vee \urcorner A$ (*legea terţiului exclus*) nu sunt teoreme în SIN1**. Ca observaţie suplimentară, să spunem că dacă păstrăm doar Ax1. – Ax.11. (împreună cu (**MP**) şi (**RS**)), sistemul mai poartă numele de ***calculul propoziţional pozitiv*** (datorat lui Hilbert). Acesta este desigur tot incomplet faţă de $\mathcal{Val}$ (**LP**), dar are avantajul că în el putem demonstra *tot ceea ce avem nevoie, în ipoteza că nu vrem să folosim negaţia*.

Arătaţi că:

(i) Sistemul este corect faţă de $\mathcal{Val}$(**LP**).

(ii) $(A \to B) \to ((A \to \rceil B) \to \rceil A)$, *legea reducerii la absurd*, este teoremă în **SIN1**.

8. Daţi o axiomatizare a teoriei formale a grupurilor, fără a folosi simbolurile care desemnează elementul neutru şi operaţia de simetrizare.