Universitatea “Alexandru Ioan Cuza”
Facultatea de Informatică
Departamentul de Invăţământ la Distanţă

Ferucio Laurenţiu Ţiplea

FUNDAMENTELE ALGEBRICE ALE INFORMATICII

2005–2006

Adresa autorului: Universitatea “Al. I. Cuza”
Facultatea de Informatică
Str. Berthelot 16
700483 - Iaşi, România
E-mail: fltiplea@mail.dntis.ro
Adresa Web: http://www.infoiasi.ro/fltiplea

Prefaţă

Materialul de faţă constituie parte integrantă a cursurilor predate de autor studenţilor Facultăţii de Informatică a Universităţii “Al.I.Cuza” din Iaşi, începând cu 1996, fiind conceput ca suport de curs.

Accentul cade în principal pe asimilarea conceptelor de bază necesare înţelegerii elementelor de informatică ce implică noţiuni cu caracter matematic, cum ar fi cele de funcţie, relaţie, inducţie matematică şi structurală, recursie, mulţime parţial ordonată, latice etc. Ultimele două capitole conţin aplicaţii consistente în informatică (criptografie şi semantica limbajelor de programare). Fiecare capitol conţine propoziţii şi leme ale căror demonstraţii sunt simple exerciţii lăsate în seama cititorului.

O extensie completă a acestui material poate fi găsită în [187].

Iaşi, 4 Octombrie, 2005
Ferucio Laurenţiu Ţiplea

Cuprins

Capitolul 1

Mulţimi, relaţii, funcţii

1.1 Mulţimi

In această secţiune vom prezenta câteva concepte de bază de teoria mulţimilor, concepte ce vor fi utilizate pe parcursul acestei lucrări. Pentru detalii cititorul este îndrumat către [82, 89, 182].

1.1.1 Ce este o mulţime?

Conceptul Cantorian de mulţime. Conceptul de mulţime stă la temelia matematicii moderne, fiind un concept larg utilizat în orice domeniu. Teoria mulţimilor (studiul abstract al mulţimilor) aşa cum o utilizăm astăzi a fost iniţiată de George Cantor în ultimul sfert al secolului 19 [23]. Abordarea lui a condus însă la *contradicţii* (numite şi *paradoxuri*), remediul "aparent" al acestora fiind abordarea axiomatică.

Conform definiţiei date de Cantor, prin *mulţime* înţelegem

> *orice colecţie de obiecte distincte şi bine definite ale intuiţiei şi gândirii noastre, considerate ca un tot* (*întreg, ca o unitate*).

Noţiunea de mulţime trebuie considerată ca un concept primitiv, suficient de bine înţeles intuitiv, care nu este precis definit dar care poate fi utilizat în definirea altor concepte particulare. Aşadar, motivaţi de "definiţia" lui Cantor, să considerăm că o mulţime este o colecţie de "obiecte" numite *elementele* mulţimii şi, să presupunem că există măcar o mulţime.

Dacă A este o mulţime şi a este un obiect (arbitrar) atunci a poate fi sau nu în mulţimea A. In primul caz vom folosi exprimarea a ***este element al mulţimii*** A sau a ***aparţine mulţimii*** A sau a ***este conţinut în*** A sau A ***conţine*** a şi vom nota $a \in A$; vom scrie $a \notin A$ dacă a nu este element al mulţimii A [1].

Menţionăm explicit că nu considerăm noţiunea de obiect ca o noţiune primară a teoriei mulţimilor. Aşadar, avem libertatea în a gândi ceea ce este un obiect. De exemplu, putem gândi anumite mulţimi ca fiind obiecte componente ale altei mulţimi.

[1] Simbolul "$\in$" a fost folosit pentru prima dată de matematicianul Giuseppe Peano [143], el fiind de fapt o variaţie grafică a primei litere a cuvântului grecesc "$\epsilon\sigma\tau\iota$" ce înseamnă "este".

Fie R mulţimea

$$R = \{x | x \text{ este mulţime şi } x \notin x\}.$$

Conform definiţiei lui Cantor, R este mulţime. Mulţimea numerelor naturale este element al mulţimii R. Este mai dificil de găsit un exemplu de mulţime ce nu este element al mulţimii R, dar aceasta nu are nici o importanţă relativ la statutul de mulţime a lui R. Insă, constatăm că are loc

$$R \in R \text{ dacă şi numai dacă } R \notin R,$$

ceea ce constituie un paradox. Acesta este aşa-numitul *paradox al lui Russell* [194].

Este natural să ne întrebăm atunci care este cauza ce conduce la acest paradox. Analizând modul de definire a mulţimii R constatăm că aceasta este bazată pe următorul principiu numit şi **Axioma abstracţiei** sau a **construcţiei de mulţimi**, introdusă de G. Frege în 1893 [58]:

> **Axioma abstracţiei** Dată o proprietate ce poate fi sau nu îndeplinită de obiecte, există o mulţime ce constă exact din obiectele ce satisfac această proprietate.

In cazul mulţimii R, proprietatea este

$$P(x): \quad x \text{ este mulţime şi } x \notin x.$$

Ca urmare, $R = \{x | P(x)\}$. La o primă analiză nu ar trebuie să fie nimic rău în a construi mulţimi printr-o astfel de axiomă. Multe mulţimi în matematică se construiesc în acest mod. De exemplu, considerând proprietatea

$$P'(x): \quad x \text{ este număr natural impar mai mic decit } 10,$$

obţinem mulţimea $A = \{1, 3, 5, 7, 9\}$. Diferenţa dintre construcţiile celor două mulţimi constă în aceea că mulţimea A este obţinută prin selectarea obiectelor dintr-o mulţime dată (cea a numerelor naturale) prin intermediul unei proprietăţi, ceea ce nu se întâmplă în cazul mulţimii R.

Descoperirea de paradoxuri în teoria Cantoriană a mulţimilor a avut efecte din cele mai neplăcute pentru mulţi matematicieni care şi-au bazat studiile şi cercetările pe o astfel de teorie. De exemplu, Richard Dedekind care începuse în 1888 să publice din studiile sale asupra teoriei numerelor – studii ce utilizau din plin teoria lui Cantor –, a fost nevoit să oprească pentru o perioadă publicarea acestora [2]. Mai tragic a fost însă cu lucrarea în două volume a lui Gottlob Frege, despre bazele aritmeticii, care tocmai fusese terminată [58] şi care utiliza Axioma abstracţiei. In cel de-al doilea volum, când Frege luase deja cunoştinţă de paradoxul lui Russell, acesta a inserat o anexă din care prezentăm mai jos un fragment (traducerea din original este după [61], pag. 234):

[2] In prefaţa la a 3-a ediţie a lucrării [38], sau în [40], pag. 449.

> "Hardly anything more unfortunate can befall a scientific writer than to have one of the foundations of his edifice shaken after the work is finished. This was the position I was placed in by a letter of Mr. Bertrand Russell, just when the printing of this volume was nearing its completion. It is a matter of my Axiom (V). I have never disguised from myself its lack of the self-evidence that belongs to the other axioms and that must be properly be demanded of a logical law ... I should gladly have dispensed with this foundation if I had known of any substitute for it. And even now I do not see how arithmetic can be scientifically established; how numbers can be apprehended as logical objects, and brought under review; unless we are permitted – at least conditionally – to pass from a concept to its extension. May I always speak of the extension of a concept – speak of a class? And if not, how are the exceptional cases recognized? ... These are the questions raised by Mr. Russell's communication."

Sistemul axiomatic Zermelo-Fraenkel. Intr-o perioadă s-a crezut că apariţia de paradoxuri în teoria creată de Cantor o pot distruge dar, aşa cum a remarcat David Hilbert, aceste paradoxuri nu au condus la altceva decât la "refondarea" acestei teorii păstrând "paradisul creat de Cantor" [3]. Refondarea teoriei a însemnat aşezarea ei pe baze axiomatice, logistice sau intuiţioniste. Dintre sistemele axiomatice propuse, sistemul *Zermelo-Fraenkel cu Axioma alegerii*, abreviat ZFC, este astăzi unul din cele mai utilizate sisteme. Restul acestei secţiuni va fi dedicat unei prezentări extrem de succinte a acestui sistem. Pentru detalii cititorul este îndrumat către [182] sau [187] (atragem atenţia asupra faptului că sistemul ZFC aşa cum este prezentat în [182] porneşte de la premisa că universul de discurs al variabilelor poate conţine obiecte ce nu sunt mulţimi, pe când abordarea pe care o vom prezenta în cele ce urmează pleacă de la premisa că universul de discurs al variabilelor conţine numai mulţimi. O discuţie detaliată asupra diferenţei dintre aceste abordări poate fi găsită în [182]).

Sistemul ZFC se construieşte peste logica cu predicate de ordinul întâi la care se adaugă două predicate noi, predicatul de *egalitate* şi predicatul binar de *apartenenţă*. Menţinăm încă de la început că apartenenţa este un predicat primitiv, care nu se defineşte. Obiectele au proprietatea de a aparţine sau nu mulţimilor, proprietate ce este primitivă.

Formulele atomice sunt de forma

$$x \in y \ \text{ şi } \ x = y,$$

pe baza cărora se construiesc noi formule prin intermediul operatorilor logici clasici şi a cuantificatorilor,

$$\varphi \wedge \psi, \ \ \varphi \vee \psi, \ \ \neg\varphi, \ \ \varphi \Rightarrow \psi, \ \ \varphi \Leftrightarrow \psi, \ \ (\forall x)\varphi \ \text{ şi } \ (\exists x)\varphi.$$

[3] "a paradise created by Cantor which nobody shall ever expel us" (conform cu [54], p.240).

Vom adopta notaţia $\varphi(x_1, \ldots, x_n)$ pentru a specifica că variabilele libere ale formulei φ sunt printre variabilele $x_1, \ldots, x_n$ (lăsând posibilitatea ca anumite variabile x_i să nu fie libere în φ sau chiar să nu apară în ea).

Un aspect fundamental îl constituie stabilirea domeniului obiectelor de studiu, domeniu din care vor lua valori variabilele, numit şi *universul de discurs*. Din punct de vedere a teoriei mulţimilor, proprietatea fundamentală prin care se poate face distincţie între obiectele universului de discurs este proprietatea de apartenenţă: un obiect poate *conţine* alte obiecte, sau nici unul. Dacă un obiect conţine obiecte atunci el va fi referit ca *mulţime*; altfel, ca *obiect individual*. Terminologia de "element" va însemna pentru noi "obiect al unei mulţimi". Este natural de a presupune că fiecare obiect al universului de discurs este element al unei mulţimi (măcar a mulţimii formate doar din obiectul în cauză). Intrebarea fundamentală ce se pune acum este următoarea: "câte" obiecte individuale, şi similar mulţimi, considerăm în univers? Trebuie să admitem că existenţa a cel puţin unui obiect este cerută atât de raţiuni filozofice cât şi practice; ea este necesară *fondării* universului. Pe de altă parte, gândind în avans la intersecţie de mulţimi, constatăm că am avea nevoie de un obiect care să reprezinte rezultatul intersecţiei a două mulţimi fără elemente comune. Este natural ca acest obiect să fie ales fără elemente, deci obiect individual, şi el să nu depindă de mulţimile sursă. Din considerente tehnice este important ca şi acest obiect să fie referit ca mulţime; ca urmare, prin mulţime vom înţelege acele obiecte ce conţin obiecte, sau acest obiect particular bine precizat (uzual, un astfel de obiect individual este numit *mulţimea vidă*). Vrem însă să accentuăm că în timp ce existenţa a cel puţin unui obiect individual este cerută de raţiuni filozofice şi practice, referirea la un obiect individual ca fiind mulţimea vidă este numai din raţiuni de convenienţă şi simplitate. Acum, avem de analizat următoarele două variante:

(1) universul de discurs conţine numai mulţimi şi, în plus, mulţimea vidă ca singur obiect individual [54, 89, 107, 190];

(2) universul de discurs conţine şi alte obiecte individuale pe lângă cel desemnat a fi mulţimea vidă [134, 15, 175, 97, 182].

Majoritatea sistemelor ZFC în varianta (1) asigură fondarea universului de discurs şi existenţa mulţimii vide

– ori prin intermediul unei axiome, uzual numită *Axioma existenţei* sau a *mulţimii vide*, care postulează existenţa unei mulţimi ce nu conţine nici un obiect (de exemplu, [82]),

– ori presupunând tacit că universul de discurs conţine cel puţin o mulţime (universul de discurs al variabilelor logicii de ordinul întâi trebuie să conţină cel puţin un obiect), de la care se deduce, pe baza Axiomei separării, existenţa unei mulţimi ce nu conţine nici un obiect (de exemplu, [107, 56]).

Ceea ce a ieşit în evidenţă este că, pentru scopuri matematice, postularea existenţei doar a unui singur obiect individual (mulţimea vidă) este suficientă [4]. Aceasta este

[4] Discuţii asupra acestui aspect pot fi găsite în [53, 136, 17]. Pe de altă parte, dacă sistemul ZFC în varianta (1) este consistent, atunci şi sistemul ZFC în varianta (2) este consistent [134, 106, 150].

varianta pe care o vom adopta şi noi (adică, (1)). Atragem atenţia asupra faptului că ne vom referi adesea la [182] unde este adoptată varianta (2) şi, ca urmare, trebuie acordată atenţie diferenţei care există în formularea unora dintre axiome.

Axiomele sistemului ZFC sunt următoarele:

1. **Axioma extensionalităţii** Două mulţimi A şi B sunt egale, şi notăm $A = B$, dacă pentru orice obiect x are loc:

$$x \in A \text{ dacă şi numai dacă } x \in B.$$

2. **Axioma de existenţă a mulţimii vide** Există mulţimi fără nici un element.

3. **Axioma separării** Pentru orice formulă $\varphi(x)$ şi mulţime U există o mulţime ce conţine toate elementele din U ce satisfac P, şi numai pe acestea.

4. **Axioma împerecherii** Pentru orice două obiecte a şi b (nu neapărat distincte) există o mulţime ce conţine obiectele a şi b, şi numai pe acestea.

5. **Axioma reuniunii** Pentru orice familie de mulţimi [5] $\mathcal{A}$ există o mulţime ce conţine elementele componente ale mulţimilor conţinute de $\mathcal{A}$, şi numai pe acestea.

6. **Axioma părţilor** Pentru orice mulţime A, există o mulţime ce conţine ca elemente toate submulţimile mulţimii A, şi numai pe acestea.

7. **Axioma regularităţii** Pentru orice mulţime nevidă A există $x \in A$ astfel încât $x \cap A = \emptyset$.

Există încă 3 axiome în sistemul ZFC, Axioma infinitului, a alegerii şi a înlocuirii. In Secţiunea 1.1.3 vom prezenta Axioma infinitului; celelalte două depăşesc cadrul lucrării şi vor fi omise (pentru detalii indicăm [182, 187]).

Vom trece acum la prezentarea câtorva proprietăţi fundamentale ce decurg de la aceste axiome.

In primul rând, egalitatea mulţimilor satisface proprietăţile:

- $A = A$; (*reflexivitate*)
- dacă $A = B$ atunci $B = A$; (*simetrie*)
- dacă $A = B$ şi $B = C$ atunci $A = C$. (*tranzitivitate*)

Definiţia 1.1.1.1. Fie A şi B două mulţimi.

(1) Spunem că A este *submulţime* a mulţimii B, şi notăm $A \subseteq B$, dacă orice element al mulţimii A este element al mulţimii B.

[5] O familie de mulţimi este o mulţime ale cărei elemente sunt mulţimi. Cum în abordarea considerată considerăm numai astfel de mulţimi, termenul de familie de mulţimi va fi utilizat pentru a atrage atenţia asupra acestui fapt, şi anume, că elementele familiei (mulţimii) considerate sunt mulţimi.

(2) Spunem că A este *submulţime proprie* a mulţimii B, şi notăm $A \subset B$, dacă $A \subseteq B$ şi $A \neq B$.

Dacă A nu este submulţime (submulţime proprie) a mulţimii B, atunci vom nota $A \nsubseteq B$ ($A \not\subset B$). Este clar că dacă $A \subset B$ atunci $A \subseteq B$.

Teorema 1.1.1.1. Fie A, B şi C mulţimi. Atunci, au loc următoarele proprietăţi:

(1) $A \subseteq A$;

(2) dacă $A \subseteq B$ şi $B \subseteq C$, atunci $A \subseteq C$;

(3) $A = B$ dacă şi numai dacă $A \subseteq B$ şi $B \subseteq A$;

(4) $A \subseteq B$ dacă şi numai dacă $A \subset B$ sau $A = B$;

(5) $A \not\subset A$;

(6) dacă $A \subset B$, atunci $B \not\subset A$;

(7) dacă $A \subset B$ şi $B \subseteq C$, sau $A \subseteq B$ şi $B \subset C$, atunci $A \subset C$.

Demonstraţie (1) şi (2) urmează direct de la Definiţia 1.1.1.1(1). (3) este o altă exprimare, utilizând incluziunea, a Axiomei extensionalităţii.

(4) Să presupunem că $A \subseteq B$. Dacă pentru orice $b \in B$ are loc $b \in A$, atunci $B \subseteq A$ şi, deci, $A = B$ (folosind (3)). Altfel, există $b \in B$ astfel încât $b \notin A$, ceea ce conduce la $A \subset B$ (conform definiţiei).

Reciproc, dacă $A \subset B$ atunci $A \subseteq B$ (de la definiţie), iar dacă $A = B$ atunci $A \subseteq B$ (de la (3)).

(5) Dacă am presupune prin contradicţie că $A \subset A$, atunci conform definiţiei ar exista $a \in A$ astfel încât $a \notin A$; contradicţie.

(6) $A \subset B$ conduce la existenţa unui element b în B care nu este în A; ca urmare, $B \not\subset A$.

(7) Să presupunem că are loc $A \subset B$ şi $B \subseteq C$. De la (4) şi (2) obţinem $A \subseteq C$. Dacă presupunem că $A = C$, atunci ipoteza se rescrie la $A \subset B$ şi $B \subseteq A$, ceea ce constituie o contradicţie. In mod similar se discută şi celălalt caz. □

Conform Teoremei 1.1.1.1(3), pentru a stabili egalitatea a două mulţimi A şi B avem de arătat că oricare din cele două mulţimi este inclusă în cealaltă. Această metodă de demonstraţie este numită adesea *demonstraţia prin dublă incluziune*.

In baza Axiomei extensionalităţii, orice două mulţimi fără nici un element sunt egale şi, deci, există o unică mulţime fără nici un element. Aceasta este numită *mulţimea vidă* şi este notată prin $\emptyset$. Este clar că $\emptyset$ este submulţime proprie a oricărei mulţimi diferite de ea însăşi şi, nici o mulţime nu este submulţime proprie a ei.

Axioma separării exclude paradoxul lui Russell formulat corespunzător acesteia. Adică, dacă considerăm o mulţime arbitrară U şi definim

$$R_U = \{x \in U | x \notin x\},$$

atunci contradicţia

$$R_U \in R_U \text{ dacă şi numai dacă } R_U \notin R_U$$

nu mai poate fi generată. In adevăr,

- dacă $R_U \in R_U$, atunci urmează $R_U \in U$ şi $R_U \notin R_U$; contradicţie;
- dacă $R_U \notin R_U$, atunci $R_U \notin U$ sau $R_U \in R_U$. Rezultă deci $R_U \notin U$.

Am obţinut astfel că $R_U \notin U$, adică, indiferent de ce mulţime U s-ar alege, mulţimea R_U nu este element al ei. Mai mult, are loc:

Teorema 1.1.1.2. Nu există nici o mulţime U care să conţină ca element orice mulţime.

Demonstraţie Pentru orice mulţime U, mulţimea R_U definită ca mai sus nu este element al mulţimii U. Ca urmare, nu poate exista o mulţime U care să conţină orice mulţime deoarece, atunci, ea ar trebui să conţină şi R_U. □

Teorema 1.1.1.2 conduce la faptul că nu există o mulţime a tuturor mulţimilor sau, altfel spus, clasa [6] tuturor mulţimilor este o clasă proprie.

Intersecţia unei familii nevide de mulţimi $\mathcal{A}$ este definită prin

$$\bigcap \mathcal{A} = \{a \in A | (\forall B \in \mathcal{A})(a \in B)\},$$

unde A este o mulţime arbitrară din $\mathcal{A}$. Este clar că $\bigcap \mathcal{A}$ există întotdeauna (cu cerinţa ca $\mathcal{A}$ să fie nevidă). Atunci când $\mathcal{A}$ este de forma $\mathcal{A} = \{A, B\}$, se notează în mod frecvent $A \cap B$ în loc de $\bigcap \mathcal{A}$.

Două mulţimi A şi B sunt numite *disjuncte* dacă $A \cap B = \emptyset$. O familie de mulţimi este numită *familie disjunctă de mulţimi* dacă mulţimele componente sunt disjuncte două câte două.

Diferenţa a două mulţimi A şi B, notată $A - B$, se obţine cu uşurinţă de la Axioma separării prin

$$A - B = \{a \in A | a \notin B\}.$$

Uneori, diferenţa $A - B$ mai poartă denumirea şi de *complementara lui* B *relativ la* (*în raport cu*) A.

Fie a şi b obiecte. Axioma extensionalităţii asigură unicitatea mulţimii ce conţine obiectele a şi b şi numai pe acestea; ea va fi notată $\{a, b\}$ (sau $\{b, a\}$), iar în cazul $a = b$ vom scrie $\{a\}$ în loc de $\{a, a\}$.

Teorema 1.1.1.3. Pentru orice x, y, z, u şi v au loc următoarele proprietăţi:

(1) $z \in \{x, y\}$ dacă şi numai dacă $z = x$ sau $z = y$;

[6] Vom folosi terminologia de *clasă* pentru colecţii de obiecte care pot să nu fie mulţimi. O *clasă proprie* nu este mulţime.

(2) $\{x, y\} = \{u, v\}$ dacă şi numai dacă $x = u$ şi $y = v$, sau $x = v$ şi $y = u$;

(3) $\{x\} = \{y\}$ dacă şi numai dacă $x = y$;

(4) $\{x\} = \{u, v\}$ dacă şi numai dacă $x = u = v$.

Demonstraţie (1), (2) şi (3) sunt imediate de la definiţii şi axiomele considerate până acum.

(4) Să presupunem întâi că $\{x\} = \{u, v\}$. De la Teorema 1.1.1.1(3) urmează că x trebuie să aparţină mulţimii $\{u, v\}$. Dacă presupunem că $x = u$, atunci, în baza aceleiaşi teoreme deducem că $v \in \{x\}$, adică $x = v$. Deci, $x = u = v$.

Reciproc, dacă $x = u = v$ atunci $\{u, v\} = \{u\} = \{x\}$. □

O consecinţă foarte importantă a Axiomei împerecherii constă în aceea că prin intermediul ei se poate introduce conceptul de *pereche ordonată* a două obiecte x şi y. Printr-o astfel de pereche ordonată se urmăreşte surprinderea următoarelor aspecte:

- obiectele x şi y sunt considerate ca un nou obiect, notat (x, y);
- în cadrul obiectului (x, y), x se consideră "primul", iar y al "doilea".

O metodă unanim acceptată de a defini astfel de obiecte în sistemul ZFC este cea propusă de Kazimierz Kuratowski [95].

Definiţia 1.1.1.2. Se numeşte *perechea ordonată* a obiectelor x şi y mulţimea notată (x, y) şi definită prin $(x, y) = \{\{x\}, \{x, y\}\}$.

Este clar că pentru orice două obiecte x şi y, perechea ordonată (x, y) există şi este unică (pe baza Axiomelor împerecherii şi extensionalităţii). Faptul că obiectul x este considerat primul iar y al doilea, în cadrul perechii ordonate (x, y), este sugerat de următoarea teoremă.

Teorema 1.1.1.4. $(x, y) = (u, v)$ dacă şi numai dacă $x = u$ şi $y = v$ [7].

Demonstraţie Să presupunem întâi că $(x, y) = (u, v)$. Dacă $x = y$, atunci $\{\{x\}\} = \{\{u\}, \{u, v\}\}$, de unde urmează $x = u = v$ (Teorema 1.1.1.3).

Să presupunem acum că $x \neq y$. Deoarece $\{x\}$ nu poate coincide cu $\{u, v\}$ (altfel am obţine $x = u = v = y$ ceea ce ar conduce la contradicţie) urmează că $\{x\} = \{u\}$ şi, de aici, se obţine $x = u$. Vom avea apoi $\{x, y\} = \{u, v\}$ care, combinată cu egalitatea precedentă, furnizează $y = v$.

Reciproc, dacă $x = y$ şi $u = v$ atunci $\{x\} = \{u\}$ şi $\{x, y\} = \{u, v\}$, ceea ce conduce la $(x, y) = (u, v)$. □

[7] O altă variantă de a defini perechea ordonată a două obiecte x şi y este cea propusă de Norbert Wiener în 1914 [195], prin care $(x, y) = \{\{\{x\}, 0\}, \{y\}\}$. Se poate arăta că şi o astfel de definiţie satisface Teorema 1.1.1.4 dar, spre deosebire de definiţia lui Kuratowski, aceasta implică un nou obiect, 0 (acesta va fi definit mai târziu).

Există variante în care noţiunea de pereche ordonată se consideră ca o noţiune primitivă şi, atunci, enunţul Teoremei 1.1.1.4 se introduce ca axiomă (a se vedea [17, 136]). Justificarea constă în faptul că marea majoritate a aplicaţiilor matematice ale acestei noţiuni utilizează Teorema 1.1.1.4 şi nu definiţia.

Evident, conceptul de pereche ordonată poate fi extins. Astfel, putem defini (x, y, z) ca fiind $((x, y), z)$. Proprietatea din Teorema 1.1.1.4 se păstrează şi pentru astfel de 3-uple, adică $(x, y, z) = (x', y', z')$ dacă şi numai dacă $x = x'$, $y = y'$ şi $z = z'$.

Ceea ce trebuie să remarcăm este că prin Axioma împerecherii putem construi mulţimi cu cel mult două elemente. Trecerea la mulţimi cu mai mult de două elemente se face prin Axioma reuniunii. Axioma extensionalităţii asigură că, pentru orice familie de mulţimi $\mathcal{A}$ există exact o mulţime ca cea postulată de Axioma reuniunii; această mulţime se numeşte *reuniunea* familiei $\mathcal{A}$ şi se notează prin $\bigcup \mathcal{A}$. Atunci când $\mathcal{A}$ este de forma $\mathcal{A} = \{A, B\}$, se notează în mod frecvent $A \cup B$ în loc de $\bigcup \mathcal{A}$.

Următoarea teoremă furnizează câteva proprietăţi de bază ale reuniunii unei familii de mulţimi.

Teorema 1.1.1.5. Fie A, B mulţimi şi $\mathcal{A}, \mathcal{B}$ două familii de mulţimi. Atunci, au loc următoarele proprietăţi:

(1) $\bigcup \emptyset = \emptyset$;

(2) $\bigcup \{A\} = A$;

(3) $\bigcup \mathcal{A} = \emptyset$ dacă şi numai dacă $\mathcal{A} = \emptyset$ sau $\mathcal{A} = \{\emptyset\}$;

(4) dacă $\mathcal{A} \subseteq \mathcal{B}$, atunci $\bigcup \mathcal{A} \subseteq \bigcup \mathcal{B}$;

(5) dacă $\mathcal{A} \in \mathcal{B}$, atunci $\mathcal{A} \subseteq \bigcup \mathcal{B}$;

(6) dacă $X \subseteq \mathcal{B}$, pentru orice $X \in \mathcal{A}$, atunci $\bigcup \mathcal{A} \subseteq \mathcal{B}$.

Posibilitatea colectării tuturor submulţimilor unei mulţimi într-o mulţime este asigurată de Axioma părţilor care, în conjuncţie cu Axioma extensionalităţii garantează unicitatea acestei mulţimi ce poartă denumirea de *mulţimea părţilor* mulţimii A şi se notează prin $\mathcal{P}(A)$.

Teorema 1.1.1.6. Fie A şi B mulţimi. Atunci, au loc următoarele proprietăţi:

(1) $\emptyset \in \mathcal{P}(A)$;

(2) $\mathcal{P}(\emptyset) = \{\emptyset\}$;

(3) dacă $A \subseteq B$ ($A \subset B$), atunci $\mathcal{P}(A) \subseteq \mathcal{P}(B)$ ($\mathcal{P}(A) \subset \mathcal{P}(B)$).

Axioma părţilor permite introducerea conceptului de produs cartezian a două mulţimi.

Definiţia 1.1.1.3. Fie A şi B două mulţimi. Numim *produsul cartezian* sau *direct* al mulţimilor A şi B, mulţimea notată $A \times B$ şi definită prin

$$A \times B = \{(x, y) | x \in A \ \wedge \ y \in B\}.$$

Existenţa produsului cartezian al mulţimilor A şi B urmează de la Axioma separării aplicate mulţimii $\mathcal{P}(\mathcal{P}(A \cup B))$,

$$A \times B = \{z \in \mathcal{P}(\mathcal{P}(A \cup B)) | \exists x \in A,\ \exists y \in B :\ z = \{\{x\}, \{x, y\}\}\},$$

iar unicitatea lui de la Axioma extensionalităţii.

Dacă una din mulţimile A sau B este vidă, atunci produsul cartezian al lor este mulţimea vidă, şi reciproc. Notăm că $A \times B$ nu este acelaşi cu $B \times A$, exceptând cazul $A = B$ sau cazul în care una din aceste două mulţimi este mulţimea vidă.

Produsul cartezian poate fi extins, în mod natural, la mai mult de două mulţimi. Considerând de exemplu mulţimile A, B şi C, putem defini

$$A \times B \times C = \{(a, b, c) | a \in A,\ b \in B,\ c \in C\}.$$

Construcţiile (a, b, c) vor fi numite *3-uple*.

Anticipând conceptul de număr natural (ceea ce nu va constitui un viciu de fond), definim A^n prin:

- $A^n = \underbrace{A \times \cdots \times A}_{n\ ori}$, dacă $n \geq 2$;
- $A^1 = A$ şi $A^0 = \{\emptyset\}$ [8].

In matematică, dar nu numai, suntem interesaţi în a construi corespondenţe (asocieri) între diverse tipuri de obiecte. Cel mai frecvent sunt întâlnite corespondenţele între două tipuri, nu neapărat distincte, de obiecte. Perechea ordonată (a, b) poate fi o alegere bună pentru a exprima corespondenţa (asocierea) dintre a şi b, mai ales atunci când dorim să surprindem şi o anumită "relaţie de precedenţă" între a şi b [9]. Ca urmare, o mulţime de perechi ordonate va modela o corespondenţă (asociere) între două tipuri de obiecte. Astfel de mulţimi vor fi numite *relaţii binare*.

Definiţia 1.1.1.4. Se numeşte *relaţie binară* orice mulţime ale cărei elemente sunt perechi ordonate [10].

Vom simplifica adesea terminologia de "relaţie binară" la cea de "relaţie", iar dacă (a, b) este un element al unei relaţii ρ atunci vom mai scrie $a \,\rho\, b$.

Dată o relaţie ρ vom nota

$$Dom(\rho) = \{a | (\exists b)(a \,\rho\, b)\}$$

[8] Anticipând câteva concepte şi notaţii care vor fi prezentate ulterior, dar cu care cititorul este probabil familiarizat, justificăm definiţia "$A^0 = \{\emptyset\}$" astfel. Aşa cum vom vedea, numerele naturale vor fi definite ca mulţimi, $0 = \emptyset$, $1 = \{0\}$, $2 = \{0, 1\}$ etc. Atunci, un n-uplu poate fi gândit ca o funcţie de la mulţimea n la mulţimea A, iar mulţimea tuturor acestor funcţii (n-uple) este notată prin A^n (în general, prin A^B se va nota mulţimea tuturor funcţiilor de la B la A). In cazul $n = 0 = \emptyset$, există o singură funcţie de la $\emptyset$ la A, şi anume funcţia vidă. Ca urmare, $A^0 = \{\emptyset\}$.

[9] O altă posibilă exprimare a asocierii dintre a şi b ar putea fi specificată prin intermediul mulţimii $\{a, b\}$. In acest caz însă se pierde "ordinea" în care sunt considerate obiectele a şi b.

[10] In limbaj logic, ρ este relaţie binară dacă $(\forall x)(x \in \rho \ \Rightarrow\ (\exists y)(\exists z)(x = (y, z)))$.

şi

$$Cod(\rho) = \{b|(\exists a)(a\,\rho\,b)\}.$$

Acestea sunt mulţimi în baza Axiomelor separării şi reuniunii. In adevăr,

$$Dom(\rho) = \{a \in \bigcup(\bigcup \rho)|(\exists b \in \bigcup(\bigcup \rho))(a\,\rho\,b)\}.$$

In mod similar putem arăta că $Cod(\rho)$ este mulţime. Mulţimea $Dom(\rho)$ se numeşte *domeniul* relaţiei ρ, iar $Cod(\rho)$, *codomeniul* relaţiei ρ.

Putem spune că ρ este relaţie dacă există două mulţimi A şi B astfel încât $\rho \subseteq A \times B$. Reciproc, orice submulţime a unui produs cartezian este relaţie. Aceasta face ca, adesea, relaţiile $\rho \subseteq A \times B$ să mai fie numite şi *relaţii de la A la B*, iar atunci când $B = A$, *relaţii* (*binare*) *pe* A.

Mulţimea vidă este relaţie (de la A la B), numită *relaţia vidă*.

Funcţiile sunt cazuri particulare de relaţii. Ele vor fi notate, cu precădere, prin f, g, h etc. (eventual indexat).

Definiţia 1.1.1.5. O relaţie binară f este numită *funcţie* dacă are loc

$$(\forall a_1, b_1, a_2, b_2)((a_1, b_1) \in f \ \wedge \ (a_2, b_2) \in f \ \wedge \ a_1 = a_2 \ \Rightarrow \ b_1 = b_2).$$

Relaţia vidă este funcţie, numită şi *funcţia vidă*.

Pentru funcţii se utilizează în mod frecvent notaţia $f(a) = b$ în loc de $(a, b) \in f$, aceasta fiind justificată prin aceea că, dat un element a, dacă $(a, b) \in f$ atunci b este unicul cu această proprietate.

Funcţiile fiind relaţii, putem vorbi de *domeniul* şi *codomeniul* acestora. Domeniul unei funcţii mai poartă denumirea şi de *domeniul de definiţie* al funcţiei. Domeniul şi codomeniul funcţiei vide sunt mulţimea vidă.

O funcţie f este numită *funcţie de la A la B* sau *funcţie definită pe A şi cu valori în B*, şi notăm $f : A \to B$, dacă $Dom(f) = A$ şi $Cod(f) \subseteq B$. Funcţia vidă este funcţie de la A la B numai dacă $A = \emptyset$.

Mulţimea tuturor funcţiilor de la A la B se notează prin $(A \to B)$ sau B^A.

Definiţia 1.1.1.6. Fie f o funcţie de la A la B.

(1) f este numită *funcţie injectivă* sau *injecţie* dacă are loc

$$(\forall a_1, b_1, a_2, b_2)((a_1, b_1) \in f \ \wedge \ (a_2, b_2) \in f \ \wedge \ b_1 = b_2 \ \Rightarrow \ a_1 = a_2).$$

(2) f este numită *funcţie surjectivă* sau *surjecţie* dacă are loc

$$(\forall b)(b \in B \ \Rightarrow \ (\exists a)(a \in A \ \wedge \ f(a) = b)).$$

(3) f este numită *funcţie bijectivă* sau *bijecţie* dacă este atât funcţie injectivă cât şi funcţie surjectivă.

Uneori funcţiile injective sunt numite *funcţii* $1-1$, iar cele surjective, *pe*. Funcţia vidă de la $\emptyset$ la B este injectivă; ea este surjectivă (deci şi bijectivă) doar dacă $B = \emptyset$.

Atunci când există o funcţie bijectivă de la o mulţime A la o mulţime B vom mai nota $A \sim B$ şi vom spune că A şi B sunt *echipotente* [11], iar dacă există o funcţie injectivă de la A la B vom scrie $A \preceq B$. Dacă $A \preceq B$ dar nu are loc $A \sim B$, atunci vom scrie $A \prec B$.

Analiza paradoxului lui Russell ridică următoarea întrebare simplă dar fundamentală: există mulţimi ce sunt elemente ale lor însăşi? Axioma regularităţii este cea care evită astfel de cazuri.

Teorema 1.1.1.7. Nu există nici o mulţime A astfel încât $A \in A$.

Demonstraţie Presupunem prin contradicţie că există o mulţime A cu proprietatea $A \in A$. Aplicăm Axioma regularităţii mulţimii $\{A\}$. Atunci, există $x \in \{A\}$ astfel încât $x \cap A = \emptyset$. Forma particulară a mulţimii $\{A\}$ conduce la faptul că x trebuie să fie A şi, atunci, $A \cap \{A\} = \emptyset$; contradicţie cu faptul că $A \in A$. □

Teorema 1.1.1.8. Nu există mulţimi A şi B astfel încât $A \in B$ şi $B \in A$.

Demonstraţie Să presupunem că există mulţimi A şi B astfel încât $A \in B$ şi $B \in A$. Aplicăm Axioma regularităţii mulţimii $\{A, B\}$. Atunci, există $x \in \{A, B\}$ astfel încât $x \cap \{A, B\} = \emptyset$. Elementul x poate fi A sau B. Dacă $x = A$, atunci $A \cap \{A, B\} = \emptyset$; contradicţie cu faptul că $B \in A$ şi $B \in \{A, B\}$. Similar se raţionează pentru cazul $x = B$. □

Axioma regularităţii este consistentă cu celelalte axiome ale sistemului ZFC şi independentă de acestea [64, 150]. Este posibil a construi sisteme ale teoriei mulţimilor care să contrazică această axiomă. Două exemple în acest sens sunt sistemul lui Lesniewski [104] şi cel al lui Quine [147].

Axioma regularităţii are consecinţe foarte naturale, aşa cum este cea din teorema următoare (care nu poate fi demonstrată pe baza celorlalte axiome).

Teorema 1.1.1.9. Fie A o mulţime. Dacă $A \subseteq A \times A$, atunci $A = \emptyset$.

Demonstraţie Presupunem, prin contradicţie, că există o mulţime nevidă A astfel încât $A \subseteq A \times A$. Atunci, elementele mulţimii $A \times A$ sunt mulţimi nevide. Ipoteza şi definiţia produsului cartezian conduc la faptul că elementele mulţimilor A şi $\bigcup A$ sunt, de asemenea, mulţimi nevide. Fie $B = A \cup \bigcup A$. Axioma regularităţii asigură existenţa unei mulţimi $x \in B$ astfel încât $x \cap B = \emptyset$. Avem de analizat următoarele două cazuri:

- $x \in A$. Atunci, $x \subseteq \bigcup A$ şi, deoarece x este mulţime nevidă, urmează că $x \cap \bigcup A \neq \emptyset$; contradicţie cu $x \cap B = \emptyset$;

[11] Terminologia de "mulţimi echipotente", care înseamnă "mulţimi cu acelaşi număr de elemente", este justificată prin aceea că o funcţie bijectivă pune în corespondenţă "unu-la-unu" elementele a două mulţimi. Echipotenţa joacă un rol important în definirea numerelor naturale, a numerelor ordinale şi cardinale.

- $x \in \bigcup A$. Conform ipotezei şi definiţiei produsului cartezian, x este ori de forma $\{a\}$ ori de forma $\{a, b\}$, unde $a, b \in A$. Deci, $x \cap A \neq \emptyset$; contradicţie cu $x \cap B = \emptyset$.

Teorema este demonstrată. □

1.1.2 Operaţii cu mulţimi

Implicit, în secţiunea anterioară au fost introduse un număr de operaţii cu mulţimi: intersecţie, diferenţă, reuniune şi produs cartezian. Vom adăuga la acestea încă câteva operaţii şi vom prezenta unele proprietăţi de bază ale lor.

Teorema 1.1.2.1. Fie A, B, şi C mulţimi. Atunci:

(1) $A \cup (B \cup C) = (A \cup B) \cup C = \bigcup\{A, B, C\}$; (*asociativitate*)

(2) $A \cup B = B \cup A$; (*comutativitate*)

(3) $A \cup A = A$; (*idempotenţă*)

(4) $A \cup \emptyset = A$;

(5) $\mathcal{P}(A) \cup \mathcal{P}(B) \subseteq \mathcal{P}(A \cup B)$.

Demonstraţie (1) Vom folosi metoda dublei incluziuni. Fie $a \in A \cup (B \cup C)$. Conform definiţiei reuniunii avem $a \in A$ sau $a \in B \cup C$. Dacă $a \in A$, atunci $a \in A \cup B$ şi, deci, $a \in (A \cup B) \cup C$. Dacă $a \in B \cup C$, atunci $a \in B$ sau $a \in C$. In cazul $a \in B$ obţinem $a \in A \cup B$ şi, deci, $a \in (A \cup B) \cup C$, iar în cazul $a \in C$ obţinem $a \in (A \cup B) \cup C$. Deci, $a \in (A \cup B) \cup C$ ceea ce arată că $A \cup (B \cup C) \subseteq (A \cup B) \cup C$. Incluziunea în sens invers se demonstrează similar.

(2) Se utilizează definiţia reuniunii şi faptul că $\{A, B\} = \{B, A\}$.

(3) Observând că $\{A, A\} = \{A\}$, ceea ce ne rămâne de demonstrat este că $\bigcup\{A\} = A$, care urmează direct de la definiţia reuniunii.

(4) Un element a este în $A \cup \emptyset$ dacă şi numai dacă $a \in A$; ca urmare $A \cup \emptyset = A$.

(5) Fie $X \in \mathcal{P}(A) \cup \mathcal{P}(B)$. Dacă $X \in \mathcal{P}(A)$ atunci $X \subseteq A$ şi este clar atunci că $X \subseteq A \cup B$; deci $X \in \mathcal{P}(A \cup B)$. Similar în cazul $X \in \mathcal{P}(B)$. □

Demonstraţiile următoarelor două teoreme sunt lăsate în seama cititorului.

Teorema 1.1.2.2. Fie A, B, şi C mulţimi. Atunci:

(1) $A \cap (B \cap C) = (A \cap B) \cap C = \bigcap\{A, B, C\}$; (*asociativitate*)

(2) $A \cap B = B \cap A$; (*comutativitate*)

(3) $A \cap A = A$; (*idempotenţă*)

(4) $A \cap \emptyset = \emptyset$;

(5) $A \cap B \subseteq A \subseteq A \cup B$;

(6) $\mathcal{P}(A) \cap \mathcal{P}(B) = \mathcal{P}(A \cap B)$.

Ca urmare a proprietăţii de asociativitate a reuniunii şi intersecţiei, putem scrie $A \cup B \cup C$ ($A \cap B \cap C$) în loc de $A \cup (B \cup C)$ ($A \cap (B \cap C)$) sau $(A \cup B) \cup C$ ($(A \cap B) \cap C$). Evident, această scriere poate fi extinsă la o reuniune (intersecţie) finită de mulţimi.

Teorema 1.1.2.3. Fie A o mulţime şi $\mathcal{C}$ o familie de mulţimi. Atunci, au loc următoarele reguli de distributivitate:

(1) $A \cap \bigcup \mathcal{C} = \bigcup \{A \cap C | C \in \mathcal{C}\} = \bigcup \mathcal{C} \cap A$;

(2) $A \cup \bigcap \mathcal{C} = \bigcap \{A \cup C | C \in \mathcal{C}\} = \bigcap \mathcal{C} \cup A$, cu condiţia ca $\mathcal{C}$ să fie nevidă.

Interpretăm proprietăţile din Teorema 1.1.2.3 prin aceea că *intersecţia este distributivă faţă de reuniune atât la stânga cât şi la dreapta*. In mod similar, *reuniunea este distributivă fată de intersecţie la stânga şi la dreapta*.

Următoarea teoremă prezintă câteva proprietăţi de bază ale diferenţei de mulţimi.

Teorema 1.1.2.4. Fie A şi B mulţimi, iar $\mathcal{C}$ o familie de mulţimi. Atunci:

(1) $A - A = \emptyset$;

(2) $\emptyset - A = \emptyset$;

(3) $A - \emptyset = A$;

(4) $A - B \subseteq A$;

(5) dacă $A \cap B = \emptyset$ atunci $A - B = A$;

(6) $A - (B - C) = (A - B) \cup (A \cap C)$;

(7) $(A - B) \cup C = (A \cup C) - (B - C)$;

(8) $(A - B) \cap C = (A \cap C) - B = A \cap (C - B)$;

(9) $A - \bigcup \mathcal{C} = \bigcap \{A - C | C \in \mathcal{C}\}$;

(10) $A - \bigcap \mathcal{C} = \bigcup \{A - C | C \in \mathcal{C}\}$, cu condiţia ca $\mathcal{C}$ să fie nevidă.

Fie U o mulţime. Complementara unei submulţimi $A \subseteq U$ în raport cu U se mai numeşte şi *complementara absolută a lui A relativ la* (*în raport cu*) U sau, mai simplu, *complementara lui A* (dar, în acest caz, U trebuie subânţeleasă din context). Ea se notează prin $\overline{A}$.

Teorema 1.1.2.5. Fie U, A şi B mulţimi astfel încât $A \cup B \subseteq U$. Atunci:

(1) $\overline{\overline{A}} = A$;

(2) $\overline{\emptyset} = U$;

(3) $\overline{U} = \emptyset$;

(4) $A \cup \overline{A} = U$;

(5) $A \cap \overline{A} = \emptyset$;

(6) $A - B = A \cap \overline{B}$;

(7) $A \subseteq B$ dacă şi numai dacă $\overline{B} \subseteq \overline{A}$

(complementara este în raport cu U).

Corolarul 1.1.2.1. (Legile lui De Morgan)
Fie U, A şi B mulţimi astfel încât $A \cup B \subseteq U$. Atunci, au loc relaţiile:

(1) $\overline{A \cup B} = \overline{A} \cap \overline{B}$;

(2) $\overline{A \cap B} = \overline{A} \cup \overline{B}$

(complementara este în raport cu U).

Definiţia 1.1.2.1. Fie A şi B două mulţimi. Numim *diferenţa simetrică* a mulţimilor A şi B mulţimea $A\Delta B = (A - B) \cup (B - A)$.

Conform Axiomei reuniunii, există o unică mulţime $A\Delta B$. Ca urmare, Definiţia 1.1.2.1 este consistentă.

Operaţiile $\cup$, $\cap$, $-$ şi Δ au fost studiate în mod sistematic pentru prima dată de către George Boole [14]. Din acest motiv, ele sunt numite astăzi *operaţii Booleene*. Ele pot fi reprezentate grafic prin aşa numitele *diagrame Venn*, ca în Figura 1.1.

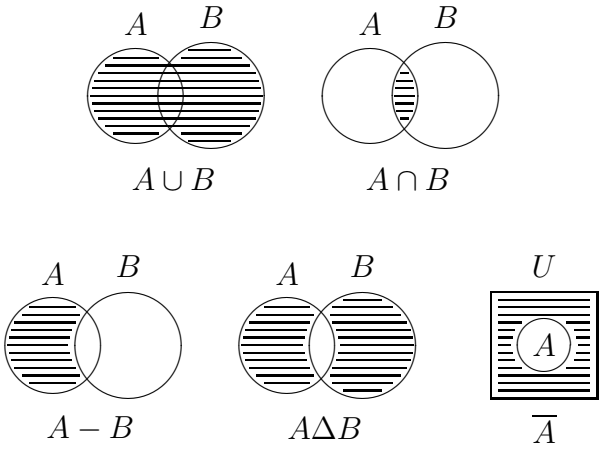

Figura 1.1: Reprezentarea operaţilor Booleene cu mulţimi prin diagrame Venn

Următoarea teoremă ne spune că produsul cartezian este *distributiv la stânga şi la dreapta faţă de reuniune şi intersecţie* (demonstraţia este lăsată în seama cititorului).

Teorema 1.1.2.6. Fie A, B, C, D mulţimi şi $\mathcal{A}$ o familie de mulţimi. Atunci, au loc următoarele proprietăţi:

(1) $A \times \bigcup \mathcal{A} = \bigcup\{A \times X | X \in \mathcal{A}\}$ şi $\bigcup \mathcal{A} \times A = \bigcup\{X \times A | X \in \mathcal{A}\}$;

(2) $A \times \bigcap \mathcal{A} = \bigcap\{A \times X | X \in \mathcal{A}\}$ şi $\bigcap \mathcal{A} \times A = \bigcap\{X \times A | X \in \mathcal{A}\}$, cu condiţia ca $\mathcal{A}$ să fie nevidă;

(3) $A \times (B - C) = (A \times B) - (A \times C)$;

(4) $(A \cap B) \times (C \cap D) = (A \times C) \cap (B \times D)$.

1.1.3 Numere naturale şi inducţie

Mulţimea numerelor naturale este, fără doar şi poate, primul exemplu de mulţime infinită la care ne-am gândi dacă am fi întrebaţi să dăm un exemplu de o astfel de mulţime. Introducerea ei este, însă, un proces destul de complex care a stat în atenţia cercetătorilor multe zeci de ani (pentru detalii indicăm [182, 187]). Problema fundamentală constă în aceea că numerele nu pot fi definite făcând apel la conceptul de număr. De exemplu, nu putem spune că 2 este proprietatea comună pe care o au toate mulţimile cu două elemente deoarece definiţia aceasta este circulară. Dar, dacă punem în evidenţă o mulţime care, intuitiv, are "două elemente", atunci o putem folosi pe aceasta pentru a defini 2. Foarte pe scurt, numerele naturale se introduc prin intermediul mulţimilor astfel:

$$0 = \emptyset,\ 1 = \{\emptyset\} = \{0\},\ 2 = \{\emptyset, \{\emptyset\}\} = \{0, 1\} \text{ etc.}$$

Dacă pentru o mulţime x notăm prin $S(x)$ mulţimea $S(x) = x \cup \{x\}$, numită *succesoarea mulţimii* x, atunci mulţimea numerelor naturale, notată prin $\mathbf{N}$, este cea mai mică mulţime cu proprietăţile:

- conţine $\emptyset$;
- dacă conţine x atunci conţine şi $S(x)$.

O mulţime cu aceste două proprietăţi poartă denumirea de *mulţime inductivă*. Ca urmare, $\mathbf{N}$ este definită ca fiind cea mai mică mulţime inductivă. Singura problemă este că nu putem demonstra existenţa unei astfel de mulţimi pe baza axiomelor prezentate. Soluţia constă în adoptarea unei noi axiome, numită Axioma infinitului

Axioma infinitului Există mulţimi inductive

In baza acestei axiome obţinem cu uşurinţă că există o cea mai mică mulţime inductivă, deci există mulţimea numerelor naturale (orice mulţime inductivă trebuie să conţine şi orice număr natural).

Ordinea pe numere naturale se obţine prin intermediul apartenenţei. Fie $<$ relaţia binară pe $\mathbf{N}$ dată prin:

$$n < m \quad \Leftrightarrow \quad n \in m,$$

pentru orice $n, m \in \mathbf{N}$. Vom arăta că $<$ este o ordine totală strictă pe $\mathbf{N}$, dar pentru aceasta vom avea nevoie mai întâi de o tehnică de demonstraţie, numită *principiul inducţiei matematice*.

Teorema 1.1.3.1. (Principiul inducţiei matematice)
Fie $P(x)$ o proprietate astfel încât:

(i) $P(0)$;

(ii) pentru orice $k \in \mathbf{N}$, $P(k)$ implică $P(S(k))$.

Atunci, P este satisfăcută de toate numerele naturale.

Demonstraţie (i) şi (ii) arată că mulţimea $A = \{k \in \mathbf{N} | P(k)\}$ este inductivă. Cum $\mathbf{N}$ este cea mai mică mulţime inductivă, $\mathbf{N} \subseteq A$, ceea ce demonstrează teorema. □

Aplicarea Principiului inducţiei în situaţii concrete constă în parcurgerea următoarelor etape:

– se alege (fixează) proprietatea P despre care se doreşte a se arăta că este satisfăcută de toate numerele naturale;

– se verifică faptul că P este satisfăcută de 0 (această etapă se numeşte *baza inducţiei*);

– se consideră un număr arbitrar $k \geq 0$, se presupune că P este satisfăcută de k (această presupunere este numită *ipoteza inductivă*), după care se verifică dacă P este satisfăcută de $S(k)$ (această etapă se numeşte *pasul inductiv*).

Dacă atât baza inducţiei cât şi pasul inductiv au fost parcurse cu succes, atunci, în baza Principiului inducţiei deducem că proprietatea P este satisfăcută de toate numerele naturale.

Demonstraţiile ce utilizează exclusiv Principiul inducţiei sau variante ale acestuia, aşa cum vom prezenta în continuare, sunt numite *demonstraţii prin inducţie* (*matematică*).

Teorema 1.1.3.2.

(1) Relaţia $<$ pe $\mathbf{N}$ este ordine totală strictă.

(2) Orice submulţime nevidă a mulţimii $\mathbf{N}$ are cel mai mic element în raport cu relaţia $<$.

Demonstraţie (1) Conform discuţiei de mai sus, ceea ce ne rămâne de arătat este că orice două numere naturale n şi m sunt *comparabile*. Demonstraţia acestui fapt o vom face prin inducţie matematică arătând că proprietatea

$$P(n): \quad n \text{ este comparabil cu orice } m \in \mathbf{N}$$

este satisfăcută de orice număr natural:

- $P(0)$: vom arăta, utilizând iarăşi inducţia matematică, că 0 este comparabil cu orice $m \in \mathbf{N}$:
  - evident, 0 este comparabil cu 0 ($0 = 0$);
  - presupunem că 0 este comparabil cu m. Ca urmare, ori $0 \in m$, ori $0 = m$. In ambele cazuri avem $0 \in S(m) = m \cup \{m\}$ şi, deci, 0 este comparabil cu $S(m)$.

  Principiul inducţiei matematice asigură atunci că 0 este comparabil cu orice număr natural m;

- presupunem că n este comparabil cu orice $m \in \mathbf{N}$. Vom arăta prin inducţie că $S(n)$ este comparabil cu orice $m \in \mathbf{N}$:
  - evident, $S(n)$ este comparabil cu 0 ($0 \in S(n)$);
  - presupunem că $S(n)$ este comparabil cu m. Ca urmare, ori $S(n) \in m$, ori $S(n) = m$, ori $m \in S(n)$. Dacă $S(n) \in m$ sau $S(n) = m$, atunci $S(n) \in m \cup \{m\} = S(m)$. Dacă $m \in S(n)$, atunci ori $m \in n$ ori $m = n$. In primul caz are loc $S(m) \in S(n)$, iar în cel de-al doilea $S(m) = S(n)$, ceea ce arată că $S(m)$ şi $S(n)$ sunt comparabile.

  Conform Principiului inducţiei, $S(n)$ este comparabil cu orice $m \in \mathbf{N}$.

Am obţinut astfel, în baza Principiului inducţiei matematice, că orice două numere naturale sunt comparabile; deci, $<$ este ordine totală strictă.

(2) Să arătăm acum că orice submulţime nevidă a mulţimii $\mathbf{N}$ are un cel mai mic element în raport cu ordinea $<$. Fie $M \subseteq \mathbf{N}$ nevidă şi $n \in M$. Mulţimea $S(n) \cap M$ este nevidă şi, deoarece $S(n)$ este număr natural, urmează că $S(n) \cap M$ admite cel mai mic element. Este uşor de văzut că acest cel mai mic element este de fapt şi cel mai mic element al mulţimii M în raport cu ordinea $<$.

Teorema este demonstrată. □

Corolarul 1.1.3.1. Dacă o submulţime de numere naturale are un element maximal, atunci acesta este unic (el fiind, astfel, cel mai mare element al acesteia).

Demonstraţie Dacă o submulţime de numere naturale ar avea mai mult de un element maximal, atunci mulţimea acestor elemente maximale ar admite un cel mai mic element care ar contrazice statutul de element maximal al acestuia. □

Faptul că orice submulţime nevidă a lui $(\mathbf{N}; <)$ are un cel mai mic element permite stabilirea unor noi variante ale Principiului inducţiei. Prezentăm întâi câteva uşoare generalizări.

Principiul inducţiei poate fi aplicat pe submulţimi nevide (bine precizate) ale lui $\mathbf{N}$. De exemplu, dacă dorim să demonstrăm că o proprietate P este satisfăcută de toate numerele naturale mai mici sau egale cu un număr fixat n, atunci avem de verificat următoarele:

(a) $P(0)$;

(b) $P(k)$ implică $P(S(k))$, pentru orice $k < n$.

In adevăr, dacă considerăm proprietatea Q dată prin $Q(k) = P(k)$, pentru $k \leq n$, şi $Q(k)$ satisfăcută pentru orice $k > n$, atunci (a) şi (b) conduc la:

(c) $Q(0)$;

(d) $Q(k)$ implică $Q(S(k))$, pentru orice $k \in \mathbf{N}$,

care în baza Principiului inducţiei asigură faptul că Q este satisfăcută de toate numerele naturale, adică $\mathbf{N} \subseteq \{k \in \mathbf{N} | Q(k)\}$. Atunci,

$$\begin{aligned} \{k \in \mathbf{N} | P(k)\} &= \{k \in \mathbf{N} | Q(k)\} \cap \{k \in \mathbf{N} | k \leq n\} \\ &\supseteq \mathbf{N} \cap \{k \in \mathbf{N} | k \leq n\} \\ &= \{k \in \mathbf{N} | k \leq n\}, \end{aligned}$$

ceea ce ne arată că P este satisfăcută de toate numerele naturale mai mici sau egale cu n. Această variantă a Principiului inducţiei poartă denumirea de *Principiul inducţiei finitare* (terminologia de "finitar" provine de la faptul că mulţimea pe care se cere verificarea proprietăţii P este finită).

Evident, se pot imagina şi alte tipuri de submulţimi pe care se poate aplica o tehnică similară. Destul de des sunt întâlnite variante de forma:

(a') $P(n_0)$ (n_0 fiind fixat a priori);

(b') $P(k)$ implică $P(S(k))$, pentru orice $k \geq n_0$,

care conduc la $\{k \in \mathbf{N} | k \geq n_0\} \subseteq \{k \in \mathbf{N} | P(k)\}$ (cititorul este invitat să argumenteze această variantă a Principiului inducţiei).

Pentru variantele pe care le vom prezenta în continuare vom utiliza din plin Teorema 1.1.3.2(2). Dacă A este o mulţime nevidă de numere naturale, atunci cel mai mic element al ei va fi notat prin $\perp_A$. Orice element $k \in A$ care nu este maximal are un succesor imediat $k' \in A$. In adevăr, mulţimea $\{a \in A | k < a\}$ este nevidă şi are cel mai mic element, care este succesorul imediat al lui k.

Propoziţia 1.1.3.1. Fie $P(x)$ o proprietate astfel încât:

(i) $P(0)$;

(ii) pentru orice $k \in \mathbf{N}$, $((\forall j \leq k)(P(j)) \Rightarrow P(S(k)))$.

Atunci, P este satisfăcută de toate numerele naturale $n \in \mathbf{N}$.

Demonstraţie Presupunem prin contradicţie că există un număr natural n ce nu satisface P. Fie A mulţimea tuturor acestor numere. A este nevidă dar nu conţine 0 (deoarece are loc (i)). Proprietatea P este satisfăcută de toate numerele naturale mai mici decât $\perp_A$ şi, atunci, (ii) conduce la faptul că P este satisfăcută de $\perp_A$; contradicţie cu $\perp_A \in A$. □

Propoziţia 1.1.3.2. Fie $A \subseteq \mathbf{N}$ şi $P(x)$ o proprietate astfel încât:

(i) $P(\perp_A)$;

(ii) pentru orice $k \in A$ ce nu este cel mai mare element al mulţimii A,

$$P(k) \Rightarrow P(k'),$$

unde k' este succesorul imediat al lui k în A.

Atunci, P este satisfăcută de toate numerele naturale $n \in A$.

Demonstraţie Considerăm proprietatea $Q(x)$ dată prin:

(1) pentru orice $x \in A$, Q este satisfăcută de x dacă şi numai dacă P este satisfăcută de x;

(2) Q este satisfăcută de orice $x \in \mathbf{N} - A$.

Utilizând Propoziţia 1.1.3.1 arătăm că proprietatea Q satisface ipotezele Principiului inducţiei:

– dacă $0 = \perp_A$, atunci Q este satisfăcută de 0 pe baza lui (1); altfel, Q este satisfăcută de 0 pe baza lui (2);

– considerăm $k \in \mathbf{N}$ ce nu este maximal şi presupunem că are loc $Q(j)$, pentru orice $j \leq k$. Dacă $S(k) \in \mathbf{N} - A$, atunci Q este satisfăcută de $S(k)$ (pe baza lui (2)). Altfel, avem de luat în considerare următoarele două cazuri:

(a) $k \in A$. Atunci, $S(k)$ este succesorul imediat al lui k în A, iar (1) şi ipoteza propoziţiei conduc la faptul că $S(k)$ satisface Q;

(b) $k \notin A$. Dacă $S(k) = \perp_A$, atunci Q este satisfăcută de $S(k)$ ca urmare a lui (1). Altfel, există un element $m \in A$ astfel încât $S(k)$ este succesorul direct al lui m în A. Numărul m satisface $m \leq k$ şi, atunci, pe baza ipotezei inductive urmează că m satisface Q. Ipoteza propoziţiei şi (1) conduc la faptul că $S(k)$ satisface Q.

Principiul inducţiei aplicat proprietăţii Q ne arată că $\mathbf{N} \subseteq \{k \in \mathbf{N} | Q(k)\}$. Deci,

$$A = \mathbf{N} \cap A \subseteq \{k \in \mathbf{N} | Q(k)\} \cap A = \{k \in \mathbf{N} | P(k)\},$$

ceea ce demonstrează propoziţia. □

Pentru Principiul din Propoziţia 1.1.3.2 se poate da o variantă ca în Propoziţia 1.1.3.1. Demonstraţia acesteia o lăsăm în seama cititorului.

Propoziţia 1.1.3.3. Fie $A \subseteq \mathbf{N}$ şi $P(x)$ o proprietate astfel încât:

(i) $P(\perp_A)$;

(ii) pentru orice $k \in A$ ce nu este cel mai mare element al mulţimii A,

$$(\forall j \leq k)(j \in A \ \wedge \ P(j)) \ \Rightarrow \ P(k'),$$

unde k' este succesorul imediat al lui k în A.

Atunci, P este satisfăcută de toate numerele naturale $n \in A$.

Definiţia 1.1.3.1. O mulţime A este numită *finită* dacă există un număr natural n astfel încât A şi n sunt echipotente. Vom mai spune în acest caz că A *are* n *elemente* şi vom nota $|A| = n$. Dacă A nu este finită vom spune că ea este *infinită*.

Secvenţele sunt "înşiruiri" finite sau infinite de elemente; ele apar frecvent în consideraţii matematice. In analiza matematică secvenţele infinite sunt uzual numite *şiruri*.

Definiţia 1.1.3.2. Se numeşte *secvenţă* de elemente peste A orice funcţie f cu domeniul un număr natural sau $\mathbf{N}$ şi cu valori în A. Dacă domeniul este un număr natural n, atunci secvenţa este numită *finită* sau *de lungime* n; altfel, ea este numită *infinită*.

Secvenţele sunt funcţii şi, ca urmare, putem vorbi despre *domeniul* şi *codomeniul* unei secvenţe. Domeniul va fi întotdeauna un număr natural sau $\mathbf{N}$. Există o unică secvenţă de lungime 0 şi anume, funcţia vidă; ea va fi numită *secvenţa vidă*.

Uzual, secvenţele infinite sunt notate prin

$$\langle a_i | i \in \mathbf{N} \rangle \ \text{ sau } \ \langle a_i | i \geq 0 \rangle \ \text{ sau } \ \langle a_i \rangle_{i \in \mathbf{N}} \ \text{ sau } \ \langle a_i \rangle_{i \geq 0},$$

iar cele finite de lungime n prin

$$\langle a_i | i < n \rangle \ \text{ sau } \ \langle a_i | i = 0, \ldots, n-1 \rangle \ \text{ sau } \ \langle a_0, \ldots, a_{n-1} \rangle \ \text{ sau } \ \langle a_i \rangle_{i=0}^{n-1},$$

unde $a_i = f(i)$, f fiind secvenţa în cauză. Uneori, croşetele "$\langle$" şi "$\rangle$" sunt înlocuite prin paranteze rotunde sau acolade, iar în cazul secvenţelor finite ele sunt eliminate cu precădere [12].

1.1.4 Recursie

Definirea operaţiilor de bază pe mulţimea numerelor naturale, cum ar fi adunarea şi înmulţirea, constituie un alt obstacol pe care trebuie să îl trecem. Menţionăm întâi că o *operaţie binară* pe o mulţime A nu este altceva decât o funcţie de la $A \times A$ cu valori în A.

Caracterul inductiv al mulţimii numerelor naturale face loc ideii definirii "inductive" de funcţii al căror domeniu este această mulţime, dar nu numai. De exemplu, adunarea poate fi definită prin

[12] Atunci când sunt utilizate parantezele rotunde, distincţia dintre secvenţe şi familii indexate de mulţimi (ce vor fi introduse în Secţiunea 1.2.4), urmează a fi dedusă din context. De fapt, trebuie să remarcăm că în cazul în care A este o familie de mulţimi, secvenţele peste A sunt cazuri particulare de familii indexate de mulţimi (mulţimea de index este un număr natural sau mulţimea $\mathbf{N}$).

- $n + 0 = n$, pentru orice $n \in \mathbf{N}$;
- $n + S(m) = S(n + m)$, pentru orice $n, m \in \mathbf{N}$.

In cazul funcţiilor, astfel de proceduri (metodologii, scheme de definiţie) sunt numite *definiţii recursive/recurente* sau *scheme de recursie/recurenţă*. In general, ele constau în:

– se defineşte funcţia pentru 0;

– dacă funcţia a fost definită pentru $n \in \mathbf{N}$, atunci se arată cum se defineşte pentru $S(n)$.

Vom spune că două funcţii f şi g sunt *compatibile* dacă $Dom(f) \subseteq Dom(g)$ şi $f(x) = g(x)$, pentru orice $x \in Dom(f)$. O mulţime A de funcţii compatibile are proprietatea că orice două funcţii ale ei sunt compatibile. Dacă A este o astfel de mulţime, atunci $\bigcup A$ este funcţie cu domeniul $\bigcup_{f \in A} Dom(f)$.

Teorema 1.1.4.1. (Teorema recursiei)
Fie A o mulţime, $a \in A$ şi $h : \mathbf{N} \times A \to A$ o funcţie. Atunci, există o unică funcţie $f : \mathbf{N} \to A$ astfel încât:

(i) $f(0) = a$;

(ii) $f(S(n)) = h(n, f(n))$, pentru orice $n \in \mathbf{N}$.

Demonstraţie Fie F mulţimea tuturor funcţiilor g al căror domeniu este un număr natural diferit de 0, cu valori în A, ce verifică:

$$(*) \begin{cases} g(0) = a, \\ g(S(x)) = h(x, g(x)), \quad \text{pentru orice } x \text{ cu } S(x) \in Dom(g). \end{cases}$$

Este uşor de văzut că F este mulţime nevidă (F conţine funcţia $g : \{0\} \to A$ dată prin $g(0) = a$).

Arătăm că orice două funcţii $g, g' \in F$ sunt compatibile. Fie $g, g' \in F$. Există numerele naturale $k, m \in \mathbf{N} - \{0\}$ astfel încât $Dom(g) = k$ şi $Dom(g') = m$. Presupunem că $k \leq m$. Prin inducţie finitară arătăm că pentru orice $0 \leq x \leq k$ are loc $g(x) = g'(x)$:

- $g(0) = a = g'(0)$;
- dacă presupunem că $g(x) = g'(x)$ pentru $x < k$, atunci

$$g(S(x)) = h(x, g(x)) = h(x, g'(x)) = g'(S(x)).$$

In baza Principiului inducţiei finitare obţinem că g şi g' sunt compatibile. Ca urmare, F este mulţime de funcţii compatibile, ceea ce conduce la faptul că există funcţia $f = \bigcup F$ cu domeniul $\bigcup_{g \in F} Dom(g)$.

Arătăm că $Dom(f) = \mathbf{N}$. Domeniul funcţiei f este submulţime a mulţimii $\mathbf{N}$. Dacă presupunem că $\mathbf{N} - Dom(f)$ este nevidă, atunci ea va avea un cel mai mic

element, fie acesta x. Este clar că $x > 0$ şi, deci, există y astfel încât $x = S(y)$. Numărul y este în domeniul funcţiei f şi, deci, există $g \in F$ astfel încât $y \in Dom(g)$. Mai mult, nu există $z \geq x$ astfel încât $z \in Dom(g)$. Adică, $Dom(g) = x$. Vom arăta că există o funcţie $g' \in F$ al cărei domeniu conţine x.

Fie $g' = g \cup \{(x, h(y, g(y)))\}$. Este clar că g' este funcţie cu domeniul

$$Dom(g') = Dom(g) \cup \{x\} = S(x).$$

Arătăm că g' satisface $(*)$:

- $g'(0) = g(0) = a$;
- fie z astfel încât $S(z) \in Dom(g')$. Dacă $S(z) \in Dom(g)$, atunci

$$g'(S(z)) = g(S(z)) = h(z, g(z)) = h(z, g'(z)).$$

  Dacă $S(z) = x$, atunci $z = y$, iar de la definiţia funcţiei g' urmează că

$$g'(S(y)) = g'(x) = h(y, g(y)) = h(y, g'(y)).$$

Ca urmare g' satisface $(*)$ şi, deci, $g' \in F$. Aceasta contrazice presupunerea conform căreia $x \notin Dom(f)$ şi, deci, $Dom(f) = \mathbf{N}$.

Arătăm că funcţia f satisface (i) şi (ii) ale teoremei. Conform definiţiei ei,

$$f(0) = g(0) = a,$$

pentru orice $g \in F$ şi, deci, f satisface (i).

Fie $x \in Dom(f)$. Atunci, există $g \in F$ astfel încât $S(x) \in Dom(g)$. Deoarece F este mulţime de funcţii compatibile urmează că

$$f(S(x)) = g(S(x)) = h(x, g(x)) = h(x, f(x)),$$

ceea ce ne arată că f satisface (ii).

Unicitatea funcţiei f se obţine astfel. Dacă ar exista o altă funcţie g ce satisface (i) şi (ii), atunci prin inducţie după $n \in \mathbf{N}$ arătăm că $f(n) = g(n)$, ceea ce va conduce la $f = g$. In adevăr, $f(0) = a = g(0)$ şi, dacă presupunem că $f(n) = g(n)$, atunci

$$f(S(n)) = h(n, f(n)) = h(n, g(n)) = g(S(n)).$$

Ca urmare, $f(n) = g(n)$ pentru orice $n \in \mathbf{N}$, ceea ce arată că $f = g$. □

Funcţiile cu domeniul $\mathbf{N}$ sunt secvenţe infinite şi reciproc. Ca urmare, Teorema recursiei poate fi reformulată în termeni de secvenţe astfel [13]:

- dată o mulţime A, $a \in A$ şi o funcţie $h : \mathbf{N} \times A \to A$, există o unică secvenţă infinită $\langle a_i | i \geq 0 \rangle$ astfel încât:

[13] Şi celelalte variante de recursie, ce vor fi prezentate pe parcursul acestui capitol, pot fi reformulate în termeni de secvenţe.

– $a_0 = a$;

– $a_{n+1} = h(n, a_n)$, pentru orice $n \in \mathbf{N}$.

Deci, a defini recursiv o funcţie cu domeniul $\mathbf{N}$ revine la a defini o secvenţă infinită în care orice element al ei, exceptând primul, este "construit" pe baza elementului anterior:

$$f(0) = a,\ f(1) = h(0, f(0)),\ f(2) = h(1, f(1)),\ \ldots$$

Uneori, este bine de gândit această definiţie şi în modul următor: iniţial (la pasul 0) funcţia f este definită prin a, la pasul 1 funcţia f este definită prin $h(0, f(0))$, la pasul 2 funcţia f este definită prin $h(1, f(1))$ etc.

Operaţiile binare, cum ar fi de exemplu adunarea, înmulţirea etc., sunt funcţii de două variabile (definite pe produsul cartezian a două mulţimi). Teorema recursiei poate fi utilizată şi pentru a defini astfel de funcţii, pornind de la următoarea remarcă. Fie $f : A \times B \to C$ o funcţie. Dacă fixăm unul din argumente iar celălalt îl păstrăm variabil, de exemplu al doilea fix şi primul variabil, atunci pentru fiecare valoare $b \in B$ dată celui de-al doilea argument obţinem o funcţie cu un singur argument, $f_b : A \to C$, cu proprietatea $f_b(a) = f(a, b)$, pentru orice $a \in A$. Atunci, a defini funcţia f revine la a defini funcţiile f_b, pentru orice $b \in B$. Dacă B este mulţimea numerelor naturale, atunci putem utiliza Teorema recursiei pentru a defini o funcţie $F : \mathbf{N} \to C^A$ astfel încât $F(b) = f_b$ pentru orice $b \in B = \mathbf{N}$; adică, F va defini funcţiile f_b pentru orice $b \in B$. Aceasta va fi de fapt ideea de demonstraţie a următoarei teoreme.

Teorema 1.1.4.2. (Varianta parametrică a Teoremei recursiei)
Fie A şi P mulţimi, iar $g : P \to A$ şi $h : P \times \mathbf{N} \times A \to A$ funcţii. Atunci, există o unică funcţie $f : P \times \mathbf{N} \to A$ astfel încât:

(i) $f(p, 0) = g(p)$, pentru orice $p \in P$;

(ii) $f(p, S(n)) = h(p, n, f(p, n))$, pentru orice $p \in P$ şi $n \in \mathbf{N}$.

Demonstraţie Fie $f_0 : P \to A$ dată prin $f_0(p) = g(p)$ pentru orice $p \in P$, şi $H : \mathbf{N} \times A^P \to A^P$ dată prin $H(n, \varphi)(p) = h(p, n, \varphi(p))$ (este uşor de văzut că aceste funcţii există). Teorema recursiei va conduce la existenţa unei unice funcţii $F : \mathbf{N} \to A^P$ astfel încât:

– $F(0) = f_0$;

– $F(S(n)) = H(n, F(n))$, pentru orice $n \in \mathbf{N}$.

Definim atunci $f : P \times \mathbf{N} \to A$ prin $f(p, n) = F(n)(p)$, pentru orice $p \in P$ şi $n \in \mathbf{N}$. f este funcţie şi arătăm că ea satisface teorema:

- $f(p, 0) = F(0)(p) = f_0(p) = g(p)$, pentru orice $p \in P$;

- $f(p, S(n)) = F(S(n))(p) = H(n, F(n))(p)$
  $= h(p, n, F(n)(p))$
  $= h(p, n, f(p, n))$,

  pentru orice $p \in P$ şi $n \in \mathbf{N}$.

Unicitatea funcţiei f se stabileşte ca în Teorema 1.1.4.1. □

Demonstraţia Teoremei 1.1.4.2 ne arată clar că a defini în manieră recursivă o funcţie $f : P \times \mathbf{N} \to A$ înseamnă a defini o secvenţa infinită de funcţii de la P la A,

$$f_0, f_1, f_2, \ldots$$

Funcţia f va fi atunci dată prin $f(p, n) = f_n(p)$, pentru orice $p \in P$ şi $n \in \mathbf{N}$. Altfel spus, funcţia f "condensează" secvenţa infinită de mai sus. Diferenţa dintre Teorema 1.1.4.1 şi Teorema 1.1.4.2 este dată de "natura" elementelor secvenţei infinite definite.

> Prezentăm o nouă demonstraţie a Teoremei 1.1.4.2, bazată pe fixarea primului argument al funcţiei f.
>
> Pentru orice $p \in P$, Teorema recursiei asigură existenţa unei unice funcţii $f_p : \mathbf{N} \to A$ astfel încât:
>
> (i) $f_p(0) = g(p)$;
>
> (ii) $f_p(S(n)) = h_p(n, f_p(n))$, pentru orice $n \in \mathbf{N}$,
>
> unde h_p este funcţia $h_p(n, x) = h(p, n, x)$, pentru orice $n, x \in \mathbf{N}$.
>
> Funcţia $f = \bigcup_{p \in P} f_p$ verifică teorema.

Teorema recursiei şi varianta ei parametrică au importanţă majoră în definirea de funcţii şi operaţii pe numere naturale, în mod recursiv. Vom ilustra aceasta arătând cum pot fi definite riguros operaţiile de bază cu numere naturale.

Teorema 1.1.4.3.

(1) Există o unică operaţie $+ : \mathbf{N} \times \mathbf{N} \to \mathbf{N}$ astfel încât:

  (a) $+(m, 0) = m$, pentru orice $m \in \mathbf{N}$;

  (b) $+(m, S(n)) = S(+(m, n))$, pentru orice $m, n \in \mathbf{N}$.

(2) Există o unică operaţie $\cdot : \mathbf{N} \times \mathbf{N} \to \mathbf{N}$ astfel încât:

  (a) $\cdot(m, 0) = 0$, pentru orice $m \in \mathbf{N}$;

  (b) $\cdot(m, S(n)) = +(\cdot(m, n), m)$, pentru orice $m, n \in \mathbf{N}$.

(3) Există o unică operaţie $\widehat{\ }$: $\mathbf{N} \times \mathbf{N} \to \mathbf{N}$ astfel încât:

  (a) $\widehat{\ }(m, 0) = 1$, pentru orice $m \in \mathbf{N}$;

  (b) $\widehat{\ }(m, S(n)) = \cdot(\widehat{\ }(m, n), m)$, pentru orice $m, n \in \mathbf{N}$.

(4) Există o unică operaţie $S' : \mathbf{N} \to \mathbf{N}$ astfel încât:

(a) $S'(0) = 0$;

(b) $S'(S(n)) = n$, pentru orice $n \in \mathbf{N}$.

(5) Există o unică operaţie $\dot{-} : \mathbf{N} \times \mathbf{N} \to \mathbf{N}$ astfel încât:

(a) $\dot{-}(m, 0) = m$, pentru orice $m \in \mathbf{N}$;

(b) $\dot{-}(m, S(n)) = S'(\dot{-}(m, n))$, pentru orice $m, n \in \mathbf{N}$.

Demonstraţie In Teorem 1.1.4.2 considerăm $A = P = \mathbf{N}$ şi:

– $g(p) = p$ şi $h(p, n, x) = S(x)$, pentru orice $p, n, x \in \mathbf{N}$ (pentru operaţia +);

– $g(p) = 0$ şi $h(p, n, x) = +(x, p)$, pentru orice $p, n, x \in \mathbf{N}$ (pentru operaţia $\cdot$);

– $g(p) = 1$ şi $h(p, n, x) = \cdot(x, p)$, pentru orice $p, n, x \in \mathbf{N}$ (pentru operaţia $\widehat{\ }$);

– $g(p) = 0$ şi $h(p, n, x) = n$, pentru orice $p, n, x \in \mathbf{N}$ (pentru operaţia S');

– $g(p) = p$ şi $h(p, n, x) = S'(x)$, pentru orice $p, n, x \in \mathbf{N}$ (pentru operaţia $\dot{-}$).

Unica funcţie a cărei existenţă este asigurată de această teoremă este întocmai $+$ sau, respectiv, $\cdot$, $\widehat{\ }$, S', $\dot{-}$. □

Operaţia $+$ ($\cdot$, $\widehat{\ }$, $\dot{-}$) este numită *operaţia de adunare* (*înmulţire*, *ridicare la putere*, *diferenţă*, *scădere aritmetică*) pe $\mathbf{N}$; uzual vom folosi notaţia infix pentru ele, adică vom scrie $m + n$ ($m \cdot n$, $m \widehat{\ } n$, $m \dot{-} n$) în loc de $+(m, n)$ ($\cdot(m, n)$, $\widehat{\ }(m, n)$, $\dot{-}(m, n)$). Semnele operaţiilor de înmulţire şi ridicare la putere se omit cu precădere, utilizându-se mn şi m^n pentru $m \cdot n$ şi, respectiv, $m \widehat{\ } n$. De la Teorema 1.1.4.3 rezultă că are loc

$$S(m) = S(+(m, 0)) = +(m, S(0)) = +(m, 1) = m + 1,$$

ceea ce permite utilizarea notaţiei $m + 1$ pentru $S(m)$, care este mult mai intuitivă şi uşor de manipulat. (a1), (a2), (i1), (i2), (p1), (p2), (d1) şi (d2) din Teorema 1.1.4.3 pot fi reformulate astfel:

(a1') $m + 0 = m$;

(a2') $m + (n + 1) = (m + n) + 1$;

(i1') $m \cdot 0 = 0$;

(i2') $m \cdot (n + 1) = (m \cdot n) + m$;

(p1') $m^0 = 1$;

(p2') $m^{n+1} = m^n \cdot m$;

(d1') $m \dot{-} 0 = m$;

(d2') $m \dot{-} (n+1) = S'(m \dot{-} n)$,

pentru orice $m, n \in \mathbf{N}$.

Introducerea mulţimii numerelor naturale, împreună cu operaţiile de bază pe acestea, constituie un obiectiv major pe care considerăm că l-am dus la bun sfârşit. Din acest punct mai departe vom presupune că cititorul este familiarizat cu proprietăţile de bază ale numerelor naturale şi operaţiile cu acestea. De asemenea, presupunem că este cunoscut modul de introducere a celorlate sisteme de numere, *întregi* ($\mathbf{Z}$), *raţionale* ($\mathbf{Q}$), *reale* ($\mathbf{R}$) şi *complexe* ($\mathbf{C}$), precum şi a operaţiilor de bază pe acestea (pentru detalii, indicăm [182]). $\mathbf{Z}^*$ denotă $\mathbf{Z} - \{0\}$, $\mathbf{Z}_+$ denotă $\{x \in \mathbf{Z} | x \geq 0\}$, iar $\mathbf{Z}_+^*$ denotă $\{x \in \mathbf{Z} | x > 0\}$. Aceste notaţii sunt extinse şi la $\mathbf{Q}$ şi $\mathbf{R}$, iar notaţia "*" şi la $\mathbf{C}$.

1.2 Relaţii şi funcţii

In Secţiunea 1.1 s-a introdus, în manieră axiomatică, conceptul de mulţime şi, bazat pe acesta, concepetele de pereche ordonată, relaţie, funcţie şi număr natural. Toate acestea sunt fundamentale în matematică, ele constituind baza tuturor celorlalte concepte matematice.

In această secţiune vom aprofunda studiul acestor concepte de bază.

1.2.1 Relaţii

Relaţiile binare (Secţiunea 1.1.1) sunt mulţimi de perechi ordonate. Mulţimea vidă este relaţie, numită *relaţia vidă*. Notaţia $a \, \rho \, b$ este utilizată frecvent pentru a specifica faptul că (a, b) este element al relaţie ρ.

Deoarece relaţiile sunt mulţimi, putem construi reuniunea, intersecţia, diferenţa şi complementara lor, care sunt relaţii; egalitatea relaţiilor este egalitate de mulţimi. $Dom(\rho)$ şi $Cod(\rho)$ desemnează domeniul şi, respectiv, codomeniul relaţie ρ.

Exemplul 1.2.1.1. Fie A şi B mulţimi.

(1) Relaţia $=_A \subseteq A \times A$ dată prin

$$=_A = \{(a, a) | a \in A\}$$

este numită *relaţia de egalitate* pe A sau *identitatea* pe A sau *diagonala* lui $A \times A$ (frecvent notată şi prin ι_A [14]).

[14] Notaţia ι_A este de preferat notaţiei $=_A$ care poate reduce lizibilitatea textului cum ar fi de exemplu în scrieri de forma "$\rho ==_A$".

(2) Relaţia $\in_A \subseteq A \times A$ dată prin

$$\in_A = \{(a,b) | a,b \in A,\ a \in b\}$$

este numită *relaţia de apartenenţă* pe A.

(3) Relaţia $\subseteq_A \subseteq A \times A$ dată prin

$$\subseteq_A = \{(a,b) | a,b \in A,\ a \subseteq b\}$$

este numită *relaţia de incluziune* pe A. Inlocuind $\subseteq$ prin $\subset$, obţinem *relaţia de incluziune strictă* pe A, notată prin $\subset_A$.

(4) Relaţia $\omega_{A,B} \subseteq A \times B$ dată prin

$$\omega_{A,B} = \{(a,b) | a \in A, b \in B\} = A \times B$$

este numită *relaţia completă* de la A la B. In cazul $A = B$, notaţia $\omega_{A,B}$ va fi simplificată la ω_A, care este numită *relaţia completă* pe A.

Atunci când mulţimea A este subînţeleasă din context, notaţia $=_A$ (ι_A, $\in_A$, $\subseteq_A$, $\subset_A$, ω_A) va fi simplificată la $=$ (ι, $\in$, $\subseteq$, $\subset$, ω).

Definiţia 1.2.1.1. Fie ρ o relaţie binară şi A o mulţime. *Restricţia relaţiei ρ la A* este relaţia binară notată $\rho|_A$ şi dată prin

$$\rho|_A = \rho \cap (A \times A).$$

Relaţia $\rho|_A$ este intersecţia a două relaţii, $\rho|_A = \rho \cap \omega_A$. Acest fapt permite dezvoltarea unor proprietăţi ale relaţiei $\rho|_A$ uzând de diverse proprietăţi ale intersecţiei de relaţii.

Evident, restricţia unei relaţii binare se poate face restrângând doar domeniul sau doar codomeniul acesteia, sau restrângându-le pe ambele dar în mod diferit. In cazul Definiţiei 1.2.1.1, atât domeniul cât şi codomeniul sunt restricţionate prin intermediul aceleiaşi mulţimi A.

Este adesea util a reprezenta grafic relaţiile binare. Reprezentarea grafică a unei relaţii ρ se face printr-un graf orientat ale cărui noduri sunt etichetate cu elementele mulţimii $Dom(\rho) \cup Cod(\rho)$. Pentru fiecare pereche $(a,b) \in \rho$ se trasează un arc de la nodul cu eticheta a la nodul cu eticheta b. In mod frecvent nodurile sunt identificate prin etichetele lor (distincţia nod-etichetă fiind esenţială atunci când noduri diferite sunt etichetate cu aceeaşi etichetă). In Figura 1.2 este reprezentată grafic relaţia

$$\rho = \{(a,a), (a,b), (b,c), (a,c), (a,d)\},$$

punând în evidenţă atât reprezentarea cu noduri etichetate cât şi cea în care nodurile sunt identificate cu etichetele lor.

Următoarea propoziţie prezintă câteva proprietăţi elementare ale domeniului şi codomeniului unei relaţii.

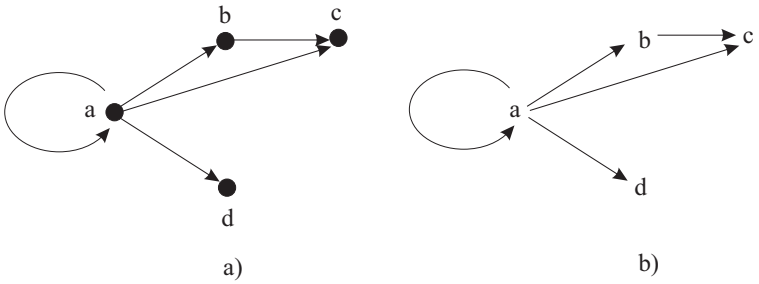

Figura 1.2: Reprezentări grafice ale aceleiaşi relaţii binare

Propoziţia 1.2.1.1. Fie ρ şi σ două relaţii binare. Atunci, au loc următoarele proprietăţi:

(1) $Dom(\rho \cup \sigma) = Dom(\rho) \cup Dom(\sigma)$;

(2) $Cod(\rho \cup \sigma) = Cod(\rho) \cup Cod(\sigma)$;

(3) $Dom(\rho \cap \sigma) \subseteq Dom(\rho) \cap Dom(\sigma)$;

(4) $Cod(\rho \cap \sigma) \subseteq Cod(\rho) \cap Cod(\sigma)$;

(5) $Dom(\rho) - Dom(\sigma) \subseteq Dom(\rho - \sigma)$;

(6) $Cod(\rho) - Cod(\sigma) = Cod(\rho - \sigma)$;

(7) dacă $\rho \subseteq \sigma$, atunci $Dom(\rho) \subseteq Dom(\sigma)$ şi $Cod(\rho) \subseteq Cod(\sigma)$.

Atragem atenţia asupra incluziunilor din Propoziţia 1.2.1.1(1)(2)(3)(4). Ele pot fi stricte. De exemplu, dacă $\rho = \{(a, b)\}$ şi $\sigma = \{(a, c)\}$, unde $b \neq c$, atunci

$$Dom(\rho \cap \sigma) = \emptyset \quad \text{dar} \quad Dom(\rho) \cap Dom(\sigma) = \{a\}.$$

Produsul şi inversa relaţiilor binare sunt "operaţii" specifice de mare importanţă în studiul acestora.

Definiţia 1.2.1.2. Fie ρ şi σ două relaţii binare. Relaţia binară notată $\rho \circ \sigma$ şi dată prin

$$\rho \circ \sigma = \{(a, c) | (\exists b)((a, b) \in \rho \ \wedge \ (b, c) \in \sigma)\}$$

este numită *produsul* relaţiilor ρ şi σ.

Este clar că pentru orice două relaţii ρ şi σ, produsul lor este relaţie binară (deci, Definiţia 1.2.1.3 este consistentă). Dacă ρ este relaţie de la A la B, iar σ de la C la D, atunci $\rho \circ \sigma$ este relaţie de la A la D. In plus, dacă $Cod(\rho) \cap Dom(\sigma) = \emptyset$, atunci $\rho \circ \sigma = \emptyset$.

Următoarea propoziţie prezintă câteva proprietăţile de bază ale produsului de relaţii.

Propoziţia 1.2.1.2. Fie ρ, σ şi θ relaţii binare, iar A şi B mulţimi. Atunci, au loc următoarele proprietăţi:

(1) $Dom(\rho \circ \sigma) \subseteq Dom(\rho)$;

(2) $Cod(\rho \circ \sigma) \subseteq Cod(\sigma)$;

(3) $\rho \circ (\sigma \circ \theta) = (\rho \circ \sigma) \circ \theta$;

(4) $\rho \circ (\sigma \cup \theta) = (\rho \circ \sigma) \cup (\rho \circ \theta)$;

(5) $(\rho \cup \sigma) \circ \theta = (\rho \circ \theta) \cup (\sigma \circ \theta)$;

(6) $\rho \circ (\sigma \cap \theta) \subseteq (\rho \circ \sigma) \cap (\rho \circ \theta)$;

(7) $(\rho \cap \sigma) \circ \theta \subseteq (\rho \circ \theta) \cap (\sigma \circ \theta)$;

(8) $\rho \circ \sigma - \rho \circ \theta \subseteq \rho \circ (\sigma - \theta)$;

(9) dacă $\sigma \subseteq \theta$, atunci $\rho \circ \sigma \subseteq \rho \circ \theta$ şi $\sigma \circ \rho \subseteq \theta \circ \rho$;

(10) $\iota_A \circ \rho \subseteq \rho$ şi $\rho \circ \iota_B \subseteq \rho$. In plus, $\iota_A \circ \rho = \rho$ dacă şi numai dacă $Dom(\rho) \subseteq A$ şi, $\rho \circ \iota_B = \rho$ dacă şi numai dacă $Cod(\rho) \subseteq B$.

Demonstraţie Vom demonstra doar (10). Fie $(a, b) \in \iota_A \circ \rho$. Atunci, există c astfel încât $(a, c) \in \iota_A$ şi $(c, b) \in \rho$. Conform definiţiei relaţiei ι_A, urmează $a = c$ şi, deci, $(a, b) \in \rho$. Am obţinut astfel incluziunea $\iota_A \circ \rho \subseteq \rho$; incluziunea $\rho \circ \iota_B \subseteq \rho$ se demonstrează similar acesteia.

Să presupunem acum că $\iota_A \circ \rho = \rho$ şi să arătăm că $Dom(\rho) \subseteq A$. Fie $a \in Dom(\rho)$. Atunci, există b astfel încât $(a, b) \in \rho$. Deoarece $\rho = \iota_A \circ \rho$, obţinem $(a, b) \in \iota_A \circ \rho$ şi, deci, va exista c astfel încât $(a, c) \in \iota_A$ şi $(c, b) \in \rho$. Conform definiţiei relaţiei ι_A avem $c = a$ şi, deci, $a \in A$. Am obţinut astfel $Dom(\rho) \subseteq A$.

Reciproc, să presupunem că $Dom(\rho) \subseteq A$. Conform cu ceea ce am demonstrat anterior ($\iota_A \circ \rho \subseteq \rho$), ne rămâne de arătat că $\rho \subseteq \iota_A \circ \rho$. Fie deci $(a, b) \in \rho$. Cum $Dom(\rho) \subseteq A$ urmează că $a \in A$ şi, atunci, putem scrie $(a, b) \in \iota_A \circ \rho$. Am obţinut astfel $\rho = \iota_A \circ \rho$.

Echivalenţa "$\rho \circ \iota_B = \rho$ dacă şi numai dacă $Cod(\rho) \subseteq B$" se demonstrează similar celei precedente. □

Atragem atenţia asupra incluziunilor din Propoziţia 1.2.1.2(1)(2). Dacă, de exemplu, există $a \in Dom(\rho)$ astfel încât

$$\{b|(a, b) \in \rho\} \cap Dom(\sigma) = \emptyset,$$

atunci $Dom(\rho \circ \sigma) \subset Dom(\rho)$. Similar, dacă există $c \in Cod(\sigma)$ astfel încât

$$Cod(\rho) \cap \{b|(b, c) \in \sigma\} = \emptyset,$$

atunci $Cod(\rho \circ \sigma) \subset Cod(\sigma)$.

Asociativitatea produsului de relaţii ne permite să scriem $\rho \circ \sigma \circ \theta$ în loc de $(\rho \circ \sigma) \circ \theta$ sau $\rho \circ (\sigma \circ \theta)$. Astfel, dacă $(a, d) \in \rho \circ \sigma \circ \theta$, atunci există b şi c astfel încât $(a, b) \in \rho$, $(b, c) \in \sigma$ şi $(c, d) \in \theta$.

Atunci când nu există pericol de confuzie semnul operaţiei de compunere, "$\circ$", va fi omis. Astfel, în loc de $\rho \circ \sigma$ vom scrie $\rho\sigma$.

Definiţia 1.2.1.3. Fie ρ o relaţie binară. *Inversa* relaţiei ρ este relaţia notată ρ^{-1} şi dată prin

$$\rho^{-1} = \{(b, a) | (a, b) \in \rho\}.$$

Inversa unei relaţii ρ există întotdeauna, iar dacă ρ este relaţie de la A la B, atunci ρ^{-1} este relaţie de la B la A. Pentru anumite relaţii inversa are o notaţie consacrată. Următorul tabel prezintă câteva dintre aceste notaţii (A este o mulţime arbitrară):

ρ	$\leq$	$<$	$\rightarrow$	$\Rightarrow$	$\subseteq_A$	$\subset_A$
ρ^{-1}	$\geq$	$>$	$\leftarrow$	$\Leftarrow$	$\supseteq_A$	$\supset_A$

Propoziţia 1.2.1.3. Fie ρ şi σ relaţii binare. Atunci, au loc următoarele proprietăţi:

(1) $Dom(\rho^{-1}) = Cod(\rho)$;

(2) $Cod(\rho^{-1}) = Dom(\rho)$;

(3) $(\rho^{-1})^{-1} = \rho$;

(4) dacă $\rho \subseteq \sigma$, atunci $\rho^{-1} \subseteq \sigma^{-1}$;

(5) $(\rho \cup \sigma)^{-1} = \rho^{-1} \cup \sigma^{-1}$;

(6) $(\rho \cap \sigma)^{-1} = \rho^{-1} \cap \sigma^{-1}$;

(7) $(\rho - \sigma)^{-1} = \rho^{-1} - \sigma^{-1}$;

(8) $(\rho \circ \sigma)^{-1} = \sigma^{-1} \circ \rho^{-1}$.

Demonstraţie Vom demonstra doar (8). Fie $(a, b) \in (\rho \circ \sigma)^{-1}$. Atunci, $(b, a) \in \rho \circ \sigma$ şi există c astfel încât $(b, c) \in \rho$ şi $(c, a) \in \sigma$. Ca urmare, $(c, b) \in \rho^{-1}$ şi $(a, c) \in \sigma^{-1}$, ceea ce arată că $(a, b) \in \sigma^{-1} \circ \rho^{-1}$. Am obţinut astfel incluziunea $(\rho \circ \sigma)^{-1} \subseteq \sigma^{-1} \circ \rho^{-1}$; incluziunea în sens invers se arată în mod similar. □

Definiţia 1.2.1.4. Fie ρ o relaţie binară, iar A şi B mulţimi.

(1) *Imaginea mulţimii A prin ρ*, notată $\rho(A)$, este mulţimea

$$\rho(A) = \{b | (\exists a \in A)(a \, \rho \, b)\}.$$

(2) *Imaginea inversă a mulţimii B prin ρ*, notată $\rho^{-1}(B)$, este mulţimea

$$\rho^{-1}(B) = \{a | (\exists b \in B)(a \, \rho \, b)\}.$$

Este clar că $\rho(A)$ şi $\rho^{-1}(B)$ există ($\rho^{-1}(B)$ este de fapt imaginea mulţimii B prin relaţia binară ρ^{-1}). Atunci când A este de forma $\{a\}$ vom nota $\rho(a)$ în loc de $\rho(\{a\})$.

Propoziţia 1.2.1.4. Fie ρ şi σ relaţii binare, iar A şi B mulţimi. Atunci, au loc următoarele proprietăţile:

(1) $\rho(A \cup B) = \rho(A) \cup \rho(B)$;

(2) dacă $A \subseteq B$, atunci $\rho(A) \subseteq \rho(B)$;

(3) $\rho(A \cap B) \subseteq \rho(A) \cap \rho(B)$;

(4) $\rho(A) - \rho(B) \subseteq \rho(A - B)$;

(5) $\rho(A) = \emptyset$ dacă şi numai dacă $Dom(\rho) \cap A = \emptyset$;

(6) $Dom(\rho) \cap A \subseteq \rho^{-1}(\rho(A))$ şi $Cod(\rho) \cap B \subseteq \rho(\rho^{-1}(B))$;

(7) $(\rho \circ \sigma)(A) = \sigma(\rho(A))$.

Demonstraţie (1) Afirmaţia se obţine pe baza echivalenţelor:

$$\begin{aligned} b \in \rho(A) \cup \rho(B) &\Leftrightarrow b \in \rho(A) \ \vee \ b \in \rho(B) \\ &\Leftrightarrow (\exists a \in A : \ a \,\rho\, b) \ \vee \ (\exists a \in B : \ a \,\rho\, b) \\ &\Leftrightarrow \exists a \in A \cup B : \ a \,\rho\, b \\ &\Leftrightarrow b \in \rho(A \cup B), \end{aligned}$$

pentru orice b.

(2) Dacă $A \subseteq B$ atunci $B = A \cup B$. Utilizând (1) obţinem

$$\rho(A) \cup \rho(B) = \rho(B),$$

ceea ce arată că $\rho(A) \subseteq \rho(B)$.

(3) Deoarece $A \cap B \subseteq A$ şi $A \cap B \subseteq B$, de la (2) urmează

$$\rho(A \cap B) \subseteq \rho(A) \ \text{ şi } \ \rho(A \cap B) \subseteq \rho(B).$$

Atunci, $\rho(A \cap B) \subseteq \rho(A) \cap \rho(B)$.

(4) Dacă $c \in \rho(A) - \rho(B)$, atunci există $a \in A$ astfel încât $a \,\rho\, c$ şi, pentru orice $b \in B$, $(b, c) \notin \rho$. Aceasta ne arată că $a \in A - B$ şi, deci, $c \in \rho(A - B)$. Ca urmare, $\rho(A) - \rho(B) \subseteq \rho(A - B)$.

(5) urmează direct de la faptul că $b \in \rho(A)$ dacă şi numai dacă există $a \in Dom(\rho) \cap A$ astfel încât $(a, b) \in \rho$.

(6) Pentru orice $a \in Dom(\rho) \cap A$, $\{b | a \,\rho\, b\} \subseteq \rho(A)$ şi, deci,

$$Dom(\rho) \cap A \subseteq \rho^{-1}(\rho(A)).$$

Similar se obţine şi incluziunea $Cod(\rho) \cap B \subseteq \rho(\rho^{-1}(B))$.

(7) Afirmaţia se obţine pe baza echivalenţelelor:

$$\begin{aligned} c \in (\rho \circ \sigma)(A) &\Leftrightarrow \exists a \in A:\ (a,c) \in \rho \circ \sigma \\ &\Leftrightarrow \exists a \in A,\ \exists b \in Cod(\rho) \cap Dom(\sigma):\ a\,\rho\,b \ \wedge\ b\,\sigma\,c \\ &\Leftrightarrow \exists b \in \rho(A):\ b\,\sigma\,c \\ &\Leftrightarrow c \in \sigma(\rho(A)), \end{aligned}$$

pentru orice c. □

Vom prezenta acum câteva tipuri importante de relaţii binare cât şi simple caracterizări ale acestora.

Definiţia 1.2.1.5. Fie ρ o relaţie binară şi A o mulţime.

(1) ρ este numită *reflexivă pe* A dacă are loc

$$(\forall a)(a \in A \ \Rightarrow\ (a,a) \in \rho).$$

(2) ρ este numită *ireflexivă pe* A dacă are loc

$$(\forall a)(a \in A \ \Rightarrow\ (a,a) \notin \rho).$$

(3) ρ este numită *simetrică pe* A dacă are loc

$$(\forall a,b)(a,b \in A \ \wedge\ (a,b) \in \rho \ \Rightarrow\ (b,a) \in \rho).$$

(4) ρ este numită *asimetrică pe* A dacă are loc

$$(\forall a,b)(a,b \in A \ \wedge\ (a,b) \in \rho \ \Rightarrow\ (b,a) \notin \rho).$$

(5) ρ este numită *antisimetrică pe* A dacă are loc

$$(\forall a,b)(a,b \in A \ \wedge\ (a,b) \in \rho \ \wedge\ (b,a) \in \rho \ \Rightarrow\ a = b).$$

(6) ρ este numită *tranzitivă pe* A dacă are loc

$$(\forall a,b,c)(a,b,c \in A \ \wedge\ (a,b) \in \rho \ \wedge\ (b,c) \in \rho \ \Rightarrow\ (a,c) \in \rho).$$

(7) ρ este numită *conexă pe* A dacă are loc

$$(\forall a,b)(a,b \in A \ \Rightarrow\ a\,\rho\,b \ \vee\ a = b \ \vee\ b\,\rho\,a).$$

(8) ρ este numită *dirijată pe* A [15] dacă are loc

$$(\forall a,b)(a,b \in A \ \Rightarrow\ (\exists\, c \in A)(a\,\rho\,c \ \wedge\ b\,\rho\,c)).$$

[15] Conceptul de relaţie dirijată apare pentru prima dată în lucrarea lui Moore şi Smith asupra unei teorii generale a conceptului de limită [133]. Acest concept de relaţie dirijată s-a dovedit ulterior de importanţă foarte mare în informatică, în studiul semanticii limbajelor de programare şi al domeniilor semantice.

(9) ρ este numită *filtrată pe* A dacă are loc

$$(\forall a, b)(a, b \in A \ \Rightarrow \ (\exists\, c \in A)(c\,\rho\, a \ \wedge \ c\,\rho\, b)).$$

(10) ρ este numită *reflexivă* (*ireflexivă, simetrică, asimetrică, antisimetrică, tranzitivă, conexă, dirijată, filtrată*) dacă ρ este reflexivă (*ireflexivă, simetrică, asimetrică, antisimetrică, tranzitivă, conexă, dirijată, filtrată*) pe mulţimea $Dom(\rho) \cup Cod(\rho)$.

Teorema 1.2.1.1. Fie ρ o relaţie binară şi $A = Dom(\rho) \cup Cod(\rho)$.

(1) ρ este reflexivă dacă şi numai dacă $\iota_A \subseteq \rho$.

(2) ρ este ireflexivă dacă şi numai dacă $\iota_A \cap \rho = \emptyset$.

(3) ρ este simetrică dacă şi numai dacă $\rho = \rho^{-1}$.

(4) ρ este antisimetrică dacă şi numai dacă $\rho \cap \rho^{-1} \subseteq \iota_A$.

(5) ρ este asimetrică dacă şi numai dacă $\rho \cap \rho^{-1} = \emptyset$.

(6) ρ este tranzitivă dacă şi numai dacă $\rho \circ \rho \subseteq \rho$.

(7) ρ este conexă dacă şi numai dacă $\rho \cup \rho^{-1} \cup \iota_A = A \times A$.

Demonstraţie Vom demonstra ca exemplu (4), celelalte rămânând în seama cititorului. Să presupunem deci că ρ este antisimetrică. Pentru orice $(a, b) \in \rho \cap \rho^{-1}$ are loc $(a, b) \in \rho$ şi $(b, a) \in \rho$. Relaţia ρ fiind antisimetrică, deducem $a = b$ şi, deci, $(a, b) \in \iota_A$. Am obţinut astfel $\rho \cap \rho^{-1} \subseteq \iota_A$. □

Dată o relaţie ρ pe A definim:

- $\rho^0 = \iota_A$;
- $\rho^{n+1} = \rho^n \circ \rho$, pentru orice $n \geq 0$;
- $\rho^+ = \bigcup_{n \geq 1} \rho^n$;
- $\rho^* = \bigcup_{n \geq 0} \rho^n$.

Corolarul 1.2.1.1. Fie ρ o relaţie pe A. Atunci,

(1) ρ^+ este cea mai mică relaţie tranzitivă pe A ce include ρ;

(2) ρ^* este cea mai mică relaţie reflexivă şi tranzitivă pe A ce include ρ.

Demonstraţie (1) Conform definiţiei, ρ^+ include ρ. In plus,

$$\rho^+ \circ \rho^+ = \bigcup_{n,m \geq 1} \rho^{n+m} \subseteq \rho^+,$$

ceea ce arată că ρ^+ este tranzitivă (Teorema 1.2.1.1(6)).

Dacă θ este o relaţie tranzitivă ce include ρ, atunci ea trebuie să includă şi ρ^2. Acum, incluzând ρ şi ρ^2, va trebui să includă şi ρ^3. Inductiv, θ trebuie să includă ρ^n, pentru orice $n \geq 1$. Deci, θ trebuie să includă ρ^+, ceea ce demonstrează (1).

(2) $\iota_A \subseteq \rho^*$ şi, deci, ρ^* este reflexivă. Restul se arată ca la (1). □

Relaţia ρ^+ este numită *închiderea tranzitivă a relaţiei* ρ, iar ρ^*, *închiderea reflexivă şi tranzitivă a relaţiei* ρ (asupra acestor relaţii vom reveni în Secţiunea 2.2).

Reprezentarea grafică a relaţiilor reflexive se simplifică, în mod frecvent, prin eliminarea arcelor de la nod la el însuşi. O simplificare mult mai consistentă se face pentru relaţii tranzitive. Dacă ρ este o relaţie tranzitivă, atunci reprezentarea grafică a ei se substituie prin reprezentarea grafică a relaţiei

$$\rho' = \rho - \rho \circ \rho.$$

De exemplu, relaţia

$$\rho = \{(a,a),(b,b),(c,c),(a,b),(b,c),(a,c)\}$$

este atât reflexivă cât şi tranzitivă. Reprezentarea grafică a ei este dată în Figura 1.3(a), iar cea simplificată în Figura 1.3(b). Atragem atenţia asupra faptului că atunci

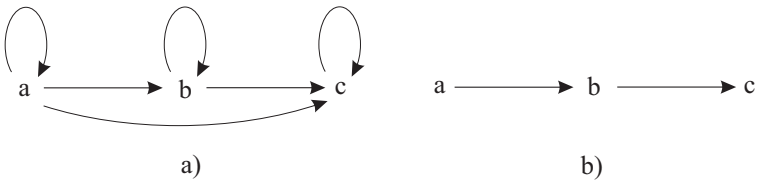

Figura 1.3: Reprezentare simplificată a unei relaţii reflexive şi tranzitive

când se fac astfel de reprezentări simplificate tipul relaţiei trebuie să rezulte clar din context.

Conceptul de relaţie binară poate fi extins la cel de *relaţie ternară* ca fiind o mulţime de 3-uple sau, ca fiind o submulţime a unui produs cartezian $A \times B \times C$. Dacă $A = B = C$, relaţia va mai fi numită *relaţie ternară pe* A.

Evident, extensia de mai sus poate fi realizată pentru orice $n \geq 2$ arbitrar, obţinându-se astfel conceptul de *relaţie* n*-ară*.

1.2.2 Relaţii de echivalenţă

Clasa relaţiilor de echivalenţă este una din cele mai importante clase de relaţii binare.

Definiţia 1.2.2.1. Fie ρ o relaţie binară şi A o mulţime. Spunem că ρ este *relaţie de echivalenţă pe* A dacă ρ este reflexivă, simetrică şi tranzitivă pe A. Atunci când $A = Dom(\rho) \cup Ran(\rho)$ vom spune că ρ este *relaţie de echivalenţă*.

Relaţia vidă este relaţie de echivalenţă numai pe mulţimea vidă (pe mulţimi nevide ea este simetrică şi tranzitivă dar nu este reflexivă).

Este uşor de văzut că dacă ρ este relaţie de echivalenţă atunci, pentru orice mulţime A, $\rho|_A$ este de asemenea relaţie de echivalenţă.

Exemplul 1.2.2.1. Fie A o mulţime nevidă. Relaţia binară $=_A$ (definită în Exemplul 1.2.1.1(1)) este relaţie de echivalenţă pe A.

Observaţia 1.2.2.1. Echipotenţa, introdusă în Secţiunea 1.1.1, verifică următoarele proprietăţi:

- $A \sim A$, pentru orice mulţime A;
- dacă $A \sim B$ atunci $B \sim A$, pentru orice mulţimi A şi B;
- dacă $A \sim B$ şi $B \sim C$ atunci $A \sim C$, pentru orice mulţimi A, B şi C.

Ca urmare, echipotenţa ar avea atributele unei relaţii de echivalenţă dar nu este relaţie de echivalenţă deoarece clasa tuturor mulţimilor, peste care s-ar considera echipotenţa ca relaţie binară, nu este mulţime. Dacă însă considerăm echipotenţa peste o familie de mulţimi $\mathcal{A}$, să o notăm prin $\sim_{\mathcal{A}}$, atunci ea devine relaţie de echivalenţă pe $\mathcal{A}$.

Definiţia 1.2.2.2. Fie ρ o relaţie de echivalenţă şi a un element. Se numeşte *clasa de echivalenţă a lui a modulo/relativ la ρ* mulţimea

$$[a]_\rho = \{b | a \,\rho\, b\}.$$

Este clar că, pentru orice a, clasa de echivalenţă a lui a modulo ρ există. Reflexivitatea asigură că această clasă este nevidă (conţine măcar pe a).

Lema 1.2.2.1. Fie ρ o relaţie de echivalenţă şi a, b două elemente. Atunci, au loc următoarele proprietăţi:

(1) $a \,\rho\, b$ dacă şi numai dacă $[a]_\rho = [b]_\rho$;

(2) $\neg(a \,\rho\, b)$ dacă şi numai dacă $[a]_\rho \cap [b]_\rho = \emptyset$.

Demonstraţie (1) Să presupunem că $a \,\rho\, b$. Fie $x \in [a]_\rho$. Urmează că $x \,\rho\, a$ (pe baza simetriei), $x \,\rho\, b$ (pe baza tranzitivităţii) şi $b \,\rho\, x$ (pe baza simetriei); deci, $x \in [b]_\rho$. Am obţinut astfel $[a]_\rho \subseteq [b]_\rho$; similar se arată şi cealaltă incluziune.

Reciproc, dacă presupunem că $[a]_\rho = [b]_\rho$, atunci $b \in [a]_\rho$ (deoarece $b \in [b]_\rho$) şi, deci, $a \,\rho\, b$.

(2) Dacă $\neg(a \,\rho\, b)$ atunci $[a]_\rho \neq [b]_\rho$ (de la (1)). Dacă mulţimile $[a]_\rho$ şi $[b]_\rho$ ar conţine elemente comune, fie c un astfel de element, atunci $a \,\rho\, c$ şi $c \,\rho\, b$ ar conduce la $a \,\rho\, b$; contradicţie.

Reciproc, dacă $[a]_\rho \cap [b]_\rho = \emptyset$, atunci $[a]_\rho \neq [b]_\rho$, iar (1) conduce la $\neg(a \,\rho\, b)$. □

Dacă ρ este o relaţie de echivalenţă pe o mulţime nevidă A, atunci Lema 1.2.2.1 ne spune că ρ împarte mulţimea A în submulţimi disjuncte (clase de echivalenţă),

fiecare submulţime fiind alcătuită din exact acele elemente ce sunt *echivalente* modulo ρ. Notăm mulţimea tuturor claselor de echivalenţă induse de relaţia ρ prin A/ρ şi o numim *mulţimea cât* sau *factor indusă de* A *şi* ρ (existenţa acestei mulţimi este asigurată de Axiomele părţilor şi separării). Adică,

$$A/\rho = \{[x]_\rho | x \in A\}.$$

Există o strânsă legătură între mulţimea partiţiilor unei mulţimi A, $Part(A)$, şi mulţimea relaţiilor de echivalenţă pe A, notată $E(A)$.

Definiţia 1.2.2.3. Fie A o mulţime nevidă iar S_1 şi S_2 două partiţii ale lui A. Spunem că S_1 *rafinează* pe S_2, şi notăm $S_1 \leq S_2$, dacă pentru orice bloc $X \in S_1$ există un bloc $Y \in S_2$ astfel încât $X \subseteq Y$.

Teorema 1.2.2.1. Fie A o mulţime nevidă.

(1) Fie S o partiţie a mulţimii A şi ρ_S relaţia binară pe A dată prin:

$$a \,\rho_S\, b \;\Leftrightarrow\; (\exists X \in S)(a, b \in X),$$

pentru orice $a, b \in A$. Atunci, ρ_S este relaţie de echivalenţă pe A.

(2) Fie ρ o relaţie de echivalenţă pe A şi S_ρ mulţimea tuturor claselor de echivalenţă induse de ρ. Atunci, S_ρ este partiţie a mulţimii A.

(3) (a) Dacă S_1 şi S_2 sunt partiţii ale mulţimii A astfel încât $S_1 \leq S_2$, atunci $\rho_{S_1} \subseteq \rho_{S_2}$.

(b) Dacă ρ_1 şi ρ_2 sunt relaţii de echivalenţă pe A astfel încât $\rho_1 \subseteq \rho_2$, atunci $S_{\rho_1} \leq S_{\rho_2}$.

(4) (a) Dacă S este partiţie a mulţimii A, atunci $S = S_{\rho_S}$.

(b) Dacă ρ este relaţie de echivalenţă pe A, atunci $\rho = \rho_{S_\rho}$.

Demonstraţie (1) şi (2) necesită doar simple verificări şi, ca urmare, vom trece la a demonstra celelalte proprietăţi.

(3)(a) Fie $(a, b) \in \rho_{S_1}$. Există atunci un bloc $X \in S_1$ astfel încât $a, b \in X$. Deoarece $S_1 \leq S_2$, va exista $Y \in S_2$ astfel încât $X \subseteq Y$. Aceasta conduce la $a, b \in Y$, adică $(a, b) \in \rho_{S_2}$. Deci, $\rho_{S_1} \subseteq \rho_{S_2}$.

Afirmaţia de la (3)(b) se obţine similar celei precedente.

(4)(a) Este suficient să arătăm că pentru orice $X \in S$ există o clasă de echivalenţă $[x]_{\rho_S}$ astfel încât $X = [x]_{\rho_S}$, şi reciproc.

Fie $X \in S$. Considerăm un element arbitrar x din X (există un astfel de element căci X este nevidă) şi arătăm că $X = [x]_{\rho_S}$. Dacă $y \in X$, atunci $x \,\rho_S y$ şi, deci, $y \in [x]_{\rho_S}$; dacă $y \in [x]_{\rho_S}$, atunci $x \,\rho_S\, y$ şi, deci, x şi y sunt în acelaşi bloc al partiţiei S. Cum $x \in X$ urmează că $y \in X$. Am demonstrat astfel că $X = [x]_{\rho_S}$.

Reciproc, dacă $[x]_{\rho_S}$ este o clasă de echivalenţă, atunci există un unic bloc X ce conţine x. Printr-un raţionament asemănător celui de mai sus se arată că $X = [x]_{\rho_S}$. De la acestea urmează $S = S_{\rho_S}$.

Afirmaţia de la (4)(b) se obţine similar celei precedente. □

Putem spune deci că relaţiile de echivalenţă pe o mulţime şi partiţiile acelei mulţimi sunt "descrieri diferite ale aceleiaşi entităţi matematice". Atunci când lucrăm cu astfel de entităţi este convenabil de a avea câte un "reprezentant" al fiecărei clase de echivalenţă. Suntem astfel conduşi la a ne întreba asupra existenţei unei mulţimi de reprezentanţi pentru o partiţie. Această chestiune a fost de altfel abordată în Secţiunea 1.1.1 şi, aşa cum am menţionat, o vom trata complet în secţiunea dedicată Axiomei alegerii.

Funcţiile injective "păstrează" relaţiile de echivalenţă. Fie A o mulţime, ρ o relaţie pe A şi $f : A \to B$ o funcţie. Notăm prin $f(\rho)$ relaţia

$$f(\rho) = \{(f(a), f(b))|(a, b) \in \rho\}.$$

Propoziţia 1.2.2.1. Fie $f : A \to B$ o funcţie şi ρ o relaţie de echivalenţă pe A. Dacă f este funcţie injectivă, atunci $f(\rho)$ este relaţie de echivalenţă pe $f(A)$.

Demonstraţie Reflexivitatea şi simetria relaţiei $f(\rho)$ se obţin imediat. Să discutăm tranzitivitatea.

Fie $(x, y), (y, z) \in f(\rho)$. Atunci, există $(a, b), (c, d) \in \rho$ astfel încât $f(a) = x$, $f(b) = y$, $f(c) = y$ şi $f(d) = z$. Injectivitatea funcţiei f conduce la $b = c$, iar tranzitivitatea relaţiei ρ conduce la $(a, d) \in \rho$ şi, deci, $(x, z) \in f(\rho)$. Deci, $f(\rho)$ este tranzitivă. Impreună cu celelalte două proprietăţi, $f(\rho)$ devide relaţie de echivalenţă pe $f(A)$. □

Atragem atenţia asupra necesităţii proprietăţii de injectivitate în a obţine tranzitivitatea relaţiei $f(\rho)$ (a se vedea demonstraţia propoziţiei). De asemenea, atragem atenţia asupra faptului că $f(\rho)$ este relaţie de echivalenţă pe $f(A)$ şi nu pe B, în mod necesar. Aceasta pentru că este posibil să se piardă proprietatea de reflexivitate.

Corolarul 1.2.2.1. Fie $f : A \to B$ o funcţie şi ρ o relaţie de echivalenţă pe A. Dacă f este funcţie bijectivă, atunci $f(\rho)$ este relaţie de echivalenţă pe B.

Fie A o mulţime şi $\theta, \rho \in E(A)$ astfel încât $\rho \subseteq \theta$. Simpla incluziune a relaţiei ρ în θ ne spune că orice clasă de echivalenţă în raport cu ρ este inclusă în exact o clasă de echivalenţă în raport cu θ. Ca urmare, o clasă de echivalenţă în raport cu θ este formată din una sau mai multe clase de echivalenţă în raport cu ρ. Grafic, această situaţie arată ca în Figura 1.4. Este justificat atunci a spune că ρ este mai *fină* decât θ.

Considerând acum mulţimea A/ρ, putem defini relaţia binară θ/ρ dată prin

$$[a]_\rho \, \theta/\rho \, [b]_\rho \quad \Leftrightarrow \quad a \, \theta \, b,$$

pentru orice $a, b \in A$.

Diferenţa între θ şi θ/ρ constă în aceea că θ acţionează pe mulţimea A, în timp ce θ/ρ acţionează pe A/ρ.

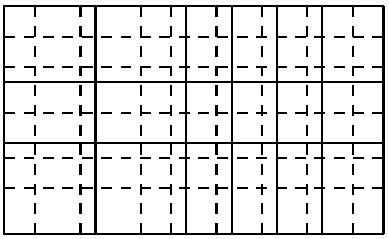

Figura 1.4: ρ este mai fină decât θ

Propoziţia 1.2.2.2. Fie A o mulţime şi ρ, θ, θ_1 şi θ_2 relaţii de echivalenţă pe A astfel încât $\rho \subseteq \theta \cap \theta_1 \cap \theta_2$. Atunci, au loc următoarele proprietăţi:

(1) $\theta/\rho \in E(A/\rho)$;

(2) orice relaţie de echivalenţă pe A/ρ este de forma θ'/ρ, unde $\theta' \in E(A)$ şi $\rho \subseteq \theta'$;

(3) $\rho/\rho = \iota_{A/\rho}$;

(4) $A^2/\rho = (A/\rho)^2$ (A^2 este relaţia binară $A \times A$ care, evident, include ρ);

(5) $\theta_1 \subset \theta_2$ dacă şi numai dacă $\theta_1/\rho \subset \theta_2/\rho$;

(6) $\theta_1 \neq \theta_2$ dacă şi numai dacă $\theta_1/\rho \neq \theta_2/\rho$.

Demonstraţie Vom demonstra (2), (3) şi (4), restul rămânând în grija cititorului.

(2) Fie ψ o relaţie de echivalenţă pe A/ρ. Definim θ' prin

$$a\,\theta'\,b \quad \Leftrightarrow \quad [a]_\rho\,\psi\,[b]_\rho,$$

pentru orice $a, b \in A$. Este uşor de văzut că θ' este relaţie de echivalenţă pe A.

Fie $a\,\rho\,b$. Atunci, $[a]_\rho\,\psi\,[b]_\rho$ deoarece ψ este reflexivă. Conform definiţiei relaţiei θ', urmează $a\,\theta'\,b$. Ca urmare, $\rho \subseteq \theta'$. Ne rămâne de arătat că $\psi = \theta'/\rho$. Aceasta urmează însă imediat de la definiţiile relaţiilor θ' şi θ'/ρ.

(3) Au loc relaţiile:

$$\begin{aligned} [a]_\rho\,\rho/\rho\,[b]_\rho &\Leftrightarrow a\,\rho\,b \\ &\Leftrightarrow [a]_\rho\,\iota_{A/\rho}\,[b]_\rho, \end{aligned}$$

pentru orice $a, b \in A$, ceea ce demonstrează egalitatea cerută.

(4) Au loc relaţiile:

$$\begin{aligned} [a]_\rho\,A^2/\rho\,[b]_\rho &\Leftrightarrow a\,A^2\,b \\ &\Leftrightarrow [a]_\rho\,(A/\rho)^2\,[b]_\rho, \end{aligned}$$

pentru orice $a, b \in A$, ceea ce demonstrează egalitatea cerută. □

Atragem atenţia asupra faptului că, în Propoziţia 1.2.2.2(3), ρ/ρ este $\iota_{A/\rho}$ şi nu ι_A/ρ.

1.2.3 Funcţii şi operaţii

Funcţiile, o clasă foarte importantă de relaţii, au fost introduse în Secţiunea 1.1.1. Astfel, s-a spus că o relaţie f este *funcţie* dacă satisface

$$(\forall a_1, b_1, a_2, b_2)((a_1, b_1) \in f \ \wedge \ (a_2, b_2) \in f \ \wedge \ a_1 = a_2 \ \Rightarrow \ b_1 = b_2).$$

Atunci când $Dom(f) = A$ şi $Cod(f) \subseteq B$ se mai spune că f este *funcţie de la A la B* sau că f este *funcţie definită pe A şi cu valori în B* şi se notează $f : A \to B$.

In informatică în special, este important de considerat şi *funcţii parţiale de la A la B*, adică funcţii f ce au proprietatea $Dom(f) \subseteq A$ şi $Cod(f) \subseteq B$. Astfel de funcţii mai sunt numite *funcţii parţial definite pe A şi cu valori în B*. Dacă $Dom(f) \subset A$, atunci se mai spune că f este *strict parţială pe A*. In contrast, dacă $Dom(f) = A$, atunci se mai spune că f este *totală pe A*. Relaţia vidă este funcţie parţială de la A la B. Aşa cum s-a spus în Secţiunea 1.1.1, ea este funcţie totală de la A la B doar dacă $A = \emptyset$.

Prin $(A \rightsquigarrow B)$ vom nota mulţimea tuturor funcţiilor parţiale de la A la B. Evident, $(A \to B) \subseteq (A \rightsquigarrow B)$.

O funcţie parţială $f : A \rightsquigarrow B$ are proprietatea că pentru orice $a \in A$ există cel mult un $b \in B$ astfel încât $f(a) = b$. Fie $a \in A$:

(1) dacă există $b \in B$ astfel încât $f(a) = b$, atunci se mai spune că b este *imaginea* lui a prin f şi că f *este definită* în a, şi se notează $f(a)\downarrow$;

(2) dacă nu există $b \in B$ astfel încât $f(a) = b$, atunci se mai spune că f *nu este definită* în a sau că f este *nedefinită* în a, şi se notează $f(a)\uparrow$.

Terminologia de funcţie "parţială" este justificată de (2).

Noţiunea de imagine a unui element printr-o funcţie parţială f poate fi extinsă la submulţimi în mod natural. De exemplu, dacă $C \subseteq A$ atunci vom numi *imaginea* mulţimii C prin f, mulţimea notată $f(C)$ şi definită prin

$$f(C) = \{b \in B | (\exists a \in C)(f(a) = b)\}$$

(a se vedea Definiţia 1.2.1.4(1)). Evident, $f(C) \subseteq B$ şi $f(C)$ poate fi chiar $\emptyset$ fără ca C să fie mulţimea vidă. Dacă $C = \emptyset$ atunci $f(C) = \emptyset$.

A specifica o funcţie parţială de la A la B înseamnă a preciza pentru fiecare element $a \in A$ dacă funcţia este definită sau nu în a; dacă ea este definită în a, atunci este necesară specificarea imaginii lui a prin respectiva funcţie.

Exemplul 1.2.3.1.

(1) Fie $A = \{1, 2, 3\}$ şi $B = \{a, b, c, d\}$. Relaţia f de la A la B dată prin

$$f(1) = a, \ f(2) = b \ \text{şi} \ f(3)\uparrow,$$

este funcţie strict parţială, iar relaţia g dată prin

$$g(1) = a, \ g(2) = b \ \text{şi} \ g(3) = c,$$

este funcţie (totală) de la A la B. Relaţia $h = \{(1, a), (1, b)\}$ nu este funcţie parţială.

(2) Funcţia parţială $f : A \rightsquigarrow B$ dată prin $f(a)\uparrow$, pentru orice $a \in A$, este numită *funcţia total nedefinită* de la A la B. Observăm că ea este de fapt relaţia vidă.

(3) Fie $A \subseteq B$. Funcţia $f : A \to B$ dată prin $f(a) = a$, pentru orice $a \in A$, este numită *funcţia incluziune*. Uneori ea se mai notează prin $f : A \hookrightarrow B$. In cazul $B = A$ funcţia incluziune se mai numeşte *funcţia identică* pe A şi se notează prin 1_A sau id_A sau chiar id. Observăm ca ea coincide cu relaţia ι_A.

(4) Relaţia completă de la A la B este funcţie parţială doar în cazul în care B conţine cel mult un element.

(5) Fie A o mulţime şi $B \subseteq A$. Funcţia $f_B : A \to \{0, 1\}$ dată prin

$$f_B(a) = \begin{cases} 1, & a \in B \\ 0, & a \in A - B, \end{cases}$$

pentru orice $a \in A$, este numită *funcţia caracteristică* a mulţimii B relativ la mulţimea A. Pentru $B = A$, f_B este funcţia *constantă* 1.

(6) Funcţiile de tipul $f : \{0, 1\}^n \to \{0, 1\}$, unde $n \geq 1$, sunt numite *funcţii booleene*.

(7) Fie $n \geq 1$, $A_1, \ldots, A_n$ mulţimi nevide şi $1 \leq i \leq n$. Funcţia

$$pr_i : A_1 \times \cdots \times A_n \to A_i$$

dată prin $pr_i(a_1, \ldots, a_n) = a_i$, pentru orice $(a_1, \ldots, a_n) \in A_1 \times \cdots \times A_n$, se numeşte funcţia de *i-proiecţie* asociată produsului $A_1 \times \cdots \times A_n$.

Dacă am considera cazul în care una dintre mulţimile $A_1, \ldots, A_n$ este mulţimea vidă, atunci funcţia de i-proiectie ar fi funcţia vidă, chiar şi atunci când A_i ar fi nevidă.

(8) Funcţiile de tipul $P : A \to \{0, 1\}$ sunt numite şi *predicate pe A*. Uzual, 0 este interpretat ca fiind valoarea de adevăr "fals", iar 1 ca fiind valoarea de adevăr "adevărat". Evident, în locul mulţimii $\{0, 1\}$ se poate alege orice altă mulţime cu două elemente.

(9) In informatică, funcţiile de tipul $f : A \to B$ date prin

$$f(a) = \begin{cases} e_1(a), & P(a) \\ e_2(a), & \text{altfel,} \end{cases}$$

pentru orice $a \in A$, unde $e_1(a)$ şi $e_2(a)$ sunt expresii ce depind de a iar P este un predicat pe A, sunt uzual notate prin

$$f(a) = \texttt{if}\ P(a)\ \texttt{then}\ e_1(a)\ \texttt{else}\ e_2(a)$$

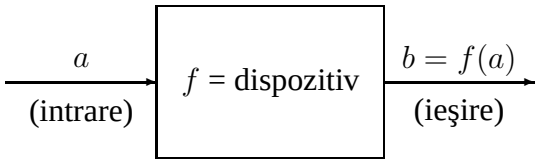

Figura 1.5: Reprezentare schematică a unei funcţii

O funcţie parţială de la A la B poate fi gândită ca un dispozitiv (Figura 1.5) care primind la intrare un element $a \in A$ funcţionează şi la ieşire emite elementul $b = f(a)$, dacă f este definită în a. In ipoteza în care f nu este definită în a, putem gândi că acest dispozitiv ori se opreşte fară a emite nimic la ieşire (de exemplu, dispozitivul se blochează) ori lucrează la nesfârşit (din punct de vedere teoretic). Un model practic, adesea considerat în literatură, este cel de automat ce poate oferi o ceaşcă de ceai sau o cafea la introducerea unei monede de un anumit tip, să spunem m_1 şi respectiv m_2 (presupunem, de exemplu, că ele au preţuri diferite). Dacă considerăm că automatul funcţionează perfect, atunci pentru m_1 el va oferi un ceai, pentru m_2, o cafea, iar pentru alte tipuri de monede se va bloca.

Funcţiile parţiale fiind relaţii, putem construi reuniunea, intersecţia şi diferenţa lor; egalitatea de funcţii parţiale este egalitate de relaţii. Evident, reuniunea a două funcţii nu este, în mod necesar, funcţie parţială. Intersecţia sau diferenţa a două funcţii este funcţie parţială.

De asemenea, putem vorbi de produs de funcţii (numit în acest caz *compunere*) şi de inversa unei funcţii. Pentru produsul a două funcţii $f : A \rightsquigarrow B$ şi $g : B \rightsquigarrow C$ vom folosi notaţia $g \circ f$ în loc de $f \circ g$ deoarece aceasta este într-un anumit sens în concordanţă cu notaţia $f(a) = b$ pentru $(a, b) \in f$:

$$(g \circ f)(a) = g(f(a)).$$

Unii autori consideră notaţia $(a)f = b$ pentru $(a, b) \in f$ şi, atunci, produsul de relaţii nu se mai schimbă notaţional:

$$(a)(f \circ g) = ((a)f)g.$$

Cum funcţiile sunt relaţii, notaţia f^n se deduce imediat de la acestea. Mai exact, dacă $f : A \rightsquigarrow A$ este o funcţie atunci:

- $f^0 = id_A$;
- $f^{n+1} = f^n \circ f = f \circ f^n$, pentru orice $n \geq 0$.

Propoziţia 1.2.3.1. Dacă $f : A \rightsquigarrow B$ şi $g : B \rightsquigarrow C$ sunt funcţii parţiale, atunci $g \circ f$ este funcţie parţială de la A la C.

Demonstraţie Fie $(a_1, b_1), (a_2, b_2) \in g \circ f$ astfel încât $a_1 = a_2$. Atunci, există c_1 şi c_2 cu proprietatea $(a_1, c_1), (a_2, c_2) \in f$ şi $(c_1, b_1), (c_2, b_2) \in g$. Egalitatea $a_1 = a_2$ combinată cu faptul că f este funcţie conduce la $c_1 = c_2$ care, la rândul ei, ne arată că $b_1 = b_2$ deoarece g este funcţie. Deci, $g \circ f$ este funcţie parţială. □

Remarcăm că $g \circ f$ nu este definită exact pentru acele elemente $a \in A$ pentru care ori $f(a)\uparrow$ ori $g(f(a))\uparrow$. Ca urmare, dacă $f = \emptyset$ sau $g = \emptyset$ atunci $g \circ f = \emptyset$, iar dacă f şi g sunt funcţii atunci $g \circ f$ este funcţie. Grafic, compunerea poate fi reprezentată ca în Figura 1.6.

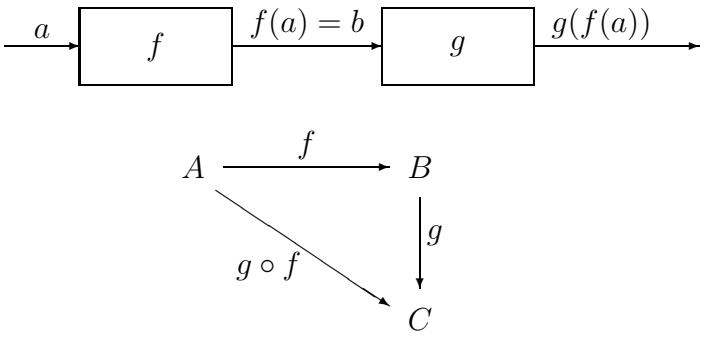

Figura 1.6: Reprezentare grafică a compunerii de funcţii

Relativ la inversa unei funcţii parţiale $f : A \rightsquigarrow B$ putem spune ca aceasta nu este neapărat funcţie parţială. De exemplu, dacă $f(a_1) = f(a_2) = b$ este o funcţie (totală) de la $A = \{a_1, a_2\}$ la $B = \{b\}$, atunci f^{-1} este relaţia $\{(b, a_1), (b, a_2)\}$ care nu este funcţie parţială.

Conceptele de injectivitate şi surjectivitate se extind şi la funcţii parţiale exact ca în Definiţia 1.1.1.6. Astfel, o funcţie parţială de la A la B este injectivă dacă are loc

$$(\forall a_1, b_1, a_2, b_2)((a_1, b_1) \in f \ \wedge \ (a_2, b_2) \in f \ \wedge \ b_1 = b_2 \ \Rightarrow \ a_1 = a_2),$$

şi este surjectivă dacă are loc

$$(\forall b)(b \in B \ \Rightarrow \ (\exists a)(a \in A \ \wedge \ f(a) = b)).$$

Ca urmare, funcţia parţială vidă de la A la B este injectivă; ea este surjectivă doar dacă $B = \emptyset$.

O funcţie parţială este bijectivă dacă este injectivă şi surjectivă. Spre deosebire de cazul funcţiilor totale bijective, inversa unei funcţii parţiale bijective este totală şi injectivă, dar nu în mod necesar surjectivă.

Funcţiile bijective de la o mulţime A la ea însăşi se mai numesc şi *permutări ale mulţimii* A. Când A este finită, $A = \{a_1, \ldots, a_n\}$, permutările $f : A \to A$ se mai notează prin

$$f = \begin{pmatrix} a_1 & \cdots & a_n \\ f(a_1) & \cdots & f(a_n) \end{pmatrix}$$

Propoziţia 1.2.3.2. Fie $f : A \rightsquigarrow B$ o funcţie parţială.

(1) Dacă f este injectivă, atunci f^{-1} este funcţie parţială injectivă.

(2) Dacă f este bijectivă, atunci f^{-1} este funcţie totală injectivă.

(3) Dacă f este totală şi bijectivă, atunci f^{-1} este totală şi bijectivă.

Demonstraţie (1) Să presupunem că f este injectivă. Fie $(b_1, a_1) \in f^{-1}$ şi $(b_2, a_2) \in f^{-1}$ astfel încât $b_1 = b_2$. Atunci, $(a_1, b_1), (a_2, b_2) \in f$ care, în baza faptului că f este injectivă conduce la $a_1 = a_2$. Deci f^{-1} este funcţie parţială.

Pentru a stabili injectivitatea funcţiei f^{-1} este suficient să observăm, în raţionamentul de mai sus, că dacă considerăm $a_1 = a_2$ atunci faptul că f este funcţie conduce la $b_1 = b_2$; adică f^{-1} este injecţie.

(2) urmează de la (1) cu remarca suplimentară că surjectivitatea funcţiei f asigură totalitatea funcţiei f^{-1}.

(3) Dacă f este totală şi bijectivă, atunci de la (2) obţinem că f^{-1} este totală şi injectivă. Totalitatea funcţiei f asigură surjectivitatea funcţiei f^{-1}. Deci, f^{-1} este totală şi bijectivă. □

Următoarea propoziţie, ce este uşor de verificat, prezntă câteva proprietăţi elementare asupra compunerii funcţiilor parţiale.

Propoziţia 1.2.3.3. Fie $f : A \rightsquigarrow B$ şi $g : B \rightsquigarrow C$ funcţii parţiale.

(1) Dacă f şi g sunt totale, atunci $g \circ f$ este totală.

(2) Dacă f şi g sunt injective, atunci $g \circ f$ este injectivă.

(3) Dacă f şi g sunt surjective, atunci $g \circ f$ este surjectivă.

(4) Dacă f şi g sunt bijective, atunci $g \circ f$ este bijectivă şi $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$.

Teorema 1.2.2.1, ce face legătură dintre $Part(A)$ şi $E(A)$ ne spune printre altele, în termeni de funcţie, că există o bijecţie ϕ între cele două mulţimi

$$\phi(S) = \rho_S,$$

pentru orice $S \in Part(A)$.

Definiţia 1.2.3.1. Fie $f : A \to B$ o funcţie.

(1) Se numeşte *invers la stânga* al funcţiei f orice funcţie $g : B \to A$ cu proprietatea $g \circ f = 1_A$.

(2) Se numeşte *invers la dreapta* al funcţiei f orice funcţie $g : B \to A$ cu proprietatea $f \circ g = 1_B$.

Propoziţia 1.2.3.4. Fie $f : A \to B$ o funcţie.

(1) Dacă f admite un invers la dreapta, atunci ea este surjecţie.

(2) Dacă $A \neq \emptyset$ şi f admite un invers la stânga, atunci f este injecţie.

Demonstraţie (1) Fie g un invers la dreapta al funcţiei f. Considerând $b \in B$ avem $g(b) \in A$ şi $f(g(b)) = b$. Deci, f este surjecţie.

(2) Fie g un invers la stânga al funcţiei f şi $a, b \in A$. Dacă presupunem că $f(a) = f(b)$, atunci $a = g(f(a)) = g(f(b)) = b$, ceea ce arată că f este injecţie. □

In cadrul Propoziţie 1.2.3.4(2) se consideră $A \neq \emptyset$ deoarece, pentru $A = \emptyset$, f este funcţia vidă care este injectivă.

Vom prezenta acum câteva rezultate de "descompunere" a funcţiilor.

Teorema 1.2.3.1. Fie $h : A \to B$ o funcţie. Atunci, există o mulţime C, o funcţie surjectivă $f : A \to C$ şi o funcţie injectivă $g : C \to B$ astfel încât $h = g \circ f$. In plus, pentru orice mulţime C', funcţie surjectivă $f' : A \to C'$ şi funcţie injectivă $g' : C' \to B$ astfel încât $h = g' \circ f'$, există o unică funcţie $d : C \to C'$ astfel încât $f' = d \circ f$ şi $g = g' \circ d$ (a se vedea diagrama din Figura 1.7).

Demonstraţie Fie $C = h(A)$, $f : A \to C$ dată prin $f(a) = h(a)$, pentru orice $a \in A$, şi $g : C \to B$ dată prin $g(c) = c$ pentru orice şi $c \in C$. Este clar că f este surjecţie, g este injecţie şi $h = g \circ f$.

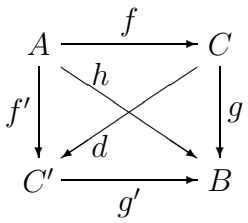

Figura 1.7: Descompunerea din Teorema 1.2.3.1

Fie C', f' şi g' ca în enunţul teoremei. Considerăm $d : C \to C'$ dată prin

$$d(c) = f'(a),$$

unde $f(a) = c$, pentru orice $c \in C$. Vom arăta că d satisface teorema. Intâi, verificăm că d este bine definită.

Deoarece f este surjecţie, pentru orice $c \in C$ există $a \in A$ astfel încât $f(a) = c$. Deci, d este definită pe C. Acum, pentru orice $a_1, a_2 \in A$ cu $f(a_1) = f(a_2)$ are loc $h(a_1) = h(a_2)$. Dar atunci, relaţia $h = g' \circ f'$ conduce la $g'(f'(a_1)) = g'(f'(a_2))$ de unde, pe baza faptului că g' este injecţie, deducem $f'(a_1) = f'(a_2)$. Ca urmare, d este bine definită pe C.

Conform definiţiei funcţiei d are loc $d \circ f = f'$. Fie acum $a \in A$ şi $c \in C$ cu $f(a) = c$. Atunci,

$$(g' \circ d)(c) = g'(d(c)) = g'(f'(a)) = h(a) = g(f(a)) = g(c),$$

ceea ce arată că $g = g' \circ d$.

Unicitatea funcţiei d decurge cu uşurinţă după cum urmează. Dacă presupunem că există o altă funcţie d' astfel încât $f' = d' \circ f$ şi $g = g' \circ d'$, atunci relaţia

$$g = g' \circ d = g' \circ d',$$

combinată cu faptul că g' este injecţie, conduce la $d = d'$. □

Teorema 1.2.3.1 nu impune nici o restricţie asupra funcţie h. Ca urmare, f poate fi şi funcţia vidă. In acest caz $A = \emptyset$, $C = \emptyset$, f este funcţia vidă de la $\emptyset$ la $\emptyset$ (deci, este surjectivă), g este funcţia vidă de la $\emptyset$ la B (deci, este injectivă), C' nu poate fi decât $\emptyset$ ca urmare a surjectivităţii funcţiei f', iar d nu este alta decât funcţia vidă de la $\emptyset$ la $\emptyset$.

Definiţia 1.2.3.2. Fie $f : A \to B$ o funcţie. Relaţia $Ker(f) \subseteq A \times A$ dată prin

$$Ker(f) = \{(a_1, a_2) | f(a_1) = f(a_2)\}$$

se numeşte *nucleul* funcţiei f.

Este uşor de verificat că pentru orice funcţie f de la A la B, $Ker(f)$ este relaţie de echivalenţă pe A. In plus,

Teorema 1.2.3.2. Fie $f : A \to B$ o funcţie. Atunci, există o bijecţie de la $A/Ker(f)$ la $f(A)$.

Demonstraţie Funcţia $h : A/Ker(f) \to f(A)$ dată prin $h([a]_{Ker(f)}) = f(a)$, pentru orice $a \in A$, este bijecţie. □

Corolarul 1.2.3.1. Orice funcţie $f : A \to B$ poate fi scrisă ca produs de 3 funcţii, $f = k \circ h \circ g$, unde $g : A \to A/Ker(f)$ este surjecţie, $h : A/Ker(f) \to f(A)$ este bijecţie şi $k : f(A) \to B$ este injecţie.

Demonstraţie Funcţia g este dată prin $g(a) = [a]_{Ker(f)}$, h este funcţia din Teorema 1.2.3.2, iar k este funcţia incluziune (a se vedea Figura 1.8). Aceste funcţii satisfac

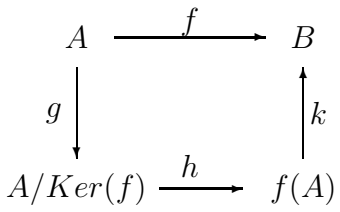

Figura 1.8: Descompunerea din Corolarul 1.2.3.1

corolarul. □

Vom prezenta în cele ce urmează câteva proprietăţi de bază referitoare la imaginea inversă a unei mulţimi printr-o funcţie (ce este caz particular a imaginii inverse a unei mulţimi printr-o relaţie – a se vedea Definiţia 1.2.1.4(2)).

Propoziţia 1.2.3.5. Fie $f : A \to B$ o funcţie şi $X, Y \subseteq B$. Atunci, au loc următoarele proprietăţi:

(1) $f^{-1}(X \cup Y) = f^{-1}(X) \cup f^{-1}(Y)$;

(2) $f^{-1}(X \cap Y) = f^{-1}(X) \cap f^{-1}(Y)$;

(3) $f^{-1}(X - Y) = f^{-1}(X) - f^{-1}(Y)$.

Demonstraţie (1) urmează de la echivalenţele

$$\begin{aligned} a \in f^{-1}(X \cup Y) &\Leftrightarrow f(a) \in X \cup Y \\ &\Leftrightarrow f(a) \in X \ \vee \ f(a) \in Y \\ &\Leftrightarrow a \in f^{-1}(X) \ \vee \ a \in f^{-1}(Y) \\ &\Leftrightarrow a \in f^{-1}(X) \cup f^{-1}(Y), \end{aligned}$$

pentru orice a.

(2) şi (3) se obţin în manieră similară. □

Propoziţia 1.2.3.6. Dacă $f : A \to B$ este o funcţie injectivă, atunci pentru orice $X, Y \subseteq A$ au loc următoarele proprietăţi:

(1) dacă $X \subset Y$, atunci $f(X) \subset f(Y)$;

(2) $f(X \cap Y) = f(X) \cap f(Y)$;

(3) $f(X - Y) = f(X) - f(Y)$.

Demonstraţie Se utilizează Propoziţia 1.2.1.4(2)(3)(4) şi proprietatea de injectivitate a funcţiei f. □

Propoziţia 1.2.3.7. Dacă $f : A \to B$ este o funcţie, $X \subseteq A$ şi $Y \subseteq f(A)$, atunci au loc următoarele proprietăţi:

(1) $f(f^{-1}(Y)) = Y$;

(2) $X \subseteq f^{-1}(f(X))$.

Demonstraţie (1) se obţine pe baza implicaţiilor

$$\begin{array}{rcl} b \in f(f^{-1}(Y)) & \Rightarrow & \exists a \in f^{-1}(Y) : \ (a,b) \in f \\ & \Rightarrow & \exists c \in Y : \ (a,c) \in f \ \wedge \ (a,b) \in f \\ & \Rightarrow & c = b \\ & \Rightarrow & b \in Y, \end{array}$$

pentru orice b, şi a Propoziţiei 1.2.1.4(6).

(2) urmează direct de la Propoziţia 1.2.1.4(6). □

Operaţiile (*parţiale*) sunt funcţii (parţiale). De exemplu, orice funcţie f de la $A \times B$ la C poate fi gândită ca o operaţie ce acţionează pe elementele a două mulţimi, A şi B, şi produce rezultate în C. Dacă operaţia este parţială atunci pot există perechi (a, b) asupra cărora operaţia să nu acţioneze. De exemplu, într-un corp, inversul multiplicativ este definit pentru elementele diferite de 0 (unitatea aditivă).

Operaţiile (parţiale) de tipul $f : A^n \to A$, unde $n \geq 0$, se numesc *operaţii* (*parţiale*) *n-are pe A*. Numărul natural n este numit *aritatea* sau *tipul* operaţie. In cazul în care $n = 0$ reamintim că A^0 este $\{\emptyset\}$ şi, ca urmare, o operaţie 0-ară (numită şi *operaţie nulară* sau *constantă* [16]) va asocia un element din A mulţimii vide. Această asociere este constantă în sensul că ea nu depinde de elementele mulţimii A. Atunci, a specifica o operaţie nulară revine la a fixa un element din A. Justificaţi de aceasta vom folosi adesea terminologia "fie $a \in A$ o operaţie nulară pe A" sau "fie a o constantă din A".

Operaţiile parţiale fiind definite prin intermediul funcţiilor parţiale, iar acestea fiind relaţii, înseamnă că operaţiile sunt relaţii. Mai exact, o operaţie parţială n-ară peste A este o relaţie $(n+1)$-ară pe A.

Operaţiile (parţiale) de tipul $f : A \to B$ sau $f : A_1 \times A_2 \to B$ pot fi specificate prin aşa numitele *tabele Cayley*. De exemplu, operaţia unară f pe $A = \{a, b\}$ dată prin $f(a) = b$ şi $f(b) = a$ poate fi specificată prin tabelul

[16] Terminologia este cea de *constantă* şi nu de *operaţie constantă*.

f	a	b
	b	a

iar operaţia binară g dată prin $g(a,a) = g(b,a) = g(b,b) = a$ şi $g(a,b) = b$ poate fi specificată prin tabelul

g	a	b
a	a	b
b	a	a

Pentru operaţii (parţiale) binare se foloseşte adesea notaţia *infix*, semnul operaţiei fiind între elemente (această notaţie am folosit-o şi la relaţii binare). De exemplu, dacă $+$ sau $\cdot$ sunt operaţii, notaţia infix asupra elementelor a şi b va fi $a + b$ şi, respectiv, $a \cdot b$. Pentru operaţii unare se foloseşte adesea şi notaţia *exponent*. De exemplu, B' poate reprezenta complementara lui B (semnul operaţiei fiind $'$).

Până acum au fost considerate un număr de operaţii pe mulţimi, relaţii şi funcţii (reuniune, intersecţie, diferenţă, produs cartezian, produs de relaţii sau funcţii, inversă etc.), precum şi unele proprietăţi ale acestora adnotate în dreapta prin cuvinte de tipul “asociativitate”, “comutativitate” etc.

Definiţia 1.2.3.3. Fie f o operaţie binară pe o mulţime A.

(1) Spunem că f este *asociativă* dacă are loc

$$f(f(a,b),a) = f(a,f(b,c)),$$

pentru orice $a, b, c \in A$.

(2) Spunem că f este *comutativă* dacă are loc

$$f(a,b) = f(b,a),$$

pentru orice $a, b \in A$.

(3) Spunem că f este *idempotentă* dacă are loc

$$f(a,a) = a,$$

pentru orice $a \in A$.

In cazul operaţiilor binare comutative, tabelul Cayley poate fi redus la jumătate considerând doar valorile de deasupra (sau de dedesubtul) diagonalei, incluzând şi diagonala. Dacă operaţia este şi idempotentă, atunci poate fi eliminată şi diagonala.

Definiţia 1.2.3.4. Fie f şi g operaţii binare pe o mulţime A.

(1) Spunem că f este *distributivă la stânga faţă de* g dacă are loc

$$f(a,g(b,c)) = g(f(a,b),f(a,c)),$$

pentru orice $a, b, c \in A$.

(2) Spunem că f este *distributivă la dreapta faţă de* g dacă are loc

$$f(g(b,c),a) = g(f(b,a),f(c,a)),$$

pentru orice $a, b, c \in A$.

1.2.4 Familii indexate de mulţimi

In această secţiune vom considera noţiunea de familie indexată de mulţimi şi vom generaliza reuniunea, intersecţia şi produsul cartezian la astfel de familii.

Definiţia 1.2.4.1. Fie A şi I două mulţimi. O *familie de elemente peste A indexată prin I*, sau *familie I-indexată peste A*, este o funcţie $f : I \to A$.

Uzual, dacă f este o familie I-indexată de elemente peste A atunci vom utiliza şi notaţia $(a_i|i \in I)$, unde $f(i) = a_i$ pentru orice $i \in I$. Atunci când I se subînţelege din context vom simplifica această notaţie la (a_i), iar dacă $I \neq \emptyset$ atunci vom spune că familia este *nevidă*. Mulţimea I este numită *mulţime de indexi* iar elementele ei, *indexi*. In cazul în care elementele familiei sunt mulţimi ($f(i)$ este mulţime pentru orice $i \in I$), vom vorbi de *familii I-indexate de mulţimi*, iar dacă mulţimile sunt disjuncte două câte două, vom spune că avem de a face cu o *familie indexată disjunctă*. Atunci când I şi A sunt subînţelese din context, sau nu este necesar a le specifica, vom simplifica terminologia renunţând la ele.

Orice familie $\mathcal{A}$ de mulţimi poate fi privită ca o familie indexată de mulţimi (peste $\mathcal{A}$) considerând $I = \mathcal{A}$ şi $f(A) = A$, pentru orice $A \in I = \mathcal{A}$. Intr-un astfel de caz vom mai scrie $\mathcal{A} = (A|A \in \mathcal{A})$. Insă, familiile indexate de mulţimi sunt oarecum mai generale decât familiile de mulţimi prin aceea că:

- referirea la o mulţime se face printr-un index;
- pentru indexi diferiţi este posibil ca mulţimile referite să fie egale (din acest punct de vedere putem gândi familiile indexate de mulţimi ca fiind colecţii de mulţimi în care anumite mulţimi pot apare de mai multe ori).

Definiţia 1.2.4.2. Fie $(A_i|i \in I)$ o familie indexată de mulţimi peste A.

(1) *Reuniunea* familiei $(A_i|i \in I)$, notată $\bigcup(A_i|i \in I)$ sau $\bigcup)i \in I A_i$, este mulţimea:
$$\bigcup(A_i|i \in I) = \{a \in A|(\exists i \in I)(a \in A_i)\}.$$

(2) Dacă $I \neq \emptyset$, atunci *intersecţia* familiei $(A_i|i \in I)$, notată $\bigcap(A_i|i \in I)$ sau $\bigcap i \in I A_i$, este mulţimea:
$$\bigcap(A_i|i \in I) = \{a \in A|(\forall i \in I)(a \in A_i)\}.$$

(3) *Produsul* (*direct*) al familiei $(A_i|i \in I)$, notat $\prod(A_i|i \in I)$ sau $\prod_{i \in I} A_i$, este definit ca fiind mulţimea tuturor funcţiilor $f : I \to \bigcup(A_i|i \in I)$ cu proprietatea $f(i) \in A_i$, pentru orice $i \in I$.

In cazul $I = \emptyset$ vom considera $\bigcap(A_i|i \in I) = A$. Observăm că dacă $I = \emptyset$, atunci $\bigcup(A_i|i \in I) = \emptyset$ şi $\prod(A_i|i \in I) = \{\emptyset\}$, iar dacă $I \neq \emptyset$ şi există $i \in I$ astfel încât $A_i = \emptyset$, atunci $\prod(A_i|i \in I) = \emptyset$. De asemenea, dacă $A_i = A$ pentru orice $i \in I$, atunci $\prod(A_i|i \in I) = A^I$, adică produsul familiei este mulţimea tuturor funcţiilor de la I la A.

Existenţa reuniunii şi intersecţiei rezultă cu uşurinţă pe baza Axiomei separării. Pentru produs se utilizează Axioma separării asupra mulţimii $\mathcal{P}(I \times \bigcup(A_i|i \in I))$.

Prezentăm în continuare câteva proprietăţi de bază ale reuniunii, intersecţiei şi produsului de familii indexate de mulţimi.

Propoziţia 1.2.4.1. Fie $(A_i|i \in I)$ şi $(B_i|i \in I)$ două familii indexate de mulţimi şi A o mulţime. Atunci, au loc următoarele proprietăţi:

(1) $\bigcap_{i\in I} A_i \subseteq A_j \subseteq \bigcup_{i\in I} A_i$, pentru orice $j \in I$;

(2) $\bigcap_{i\in I}(A_i \cap B_i) = \bigcap_{i\in I} A_i \cap \bigcap_{i\in I} B_i$;

(3) $\bigcup_{i\in I}(A_i \cup B_i) = \bigcup_{i\in I} A_i \cup \bigcup_{i\in I} B_i$;

(4) $\bigcap_{i\in I} A_i \cup \bigcap_{i\in I} B_i = \bigcap_{i,j\in I}(A_i \cup B_j) \subseteq \bigcap_{i\in I}(A_i \cup B_i)$;

(5) $\bigcup_{i\in I}(A_i \cap B_i) \subseteq \bigcup_{i,j\in I}(A_i \cap B_j) = \bigcup_{i\in I} A_i \cap \bigcup_{i\in I} B_i$;

(6) $A - \bigcap_{i\in I} A_i = \bigcup_{i\in I}(A - A_i)$;

(7) $A - \bigcup_{i\in I} A_i = \bigcap_{i\in I}(A - A_i)$;

(8) $\bigcap_{i\in I}(A \cup A_i) = A \cup \bigcap_{i\in I} A_i$;

(9) $\bigcup_{i\in I}(A \cap A_i) = A \cap \bigcup_{i\in I} A_i$;

(10) dacă $A \subseteq A_i$, pentru orice $i \in I$, atunci $A \subseteq \bigcap_{i\in I} A_i$;

(11) dacă $A_i \subseteq A$, pentru orice $i \in I$, atunci $\bigcup_{i\in I} A_i \subseteq A$.

Demonstraţie Vom demonstra ca exemplu doar (6), care se obţine pe baza echivalenţelor:

$$\begin{aligned} a \in A - \textstyle\bigcap_{i\in I} A_i &\Leftrightarrow a \in A \wedge a \notin \textstyle\bigcap_{i\in I} A_i \\ &\Leftrightarrow a \in A \wedge (\exists i \in I : a \notin A_i) \\ &\Leftrightarrow \exists i \in I : a \in A - A_i \\ &\Leftrightarrow a \in \textstyle\bigcup_{i\in I}(A - A_i), \end{aligned}$$

pentru orice a. □

Fie $(A_i|i \in I)$ o familie indexată de mulţimi. Direct de la definiţie urmează că $\bigcup_{i\in I} A_i$ este cea mai mică mulţime, în sensul incluziunii, ce include toate mulţimile A_i, iar $\bigcap_{i\in I} A_i$ este cea mai mare mulţime, în sensul incluziunii, ce este inclusă în fiecare mulţime A_i. Este util de menţionat acest rezultat sub forma unei leme.

Lema 1.2.4.1. Fie $(A_i|i \in I)$ o familie indexată de mulţimi. Atunci, $\bigcup_{i\in I} A_i$ este unica mulţime A ce satisface proprietăţile:

(i) $A_i \subseteq A$, pentru orice $i \in I$;

(ii) dacă B este o mulţime ce satisface $A_i \subseteq B$, pentru orice $i \in I$, atunci $A \subseteq B$.

Similar, $\bigcap_{i\in I} A_i$ este unica mulţime A ce satisface proprietăţile:

(i') $A \subseteq A_i$, pentru orice $i \in I$;

(ii') dacă B este o mulţime ce satisface $B \subseteq A_i$, pentru orice $i \in I$, atunci $B \subseteq A$.

Demonstraţie De la definiţii şi Propoziţia 1.2.4.1(1)(10)(11). □

Propoziţia 1.2.4.2. Fie $(B_j|j \in J)$ o familie indexată de mulţimi şi I reuniunea acestei familii. Atunci, pentru orice familie indexată de mulţimi $(A_i|i \in I)$, au loc următoarele proprietăţi:

(1) $\bigcup_{i\in I} A_i = \bigcup_{j\in J}(\bigcup_{i\in B_j} A_i)$;

(2) $\bigcap_{i\in I} A_i = \bigcap_{j\in J}(\bigcap_{i\in B_j} A_i)$.

Demonstraţie Vom demonstra (1) (relaţia (2) se obţine în manieră similară). Notăm $S_j = \bigcup_{i\in B_j} A_i$, pentru orice $j \in J$, şi $S = \bigcup_{i\in I} A_i$. Avem de arătat că are loc $S = \bigcup_{j\in j} S_j$.

S este cea mai mică mulţime ce include mulţimile A_i, $i \in I$. De asemenea, pentru orice $i \in I$ există $j \in J$ astfel încât $A_i \subseteq S_j$, ceea ce conduce la faptul că $\bigcup_{j\in J} S_j$ include toate mulţimile A_i, $i \in I$. Minimalitatea mulţimii S (Lema 1.2.4.1) conduce atunci la $S \subseteq \bigcup_{j\in J} S_j$.

Reciproc, $\bigcup_{j\in J} S_j$ este cea mai mică mulţime ce include mulţimile S_j, pentru orice $j \in J$. Însă, pentru orice $j \in J$, $S_j \subseteq S$. Minimalitatea mulţimii $\bigcup_{j\in J} S_j$ (Lema 1.2.4.1) conduce la $\bigcup_{j\in J} S_j \subseteq S$, care combinată cu incluziunea anterioară furnizează (1). □

Propoziţia 1.2.4.3. Fie $(A_i|i \in I)$ o familie indexată de mulţimi şi f o permutare a mulţimii I. Atunci, au loc următoarele proprietăţi:

(1) $\bigcup_{i\in I} A_i = \bigcup_{i\in I} A_{f(i)}$;

(2) $\bigcap_{i\in I} A_i = \bigcap_{i\in I} A_{f(i)}$.

Demonstraţie Ca şi în cazul Propoziţiei 1.2.4.2, vom demonstra doar (1). Notăm $S = \bigcup_{i\in I} A_{f(i)}$. Vom arăta că S este cea mai mică mulţime ce include mulţimile A_i, $i \in I$. Pentru orice $i \in I$, S include A_j, unde $j = f^{-1}(i)$ şi, deci, va include şi $A_{f(j)}$ (care este de fapt A_i). Dacă există o altă mulţime B care include toate mulţimile A_i, atunci ea include şi S conform definiţiei acesteia şi a noţiunii de permutare. Lema 1.2.4.1 conduce atunci la (1). □

Propoziţia 1.2.4.4. Fie $(B_j|j \in J)$ o familie indexată de mulţimi, I reuniunea ei şi

$$K = \{C \in \mathcal{P}(I)|(\forall j \in J)(C \cap B_j \neq \emptyset)\}.$$

Atunci, pentru orice familie indexată de mulţimi $(A_i|i \in I)$, au loc următoarele proprietăţi:

(1) $\bigcap_{j\in J}(\bigcup_{i\in B_j} A_i) = \bigcup_{C\in K}(\bigcap_{i\in C} A_i)$;

(2) $\bigcup_{j\in J}(\bigcap_{i\in B_j} A_i) = \bigcap_{C\in K}(\bigcup_{i\in C} A_i)$.

Demonstraţie Demonstrăm doar (1), (2) obţinându-se în manieră similară. Fie $C \in K$ şi $j \in J$. Conform definiţiei mulţimii K avem $C \cap B_j \neq \emptyset$. Pentru orice $i \in C \cap B_j$, Propoziţia 1.2.4.1(1) conduce la

$$\bigcap_{i\in C} A_i \subseteq A_i \subseteq \bigcup_{i\in B_j} A_i.$$

Cum aceaste incluziuni este safisfăcută pentru orice $j \in J$, utilizând Propoziţia 1.2.4.1(10) obţinem

$$\bigcap_{i\in C} A_i \subseteq \bigcap_{j\in J}(\bigcup_{i\in B_j} A_i)$$

şi, apoi, de la punctul (11) al aceleiaşi propoziţii deducem

$$\bigcup_{C\in K}(\bigcap_{i\in C} A_i) \subseteq \bigcap_{j\in J}(\bigcup_{i\in B_j} A_i).$$

Pentru a demonstra incluziunea în sens invers considerăm un element arbitrar a din $\bigcap_{j\in J}(\bigcup_{i\in B_j} A_i)$ şi fie $C = \{i \in I | a \in A_i\}$. Pentru orice $j \in J$, a este în $\bigcup_{i\in B_j} A_i$ şi, deci, există $i \in B_j$ astfel încât $a \in A_i$. Deci, $i \in C$, ceea ce arată că $C \cap B_j \neq \emptyset$ şi, aşadar, $C \in K$. Urmează acum că $a \in A_i$, pentru orice $i \in C$, adică $a \in \bigcap_{i\in C} A_i$. Am obţinut astfel că $a \in \bigcup_{C\in K}(\bigcap_{i\in C} A_i)$, şi (1) este demonstrată. □

Ne îndreptăm acum atenţia asupra unor proprietăţi ale imaginilor şi imaginilor inverse ale reuniunilor şi intersecţiilor de familii indexate de mulţimi.

Propoziţia 1.2.4.5. Fie $(A_i | i \in I)$ o familie indexată de mulţimi şi f o funcţie. Atunci, au loc următoarele proprietăţi:

(1) $f(\bigcup_{i\in I} A_i) = \bigcup_{i\in I} f(A_i)$;

(2) $f(\bigcap_{i\in I} A_i) \subseteq \bigcap_{i\in I} f(A_i)$. Dacă f este injectivă atunci relaţia are loc prin egalitate.

Demonstraţie Au loc echivalenţele

$$\begin{aligned} b \in f(\textstyle\bigcup_{i\in I} A_i) &\Leftrightarrow \exists a \in \bigcup_{i\in I} A_i : \ b = f(a) \\ &\Leftrightarrow \exists i \in I, \ \exists a \in A_i : \ b = f(a) \\ &\Leftrightarrow \exists i \in I : \ b \in f(A_i) \\ &\Leftrightarrow b \in \textstyle\bigcup_{i\in I} f(A_i), \end{aligned}$$

pentru orice b, ceea ce demonstrează (1). Similar se arată şi (2). □

Demonstraţia următoarei propoziţii este lăsată în grija cititorului.

Propoziţia 1.2.4.6. Fie $(A_i | i \in I)$ o familie indexată de mulţimi şi f o funcţie. Atunci, au loc următoarele proprietăţi:

(1) $f^{-1}(\bigcup_{i\in I} A_i) = \bigcup_{i\in I} f^{-1}(A_i)$;

(2) $f^{-1}(\bigcap_{i\in I} A_i) = \bigcap_{i\in I} f^{-1}(A_i)$.

Definiţia 1.2.4.3. Fie $(A_i|i \in I)$ o familie indexată nevidă de mulţimi şi $i \in I$. *Funcţia de i-proiecţie* asociată familiei $(A_i|i \in I)$ este funcţia

$$pr_i : \prod(A_i|i \in I) \to A_i$$

dată prin $pr_i(f) = f(i)$, pentru orice $f \in \prod(A_i|i \in I)$.

Dacă există $j \in I$ astfel încât $A_j = \emptyset$, atunci $\prod(A_i|i \in I) = \emptyset$ şi, deci, pr_i este funcţia vidă, pentru orice $i \in I$. Altfel, pr_i sunt funcţii surjective.

Teorema 1.2.4.1. (Proprietatea de universalitate a produsului)
Fie $(A_i|i \in I)$ o familie indexată nevidă de mulţimi, A o mulţime nevidă şi g_i funcţii de la A la A_i, pentru orice $i \in I$. Atunci, există o unică funcţie $f : A \to \prod(A_i|i \in I)$ astfel încât $g_i = pr_i \circ f$.

Demonstraţie Pentru orice $a \in A$ fie $x_a \in \prod(A_i|i \in I)$ dată prin $x_a(i) = g_i(a)$, pentru orice $i \in I$. Definim funcţia f prin $f(a) = x_a$, pentru orice $a \in A$, şi constatăm cu uşurinţă că ea satisface proprietatea $g_i = pr_i \circ f$, pentru orice $i \in I$.

Unicitatea funcţiei f se obţine astfel. Dacă ar exista o altă funcţie f' cu proprietatea $g_i = pr_i \circ f'$, pentru orice $i \in I$, atunci ar urma $pr_i \circ f = pr_i \circ f'$, adică $pr_i(f(a)) = pr_i(f'(a))$, pentru orice $a \in A$ şi $i \in I$. Dar aceasta nu înseamnă altceva decât că are loc $f = f'$. □

1.2.5 Relaţii de ordine

O altă clasă importantă de relaţii binare, pe lângă cea a relaţiilor de echivalenţă şi a funcţiilor, este cea a relaţiilor de ordine.

Definiţia 1.2.5.1. Fie ρ o relaţie binară pe o mulţime A.

(1) ρ este numită *relaţie de pre-ordine* sau *quasi-ordine pe* A dacă ρ este reflexivă şi tranzitivă pe A şi, în acest caz, cuplul $(A;\rho)$ se numeşte *mulţime pre-ordonată* sau *quasi-ordonată*.

(2) ρ este numită *relaţie de ordine parţială pe* A dacă ρ este reflexivă, antisimetrică şi tranzitivă pe A şi, în acest caz, cuplul $(A;\rho)$ se numeşte *mulţime parţial ordonată* (abreviat, mpo).

(3) ρ este numită *relaţie de ordine parţială strictă pe* A dacă ρ este ireflexivă şi tranzitivă pe A şi, în acest caz, cuplul $(A;\rho)$ se numeşte *mulţime parţial ordonată strict* (abreviat, mpos).

(4) ρ este numită *relaţie de ordine totală pe* A dacă ρ este ordine parţială şi conexă pe A şi, în acest caz, cuplul $(A;\rho)$ se numeşte *mulţime total ordonată* (abreviat, mto) sau *mulţime liniar ordonată* sau *lanţ*.

(5) ρ este numită *relaţie de ordine totală strictă pe* A dacă ρ este ordine parţială strictă şi conexă pe A şi, în acest caz, cuplul $(A;\rho)$ se numeşte *mulţime total ordonată strict* (abreviat, mtos).

Nu este dificil de văzut că ireflexivitatea şi tranzitivitatea implică antisimetria, iar asimetria este echivalentă cu antisimetria plus ireflexivitatea. Ca urmare, putem spune că relaţiile de ordine parţială strictă sunt relaţii asimetrice şi tranzitive. Relaţiile definite mai sus vor fi referite în general ca fiind *relaţii de ordine*.

Relaţia vidă satisface oricare dintre proprietăţile din Definiţia 1.2.5.1. Ca urmare, perechea $(\emptyset;\emptyset)$ este mpo, mto etc.

Definiţia 1.2.5.2.

(1) Se numeşte *mulţime dirijată* orice cuplu $(A;\rho)$ format dintr-o mulţime nevidă A şi o relaţie ρ dirijată pe A.

(2) Se numeşte *mulţime filtrată* orice cuplu $(A;\rho)$ format dintr-o mulţime nevidă A şi o relaţie ρ filtrată pe A.

Este uşor de văzut că orice mulţime total ordonată nevidă este atât dirijată cât şi filtrată.

Frecvent, conceptul de mulţime dirijată (filtrată) este cuplat cu unul din conceptele din Definiţia 1.2.5.1(1)(2)(3). Astfel, o *mulţime parţial ordonată dirijată* este o mulţime parţial ordonată ce este şi dirijată.

Reprezentarea grafică a perechilor $(A;\rho)$, unde ρ poate fi o relaţie binară arbitrară pe A, se face ca în Secţiunea 1.2.1 cu deosebirea că nodurile grafului nu sunt date de $Dom(\rho)\cup Cod(\rho)$ ci de A. Ca urmare, graful asociat relaţiei poate conţine *noduri izolate* (noduri ce nu sunt extremităţi ale nici unui arc). Evident, în cazul în care ρ este reflexivă şi/sau tranzitivă se poate recurge la simplificarea reprezentării grafice aşa cum a fost menţionat în Secţiunea 1.2.1.

Exemplul 1.2.5.1. Fie A o mulţime nevidă. Relaţia de incluziune pe A, $\subseteq_A$, este reflexivă, antisimetrică şi tranzitivă. Ca urmare, $(A;\subseteq_A)$ este mpo.

Structurile $(A;\rho)$ din Definiţia 1.2.5.1 se generalizează la *structuri relaţionale* ce vor fi prezentate în secţiunea 3.1.

Capitolul 2

Inchideri

Inchiderea unei mulţimi la o familie de constructori este una din operaţiile de bază în matematică şi informatică. Dacă din punct de vedere pur matematic suntem adesea interesaţi doar în existenţa închiderii (ca fiind intersecţia tuturor mulţimilor ce includ mulţimea în cauză şi sunt închise la familia respectivă de constructori), din punct de vedere informatic lucrurile stau puţin altfel. Ne interesează nu numai existenţa închiderii dar şi o manieră constructivă de obţinere a obiectelor închiderii. Mai suntem interesaţi şi într-o ordinea de aplicare a constructorilor, atunci când este posibil de stabilit o astfel de ordine, şi ne mai interesează şi metode de demonstraţie de proprietăţi ale obiectelor închiderii. Mai sunt şi alte probleme de care suntem interesaţi, cum ar fi unicitatea construcţiei obiectelor închiderii. Toate acestea vor fi discutate în această secţiune, urmând în principal [182].

2.1 Inchideri. Inducţie structurală

Incepem secţiunea prin a stabili câteva convenţii ce vor fi utilizate ori de câte ori va fi vorba despre închideri.

In general, vom considera relaţii r de la V^n la V, unde V este o mulţime şi n este un număr natural. Reamintim că pentru $n = 0$ avem, prin convenţie matematică, $V^0 = \{\emptyset\}$. Ca urmare, a specifica o relaţie de la V^0 la V revine la a specifica o submulţime a mulţimii V. In cazul $n = 1$, perechile $((a_1), a) \in r$ vor fi notate simplificat prin (a_1, a).

Vom fi interesaţi în a realiza închideri la mulţimi $\mathcal{R}$ de relaţii. Din raţiuni tehnice vom considera că fiecare relaţie $r \in \mathcal{R}$ este de la V^{n_r} la V, unde n_r este un număr natural. Pentru $A \subseteq V$ şi $r \in \mathcal{R}$ vom nota prin $r(A^{n_r})$ mulţimea

$$r(A^{n_r}) = \begin{cases} \{a \in V | (a) \in r\}, & \text{dacă } n_r = 0 \\ \{a \in V | (\exists a_1, \ldots, a_{n_r} \in A)(((a_1, \ldots, a_{n_r}), a) \in r)\}, & \text{altfel,} \end{cases}$$

iar prin $\mathcal{R}(A)$, mulţimea

$$\mathcal{R}(A) = \bigcup_{r \in \mathcal{R}} r(A^{n_r}).$$

Definiţia 2.1.1. Fie A o mulţime şi $\mathcal{R}$ o mulţime de relaţii.

(1) Spunem că A este *închisă la* $r \in \mathcal{R}$ dacă $r(A^{n_r}) \subseteq A$.

(2) Spunem că o mulţime B este *închiderea mulţimii* A *la mulţimea de relaţii* $\mathcal{R}$, sau că este $\mathcal{R}$*-închiderea mulţimii* A, şi notăm $B = \mathcal{R}[\![A]\!]$, dacă B este cea mai mică mulţime, în sensul incluziunii, ce include A şi este închisă la fiecare relaţie $r \in \mathcal{R}$.

In cazul în care $\mathcal{R}$ este formată dintr-un singur element r, vom mai scrie $r[\![A]\!]$ în loc de $\mathcal{R}[\![A]\!]$.

Următoarea teoremă ne va arăta că închiderea unei mulţimi la o mulţime de relaţii există întotdeauna. In plus, demonstraţia acestei teoreme ne va furniza şi o metodă de determinare a închiderii.

Teorema 2.1.1. Fie A o mulţime şi $\mathcal{R}$ o mulţime de relaţii pe V. Atunci, există o unică mulţime B ce este închiderea mulţimii A la $\mathcal{R}$.

Demonstraţie Fie şirul de mulţimi:

– $B_0 = A$;

– $B_{m+1} = B_m \cup \mathcal{R}(B_m)$, pentru orice $m \in \mathbf{N}$.

Este clar că au loc incluziunile

$$A = B_0 \subseteq B_1 \subseteq \ldots \subseteq B_m \subseteq \ldots .$$

Fie $B = \bigcup_{m \geq 0} B_m$. Arătăm că B satisface teorema. B include A şi

$$\begin{array}{rcl} \mathcal{R}(B) & = & \mathcal{R}(\bigcup_{m \geq 0} B_m) \\ & = & \bigcup_{m \geq 0} \mathcal{R}(B_m) \\ & \subseteq & \bigcup_{m \geq 0} B_m \\ & = & B, \end{array}$$

ceea ce arată că B este închisă la $\mathcal{R}$.

Fie B' o mulţime ce include A şi este închisă la $\mathcal{R}$. Prin inducţie matematică se arată că, pentru orice $m \geq 0$, are loc $B_m \subseteq B'$ şi, deci,

$$B = \bigcup_{m \geq 0} B_m \subseteq B'.$$

Ca urmare, B este cea mai mică mulţime ce include A şi este închisă la $\mathcal{R}$, fiind unica mulţime cu aceste proprietăţi. □

Observaţia 2.1.1.

(1) Demonstraţia Teoremei 2.1.1 ne arată că închiderea mulţimii A la mulţimea $\mathcal{R}$ de relaţii este limita (supremum) şirului de mulţimi:

– $B_0 = A$;

– $B_{m+1} = B_m \cup \mathcal{R}(B_m)$, pentru orice $m \geq 0$,

adică, $B = \bigcup_{m \geq 0} B_m$.

(2) Dacă considerăm mulţimea

$$\mathcal{A} = \{X \subseteq V | A \subseteq X \ \wedge \ X \text{ este închisă la } \mathcal{R}\},$$

atunci constatăm că aceasta este nevidă ($V \in \mathcal{A}$) şi $\bigcap \mathcal{A}$ este închiderea mulţimii A la $\mathcal{R}$. Ca urmare, putem spune că închiderea mulţimii A la $\mathcal{R}$ este intersecţia tuturor submulţimilor X ale mulţimii V, ce include A şi sunt închise la $\mathcal{R}$.

Următoarea teoremă, cunoscută ca fiind *Principiul inducţiei structurale*, poate fi gândită ca echivalenta Principiului inducţie matematice pentru mulţimi definite prin închidere. Ea este un instrument matematic foarte important prin care se pot demonstra anumite proprietăţi ale elementelor închiderii unei mulţimi.

Teorema 2.1.2. (Principiul inducţiei structurale)
Fie $B = \mathcal{R} \llbracket A \rrbracket$ şi P o proprietate astfel încât:

(i) $P(a)$, pentru orice $a \in A$;

(ii) $(P(a_1) \wedge \cdots \wedge P(a_{n_r}) \ \Rightarrow \ P(a))$, pentru orice $r \in \mathcal{R}$ şi $a_1, \ldots, a_{n_r}, a \in B$ cu $((a_1, \ldots, a_{n_r}), a) \in r$.

Atunci, P este satisfăcută de toate elementele $b \in B$.

Demonstraţie Fie $B' = \{b \in B | P(b)\}$. Atunci, $A \subseteq B'$ (de la (i)) şi B' este închisă la $\mathcal{R}$ (de la (ii)). Ca urmare, $B \subseteq B'$. □

Observaţia 2.1.2.

(1) Se poate da o demonstraţie a Teoremei 2.1.2 pornind de la remarca că B este supremul şirului de mulţimi $\langle B_m | m \geq 0 \rangle$ descris în Observaţia 2.1.1. Astfel, printr-o simplă inducţie matematică după $m \geq 0$ se arată că proprietatea P este satisfăcută de orice element $b \in B_m$. Ca urmare, P va fi satisfăcută de orice $b \in B$.

(2) Punctul (1) al acestei observaţii ne arată, suplimentar, că ceea ce se poate demonstra prin inducţie structurală se poate demonstra şi prin inducţie matematică. Ca urmare, inducţia structurală nu este "mai puternică" decât inducţia matematică. Insă, utilizarea inducţiei structurale acolo unde este cazul conduce, în general, la simplificarea tehnică a demonstraţiilor oferind un plus de naturaleţe şi eleganţă.

Un aspect important ce trebuie discutat este cel legat de "ordinea" în care se face închiderea unei mulţimi atunci când sunt considerate mai mult de o relaţie. In cadrul şirului de mulţimi din demonstraţia Teoremei 2.1.1 nu este impusă o anume ordine, toate relaţiile fiind aplicate la fiecare pas. Am putea să ne întrebăm: dacă realizăm întâi închiderea mulţimii A la r_0, apoi a mulţimii $r_0 \llbracket A \rrbracket$ la r_1 etc., obţinem $\{r_i | i \geq 0\} \llbracket A \rrbracket$? Răspunsul este negativ, în general, iar în secţiunea următoare vom prezenta câteva exemple în acest sens.

2.2 Inchideri ale unei relaţii binare

In matematică se consideră frecvent închideri reflexive, simetrice, tranzitive etc. ale unor relaţii binare date. Acestea nu sunt altceva decât cazuri particulare ale Definiţiei 2.1.1.

Definiţia 2.2.1. Fie ρ o relaţie binară pe A şi P o proprietate ca în Definiţia 1.2.1.5. Cea mai mică relaţie binară ρ' pe A ce include ρ şi are proprietatea P este numită *închiderea* P a relaţiei ρ.

Este clar că dacă o relaţie binară ρ pe A are proprietatea P, atunci închiderea P a ei este chiar ρ. Vom nota prin $r(\rho)$ ($s(\rho)$, $t(\rho)$) închiderea reflexivă (simetrică, tranzitivă) a relaţiei ρ.

In Secţiunea 1.2.1 am definit relaţia ρ^+ şi am arătat că ea este cea mai mică relaţie ce include ρ şi este tranzitivă. Ca urmare, $t(\rho) = \rho^+$, ceea ce justifică terminologia adoptată pentru ρ^+ în respectiva secţiune. De asemenea, am introdus relaţia ρ^* şi am arătat că aceasta este cea mai mică relaţie ce include ρ, este reflexivă şi tranzitivă. Vom vedea că această relaţie este închiderea reflexivă şi tranzitivă a relaţiei ρ, justificând astfel terminologia adoptată şi pentru această relaţie în Secţiunea 1.2.1.

Inchiderea la echivalenţă a relaţiei ρ o vom nota prin $equiv(\rho)$ sau $\equiv_\rho$. Atunci când vom realiza închideri multiple ale unei relaţii vom omite parantezele înţelegând că ordinea de realizare a închiderilor este de la dreapta la stânga. De exemplu, vom folosi scrierea $trs(\rho)$ şi terminologia de închiderea simetrică reflexivă şi tranzitivă a relaţiei ρ pentru $t(r(s(\rho)))$.

Teorema 2.2.1. Fie ρ o relaţie binară pe o mulţime A. Atunci, au loc următoarele proprietăţi:

(1) $r(\rho) = \rho \cup \iota_A$;

(2) $s(\rho) = \rho \cup \rho^{-1}$;

(3) $t(\rho) = \rho^+ = \bigcup_{n \geq 1} \rho^n$;

(4) $rt(\rho) = \rho^* = \bigcup_{n \geq 0} \rho^n$.

Demonstraţie (1) Orice relaţie reflexivă pe A include ι_A şi, ca urmare, orice relaţie reflexivă ce include ρ va include şi $\rho \cup \iota_A$. Deci, cea mai mică relaţie reflexivă ce include ρ este $\rho \cup \iota_A$, adică, $r(\rho) = \rho \cup \iota_A$.

(2) Orice relaţie simetrică ce include ρ trebuie să includă şi ρ^{-1} (conform definiţiei). Ca urmare, cea mai mică relaţie simetrică ce include ρ este $\rho \cup \rho^{-1}$, adică, $s(\rho) = \rho \cup \rho^{-1}$.

(3) urmează de la discuţia de mai sus.

(4) $\rho^* = r(t(\rho)) = r(\rho^+) = \rho^+ \cup \iota_A = \rho^+ \cup \rho^0 = \bigcup_{n \geq 0} \rho^n$. □

Inchiderea tranzitivă a unei relaţii binare ρ pe A este caz particular al Definiţiei 2.1.1. In adevăr, dacă considerăm o relaţie θ ce conţine toate 2-uplurile de forma $(((a, b), (b, c)), (a, c))$, unde $a, b, c \in A$, atunci pe baza Observaţiei 2.1.1(1) şi a Teoremei 2.2.1(3) obţinem $\theta \, [\![\, \rho \,]\!] = \rho^+ = t(\rho)$.

Similar, se poate arăta că şi închiderea reflexivă (simetrică) este caz particular al Definiţiei 2.1.1.

Demonstraţia următoarei propoziţii rămâne în grija cititorului.

Propoziţia 2.2.1. Fie ρ şi σ relaţii binare pe o mulţime A. Atunci, au loc următoarele proprietăţi:

(1) $r(\rho \cup \sigma) = r(\rho) \cup r(\sigma)$;

(2) $s(\rho \cup \sigma) = s(\rho) \cup s(\sigma)$;

(3) $t(\rho \cup \iota_A) = t(\rho) \cup \iota_A$;

(4) $(\rho^n)^{-1} = (\rho^{-1})^n$, pentru orice $n \geq 1$;

(5) $(\bigcup_{n\geq 1} \rho^n)^{-1} = \bigcup_{n\geq 1}(\rho^{-1})^n$.

Teorema 2.2.2. Fie ρ o relaţie binară. Atunci, au loc următoarele proprietăţi:

(1) $sr(\rho) = rs(\rho)$;

(2) $tr(\rho) = rt(\rho)$;

(3) dacă ρ este simetrică, atunci $t(\rho)$ este simetrică.

Demonstraţie Vom folosi din plin Propoziţia 2.2.1.

(1) $sr(\rho) = s(\rho \cup \iota_A) = s(\rho) \cup \iota_A = r(s(\rho)) = rs(\rho)$.

(2) $tr(\rho) = t(\rho \cup \iota_A) = t(\rho) \cup \iota_A = r(t(\rho)) = rt(\rho)$.

(3) Vom arăta că $t(\rho) = t(\rho)^{-1}$ utilizând faptul că $\rho = \rho^{-1}$ (ρ este simetrică). Are loc,

$$(t(\rho))^{-1} = (\bigcup_{n\geq 1} \rho^n)^{-1} = \bigcup_{n\geq 1}(\rho^{-1})^n = \bigcup_{n\geq 1}(\rho)^n = t(\rho),$$

ceea ce stabileşte simetria relaţiei $t(\rho)$. □

Combinând Teorema 2.2.1(4) cu Teorema 2.2.2(2) obţinem că $tr(\rho)$ este închiderea reflexivă şi tranzitivă a relaţiei ρ, ea fiind de fapt ρ^*.

Revenim acum la problema menţionată la sfârşitul secţiunii anterioare. Aşa cum am spus, ordinea în care se realizează închiderea unei mulţimi este foarte importantă. Punctele (1) şi (2) ale Teoremei 2.2.2 ne arată că ordinea închiderii la simetrie/tranzitivitate şi reflexivitate poate fi permutată fără a afecta rezultatul final, în timp ce punctul (3) ne spune că închiderea tranzitivă nu "distruge" simetria. Inchiderea simetrică poate distruge însă tranzitivitatea, aşa cum ne arată următorul exemplu. Fie

$$\rho = \{(1,1),(1,2),(1,3),(2,2),(3,3)\} \subseteq \{1,2,3\} \times \{1,2,3\}.$$

Au loc relaţiile:

$$st(\rho) = \{(1,1),(2,2),(3,3),(1,2),(2,1),(1,3),(3,1)\},$$

$$ts(\rho) = \{(1,1),(2,2),(3,3),(1,2),(2,1),(1,3),(3,1),(2,3),(3,2)\},$$

care ne arată că $st(\rho) \neq ts(\rho)$.

Teorema 2.2.3. Fie ρ o relaţie binară pe o mulţime A. Atunci,

$$\equiv_\rho = tsr(\rho) = trs(\rho) = rts(\rho).$$

Demonstraţie Utilizând Teorema 2.2.2 obţinem şirul de egalităţi

$$tsr(\rho) = trs(\rho) = rts(\rho)$$

şi faptul că $tsr(\rho)$ este relaţie de echivalenţă. In plus, $\rho \subseteq tsr(\rho)$.

Dacă σ este o relaţie de echivalenţă pe A ce include ρ, atunci

$$r(\rho) \subseteq r(\sigma) = \sigma, \;\; sr(\rho) \subseteq s(\sigma) = \sigma \;\text{ şi }\; tsr(\rho) \subseteq t(\sigma) = \sigma;$$

ca urmare, $tsr(\rho)$ este cea mai mică echivalenţă pe A ce include ρ. □

Teorema anterioară ne spune că pentru a realiza închiderea la echivalenţă a unei relaţii ρ este important de a realiza întâi închiderea simetrică şi apoi închiderea tranzitivă; închiderea reflexivă poate fi realizată oricând.

2.3 Definiţii inductive/recursive

O construcţie inductivă de mulţimi are ca scop construcţia unei mulţimi pornind de la o mulţime dată (de elemente de bază) şi adăugând, pas cu pas, noi elemente.

Considerăm următorul exemplu din logica matematică. Fie A o mulţime nevidă astfel încât $A \cap \{\neg, \vee, \wedge, (,)\} = \emptyset$. *Formulele propoziţionale peste* A sunt definite prin:

(a) orice element al mulţimii A este formulă propoziţională;

(b) dacă w_1 şi w_2 sunt formule propoziţionale, atunci

$$(\neg w_1),\; (w_1 \vee w_2) \text{ şi } (w_1 \wedge w_2)$$

sunt formule propoziţionale;

(c) formulele propoziţionale sunt definite numai ca la (a) sau (b).

Un cititor pretenţios din punct de vedere a rigorii matematice se poate întreba: este aceasta o "definiţie" matematică a noţiunii de formulă propoziţională (peste A)? Există o mulţime a tuturor formulelor propoziţionale (peste A)? Cum pot fi demonstrate anumite proprietăţi ale formulelor propoziţionale?

Pentru a răspunde acestor întrebări, vom rescrie "definiţia" de mai sus în termeni de închidere a unei mulţimi. Considerăm relaţiile r_0 şi r_1 ce conţin toate perechile de forma $(x, (\neg x))$ şi respectiv, $((x, y), (x \vee y))$ şi $((x, y), (x \wedge y))$, unde $x, y \in (A \cup \{\neg, \vee, \wedge, (,)\})^+$. Atunci, presupunând că există mulţimea tuturor formulelor propoziţionale peste A, fie aceasta $FP(A)$, punctele (a), (b) şi (c) ne spun că:

(a') B include A (de la (a));

(b') B este închisă la r_0 şi r_1 (de la (b));

(c') B este cea mai mică mulţime, în sensul incluziunii, cu proprietăţile (a') şi (b') (de la (c)).

Adică, $FP(A)$ este închiderea mulţimii A la mulţimea $\{r_0, r_1\}$ de relaţii, mulţime ce există în baza Teoremei 2.1.1. Ca urmare, $FP(A)$ există şi este reuniunea şirului de mulţimi $\langle B_m | m \geq 0 \rangle$ dat prin:

- $B_0 = A$;
- $B_{m+1} = B_m \cup \{(\neg w_1), (w_1 \vee w_2), (w_1 \wedge w_2) | w_1, w_2 \in B_m\}$, pentru orice $m \geq 0$.

Principiul inducţiei structurale poate fi aplicat mulţimii $FP(A)$ pentru a demonstra anumite proprietăţi ale elementelor acesteia. Astfel, pentru a arăta că în orice formulă propoziţională $w \in FP(A)$ numărul de paranteze "(" este egal cu numărul de paranteze ")", avem de verificat următoarele:

- dacă $w \in A$, atunci proprietatea este satisfăcută;
- dacă w este de forma $w = (\neg w_1)$ sau $w = (w_1 \vee w_2)$ sau $w = (w_1 \wedge w_2)$, şi presupunem proprietatea adevărată pentru w_1 şi w_2, atunci ea va fi adevărată şi pentru w.

Discuţia purtată până acum conduce la următoarea definiţie. Spunem că o mulţime B este *definită inductiv* dacă există o mulţime A şi o mulţime $\mathcal{R}$ de relaţii astfel încât $B = \mathcal{R}[\![A]\!]$ [1]. Aceasta nu este o nouă definiţie; nu am făcut altceva decât să atribuim o noua terminologie noţiunii de închidere a unei mulţimi, să scoatem în evidenţă faptul că închiderea unei mulţimi poate acţiona ca metodă de definiţie de mulţimi şi, să justificăm "formularea" utilizată frecvent în descrierea inductivă a (obiectelor) unei mulţimi.

Fie $B = \mathcal{R}[\![A]\!]$. Observaţia 2.1.1(1) ne spune că pentru orice $b \in B$ există o secvenţă

$$a_1, \ldots, a_i, \ldots, a_n = b$$

astfel încât, pentru orice $1 \leq i \leq n$, are loc

- $a_i \in A$, sau
- există $r \in \mathcal{R}$ cu $n_r = 0$ şi $(a) \in r$, sau
- există $r \in \mathcal{R}$ şi $j_1, \ldots, j_{n_r} < i$ astfel încât $((a_{j_1}, \ldots, a_{j_{n_r}}), a) \in r$.

O astfel de secvenţă poartă denumirea de *construcţie/definiţie inductivă* a lui b. Ca urmare, B este mulţimea tuturor elementelor ce au cel puţin o construcţie inductivă de la A şi $\mathcal{R}$.

[1] Unii autori [112, 49] atribuie terminologia de *constructor* relaţiilor ce intervin într-o astfel de definiţie. Analiza exemplului de mai sus credem că justifică cititorului această terminologie.

Un alt aspect important pe care trebuie să-l discutăm este cel legat de definiţia prin recursie/recurenţă a unor funcţii al căror domeniu este o mulţime definită inductiv. Observaţia de la care plecăm constă în faptul că o funcţie este o relaţie, deci o mulţime şi, atunci, a defini recursiv o funcţie revine la a defini inductiv o mulţime. Dar, să fixăm întâi cu exactitate problematica pe care o urmărim.

Fie B o mulţime definită inductiv de A şi $\mathcal{R}$, V o mulţime, g o funcţie de la A la V şi h o funcţie ce asociază fiecărei relaţii r o funcţie $h(r)$ (parţială sau totală) de la V^{n_r} la V. Ne punem problema existenţei unei funcţii $f : B \to V$ cu proprietăţile:

(i) $f(a) = g(a)$, pentru orice $a \in A$;

(ii) $f(a) = h(r)(f(a_1), \ldots, f(a_{n_r}))$, pentru orice $a, a_1, \ldots, a_{n_r}$ ce satisface $((a_1, \ldots, a_{n_r}), a) \in r$ şi $h(r)(f(a_1), \ldots, f(a_{n_r}))\downarrow$

(egalitatea de la (ii) trebuie înţeleasă ca egalitate de funcţii parţial definite. Astfel, dacă $h(r)$ nu este definită pe $(f(a_1), \ldots, f(a_{n_r}))$, atunci f nu va fi definită pe a).

Evident, pentru ca o astfel de funcţie să existe este necesar ca, pentru orice $((a_1, \ldots, a_{n_r}), a) \in r$ şi $((a'_1, \ldots, a'_{n_{r'}}), a) \in r'$ să avem

(iii) $h(r)(f(a_1), \ldots, f(a_{n_r})) = h(r')(f(a'_1), \ldots, f(a'_{n_{r'}}))$.

O astfel de condiţie, pe care am dori-o satisfăcută a priori de definiţia funcţiei f, implică însăşi funcţia f. Ea va fi însă satisfăcută dacă presupunem că pentru orice $a \in B$, ori $a \in A$, ori există o unică relaţie r şi un unic n_r-uplu $(a_1, \ldots, a_{n_r})$ astfel încât $((a_1, \ldots, a_{n_r}), a) \in r$. De fapt, aceasta din urmă va fi ipoteza în care vom lucra.

Aşa cum am spus, o funcţie este o relaţie şi, deci, o mulţime. Atunci, a defini f cu proprietăţile (i) şi (ii) revine la a defini o mulţime $f \subseteq B \times V$ astfel încât:

(a) $(a, g(a)) \in f$, pentru orice $a \in A$;

(b) dacă $(a_1, b_1), \ldots, (a_{n_r}, b_{n_r}) \in f$, $((a_1, \ldots, a_{n_r}), a) \in r$ şi $h(r)(b_1, \ldots, b_{n_r})\downarrow$, atunci $(a, h(r)(b_1, \ldots, b_{n_r})) \in f$;

(c) f este cea mai mică mulţime, în sensul incluziunii, cu proprietăţile (a) şi (b).

Următoarea lemă va stabili existenţa unei astfel de mulţimi.

Lema 2.3.1. Fie B mulţimea definită inductiv de A şi $\mathcal{R}$, V o mulţime, g o funcţie de la A la V şi h o funcţie ce asociază fiecărei relaţii $r \in \mathcal{R}$ o funcţie $h(r)$ (parţială sau totală) de la V^{n_r} la V. Atunci, există o unică mulţime $f \subseteq B \times V$ cu proprietăţile (a), (b) şi (c).

Demonstraţie Fie mulţimea $A' = \{(a, g(a)) | a \in A\}$. Pentru fiecare relaţie r considerăm o nouă relaţie r' astfel încât

$$(((a_1, b_1), \ldots, (a_{n_r}, b_{n_r})), (a, h(r)(b_1, \ldots, b_{n_r}))) \in r'$$

dacă şi numai dacă

$$((a_1, \ldots, a_{n_r}), a) \in r, \;\; b_1, \ldots, b_{n_r} \in V \;\text{ şi }\; h(r)(b_1, \ldots, b_{n_r})\downarrow .$$

Fie $\mathcal{R}' = (r'|r \in \mathcal{R})$. Există atunci o unică mulţime B' ce este închiderea mulţimii A' la $\mathcal{R}'$. Pe baza definiţiei închiderii şi a Principiului inducţiei structurale se obţine cu uşurinţă că $f = B'$ este unica mulţime ce satisface lema. □

Mulţimea f din Lema 2.3.1 nu este, în mod necesar, funcţie.

Definiţia 2.3.1. O mulţime B spunem că este *liber inductiv definită* de A şi $\mathcal{R}$ dacă B este definită inductiv de A şi $\mathcal{R}$ şi, pentru orice $a \in B$,

- ori $a \in A$,
- ori există o unică relaţie $r \in \mathcal{R}$ şi un unic n_r-uplu $(a_1, \ldots, a_{n_r})$ astfel încât $((a_1, \ldots, a_{n_r}), a) \in r$.

Teorema 2.3.1. (Teorema recursiei)
Fie B o mulţime definită inductiv de A şi $\mathcal{R}$, V o mulţime, g o funcţie de la A la V şi h o funcţie ce asociază fiecărei c-relaţii $r \in \mathcal{R}$ o funcţie $h(r)$ (parţială sau totală) de la V^{n_r} la V. Dacă B este liber inductiv definită de A şi $\mathcal{R}$, atunci există o unică funcţie (parţială) $f : B \to V$ astfel încât:

(i) $f(a) = g(a)$, pentru orice $a \in A$;

(ii) $f(a) = h(r)(f(a_1), \ldots, f(a_{n_r}))$, pentru orice $a, a_1, \ldots, a_{n_r}$ ce satisface $((a_1, \ldots, a_{n_r}), a) \in r$ şi $h(r)(f(a_1), \ldots, f(a_{n_r}))\downarrow$.

(egalitatea de la (ii) este înţeleasă ca egalitate de funcţii parţial definite).

Demonstraţie Lema 2.3.1 asigură existenţa unei mulţimi $f \subseteq B \times V$ cu proprietăţile (a), (b) şi (c). Prin inducţie structurală şi utilizând ipoteza (B este liber inductiv definită) se arată că f satisface teorema. □

O uşoară modificare a demonstraţiei Lemei 2.3.1 permite stabilirea următoarei variante a Teoremei recursiei.

Teorema 2.3.2. Fie B o mulţime definită inductiv de A şi $\mathcal{R}$, V o mulţime, g o funcţie de la A la V şi h o funcţie ce asociază fiecărei relaţii $r \in \mathcal{R}$ o funcţie $h(r)$ (parţială sau totală) de la $B^{n_r} \times V^{n_r}$ la V. Dacă B este liber inductiv definită de A şi $\mathcal{R}$, atunci există o unică funcţie (parţială) $f : B \to V$ astfel încât:

(i) $f(a) = g(a)$, pentru orice $a \in A$;

(ii) $f(a) = h(r)(a_1, \ldots, a_{n_r}, f(a_1), \ldots, f(a_{n_r}))$, pentru orice $a, a_1, \ldots, a_{n_r}$ ce satisface $((a_1, \ldots, a_{n_r}), a) \in r$ şi $h(r)(a_1, \ldots, a_{n_r}, f(a_1), \ldots, f(a_{n_r}))\downarrow$.

Teoremele 2.3.1 şi t2:p540 sunt de importanţă crucială ori de câte ori avem de extins funcţii definite pe o mulţime A la funcţii definite pe închiderea mulţimii A la anumiţi operatori.

Capitolul 3

Sisteme relaţionale şi algebre universale

Structurile relaţionale şi algebrele universale constituie un cadru general prin care pot fi introduse mulţimile parţial ordonate, semigrupurile, grupurile etc., şi prin care pot fi studiate proprietăţi comune ale acestora.

3.1 Sisteme relaţionale

De multe ori apare necesitatea de a considera cupluri formate dintre o mulţime de bază şi anumite relaţii pe acea mulţime. Aceste cupluri se numesc *sisteme relaţionale*.

Definiţia 3.1.1. Se numeşte *sistem relaţional* orice cuplu $\mathcal{R} = (A; R)$, unde A este o mulţime arbitrară iar R este o familie de relaţii pe A.

Dacă $\mathcal{R} = (A; R)$ este un sistem relaţional şi $R = \{\rho_1, \ldots, \rho_n\}$, $n \geq 1$, atunci vom mai nota $\mathcal{R}$ prin $(A; \rho_1, \ldots, \rho_n)$. Mulţimea A va fi numită *mulţimea suport* a sistemului relaţional $\mathcal{R}$. In cazul particular în care R este formată doar dintr-o singură relaţie binară ρ vom spune că $(A; R)$ este o *structură* (*relaţională*), ce va fi notată simplificat prin $(A; \rho)$.

Structurile introduse în Definiţia 1.2.5.1 sunt cazuri particulare de structuri relaţionale.

Reprezentarea grafică a unei structuri relaţionale se face aşa cum este menţionat în Secţiunea 1.2.5.

Definiţia 3.1.2. Fie $(A; \rho)$ o structură relaţională. $(B; \sigma)$ este numită *substructură* a structurii $(A; \rho)$ dacă $B \subseteq A$ şi $\sigma = \rho_{|B}$.

Observăm că o substructură $(B; \sigma)$ a unei structuri $(A; \rho)$ este complet determinată de submulţimea B. Ca urmare, ne vom referi uneori la substructuri ale unei structuri ca fiind submulţimi ale mulţimii suport a acestora.

Atunci când pentru o anumită structură relaţională este adoptată o terminologie specifică, cum ar fi de exemplu cea de "mulţime parţial ordonată", terminologia de substructură relaţională va fi modificată corespunzător, cum ar fi de exemplu "submulţime parţial ordonată", abreviat *sub-mpo*.

Definiţia 3.1.3. Fie $(A;\rho)$ o structură, $a,b\in A$ şi $B\subseteq A$.

(1) Se numeşte *lanţ al structurii* $(A;\rho)$ orice submulţime $B\subseteq A$ cu proprietatea că $(B;\rho|_B)$ este lanţ.

(2) Se numeşte *submulţime dirijată a structurii* $(A;\leq)$ orice submulţime nevidă $B\subseteq A$ cu proprietatea că $(B;\rho|_B)$ este mulţime dirijată.

(3) Se numeşte *submulţime filtrată a structurii* $(A;\leq)$ orice submulţime nevidă $B\subseteq A$ cu proprietatea că $(B;\rho|_B)$ este mulţime filtrată.

(4) Mulţimea $[a,b]=\{x\in A|a\leq x\leq b\}$ este numită *segmentul* sau *intervalul indus de* a *şi* b *în* M.

(5) Mulţimea $B\uparrow=\{x\in A|(\exists a\in B)(a\leq x)\}$ este numită *mulţimea succesorilor mulţimii* B.

(6) Mulţimea $B\downarrow=\{x\in A|(\exists a\in B)(x\leq a)\}$ este numită *mulţimea predecesorilor mulţimii* B.

Atunci când $B=\{a\}$ vom scrie mai simplu $a\uparrow$ ($a\downarrow$) în loc de $\{a\}\uparrow$ ($\{a\}\downarrow$). Nu trebuiesc confundate aceste notaţii cu $f(a)\uparrow$ sau $f(a)\downarrow$.

Este uşor de văzut că are loc $[a,b]=a\uparrow\cap b\downarrow$, pentru orice a şi b.

Conceptul de morfism de structuri este unul din cele mai importante concepte în teoria structurilor relaţionale.

Definiţia 3.1.4. Fie $M=(A;\rho)$ şi $N=(B;\sigma)$ două structuri. Numim *homomorfism* sau *morfism* de la M la N orice funcţie $f:A\to B$ pentru care are loc:

$$(\forall x,y\in A)(x\,\rho\,y\ \Rightarrow\ f(x)\,\sigma\,f(y)).$$

Homomorfismele de la o structură la ea însăşi se mai numesc şi *endomorfisme*.

In Figura 3.1 sunt reprezentate grafic 3 structuri. Funcţia f dată prin

$$f(a)=x=f(b),\ \ f(c)=y\ \text{ şi }\ f(d)=z$$

este un morfism de la structura din Figura 3.1(a) la structura din Figura 3.1(b), dar este uşor de văzut că nu există nici un morfism de la structura din Figura 3.1(b) la cea din Figura 3.1(c).

Morfismele se mai numesc şi *funcţii monotone* [1]. Ele au proprietatea că păstrează relaţia de ordine pe domeniului de definiţie, ceea ce in Definiţia 3.1.4 este

[1]Unii autori folosesc termenul de "funcţie monotonă" doar pentru morfisme între mpo. Considerăm însă că este potrivit a utiliza această terminologie şi pentru morfisme între structuri arbitrare, aşa cum s-a introdus în Definiţia 3.1.4.

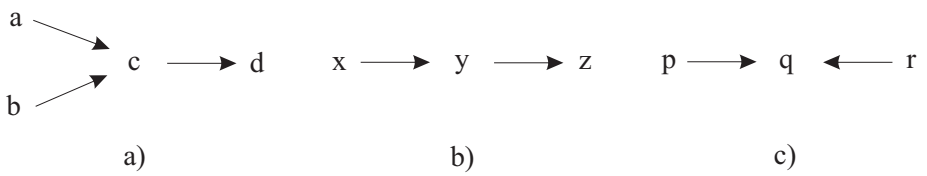

Figura 3.1: Structuri relationale

specificat prin "$x\,\rho\,y \;\Rightarrow\; f(x)\,\sigma\,f(y)$". Ca o consecinţă importantă, funcţiile monotone *păstrează lanţurile* (*submulţimile dirijate*, *submulţimile filtrate*) în sensul că, dacă $f : M \to N$ este funcţie monotonă de la M la N atunci pentru orice lanţ (submulţime dirijată, submulţime filtrată) B în M, $f(B)$ este lanţ (submulţime dirijată, submulţime filtrată) în N.

In cazul în care un morfism f are proprietatea

$$(\forall x, y \in A)(x \neq y \;\wedge\; x\,\rho\,y \;\Rightarrow\; f(x) \neq f(y) \;\wedge\; f(x)\,\sigma\,f(y))$$

vom spune că f este *morfism strict* (*funcţie monotonă strictă*). Mulţimea tuturor funcţiilor monotone de la M la N va fi notată prin $(M \to_m N)$ sau $(A \to_m B)$ atunci când relaţiile ρ şi σ sunt subînţelese din context.

Definiţia 3.1.5. Fie $M = (A; \rho)$ şi $N = (B; \sigma)$ două structuri. Numim *izomorfism* de la M la N orice funcţie bijectivă $f : A \to B$ pentru care f este morfism de la M la N şi f^{-1} este morfism de la N la M.

Izomorfismele de la o structură la ea însăşi se mai numesc şi *automorfisme*.

Propoziţia 3.1.1. Fie $M = (A; \rho)$ şi $N = (B; \sigma)$ două structuri. O funcţie f de la M la N este izomorfism de structuri dacă şi numai dacă f este funcţie bijectivă şi are loc

$$(\forall x, y \in A)(x\,\rho\,y \;\Leftrightarrow\; f(x)\,\sigma\,f(y)).$$

Demonstraţie Dacă f este izomorfism de la M la N atunci f este funcţie bijectivă şi au loc proprietăţile

$$(\forall x, y \in A)(x\,\rho\,y \;\Rightarrow\; f(x)\,\sigma\,f(y))$$

şi

$$(\forall x', y' \in B)(x'\,\sigma\,y' \;\Rightarrow\; f^{-1}(x')\,\rho\,f^{-1}(y')).$$

Alegând $x' = f(x)$ şi $y' = f(y)$ în cea de a doua relaţie, şi combinând cu prima, obţinem

$$(\forall x, y \in A)(x\,\rho\,y \;\Leftrightarrow\; f(x)\,\sigma\,f(y)).$$

Demonstraţia în sens invers decurge similar celei de mai sus. □

Este clar că dacă f este izomorfism, atunci atât f cât şi f^{-1} sunt morfisme stricte. Dacă există cel puţin un izomorfism de la M la N atunci vom spune că M şi N sunt *izomorfe*, şi vom nota $M \cong N$. In plus, dacă f este un izomorfism de la M la N şi dorim să specificăm aceasta atunci vom folosi notaţia $M \cong_f N$.

Propoziţia 3.1.2. Compunere de morfisme (izomorfisme) de structuri este morfism (izomorfism) de structuri.

Demonstraţie Fie $M = (A; \rho)$, $N = (B; \sigma)$ şi $P = (C; \theta)$ trei structuri, iar $f : A \to B$ şi $g : B \to C$ morfisme. Fie $a, b \in A$ cu $a \,\rho\, b$. Faptul că f este morfism conduce la $f(a) \,\sigma\, f(b)$, care combinată cu faptul că g este morfism conduce la $g(f(a)) \,\theta\, g(f(b))$. Deci, $g \circ f$ este morfism.

Demonstraţia decurge similar în cazul izomorfismelor. □

Fie $M = (A; \rho)$ şi $N = (B; \sigma)$ două structuri şi f un morfism de la M la N. Mulţimea $A/_{Ker(f)}$ poate fi organizată ca o structură considerând relaţia binară θ dată prin:

$$[x]_{Ker(f)} \,\theta\, [y]_{Ker(f)} \;\Leftrightarrow\; f(x) \,\sigma\, f(y),$$

pentru orice $x, y \in A$. Vom nota această structură prin $M/_{Ker(f)}$.

Propoziţia 3.1.3. Fie f un morfism de la $M = (A; \rho)$ la $N = (B; \sigma)$. Funcţia $g : A \to A/_{Ker(f)}$ dată prin $g(x) = [x]_{Ker(f)}$, pentru orice $x \in A$, este morfism de la M la $M/_{Ker(f)}$.

Demonstraţie Fie $a, b \in A$ cu $a \,\rho\, b$. Atunci, $f(a) \,\sigma\, f(b)$, ceea ce înseamnă $[a]_{Ker(f)} \,\theta\, [b]_{Ker(f)}$. Deci, g este morfism. □

3.2 Mulţimi parţial ordonate

Mulţimile parţial ordonate sunt structuri de importanţă majoră. In această secţiune vom prezenta câteva din conceptele şi proprietăţile de bază asupra acestora.

3.2.1 Concepte de bază

Fie $M = (A; \rho)$ o mpo. Cel mai adesea relaţia ρ se notează prin $\leq$. Prin $<$ vom nota relaţia $\leq -\iota_A$. Inversa relaţiei $\leq$ ($<$) va fi notată prin $\geq$ ($>$).

Dacă $a, b \in A$ şi $a \leq b$ sau $b \leq a$, atunci vom spune că a şi b sunt *comparabile* (în raport cu $\leq$); altfel, ele sunt numite *incomparabile* (în raport cu $\leq$). Dacă $a \leq b$, atunci vom spune că a *precede* b sau că b *succede* a. In ipoteza suplimentară în care $a \neq b$ şi nu există c astfel încât $a < c < b$, vom spune că a *precede imediat* pe b sau că b *succede imediat* pe a; a este numit *predecesor imediat* al lui b, iar b este numit *succesor imediat* al lui a.

Este uşor de văzut că o ordine parţială ρ este totală (pe A) dacă şi numai dacă $\rho \cup \rho^{-1} = A \times A$. Intr-un lanţ, orice două elemente sunt comparabile. Se mai foloseşte adesea şi terminologia de *antilanţ* pentru mpo în care orice două elemente sunt incomparabile.

In cazul mulţimilor parţial ordonate, pe lângă reprezentarea grafică introdusă în Secţiunea 1.2.5, o altă reprezentarea frecvent întâlnită este cea prin *diagrame*

Hasse [2]. Aceste diagrame se contruiesc similar reprezentărilor specifice mpo utilizate de noi dar cu deosebirea că arcele ce unesc nodurile grafului sunt neorientate. In acest caz, orientarea este suplinită desenând nodul ce succede un alt nod mai sus decât acesta, pe verticală. De exemplu, în Figura 3.2 sunt reprezentate prin diagrame Hasse 4 mpo. Ultimele două reprezentări sunt ale aceleiaşi mpo a cărei relaţie binară este

$$\{(a,a),(b,b),(c,c),(d,d),(e,e),(f,f),(a,d),(a,e),(b,d),(b,f),(c,e),(c,f)\}.$$

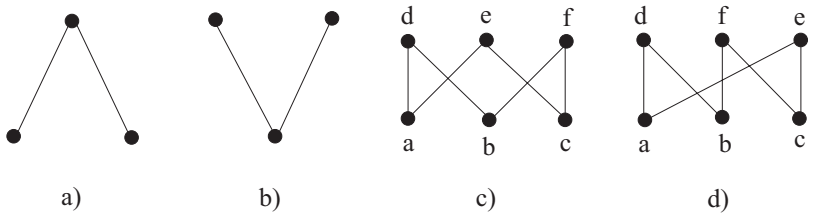

Figura 3.2: Diagrame Hasse

Definiţia 3.2.1.1. Fie $M = (A; \leq)$ o mpo şi $B \subseteq A$.

(1) Un element $a \in A$ este numit *majorant* al mulţimii B dacă $b \leq a$, pentru orice $b \in B$.

(2) Un element $a \in A$ este numit *cel mai mic majorant* al mulţimii B dacă el este majorant al lui B şi pentru orice alt majorant a' al mulţimii B are loc $a \leq a'$.

(3) B este numită *majorată* sau *mărginită superior* dacă există cel puţin un majorant al ei.

(4) Un element $b \in B$ este numit *maximal* (în B) dacă pentru orice $b' \in B$, $b' \leq b$ sau b şi b' sunt incomparable.

(5) Un element $b \in B$ este numit *cel mai mare element* (al mulţimii B) dacă $b' \leq b$, pentru orice $b' \in B$.

(6) Un element $a \in A$ este numit *minorant* al mulţimii B dacă $a \leq b$, pentru orice $b \in B$.

(7) Un element $a \in A$ este numit *cel mai mare minorant* al mulţimii B dacă el este minorant al lui B şi pentru orice alt minorant a' al mulţimii B are loc $a' \leq a$.

(8) B este numită *minorată* sau *mărginită inferior* dacă există cel puţin un minorant al ei.

[2] După numele matematicianului german Helmut Hasse (1898–1979) care le-a utilizat pentru prima dată cu scopul reprezentării grafice a mpo.

(9) Un element $b \in B$ este numit *minimal* (în B) dacă pentru orice $b' \in B$, $b \leq b'$ sau b şi b' sunt incomparabile.

(10) Un element $b \in B$ este numit *cel mai mic element* (al mulţimii B) dacă $b \leq b'$, pentru orice $b' \in B$.

(11) B este numită *mărginită* dacă este mărginită inferior şi superior.

Fie $M = (A; \leq)$ o mpo şi $B \subseteq A$. Vom nota prin B^+ mulţimea tuturor majoranţilor mulţimii B în M, şi prin B^- mulţimea tuturor minoranţilor mulţimii B în M. Atragem atenţia asupra faptului că $B^+ \subseteq B\uparrow$, dar incluziunea poate fi strictă (similar pentru $B^- \subseteq B\downarrow$).

Este clar că dacă există cel mai mic majorant (cel mai mare minorant) al mulţimii B, atunci acesta este unic, el fiind cel mai mic element al mulţimii B^+ (cel mai mare element al mulţimii B^-). Acest element, atunci când există, va fi notat prin $lub_M(B)$ ($glb_M(B)$) sau $sup_M(B)$ ($inf_M(B)$) sau $\vee_M(b|b \in B)$ ($\wedge_M(b|b \in B)$) şi va mai fi numit şi *supremum* (*infimum*) mulţimii B. In cazul în care B are două elemente, $B = \{b_1, b_2\}$, ultima notaţie va fi simplificată la $b_1 \vee_M b_2$ ($b_1 \wedge_M b_2$). Similar, cel mai mic (cel mai mare) element al mulţimii B, dacă există, este unic; el se notează (atunci când există) prin $\bot_{M,B}$ ($\top_{M,B}$) sau $min_M(B)$ ($max_M(B)$). Indicele M va fi întotdeauna eliminat atunci când se subânţelege din context.

Dacă elementul $a \in A$ are proprietatea $b < a$ pentru orice $b \in B$, atunci el este numit şi *majorant strict al* (*margine superioară strictă a*) mulţimii B, iar B spunem că este *majorată strict* (*mărginită superior strict*). Dacă există cel mai mic majorant strict al mulţimii B, atunci acesta este numit şi *succesor imediat* al lui B. Orice majorant strict este majorant, dar o mulţime poate avea majoranţi fără a avea majoranţi stricţi. Noţiunile de *minorant strict* şi mulţime *minorată strict* (*mărginită inferior strict*) se introduc în manieră similară. Cel mai mare minorant strict al unei mulţimi, atunci când există, este numit şi *precedesor imediat* al acelei mulţimi.

Exemplul 3.2.1.1. Fie A o mulţime. Pentru orice sistem nevid S peste A, cuplul $(S; \subseteq_S)$ este mpo pentru care $\bigcap S$ este cel mai mic element, iar $\bigcup S$ este cel mai mare element (în cazul $S = \mathcal{P}(A)$, $\bigcap S = \emptyset$ şi $\bigcup S = A$).

Utilizând conceptele introduse în Definiţia 3.2.1.1, mpo dirijate (filtrate) pot fi definite ca fiind mpo ce au proprietatea că orice submulţime cu unul sau două elemente admite cel puţin un majorant (minorant). Evident, această proprietate este echivalentă cu a spune că orice submulţime finită şi nevidă admite cel puţin un majorant (minorant).

Orice mpo care are cel mai mare element (cel mai mic element) este dirijată (filtrată).

Am spus în Secţiunea 3.1 că funcţiile monotone păstrează lanţurile şi submulţimile dirijate şi filtrate. Acest concept de păstrare a unei proprietăţi poate fi extins. De exemplu, putem spune că funcţia $f : M \to M'$ *păstrează supremum submulţimilor nevide* dacă pentru orice submulţime nevidă $B \subseteq A$ pentru care există $sup_M(B)$,

există şi $sup_{M'}(f(B))$ şi are loc $sup_{M'}(f(B)) = f(sup_M(B))$. In manieră similară putem discuta despre funcţii ce păstrează infimum etc.

Fie A o mulţime şi $PO(A)$ mulţimea tuturor ordinilor parţiale pe A. Această mulţime este nevidă deoarece $\iota_A \in PO(A)$, şi poate fi structurată ca o mpo prin relaţia de incluziune pe relaţii de ordine parţiale

$$\rho \leq \theta \quad \Leftrightarrow \quad \rho \subseteq \theta,$$

pentru orice $\rho, \theta \in PO(A)$.

Propoziţia 3.2.1.1. Fie ρ o ordine parţială pe o mulţime A. ρ este totală dacă şi numai dacă este element maximal al mulţimii parţial ordonate $(PO(A); \leq)$.

Demonstraţie Fie $\rho \in PO(A)$ o ordine totală. Presupunem prin contradicţie că ρ nu este element maximal în $(PO(A); \leq)$. Atunci, există $\theta \in PO(A)$ astfel încât $\rho < \theta$. Adică, $\rho \subset \theta$ şi, deci, există $a, b \in A$ astfel încât $a \neq b$ şi $(a, b) \in \theta - \rho$. In plus, $(b, a) \notin \rho$ deoarece, altfel, am avea $(a, b) \in \theta$ şi $(b, a) \in \theta$ de unde ar urma $a = b$. Deci, ρ nu este ordine totală pe A; contradicţie.

Reciproc, presupunem că ρ este element maximal al mulţimii parţial ordonate $(PO(A); \leq)$ dar nu este ordine totală. Ca urmare, există $a, b \in A$ astfel încât $a \neq b$, $(a, b) \notin \rho$ şi $(b, a) \notin \rho$. Este uşor de văzut că există o ordine parţială θ pe A ce include $\rho \cup \{(a, b)\}$. Dar atunci, θ extinde strict relaţia ρ; contradicţie cu maximalitatea relaţiei ρ. □

Teorema 3.2.1.1. (Teorema de reprezentare a mpo)
Orice mpo $(A; \leq)$ este izomorfă cu o mpo de forma $(S; \subseteq_S)$, unde S este un sistem peste A.

Demonstraţie Fie $(A; \leq)$ o mpo. Pentru orice element $a \in A$ considerăm mulţimea $A_a = \{b \in A | b \leq a\}$ şi fie $S = \{A_a | a \in A\}$. Cuplul $(S; \subseteq_S)$ este mulţime parţial ordonată. Definim $f : A \to S$ prin $f(a) = A_a$. Această aplicaţie este bijecţie şi are loc

$$(\forall a, b \in A)(a \leq b \Leftrightarrow A_a \subseteq_S A_b),$$

ceea ce ne arată că f este izomorfism între $(A; \leq)$ şi $(S; \subseteq_S)$. □

3.2.2 Dualitate

Este uşor de văzut că dacă ρ este relaţie de ordine parţială pe o mulţime A, atunci ρ^{-1} este de asemenea relaţie de ordine parţială pe A.

Definiţia 3.2.2.1. Fie $M = (A; \leq)$ o mpo. Mulţimea parţial ordonată

$$M^{-1} = (A; \geq)$$

este numită *duala* mulţimii parţial ordonate M.

Este util de remarcat că anumite concepte valide pentru mpo au o contraparte pentru dual㠄. De exemplu, dacă a este un majorant pentru submulţimea B a mpo $(A;\leq)$, atunci a este minorant pentru B în duala $(A;\geq)$. Următorul tabel prezintă conceptele duale întâlnite până acum (alte concepte duale vor fi întâlnite pe parcursul lucrării).

Concept	Concept dual
majorant	minorant
minorant	majorant
mulţime mărginită superior	mulţime mărginită inferior
mulţime mărginită inferior	mulţime mărginită superior
cel mai mic majorant	cel mai mare minorant
cel mai mare minorant	cel mai mic majorant
element maximal	element minimal
element minimal	element maximal
cel mai mare element	cel mai mic element
cel mai mic element	cel mai mare element

Dacă toate conceptele ce apar într-o afirmaţie asupra unei mpo M sunt înlocuite prin conceptul dual corespunzător, atunci se obţine ceea ce se numeşte *afirmaţia dual㠄* afirmaţiei date. Dacă afirmaţia iniţială este validă în M, atunci afirmaţia dual㠄 va fi validă în duala lui M. Acesta este aşa numitul *Principiu al dualităţii pentru mpo* în baza căruia multe din demonstraţii pot fi reduse la jumătate.

3.2.3 Proprietăţi de bază ale supremum şi infimum

Vom prezenta în continuare câteva simple proprietăţi referitoare la cel mai mic majorant al unei submulţimi într-o mpo, proprietăţi ce vor avea multiple aplicaţii în secţiunile următoare. Evident, toate acestea pot fi dualizate conducând astfel la proprietăţi similare pentru cel mai mare minorant.

Propoziţia 3.2.3.1. Fie $M = (A;\leq)$ o mpo. Atunci, au loc următoarele proprietăţi:

(1) există $sup(\emptyset)$ dacă şi numai dacă M are cel mai mic element. In plus, dacă există cel mai mic element atunci $sup(\emptyset) = \bot_M = inf(A)$;

(2) există $inf(\emptyset)$ dacă şi numai dacă M are cel mai mare element. In plus, dacă există cel mai mare element atunci $inf(\emptyset) = \top_M = sup(A)$.

Demonstraţie (1) Dacă există $sup(\emptyset)$ atunci mulţimea de majoranţi ai mulţimii vide este nevidă şi $sup(\emptyset)$ este cel mai mic element al acesteia. Insă, observăm că mulţimea de majoranţi ai mulţimii vide este A (în limbaj logic, faptul că $a \in A$ este majorant al mulţimii vide se exprimă prin

$$(\forall x)(x \in \emptyset \;\Rightarrow\; x \leq a).$$

Cum "$x \in \emptyset$" este falsă, deducem că a este majorant pentru $\emptyset$). Ca urmare, existenţa $sup(\emptyset)$ conduce la existenţa celui mai mic element al mulţimii A, deci a mpo M.

Reciproc, dacă există $\perp_M$ atunci mulţimea de majoranţi ai mulţimii vide este nevidă deoarece conţine $\perp_M$ care, evident, este şi $sup(\emptyset)$.

Este clar că dacă există $\perp_M$ atunci $sup(\emptyset) = \perp_M = inf(A)$.

(2) se demonstrează similar proprietăţii de la (1). □

Atragem explicit atenţia asupra faptului că supremum mulţimii vide poate exista doar dacă mulţimea vidă este considerată submulţime a unei mpo nevide. Altfel spus, $sup(\emptyset)$ în $(\emptyset; \emptyset)$ nu există.

Următoarea propoziţie se obţine cu uşurinţă de la definiţii şi Propoziţia 3.2.3.1.

Propoziţia 3.2.3.2. Fie $M = (A; \leq)$ o mpo. Atunci, următoarele afirmaţii sunt echivalente:

(1) pentru orice submulţime $B \subseteq A$ există $sup(B)$;

(2) există $\perp_M$ şi pentru orice submulţime nevidă $B \subseteq A$ există $sup(B)$;

(3) pentru orice submulţime $B \subseteq A$ există $inf(B)$;

(4) există $\top_M$ şi pentru orice submulţime nevidă $B \subseteq A$ există $inf(B)$;

(5) Pentru orice submulţime $B \subseteq A$ există $sup(B)$ şi $inf(B)$;

(6) există $\perp_M$ şi $\top_M$, şi pentru orice submulţime nevidă $B \subseteq A$ există $sup(B)$ şi $inf(B)$.

Propoziţia 3.2.3.3. Fie $M = (A; \leq)$ a mpo şi $B \subseteq A$. Atunci, au loc următoarele proprietăţi:

(1) dacă există $sup(B)$ atunci există $inf(B^+)$, şi reciproc. In plus, dacă există $sup(B)$, atunci $sup(B) = inf(B^+)$;

(2) dacă există $inf(B)$, atunci $B^- = inf(B)^-$;

(3) dacă există $sup(B)$, atunci $(B^+)^- = sup(B)^-$

(s-a notat $inf(B)^-$ în loc de $\{inf(B)\}^-$, şi $sup(B)^-$ în loc de $\{sup(B)\}^-$).

Demonstraţie Vom demonstra doar (1). Să presupunem că există $sup(B)$. Acesta este cel mai mic majorant al mulţimii B^+ şi, ca urmare, el coincide cu $inf(B^+)$. □

Evident, Propoziţia 3.2.3.3 poate fi dualizată.

Următoarea lema va avea aplicaţii majore în Secţiunea 5.1, dar este importantă şi ca rezultat de sine stătător.

Lema 3.2.3.1. (minsup–majinf)
Fie $M = (A; \leq)$ o mpo. Pentru orice submulţime nevidă $B \subseteq A$ cu proprietatea $B^- \neq \emptyset$ are loc:

$$(\forall C \subseteq B^-)((\exists\, sup(C)) \Rightarrow sup(C) \in B^-).$$

Demonstraţie Fie $C \subseteq B^-$ astfel încât există $sup(C)$.

Dacă $C = \emptyset$ atunci $sup(C) = \perp_M$, ceea ce ne spune că $sup(C) \in B^-$. Să presupunem că C este nevidă. Orice element din B este majorant pentru C. Cum $sup(C)$ este cel mai mic majorant al mulţimii C, el va fi mai mic decât orice element din B. Ca urmare, $sup(C) \in B^-$. □

Interpretăm această foarte importantă lemă prin aceea că mulţimea minoranţilor unei submulţimi nevide este închisă la supremum (de aici provine prima parte, *minsup*, din denumirea lemei). Prin dualizare, obţinem că mulţimea majoranţilor unei submulţimi nevide este închisă la infimum (de aici provine a doua parte, *majinf*, din denumirea lemei).

Propoziţia 3.2.3.4. Fie $M = (A; \leq)$ o mpo şi $B = \{a_{ij} \in A | i \in I,\ j \in J\}$, unde I şi J sunt mulţimi nevide. Dacă au loc:

(1) există $b_i = sup(\{a_{ij} | j \in J\})$, pentru orice $i \in I$;

(2) există $u = sup(\{b_i | i \in I\})$,

atunci există $sup(\{a_{ij} | i \in I,\ j \in J\})$ şi $sup(\{a_{ij} | i \in I,\ j \in J\}) = u$.

Demonstraţie Presupunem că au loc (1) şi (2). Atunci, este clar că u este majorant al mulţimii $\{a_{ij} | i \in I,\ j \in J\}$.

Dacă v este un alt majorant al acestei mulţimi atunci, pentru orice i, $b_i \leq v$ deoarece b_i este cel mai mic majorant al mulţimii $\{a_{ij} | j \in J\}$ (v fiind majorant al acestei mulţimi). Cum u este cel mai mic majorat al mulţimii $\{b_i | i \in I\}$, urmează $u \leq v$. Deci, $u = sup(\{a_{ij} | i \in I,\ j \in J\})$. □

Corolarul 3.2.3.1. Fie $M = (A; \leq)$ o mpo şi $B = \{a_{ij} \in A | i \in I,\ j \in J\}$, unde I şi J sunt mulţimi nevide. Dacă au loc:

(1) există $b_i = sup(\{a_{ij} | j \in J\})$, pentru orice $i \in I$;

(2) există $u = sup(\{b_i | i \in I\})$;

(3) există $c_j = sup(\{a_{ij} | i \in I\})$, pentru orice $j \in J$;

(4) există $v = sup(\{c_j | j \in J\})$,

atunci există $sup(\{a_{ij} | i \in I,\ j \in J\})$ şi $sup(\{a_{ij} | i \in I,\ j \in J\}) = u = v$.

Demonstraţie Direct de la Propoziţia 3.2.3.4. □

Deci, calculul supremului unei mulţimi dublu indexate se poate face calculând supremum după unul din indexi, şi apoi după celălalt (atunci când aceştia există).

Corolarul 3.2.3.2. Fie $M = (A; \leq)$ o mpo şi $X, Y \subseteq A$ două submulţimi ale lui A. Dacă există $sup(X)$, $sup(Y)$ şi $sup(\{sup(X), sup(Y)\})$, atunci există şi $sup(X \cup Y)$ şi acesta este $sup(\{sup(X), sup(Y)\})$.

Demonstraţie Evident, putem presupune că atât X cât şi Y sunt nevide. Corolarul poate fi demonstrat direct, similar Propoziţiei 3.2.3.4, dar poate fi obţinut şi drept caz particular al acesteia considerând

$$X = \{a_{1j} | j \in J\}$$

şi

$$Y = \{a_{2k} | k \in K\}$$

cu $J \subseteq K$ sau $K \subseteq J$. Mai mult, putem presupune că are loc $J = K$. In adevăr, dacă am presupune că $K \subset J$ atunci, repetând un element din Y şi indexându-l cu indexi $2k$ cu $k \in J - K$, obţinem o nouă mulţime $Y' = \{a_{2j} | j \in J\}$ pentru care $sup(Y') = sup(Y)$ şi $X \cup Y = X \cup Y'$. □

Definiţia 3.2.3.1. Fie $M = (A; \leq)$ o mpo şi $B, C \subseteq A$. Spunem că C este cofinală în B dacă pentru orice $b \in B$ există $c \in C$ astfel încât $b \leq c$.

Orice submulţime este cofinală în $\emptyset$, dar $\emptyset$ nu este cofinală în nici o submulţime nevidă.

Propoziţia 3.2.3.5. Fie $M = (A; \leq)$ o mpo şi B, C submulţimi ale lui A astfel încât C este cofinală în B. Dacă există $sup(B)$ şi $sup(C)$, atunci $sup(B) \leq sup(C)$.

Demonstraţie Deoarece C este cofinală în B, $sup(C)$ este majorant pentru B. Deci, $sup(B) \leq sup(C)$. □

Propoziţia 3.2.3.6. Fie $M = (A; \leq)$ o mpo şi B, C submulţimi ale lui A astfel încât C este cofinală în B şi $C \subseteq B$. Dacă există unul din $sup(B)$ sau $sup(C)$ atunci există şi celălalt, şi ele sunt egale.

Demonstraţie Se observă că $B^+ = C^+$, de la care urmează propoziţia. □

Propoziţia 3.2.3.7. Fie $f : M \to M'$ un morfism de mpo şi $B \subseteq A$.

(1) Dacă există $sup(B)$, atunci $f(sup(B))$ este majorant pentru $f(B)$. Dacă în plus există şi $sup(f(B))$, atunci $sup(f(B)) \leq' f(sup(B))$.

(2) Dacă există $sup(B)$ şi f este izomorfism, atunci există şi $sup(hfB))$ şi are loc $sup(f(B)) = f(sup(B))$.

Demonstraţie (1) Presupunem că există $sup(B)$. Pentru orice element $a \in B$ are loc $a \leq sup(B)$. Cum f este morfism, urmează $f(a) \leq' f(sup(B))$. Ca urmare, $f(sup(B))$ este majorant pentru $f(B)$.

Să presupunem că există $sup(f(B))$. Cum acesta este cel mai mic majorant pentru $f(B)$, are loc $sup(f(B)) \leq' f(sup(B))$.

(2) Presupunem că există $sup(B)$ şi f este izomorfism. Conform punctului (1), $f(sup(B))$ este majorant pentru $f(B)$. Dacă ar exista un alt majorant a' pentru $f(B)$ care să satisfacă $a' <' f(sup(B))$, atunci am obţine că $f^{-1}(a')$ este majorant pentru $f^{-1}(f(B)) = B$ şi $f^{-1}(a') < f^{-1}(f(sup(B))) = sup(B)$, ceea ce ar contrazice faptul că $sup(B)$ este cel mai mic majorant pentru B (s-a folosit faptul că f este bijecţie şi f^{-1} este morfism strict).

Deci, $f(sup(B))$ este cel mai mic majorant pentru $f(B)$, ceea ce înseamnă că are loc $sup(f(B)) = f(sup(B))$. □

Corolarul 3.2.3.3. Fie $f : M \to M'$ un morfism de mpo.

(1) Dacă există $\perp_M$ şi $\perp_{M'}$, atunci $\perp_{M'} \leq' f(\perp_M)$.

(2) Dacă există $\perp_M$ şi f este izomorfism, atunci există şi $\perp_{M'}$ şi $\perp_{M'} = f(\perp_M)$.

Demonstraţie Atât (1) cât şi (2) pot fi demonstrate direct sau utilizând Propoziţia 3.2.3.7 cu $B = \{\perp_M\}$. □

Putem spune deci că izomorfismele păstrează supremum submulţimilor atunci când acesta există. In particular, izomorfismele păstrează cel mai mic element atunci când acesta există.

3.2.4 Construcţii de mpo

Arătăm în continuare cum putem construi noi mulţimi parţial ordonate pornind de la mulţimi parţial ordonate date. Vom spune ca o familie de mpo (indexată sau nu) este *familie disjunctă de mpo* dacă mulţimile suport sunt disjuncte două câte două.

Mpo plate. Fie A o mulţime. Considerăm un nou element, notat $\perp_A$ sau $\perp$ atunci când A este subînţeleasă din context ($\perp_A \notin A$), şi fie $A_\perp = A \cup \{\perp_A\}$.

Definiţia 3.2.4.1. Fie A o mulţime. *Mulţimea parţial ordonată plată indusă de* A este mulţimea parţial ordonată $(A_\perp; \leq)$, unde $\leq$ este dată prin:

$$(\forall x, y \in A_\perp)(x \leq y \;\Leftrightarrow\; x = y \;\vee\; (x = \perp_A \;\wedge y \in A)).$$

Se verifică cu uşurinţă că, în adevăr, $(A_\perp; \leq)$ este mpo. Elementul $\perp_A$ este cel mai mic element al mulţimii parţial ordonate $(A_\perp; \leq)$, şi orice element $a \in A$ este element maximal. $(A_\perp; \leq)$ are cel mai mare element dacă şi numai dacă $|A| \leq 1$.

Această construcţie este importantă în special atunci când A are cel puţin 2 elemente. Prin intermediul ei se introduce un cel mai mic element, elementele mulţimii A fiind tratate egal.

Dacă $\perp$ este un element arbitrar, atunci vom nota prin $\perp\!\!\!\perp$ mulţimea parţial ordonată $\perp\!\!\!\perp = (\{\perp\}; \{(\perp, \perp)\})$.

Pe mpo plate proprietatea de monotonie admite caracterizări foarte simple.

Propoziţia 3.2.4.1. Fie $f : A_\perp \to B_\perp$. Atunci, f este monotonă dacă şi numai dacă $f(\perp_A) = \perp_B$ sau există $c \in B_\perp$ astfel încât $f(a) = c$ pentru orice $a \in A_\perp$.

Demonstraţie Urmează direct de la definiţia ordinii parţiale pe mpo plate. □

Intersecţie de mpo. *Intersecţia* unei familii $((A_i; \leq_i) | i \in I)$ de mpo, notată prin $\bigcap_{i \in I}(A_i; \leq_i)$, este definită ca fiind

$$\bigcap_{i \in I}(A_i; \leq_i) = (\bigcap_{i \in I} A_i; \bigcap_{i \in I} \leq_i).$$

Este cât se poate de clar că intersecţie de mpo este mpo. De asemenea, este uşor de verificat că intersecţia unei familii de sub-mpo ale unei mpo M este sub-mpo a mpo M.

Intersecţia mpo din Figura 3.3(a)(b) este reprezentată grafic în Figura 3.3(c).

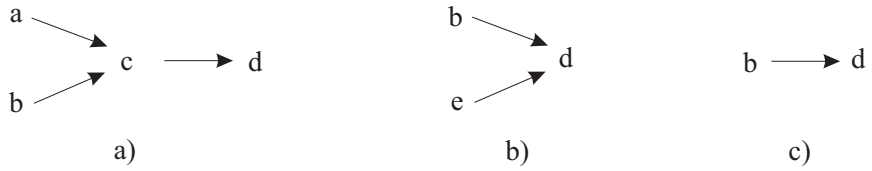

Figura 3.3: Intersecţie de mpo

Reuniune de mpo. *Reuniunea* unei familii $((A_i; \leq_i)|i \in I)$ de mpo, notată prin $\bigcup_{i\in I}(A_i; \leq_i)$, este definită ca fiind

$$\bigcup_{i\in I}(A_i; \leq_i) = (\bigcup_{i\in I} A_i; \bigcup_{i\in I} \leq_i).$$

Spre deosebire de intersecţie, reuniunea unei familii de mpo poate să nu mai fie mpo. Dacă însă familia este disjunctă, atunci şi reuniunea este mpo.

Reuniunea mpo din Figura 3.3(a)(b) este reprezentată grafic în Figura 3.4.

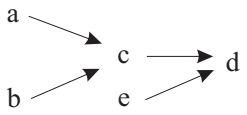

Figura 3.4: Reuniune de mpo

Este clar că reuniunea de mpo, atunci când este definită, este asociativă.

Sume de mpo. Introducerea conceptului de *sumă* de mpo necesită, în prealabil, demonstrarea următorului rezultat.

Propoziţia 3.2.4.2. Fie $\mathcal{I} = (I; \leq)$ o mto şi $((A_i; \leq_i)|i \in I)$ o familie disjunctă de mpo. Atunci, relaţia binară $\leq'$ pe $\bigcup_{i\in I} A_i$ dată prin

$$x \leq' y \;\Leftrightarrow\; (\exists i, j \in I)(x \in A_i \wedge y \in A_j \wedge (\text{ori } i < j \text{ ori } (i = j \wedge x \leq_i y)))$$

este relaţie de ordine parţială.

Demonstraţie Vom verifica reflexivitatea, antisimetria şi tranzitivitatea relaţiei $\leq'$. Fie $x \in \bigcup_{i\in I} A_i$. Deoarece mulţimile acestei familii sunt disjuncte urmează că există un unic $i \in I$ astfel încât $x \in A_i$. Reflexivitatea relaţiei $\leq_i$ conduce la $x \leq' x$, adică $\leq'$ este reflexivă.

Considerăm $x, y \in \bigcup_{i\in I} A_i$ astfel încât $x \leq' y$ şi $y \leq' x$. Utilizând iarăşi faptul că mulţimile acestei familii sunt disjuncte obţinem că există două unice elemente $i, j \in I$ astfel încât $x \in A_i$ şi $y \in A_j$. Cuplul $(I; \leq)$ este mulţime total ordonată şi, deci, are loc doar unul din următoarele cazuri: $i < j$, $i = j$ sau $i > j$. Cazul $i < j$ este imposibil deoarece $y \leq' x$ şi similar, cazul $i > j$. Urmează atunci că $i = j$ şi, pe baza antisimetriei relaţiei $\leq_i$ obţinem $x = y$. Deci, $\leq'$ este antisimetrică.

Tranzitivitatea relaţiei $\leq'$ se stabileşte în mod similar. □

Această propoziţie asigură consistenţa conceptului de *sumă ordonată* de mpo.

Definiţia 3.2.4.2. Fie $\mathcal{I} = (I; \leq)$ o mto şi $((A_i; \leq_i)|i \in I)$ o familie disjunctă de mpo. *Suma ordonată* a acestei familii, notată $\sum^o_{i\in I}(A_i; \leq_i)$ sau $\oplus_{i\in I}(A_i; \leq_i)$, este definită ca fiind mulţimea parţial ordonată

$$\sum_{i\in I}^{o}(A_i; \leq_i) = (\bigcup_{i\in I} A_i; \leq'),$$

unde $\leq'$ este relaţia de ordine parţială din Propoziţia 3.2.4.2.

Suma ordonată a două mpo disjuncte $(A_1; \leq_1)$ şi $(A_2; \leq_2)$, considerate în această ordine, va mai fi notată prin

$$(A_1; \leq_1) \oplus (A_2; \leq_2),$$

notaţie care se extinde în mod natural la un număr finit arbitrar de mpo disjuncte.

Dacă A este o mulţime, atunci putem scrie

$$(A_\perp; \leq) = \perp\!\!\!\perp \oplus (A; \iota_A) = \perp\!\!\!\perp \oplus (\bigcup_{a\in A}(\{a\}; \iota_{\{a\}})).$$

Constatăm că operatorul de sumă ordonată este asociativ.

Suma ordonată a mpo din Figura 3.5(a)(b) este reprezentată grafic în Figura 3.5(c).

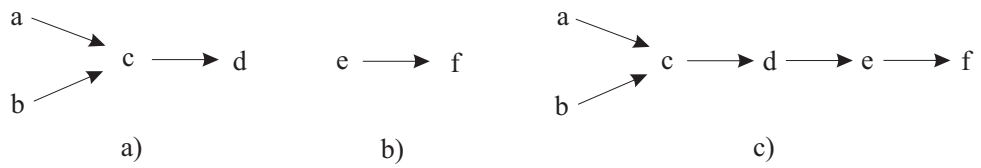

Figura 3.5: Sumă de mpo

Produse de mpo. Fie $n \geq 1$ un număr natural şi $(A_i; \leq_i)$ mpo, $1 \leq i \leq n$. Produsul cartezian $A_1 \times \cdots \times A_n$ poate fi organizat ca mpo considerând relaţia binară $\leq$ dată prin

$$(a_1, \ldots, a_n) \leq (b_1, \ldots, b_n) \quad \Leftrightarrow \quad (\forall 1 \leq i \leq n)(a_i \leq_i b_i),$$

pentru orice $(a_1, \ldots, a_n), (b_1, \ldots, b_n) \in A_1 \times \cdots \times A_n$.

Este trivial de arătat că această relaţie binară este ordine parţială pe mulţimea $A_1 \times \cdots \times A_n$. Ca urmare, $(A_1 \times \cdots \times A_n; \leq)$ este mpo. Vom nota această mpo prin $\times^n_{i=1}(A_i; \leq i)$ sau $(A_1; \leq_1) \times \cdots \times (A_n; \leq_n)$ şi o vom numi *produsul cartezian* al mpo $(A_1; \leq_1), \ldots, (A_n; \leq_n)$.

In Figura 3.6(c) este reprezentat grafic produsul cartezian al mpo din Figura 3.6(a)(b).

Propoziţia 3.2.4.3. Fie $f : A^n_\perp \to B_\perp$, unde $n \geq 1$. Dacă f este monotonă atunci $f(\perp_A, \ldots, \perp_A) = \perp_B$ sau există $c \in B_\perp$ astfel încât $f(a) = c$ pentru orice $a \in A^n_\perp$.

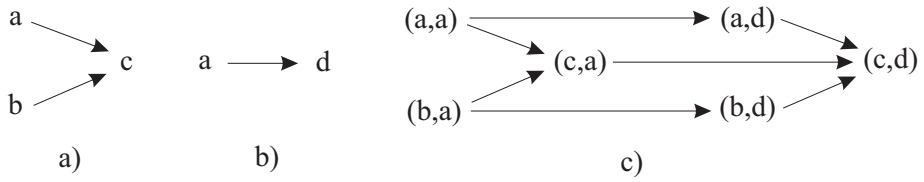

Figura 3.6: Produs cartezian de mpo

Demonstraţie Dacă $f(\perp_A, \ldots, \perp_A) = c$ şi $c \neq \perp_B$, atunci se arată cu uşurinţă (utilizând ordinea parţială pe mpo plate) că $f(a) = c$ pentru orice $a \in A_\perp^n$. □

Reciproca Propoziţiei 3.2.4.3 nu este în general adevărată dacă $n \geq 2$. De exemplu, funcţia $f : \mathbf{R}_\perp^2 \to \mathbf{R}_\perp$ dată prin

$$f(x,y) = \begin{cases} x/y, & \text{dacă } x, y \in \mathbf{R} \text{ şi } y \neq 0 \\ 0, & \text{dacă } x \in \mathbf{R} \text{ şi } y = \perp \\ \perp, & \text{altfel,} \end{cases}$$

unde x/y este împărţirea uzuală pe $\mathbf{R}$, satisface $f(\perp, \perp) = \perp$ dar f nu este monotonă deoarece $f(1, \perp) = 0 \not\leq 1 = f(1,1)$.

Propoziţia 3.2.4.3 ne spune că o funcţie ne-constantă $f : A_\perp^n \to B_\perp$ trebuie să satisfacă în mod necesar $f(\perp_A, \ldots, \perp_A) = \perp_B$ pentru a putea fi monotonă. Evident, aceasta nu este suficient.

Definiţia 3.2.4.3. O funcţie $f : A_\perp^n \to B_\perp$, unde A şi B sunt două mulţimi arbitrare şi $n \geq 1$, este numită *extinsă natural* dacă $f(a_1, \ldots, a_n) = \perp$ ori de câte ori există $1 \leq i \leq n$ astfel încât $a_i = \perp_A$, pentru orice $(a_1, \ldots, a_n) \in A^n$.

Propoziţia 3.2.4.4. Orice funcţie extinsă natural este monotonă.

Demonstraţie Urmează direct de la definiţia ordinii parţiale pe mpo plate. □

Oricărei funcţii $f : A^n \to B$, unde A şi B sunt două mulţimi arbitrare şi $n \geq 1$, i se poate asocia în mod unic o *extensie naturală* $f' : A_\perp^n \to B_\perp$ dată prin

$$f'(a_1, \ldots, a_n) = \begin{cases} f(a_1, \ldots, a_n), & \text{dacă } (\forall 1 \leq i \leq n)(a_i \neq \perp_A) \\ \perp_B, & \text{altfel} \end{cases}$$

pentru orice $(a_1, \ldots, a_n) \in A^n$.

Exemplul 3.2.4.1.

(1) Funcţia $f : \mathbf{R}_\perp^2 \to \mathbf{R}_\perp$ dată prin

$$f(x,y) = \begin{cases} x/y, & \text{dacă } (x, y \in \mathbf{R} \ \wedge \ y \neq 0) \\ \perp_{\mathbf{R}}, & \text{altfel,} \end{cases}$$

este extinsă natural şi, deci, este monotonă.

(2) Predicatul de egalitate $=: A^2 \to \{0,1\}$ poate fi extins la $A_\perp$ astfel:

1. natural, ceea ce va conduce la faptul că această extensie este funcţie monotonă. Uzual, extensia naturală a acestui predicat se notează tot prin $=$ sau prin $=_w$ şi se mai numeşte *predicatul de egalitate slabă*;
2. $=_s: A_\perp^2 \to \{0,1\}_\perp$ prin

$$=_s (a,b) \stackrel{def}{=} \begin{cases} 1, & \text{dacă } ((a,b \in A \ \wedge \ a=b) \ \vee (a=b=\perp_A)) \\ 0, & \text{altfel,} \end{cases}$$

pentru orice $a, b \in A_\perp$. $=_s$ se mai numeşte şi *predicatul de egalitate tare*.

Este uşor de văzut că $=_s$ nu este o funcţie monotonă deoarece

$$=_s (\perp, a) \not\leq =_s (a,a),$$

pentru orice $a \in A$ (presupunând că A este nevidă).

(3) Fie funcţia $if\_then\_else : \{0,1\} \times A \times A \to A$ dată prin

$$if\_then\_else(b,x,y) = \begin{cases} x, & \text{dacă } b=1 \\ y, & \text{dacă } b=0, \end{cases}$$

pentru orice $(b,x,y) \in \{0,1\} \times A \times A$. Uzual, $if\_then\_else(b,x,y)$ se notează prin $if\ b\ then\ x\ else\ y$.

Extensia acestei funcţii la $\{0,1\}_\perp \times A_\perp \times A_\perp$, notată tot prin if_then_else şi dată prin

$$if\_then\_else(b,x,y) = \begin{cases} x, & \text{dacă } b=1 \\ y, & \text{dacă } b=0 \\ \perp_A, & \text{dacă } b=\perp_{\{0,1\}} \end{cases}$$

este funcţie monotonă chiar dacă nu este extensie naturală. Aceasta se poate arăta cu uşurinţă luând în discuţie cele 3 cazuri posibile pentru b.

Produsul cartezian al unei familii finite de mulţimi este generalizat prin intermediul produsului direct la familii arbitrare de mulţimi. Aceaşi generalizare se poate aplica şi pentru mpo. Inainte însă de a defini *produsul direct* al unei familii $((A_i;\leq_i)|i \in I)$ de mpo reamintim că produsul direct al familiei $(A_i|i \in I)$ este mulţimea tuturor aplicaţiilor f de la I la $\bigcup_{i\in I} A_i$ cu proprietatea $f(i) \in A_i$ pentru orice $i \in I$. Următoarea propoziţie, a cărei demonstraţie este imediată, va asigura consistenţa noţiunii de produs direct de mpo.

Propoziţia 3.2.4.5. Fie $((A_i;\leq_i)|i \in I)$ o familie de mpo. Atunci, relaţia binară $\leq'$ pe $\prod_{i\in I} A_i$ dată prin

$$f \leq' g \ \Leftrightarrow \ (\forall i \in I)(f(i) \leq_i g(i)),$$

este relaţie de ordine parţială.

Definiţia 3.2.4.4. Fie $((A_i;\leq_i)|i\in I)$ o familie de mulţimi parţial ordonate. *Produsul direct* al familiei $((A_i;\leq_i)|i\in I)$, notat $\prod^o_{i\in I}(A_i;\leq_i)$ sau $\otimes_{i\in I}(A_i;\leq_i)$, este definit ca fiind mulţimea parţial ordonată

$$\prod_{i\in I}^{o}(A_i;\leq_i)=(\prod_{i\in I}A_i;\leq'),$$

unde $\leq'$ este relaţia de ordine parţială din Propoziţia 3.2.4.5.

Cazul familiilor nedisjuncte. Aşa cum observăm, proprietatea de disjunctivitate a familiei $((A_i;\leq_i)|i\in I)$ este esenţială în definiţia reuniunii şi a sumei ordonate. In cazul în care această proprietate nu este asigurată se poate recurge la diverse variante prin care să se poată introduce conceptul de reuniune şi sumă ordonată de mpo. Una din variantele des întâlnite este de a considera $A_i\times\{i\}$, pentru orice $i\in I$, şi de a defini reuniunea sau produsul având în vedere aceste mulţimi. Astfel, definim *reuniunea disjunctă* a familiei $((A_i;\leq_i)|i\in I)$, notată $\biguplus_{i\in I}(A_i;\leq_i)$, prin

$$\biguplus_{i\in I}(A_i;\leq_i)=(A;\leq'),$$

unde $A=\bigcup_{i\in I}(A_i\times\{i\})$ iar relaţia $\leq'$ este definită prin

$$x\leq' y\ \Leftrightarrow\ (\exists i\in I)(x=(a,i)\ \wedge\ y=(b,i)\ \wedge\ a\leq_i b),$$

pentru orice $x,y\in A$ (şi în acest caz se arată cu uşurinţă că $\leq'$ este ordine parţială pe $\bigcup_{i\in I}(A_i\times\{i\})$).

In manieră similară cititorul poate defini *suma ordonată disjunctă*, notată prin $\sum^{od}_{i\in I}(A_i;\leq_i)$.

3.3 Latici

Structurile laticeale, ce îşi au originea în studiile lui George Boole asupra logicii [13], sunt cazuri particulare de mulţimi parţial ordonate ce apar în cele mai variate domenii: matematică, fizică, informatică, biologie, geologie etc. Cunoaşterea proprietăţilor de bază ale acestora este nu numai benefică dar şi necesară.

3.3.1 Laticea ca mulţime parţial ordonată

Laticile pot fi introduse în două moduri (echivalente): ca mulţimi parţial ordonate sau ca algebre. In această secţiune vom discuta prima variantă.

Definiţia 3.3.1.1. Fie $M=(A;\leq)$ o mpo.

(1) M este numită *inf-semilatice* dacă pentru orice două elemente $a,b\in A$ există $inf(\{a,b\})$.

(2) M este numită *sup-semilatice* dacă pentru orice două elemente $a, b \in A$ există $sup(\{a, b\})$.

(3) M este numită *latice* dacă este atât inf-semilatice cât şi sup-semilatice.

Observăm că perechea $(\emptyset; \emptyset)$ este atât inf-semilatice cât şi sup-semilatice şi latice. Ea va fi numită *laticea vidă*. De asemenea, observăm că sup-semilaticile nevide sunt mpo dirijate, iar inf-semilaticile nevide sunt mpo filtrate.

Exemplul 3.3.1.1.

(1) Orice lanţ este latice.

(2) Dacă A este o mulţime arbitrară, atunci $(\mathcal{P}(A); \subseteq)$ este latice.

(3) Orice familie de mulţimi ce este închisă la reuniune şi intersecţie este latice (în raport cu incluziunea).

Pentru latici se utilizează în mod uzual reprezentarea grafică prin diagrame Hasse.

Propoziţia 3.3.1.1. Fie $M = (A; \leq)$ o mpo. M este inf-semilatice dacă şi numai dacă pentru orice submulţime finită şi nevidă $B \subseteq A$ există $inf(B)$.

Demonstraţie Este clar că dacă pentru orice submulţime finită şi nevidă $B \subseteq A$ există $inf(B)$ atunci M este inf-semilatice.

Reciproca se obţine prin aplicarea repetată a Corolarului 3.2.3.2. □

Propoziţia 3.3.1.1 poate fi dualizată pentru cazul sup-semilaticilor şi, împreună conduc la faptul că o mpo M este latice dacă şi numai dacă pentru orice submulţime finită şi nevidă $B \subseteq A$ există $inf(B)$ şi $sup(B)$.

O generalizare naturală a conceptului de latice, prin prisma Propoziţiei 3.3.1.1, este următoarea.

Definiţia 3.3.1.2. Fie $M = (A, \leq)$ o mpo.

(1) M este numită *inf-semilatice completă* dacă pentru orice submulţime nevidă $B \subseteq A$ există infimum.

(2) M este numită *sup-semilatice completă* dacă pentru orice submulţime nevidă $B \subseteq A$ există supremum.

(3) M este numită *latice completă* dacă pentru orice submulţime nevidă $B \subseteq A$ există infimum şi supremum.

Şi în acest caz observăm că structura $(\emptyset; \emptyset)$ este atât inf-semilatice completă cât şi sup-semilatice completă şi latice completă.

Observaţia 3.3.1.1. Fie $M = (A; \leq)$ o mpo nevidă. Dacă M este inf-semilatice completă atunci ea are cel mai mic element, dacă M este sup-semilatice completă atunci ea are cel mai mare element, iar dacă M este latice completă atunci ea are atât cel mai mic cât şi cel mai mare element. In mod uzual, in cadrul laticilor, cel mai mic element se mai notează prin 0 (atunci când există), iar cel mai mare element se mai notează prin 1 (atunci când există).

Următoarea proprietate urmează direct de la definiţii şi Propoziţia 3.3.1.1.

Propoziţia 3.3.1.2. Orice inf-semilatice (sup-semilatice, latice) finită este inf-semilatice (sup-semilatice, latice) completă.

Utilizând echivalenţele din Propoziţia 3.2.3.2, demonstraţia următoarei teoreme este imediată.

Teorema 3.3.1.1. Fie $M = (A; \leq)$ o mpo nevidă. Atunci, următoarele afirmaţii sunt echivalente:

(1) M este latice completă;

(2) Pentru orice submulţime $B \subseteq A$ există $inf(B)$;

(3) M este inf-semilatice completă ce are cel mai mare element.

Evident, Teorema 3.3.1.1 poate fi dualizată înlocuind “inf” cu “sup” şi “cel mai mare element” cu “cel mai mic element”.

Exemplul 3.3.1.2. (Laticea completă a submulţimilor unei mulţimi)
Fie A o mulţime şi $S \subseteq \mathcal{P}(A)$ un sistem peste A astfel încât $A \in S$ şi S este închis la intersecţii de familii nevide cu elemente din S. In baza Teoremei 3.3.1.1, $(S; \subseteq)$ este latice completă. In particular, $(\mathcal{P}(A); \subseteq)$ este latice completă, numită *laticea* (*completă a*) *submulţimilor mulţimii* A.

Exemplul 3.3.1.3. (Laticea completă a relaţiilor de echivalenţă)
Mulţimea relaţiilor de echivalenţă $E(A)$ peste o mulţime A, cu incluziunea, formează latice completă. In adevăr, fie $(\rho_i | i \in I)$ o familie nevidă de relaţii de echivalenţă pe A. Este trivial de verificat că $\bigcap_{i \in I} \rho_i$ este relaţie de echivalenţă pe A. In plus, ea este $inf(\rho_i | i \in I)$.

Cum A^2 este cea mai mare relaţie de echivalenţă pe A, în baza Teoremei 3.3.1.1, $(E(A), \subseteq)$ este latice completă.

Evident, este interesant de ştiut cine este $sup(\rho_i | i \in I)$. Putem răspunde foarte simplu prin: $sup(\rho_i | i \in I)$ este închiderea la echivalenţă a relaţiei $\bigcup_{i \in I} \rho_i$. Dacă apelăm la modul de construcţie a închiderii, obţinem relaţia θ dată prin

$$x \, \theta \, y \;\Leftrightarrow\; (\exists x_0, \ldots, x_n)(x_0 = x \wedge y = x_n \wedge (\forall 1 \leq j \leq n)(\exists i_j \in I)(x_{j-1} \, \rho_{i_j} \, x_j)),$$

pentru orice $x, y \in A$, care este $sup(\rho_i | i \in I)$.

Diagrama din Figura 3.7 prezintă schematic relaţia dintre inf-semilatici complete, sup-semilatici complete şi latici complete.

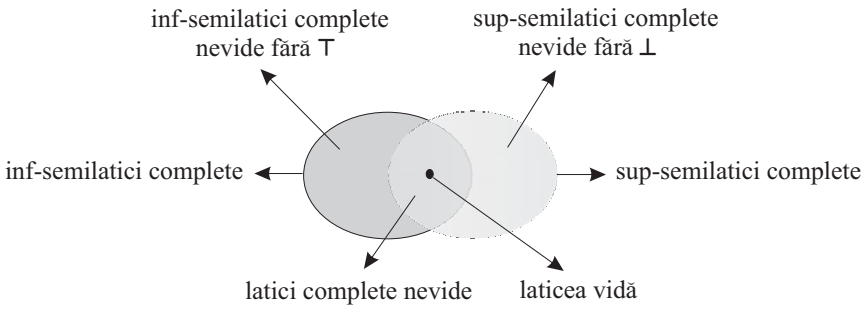

Figura 3.7: Inf-semilatici complete, sup-semilatici complete şi latici complete

3.3.2 Laticea ca structură algebrică

Fie $M = (A; \leq)$ o latice. Definiţia laticii cât şi faptul că supremum este unic asigură consistenţa definiţiei unei operaţii binare pe A, notată $\vee$ [3] şi dată prin

$$\vee(a, b) = sup(\{a, b\}),$$

pentru orice $a, b \in A$. In mod similar putem defini operaţia binară $\wedge$ dată prin

$$\wedge(a, b) = inf(\{a, b\}),$$

pentru orice $a, b \in A$. Este uşor de văzut că aceste două operaţii satisfac următoarele proprietăţi:

- $a \vee b = b \vee a$ şi $a \wedge b = b \wedge a$ (comutatitivate)
- $a \vee (b \vee c) = (a \vee b) \vee c$ şi $a \wedge (b \wedge c) = (a \wedge b) \wedge c$ (asociativitate)
- $a \wedge (a \vee b) = a$ şi $a \vee (a \wedge b) = a$ (absorbţie)

pentru orice $a, b \in A$. In plus,

- $a \leq b \;\Leftrightarrow\; a \vee b = b \;\Leftrightarrow\; a \wedge b = a,$

pentru orice $a, b \in A$.

Proprietatea de absorbţie conduce la idempotenţa operaţiilor $\vee$ şi $\wedge$. In adevăr,

$$a \vee a = a \vee (a \wedge (a \vee a)) = a,$$

pentru orice a (prima egalitate se obţine înlocuind al doilea a din membrul stâng al egalităţii prin $a \wedge (a \vee a)$, iar a doua egalitate se obţine aplicând absorbţia în forma $a \vee (a \wedge b) = a$). In mod similar se arată că are loc $a \wedge a = a$.

Este demn de remarcat că pentru orice triplet $(A; \vee, \wedge)$, unde A este o mulţime iar $\vee$ şi $\wedge$ sunt două operaţii binare pe A ce satisfac proprietăţile de comutativitate, asociativitate şi absorbţie (ca mai sus), relaţia binară $\leq$ dată prin

$$a \leq b \quad \Leftrightarrow \quad a \vee b = b,$$

pentru orice $a, b \in A$, structurează A ca o latice. Teorema de mai jos ne arată aceasta.

[3] Această notaţie nu trebuie confundată cu notaţia pentru supremum introdusă în Secţiunea 3.2. De fapt, am evitat să utilizăm acea notaţie până acum întocmai pentru a nu crea confuzii. Această remarcă va fi valabilă şi pentru notaţia $\wedge$ ce urmează a fi introdusă ca notaţie de operaţie binară.

Teorema 3.3.2.1.

(1) Fie $M = (A; \leq)$ o latice. Atunci, structura $M^a = (A; \vee, \wedge)$, unde $\vee$ şi $\wedge$ sunt operaţiile binare pe A date prin

$$\vee(a, b) = sup(\{a, b\}) \text{ şi } \wedge(a, b) = inf(\{a, b\}),$$

pentru orice $a, b \in A$, verifică proprietăţile de comutativitate, asociativitate şi absorbţie. In plus,

$$a \leq b \quad \Leftrightarrow \quad a \vee b = b \quad \Leftrightarrow \quad a \wedge b = a,$$

pentru orice $a, b \in A$.

(2) Fie $M = (A; \vee, \wedge)$ o structură formată dintr-o mulţime A şi două operaţii binare pe A ce verifică proprietăţile de comutativitate, asociativitate şi absorbţie. Atunci, cuplul $M^o = (A; \leq)$, unde $\leq$ este relaţia binară dată prin

$$a \leq b \quad \Leftrightarrow \quad a \vee b = b,$$

pentru orice $a, b \in A$, este latice. In plus, $sup(\{a, b\}) = a \vee b$ şi $inf(\{a, b\}) = a \wedge b$, pentru orice $a, b \in A$.

(3) Fie $M = (A; \leq)$ o latice. Atunci, $(M^a)^o = M$.

(4) Fie $M = (A; \vee, \wedge)$ o structură ca la (2). Atunci, $(M^o)^a = M$.

Demonstraţie Lăsăm (1), (3) şi (4) în seama cititorului şi ne vom ocupa de (2).

Reflexivitatea relaţiei binară $\leq$ decurge imediat de la proprietatea de idempotenţă a operaţiei $\vee$ (proprietate indusă de absorbţie). In adevăr, relaţia $a \vee a = a$ ne spune că $a \leq a$, pentru orice a.

Fie $a, b \in A$ astfel încât $a \leq b$ şi $b \leq a$. Atunci, $a = a \vee b$ şi $b = b \vee a$, iar comutativitatea conduce la

$$a = a \vee b = b \vee a = b,$$

care stabileşte antisimetria relaţiei $\leq$.

Fie $a, b, c \in A$ astfel încât $a \leq b$ şi $b \leq c$. Atunci, $a = a \vee b$ şi $b = b \vee c$, iar asociativitate conduce la

$$a \vee c = a \vee (b \vee c) = (a \vee b) \vee c = b \vee c = c,$$

care ne arată că $\leq$ este tranzitivă.

Ca urmare, $\leq$ este ordine parţială pe A.

Fie $a, b \in A$. Vom arăta că există $sup(\{a, b\})$ şi $inf(\{a, b\})$. Fie $c = a \vee b$ (acest element există deoarece $\vee$ este operaţie binară pe A, deci definită pentru orice două elemente din A). Arătăm că c este majorant pentru $\{a, b\}$. In adevăr,

$$a \vee c = a \vee (a \vee b) = (a \vee a) \vee b = a \vee b = c,$$

ceea ce ne arată că are loc $a \leq c$. In mod similar obţinem $b \leq c$, ceea ce conduce la faptul că c este majorant pentru $\{a, b\}$.

Dacă d este un alt majorant pentru $\{a, b\}$, atunci $d = a \vee d$, $d = b \vee d$ şi

$$d = a \vee d = a \vee (b \vee d) = (a \vee b) \vee d = c \vee d,$$

care ne arată că $c \leq d$. Deci, $c = sup(\{a, b\})$. In mod similar se obţine $inf(\{a, b\}) = a \wedge b$. □

Ca urmare, o latice poate fi vazută atât ca mulţime parţial ordonată cât şi ca structură algebrică. In plus, atunci când lucrăm cu latici putem folosi $\vee$ ($\wedge$) atât pentru a specifica supremum (infimum) cât şi ca operaţie binară.

Privind laticea ca algebră universală putem recurge la specificări ale acesteia prin intermediul diagramelor operaţiilor $\vee$ şi $\wedge$. Cum aceste operaţii sunt comutative şi idempotente, tabelele lor pot fi reduse la jumătate, renunţând şi la diagonală. Ca urmare, ambele tabele pot fi cumulare în unul singur (fără diagonală). De exemplu, tabelul

$\vee \wedge$	0	a	b	1
0		0	0	0
a	a		0	a
b	b	1		b
1	1	1	1	

specifică laticea ale cărei operaţii sunt date prin $x \vee x = x = x \wedge x$, pentru orice x, $0 \vee a = a$, $0 \wedge a = 0$ etc.

Vom prezenta în continuare o serie de proprietăţi simple ce au loc în latici.

Propoziţia 3.3.2.1. Fie $M = (A; \leq)$ o latice şi $a, b_i \in A$ pentru orice $1 \leq i \leq n$ cu $n \geq 1$ număr natural. Dacă $a \leq b_i$ pentru orice i, atunci $a \leq \bigwedge_{i=1}^{n} b_i$. Similar, dacă $b_i \leq a$ pentru orice i, atunci $\bigvee_{i=1}^{n} b_i \leq a$.

Demonstraţie Dacă $a \leq b_i$ atunci a este minorant al lui b_i, pentru orice i. Ca urmare, $a \leq \bigwedge_{i=1}^{n} b_i$ deoarece $\bigwedge_{i=1}^{n} b_i$ este cel mai mare minorant al mulţimii $\{b_i | 1 \leq i \leq n\}$. Similar pentru cealaltă proprietate (sau prin dualizare). □

Propoziţia 3.3.2.2. (Proprietăţi de idempotenţă)
Fie $M = (A; \leq)$ o latice. Atunci, pentru orice $a \in A$ şi $n \geq 1$ număr natural au loc proprietăţile de idenpotenţă $\bigvee_{i=1}^{n} a = a$ şi $\bigwedge_{i=1}^{n} a = a$.

Demonstraţie Pentru $n = 2$ proprietatea a fost deja demonstrată. Cazul general se poate obţine cu uşurinţă prin inducţie matematică. □

Propoziţia 3.3.2.3. Fie $M = (A; \leq)$ o latice şi $a_i, b_i \in A$ pentru orice $1 \leq i \leq n$ cu $n \geq 1$ număr natural. Dacă $a_i \leq b_i$ pentru orice i, atunci $\bigvee_{i=1}^{n} a_i \leq \bigvee_{i=1}^{n} b_i$ şi $\bigwedge_{i=1}^{n} a_i \leq \bigwedge_{i=1}^{n} b_i$.

Demonstraţie Vom demonstra propoziţia pentru $n = 2$.

Inegalitatea $a_1 \leq b_1$ conduce la $a_1 \vee b_1 = b_1$, iar $a_2 \leq b_2$ conduce la $a_2 \vee b_2 = b_2$. Atunci,

$$(a_1 \vee a_2) \vee (b_1 \vee b_2) = (a_1 \vee b_1) \vee (a_2 \vee b_2) = b_1 \vee b_2,$$

care ne arată că $a_1 \vee a_2 \leq b_1 \vee b_2$ (s-a utilizat asociativitatea operatorului $\vee$).

Cea de a doua inegalitate din enunţul propoziţiei se obţine în mod similar utilizând $a_1 \wedge b_1 = a_1$ şi $a_2 \wedge b_2 = a_2$. □

Propoziţia 3.3.2.4. (Proprietatea min-max)
Fie $M = (A; \leq)$ o latice, şi $a_{ij} \in A$ pentru orice $1 \leq i \leq m$ şi $1 \leq j \leq n$, unde $m, n \geq 1$. Atunci, are loc:

$$\bigvee_{j=1}^{n}(\bigwedge_{i=1}^{m} a_{ij}) \leq \bigwedge_{i=1}^{m}(\bigvee_{j=1}^{n} a_{ij}).$$

Demonstraţie Vom demonstra proprietatea, ca exemplu, pentru $m = 2$ şi $n = 3$ (cazul general fiind similar acestuia). Avem de arătat că are loc

$$(a_{11} \wedge a_{21}) \vee (a_{12} \wedge a_{22}) \vee (a_{13} \wedge a_{23}) \leq (a_{11} \vee a_{12} \vee a_{13}) \wedge (a_{21} \vee a_{22} \vee a_{23}),$$

ceea ce poate fi redus la a arăta că are loc

$$(a_{11} \wedge a_{21}) \vee (a_{12} \wedge a_{22}) \vee (a_{13} \wedge a_{23}) \leq a_{11} \vee a_{12} \vee a_{13}$$

şi

$$(a_{11} \wedge a_{21}) \vee (a_{12} \wedge a_{22}) \vee (a_{13} \wedge a_{23}) \leq a_{21} \vee a_{22} \vee a_{23}$$

(conform Propoziţiei 3.3.2.1). Prima inegalitate este indusă de $a_{11} \wedge a_{21} \leq a_{11}$, $a_{12} \wedge a_{22} \leq a_{12}$ şi $a_{13} \wedge a_{23} \leq a_{13}$ prin aplicarea Propoziţiei 3.3.2.3.

In mod similar se obţine şi cea de a doua inegalitate. □

Dacă gândim elementele a_{ij} din Propoziţia 3.3.2.4 ca fiind distribuite într-o matrice (cu notaţia uzuală), atunci proprietatea min-max ne spune că supremum infimurilor calculate pe coloane este mai mic cel mult egal cu infimum supremurilor calculate pe linii.

Corolarul 3.3.2.1. (Inegalităţi de distributivitate)
In orice latice $M = (A; \leq)$ au loc proprietăţile

$$a \vee (b \wedge c) \leq (a \vee b) \wedge (a \vee c)$$

şi

$$a \wedge (b \vee c) \geq (a \wedge b) \vee (a \wedge c),$$

pentru orice $a, b, c \in A$.

Demonstraţie Este suficient de demonstrat doar una din aceste proprietăţi deoarece cealaltă se obţine prin dualizare. Ca urmare vom demonstra prima proprietate, care de fapt decurge imediat din Propoziţia 3.3.2.4 considerând $m = n = 2$, $a_{11} = a_{21} = a$, $a_{12} = b$ şi $a_{22} = c$. □

Intr-o latice, elementul $(a \wedge b) \vee (b \wedge c) \vee (c \wedge a)$ se numeşte *mediana* elementelor a, b şi c. El satisface următoarea proprietate:

Corolarul 3.3.2.2. (Proprietatea mediană)
In orice latice $M = (A; \leq)$ are loc proprietatea

$$(a \wedge b) \vee (b \wedge c) \vee (c \wedge a) \leq (a \vee b) \wedge (b \vee c) \wedge (c \vee a),$$

pentru orice $a, b, c \in A$.

Demonstraţie Direct de la Propoziţia 3.3.2.4 alegând $m = n = 3$, $a_{11} = a_{13} = a_{31} = a$, $a_{12} = a_{21} = a_{22} = b$ şi $a_{23} = a_{32} = a_{33} = c$. □

Corolarul 3.3.2.3. (Inegalităţi modulare)
In orice latice $M = (A; \leq)$ au loc proprietăţile

$$a \leq c \;\Rightarrow\; a \vee (b \wedge c) \leq (a \vee b) \wedge c$$

şi

$$a \geq c \;\Rightarrow\; a \wedge (b \vee c) \geq (a \wedge b) \vee c,$$

pentru orice $a, b, c \in A$.

Demonstraţie Vom demonstra doar prima proprietate. Presupunând $a \leq c$ obţinem $a \vee c = c$ şi

$$\begin{aligned} a \vee (b \wedge c) &\leq (a \vee b) \wedge (a \vee c) \\ &= (a \vee b) \wedge c \end{aligned}$$

(inegalitatea urmează de la Corolarul 3.3.2.1). □

3.4 Algebre universale dintr-un punct de vedere elementar

Scopul acestei secţiuni este de a face o simplă dar unitară introducere în teoria structurilor algebrice, cum ar fi semigrupurile, monoizii, grupurile etc. Caracterul unitar va consta în aceea că vom introduce conceptul de algebră universală ca un cuplu format dintr-o mulţime şi un număr arbitrar de operaţii definite pe acea mulţime, după care vom discuta conceptele de subalgebră, homomorfism, congruenţă etc. Toate acestea pot fi apoi translatate cu mare uşurinţă la semigrupuri, monoizi, grupuri etc.

3.4.1 Algebre

Definiţia 3.4.1. Se numeşte *algebră universală* orice cuplu (A, F) format dintr-o mulţime nevidă A şi o mulţime nevidă F de operaţii pe A, fiecare operaţie având asociată o anumită aritate (ce poate fi şi 0).

Terminologia de "algebră universală" va fi simplificată frecvent la cea de "algebră".

Fie (A, F) o algebră. Mulţimea A este numită *mulţimea suport* sau *suportul* algebrei. Dacă suportul este finit atunci vom spune că algebra este *finită*, iar dacă conţine doar un element, atunci vom spune că algebra este *trivială*. Atunci când F este finită, de exemplu $F = \{f_1, \ldots, f_k\}$, vom mai nota algebra prin $(A, f_1, \ldots, f_k)$. Intr-un astfel de caz, n_i va desemna aritatea operaţiei f_i, pentru orice $1 \leq i \leq k$. Operaţiile 0-are vor fi numite *constantele algebrei*.

Exemplul 3.4.1. Anticipăm câteva din structurile algebrice de bază ce pot fi prezentate în termeni de algebră universală. Probabil că cititorul s-a întâlnit deja cu aceste concepte; ele vor fi studiate în detaliu în capitolele următoare.

(1) Un *semigrup* este o algebra $(A, \cdot)$ cu o singură operaţie binară şi asociativă $\cdot$.

(2) Un *monoid* este o algebra $(A, \cdot, e)$, unde $\cdot$ este o operaţie binară asociativă pe A, iar e este o operaţie nulară pe A ce satisface

$$e \cdot a = a \cdot e = a,$$

pentru orice $a \in A$. Constanta e se mai notează şi prin 1_M şi se numeşte *unitatea monoidului*. Este uşor de văzut că ea este unica constantă ce satisface proprietatea de mai sus. In adevăr, dacă am presupune că mai există un element e' ce satisface $e' \cdot a = a \cdot e' = a$, pentru orice $a \in M$, atunci are loc

$$e' = e' \cdot e = e$$

(prima egalitate urmează de la faptul că e este unitate, iar a doua de la faptul că e' este unitate). Deci, unitatea monoidului este unică.

Monoizii sunt adesea întâlniţi şi sub denumirea de *semigrupuri cu unitate*.

(3) Un *grup* este o algebra $(A, \cdot, ', e)$, unde $\cdot$ este o operaţie binară asociativă pe A, e este o operaţie nulară pe A ce satisface proprietatea de la (2), iar $'$ este o operaţie unară pe A pentru care are loc:

$$a \cdot a' = a' \cdot a = e,$$

pentru orice $a \in A$. e se numeşte *unitatea grupului*, este unică, şi se mai notează prin 1_G. Elementul a' se mai numeşte *inversul lui* a; el este unic, ceea ce se poate vedea cu uşurintă. De exemplu, dacă presupunem că ar mai exista încă un element b ce ar satisface $a \cdot b = b \cdot a = e$, atunci

$$b = b \cdot e = b \cdot (a \cdot a') = (b \cdot a) \cdot a' = e \cdot a' = a'.$$

(4) Un semigrup (monoid, grup) pentru care operaţia binară este comutativă se numeşte *semigrup* (*monoid*, *grup*) *comutativ* sau *abelian* [4].

[4] Denumirea de grup "abelian" provide de la numele matematicianului norvegian Niels Abel.

(5) Un *inel* este o algebra $(A, +, -, 0, \cdot)$, unde $+$ şi $\cdot$ sunt operaţii binare, $-$ este o operaţie unară, iar 0 este operaţie nulară pe A ce satisfac proprietăţile:

(i) $(A, +, -, 0)$ este grup comutativ;

(ii) $(A, \cdot)$ este semigrup;

(iii) $a \cdot (b+c) = a \cdot b + a \cdot c$ şi $(b+c) \cdot a = b \cdot a + c \cdot a$, pentru orice $a, b, c \in A$.

Uzual, operaţiile $+$ şi $\cdot$ sunt numite *adunarea* şi, respectiv, *înmulţirea* (dar nu sunt, în mod necesar, operaţiile de adunare şi înmulţire pe mulţimi uzuale ca cea a numerelor întregi). Prima parte a proprietăţii de la (iii) este numită proprietatea de *distributivitate la stânga a înmulţirii faţă de adunare*, iar cea ce a doua parte, proprietatea de *distributivitate la dreapta a înmulţirii faţă de adunare*.

Atunci câd semigrupul $(A, \cdot)$ este comutativ, inelul este numit *inel comutativ*.

(6) Un *inel cu unitate* este o algebra $(A, +, -, 0, \cdot, 1)$ definită ca la (5) dar cu diferenţa că $(A, \cdot, 1)$ este monoid. Dacă acest monoid este comutativ, atunci inelul este numit *inel comutativ cu unitate*.

Atunci când monoizii, grupurile şi inele sunt formate doar din unitate, ele sunt triviale (în cazul inelelor cu unitate, proprietatea de a fi trivial forţează egalitatea între cele două unităţi).

Ca o remarcă generală, atunci când operţia binară ca fi notată multiplicativ (prin $\cdot$, $\circ$ sau $*$), operaţia unară corespunzătoare va fi notată prin $'$, iar cea nulară, prin e sau uneori 1 (eventual indexate). Atunci când operţia binară va fi notată aditiv (prin $+$), operaţia unară corespunzătoare va fi notată prin $-$, iar cea nulară, prin 0. In acest caz, $a - b$ va reprezenta $a + (-b)$.

Definiţia 3.4.2. Spunem că două algebre (A, F) şi (A', F') au *acelaşi tip* dacă există o bijecţie h între F şi F' ce păstrează aritatea operaţiilor, adică aritatea operaţie $f \in F$ este aceeaşi cu aritatea operaţiei $h(f) \in F'$, pentru orice $f \in F$.

Atunci când vom spune că (A, F) şi (A', F') au acelaşi tip vom presupune implicit că $f' \in F'$ este corespondenta operaţiei $f \in F$ printr-o bijecţie h ce păstrează tipul operaţiilor (adică, f' denotă $h(f)$).

In cazul algebrelor cu un număr finit de operaţii, pe lângă convenţia de mai sus, vom adopta şi următoarea convenţie. Dacă $(A, f_1, \ldots, f_k)$ şi $(A', g_1, \ldots, g_k)$ sunt de acelaşi tip, atunci vom presupune că f_i şi g_i au aceeaşi aritate n_i, pentru orice $1 \leq i \leq k$.

Semigrupurile (monoizii, grupurile, inelele) sunt algebre de acelaşi tip. O mică discuţie este necesară în cazul inelelor care au două operaţii binare. Asocierea operaţiilor trebuie înţeleasă în conformitate cu ordinea lor în 5-uplul ce defineşte inelul (de altfel, aşa cum s-a specificat în convenţia adoptată mai sus).

3.4.2 Subalgebre. Ordin

Definiţia 3.4.3. Fie (A, F) şi (A', F') două algebre de acelaşi tip. Spunem că (A', F') este *subalgebră* a algbrei (A, F) dacă $A' \subseteq A$ şi $f' = f|_{A'}$, pentru orice $f \in F$.

Definiţia 3.4.4. Fie (A, F) o algebră şi $X \subseteq A$. Spunem că X este *închisă în* (A, F) dacă are loc:

$$(\forall f \in F)(\forall a_1, \ldots, a_{ar(f)} \in A)(a_1, \ldots, a_{ar(f)} \in X \ \Rightarrow \ f(a_1, \ldots, a_{ar(f)}) \in X)$$

(în cazul $ar(f) = 0$, cerinţa se reduce la $f \in X$).

Observaţia 3.4.1.

(1) Orice subalgebră sau submulţime închisă conţine toate constantele algebrei gazdă.

(2) Dacă (A', F') este subalgebră în (A, F), atunci A' este închisă în (A, F).

Reciproc, dacă $A' \subseteq A$ este o submulţime închisă în (A, F), atunci ea poate fi structurată ca o subalgebră a algebrei (A, F) considerând pentru orice $f \in F$ operaţia $f' : (A')^{ar(f)} \to A'$ dată prin $f' = f|_{A'}$. Mulţimea tuturor acestor operaţii, notată F', împreună cu A' formează o subalgebră a algebrei (A, F).

Este important de menţionat că există o diferenţă între operaţia f' definită ca mai sus şi $f|_{A'}$. Prima are codomeniul A', pe când a doua are codomeniul A. Evident, această diferenţă nu este semnificativă, motiv pentru care vom identifica adesea subalgebrelele unei algebre cu submulţimile închise în acea algebră împreună cu restricţiile operaţilor algebrei gazdă la acea submulţime.

Următoarea propoziţie urmează imediat de la definiţii.

Propoziţia 3.4.1. Intersecţia oricărei familii nevide de submulţimi închise (subalgebre) ale unei algebre (A, F) este submulţime închisă (subalgebră) a algebrei (A, F).

Fie (A, F) o algebră şi $X \subseteq A$. Conform teoriei închiderii, închiderea acestei mulţimii în algebra (A, F), notată $\langle X \rangle_{(A,F)}$, este dată prin

$$\langle X \rangle_{(A,F)} = \bigcup_{m \geq 0} B_m$$

unde:

- $B_0 = X$,
- $B_{m+1} = B_m \cup \bigcup_{f \in F} f(B_m)$, pentru orice $m \geq 0$.

Este clar că $\langle X \rangle_{(A,F)}$ este închisă în (A, F) şi, deci, ea defineşte o subalgebră a algebrei (A, F). Aceasta se numeşte *subalgebra generată de* X, şi este cea mai mică subalgebră ce include X

$$\langle X \rangle_{(A,F)} = \bigcap \{B | X \subseteq B \subseteq A, \ B \text{ închisă în } (A, F)\},$$

fiind intersecţia tuturor subalgebrelor ce includ X.

Apelând iarăşi la teoria închiderii, un element a este în $\langle X\rangle_{(A,F)}$ dacă şi numai dacă există o secvenţă

$$x_1, \ldots, x_n = a$$

astfel încât, pentru orice i, are loc:

- $x_i \in X$, sau
- $(\exists f \in F)(\exists i_1, \ldots, i_{ar(f)} < i)(x_i = f(x_{i_1}, \ldots, x_{i_{ar(f)}}))$.

Definiţia 3.4.5.

(1) Spunem că o algebră (A, F) este *generată de* $X \subseteq A$ dacă are loc $A = \langle X\rangle_{(A,F)}$.

(2) Spunem că (A, F) este *finit generată* dacă există o submulţime finită $X \subseteq A$ astfel încât (A, F) este generată de X.

Dacă X generează algebra (A, F), atunci X se numeşte *mulţime de generatori* a algebrei (A, F), iar elementele ei, *generatori* ai algebrei.

O algebră generată doar de un singur element al ei se numeşte *algebră ciclică*. Particularizând, obţinem conceptele de *semigrup ciclic*, *monoid ciclic* şi *grup ciclic*.

Exemplul 3.4.2.

(1) $(M_2, \circ, e_2)$ este submonoid al monoidului $(M_1, \cdot, e_1)$ dacă au loc proprietăţile $M_2 \subseteq M_1$, $\circ = \cdot|_{M_2}$ şi $e_2 = e_1$.

(2) $(G_2, \circ, '', e_2)$ este subgrup al grupului $(G_1, \cdot, ', e_1)$ dacă au loc proprietăţile $G_2 \subseteq G_1$, $\circ = \cdot|_{G_2}$, $'' = ' |_{G_2}$ şi $e_2 = e_1$.

Este uşor de văzut că putem renunţa la cerinţa "$e_2 = e_1$" deoarece aceasta se obţine combinând primele 3 cerinţe:

$$e_2 = a \circ a'' = a \cdot a' = e_1,$$

pentru orice $a \in G_2$.

Definiţia 3.4.6. Fie (A, F) o algebră.

(1) Spunem că (A, F) *este de ordin* ∞ sau că *are ordinul* ∞ dacă A este mulţime infinită. Altfel, spunem că algebra *este de ordin finit* sau că *are ordinul finit* sau că *este de ordin* $|A|$ sau că *are ordinul* $|A|$.

(2) *Ordinul* unui element $a \in A$, notat $ord_{(A,F)}(a)$, este definit ca fiind ordinul subalgebrei generate de a.

Următoarea propoziţie urmează imediat de la definiţii.

Propoziţia 3.4.2. Intersecţia oricărei familii nevide de subalgebre ale unei algebre (A, F) este subalgebră a algebrei (A, F).

Vom încheia subsecţiunea printr-un rezultat important ce poate fi utilizat în demonstrarea de proprietăţi în algebre.

Teorema 3.4.1. (Principiul inducţiei structurale pentru algebre)
Fie (A, F) o algebră generată de o parte a sa X. Dacă P este o proprietate referitoare la elementele algebrei (A, F) astfel încât:

(1) $P(x)$, pentru orice $x \in X$;

(2) $(P(a_1) \wedge \cdots \wedge P(a_{ar(f)}) \Rightarrow P(f(a_1, \ldots, a_{ar(f)})))$, pentru orice $f \in F$ şi $a_1, \ldots, a_{ar(f)} \in A$.

atunci $P(a)$, pentru orice $a \in A$.

Demonstraţie Direct de la principiul inducţiei structurale pentru nulţimi inductiv definite. □

Exemplul 3.4.3.

(1) In cazul semigrupurilor, principiul inducţiei structurale capătă următoarea formă. Fie $(S, \cdot)$ un semigrup generat de $X \subseteq S$ şi P o proprietate referitoare la elementele lui. Dacă

(1) $P(x)$, pentru orice $x \in X$;
(2) $(\forall a, b \in S)(P(a) \wedge P(b) \Rightarrow P(a \cdot b))$,

atunci $P(a)$, pentru orice $a \in S$.

(2) In cazul grupurilor, principiul inducţiei structurale poate fi pus în următoarea formă. Fie $(G, \cdot, ', e)$ un grup generat de $X \subseteq S$ şi P o proprietate referitoare la elementele lui. Dacă

(1) $P(x)$, pentru orice $x \in X$, şi $P(e)$;
(2) $(\forall a, b \in S)(P(a) \wedge P(b) \Rightarrow P(a \cdot b) \wedge P(a'))$,

atunci $P(a)$, pentru orice $a \in G$.

Diferenţa faţă de forma din teoremă constă în aceea că se verifică $P(e)$ la pasul (1) şi nu la pasul (2).

3.4.3 Homomorfisme şi congruenţe

Definiţia 3.4.7. Fie (A, F) şi (A', F') două algebre de acelaşi tip. Un *homomorfism* de la (A, F) la (A', F') este o aplicaţie $h : A \to A'$ ce satisface

$$h(f(a_1, \ldots, a_{ar(f)})) = f'(h(a_1), \ldots, h(a_{ar(f)})),$$

pentru orice $f \in F$ şi $a_1, \ldots, a_{ar(f)} \in A$ (atunci când f este o constantă, proprietatea de homomorfism se reduce $h(f) = f'$).

Exemplul 3.4.4.

(1) Un homorfism h de la semigrupul $(S_1, \cdot)$ la semigrupul $(S_2, *)$ satisface

$$h(a \cdot b) = h(a) * h(b),$$

pentru orice $a, b \in S$.

(2) Un homorfism h de la monoidul $(M_1, \cdot, 1_{M_1})$ la monoidul $(M_2, *, 1_{M_2})$ satisface

(i) $h(a \cdot b) = h(a) * h(b)$;

(ii) $h(1_{M_1}) = h(1_{M_2})$,

pentru orice $a, b \in M_1$.

(3) Un homorfism h de la grupul $(G_1, \cdot, ', 1_{G_1})$ la grupul $(G_2, *, '', 1_{G_2})$ satisface

(i) $h(a \cdot b) = h(a) * h(b)$;

(ii) $h(a') = (h(a))''$;

(iii) $h(1_{G_1}) = 1_{G_2}$,

pentru orice $a, b \in G_1$.

Proprietăţile grupului fac ca cea de a doua cerinţă să nu fie necesară. In adevăr, relaţia

$$h(a \cdot a') = h(1_{G_1}) = h(a' \cdot a)$$

conduce la

$$h(a) * h(a') = 1_{G_2} = h(a') * h(a)$$

de la care urmează $h(a') = (h(a))''$, pentru orice $a \in G_1$.

Interesant este că nici cea de a treia cerinţă nu este necesară deoarece

$$h(1_{G_1}) = h(1_{G_1} \cdot 1_{G_1}) = h(1_{G_1}) * h(1_{G_1}),$$

de la care obţinem $h(1_{G_1}) = 1_{G_2}$ dacă aplicăm $(h(1_{G_1}))''$.

Uzual, homomorfismele injective sunt numite *monomorfisme*, homomorfismele surjective sunt numite *epimorfisme*, iar homorfismele bijective sunt numite *izomorfisme*. Un homorfism de la o algebră (A, F) la ea însăşi este numit *endomorfism*. Mulţimea tuturor endomorfismelor algebrei (A, F) se notează prin $End(A, F)$. Endomorfismele care sunt şi izomorfisme se mai numesc *automorfisme*. Mulţimea tuturor automorfismelor algebrei (A, F) se notează prin $Aut(A, F)$.

Următoarele propoziţii urmează imediat de la definiţii.

Propoziţia 3.4.3.

(1) Compunere de homomorfisme este homomorfism.

(2) Dacă h este izomorfism de la (A, F) la (A', F'), atunci h^{-1} este izomorfism de la (A', F') la (A, F).

(3) $End(A, F)$, cu compunerea funcţiilor, formează monoid.

(4) $Aut(A, F)$, cu compunerea funcţiilor, formează grup.

Grupul $Aut(A, F)$ se numeşte *grupul automorfismelor* algebrei (A, F).

Propoziţia 3.4.4. Fie (A, F) şi (A', F') algebre de acelaşi tip şi $h : A \to B$ un homomorfism.

(1) Dacă $C \subseteq A$ este închisă în (A, F), atunci $h(C)$ este închisă în (A', F').

(2) Dacă $C \subseteq A'$ este închisă în (A', F'), atunci $h^{-1}(C)$ este închisă în (A, F).

In baza acestei propoziţii este clar că imaginea unei subalgebre printr-un homomorfism este subalgebră. Similar, imaginea inversă a unei subalgebre printr-un homomorfism este subalgebră.

Propoziţia 3.4.5. Fie (A, F) şi (A', F') algebre de acelaşi tip şi $h_1, h_2 : A \to A'$ două homomorfisme. Dacă (A, F') este generată de X şi $h_1(x) = h_2(x)$ pentru orice $x \in X$, atunci $h_1 = h_2$.

Demonstraţie Vom demonstra propoziţia prin inducţie structurală. Conform ipotezei, ceea ce ne rămâne de arătat este că dacă $f \in F$ şi $h_1(a_i) = h_2(a_i)$, pentru orice $a_i \in A$ şi $1 \leq i \leq ar(f)$, atunci

$$h_1(f(a_1, \ldots, a_{ar(f)}) = h_2(f(a_1, \ldots, a_{ar(f)}).$$

Aceasta urmează imediat de la ipoteza inductivă şi proprietatea de homomorfism a funcţiilor h_1 şi h_2. □

Deci, două homomorfisme ce coincid pe o mulţime de generatori a unei algebre vor coincide pe întreaga algebră.

Definiţia 3.4.8. Fie (A, F) o algebră. O *congruenţă* în (A, F) este o relaţie de echivalenţă ρ pe A ce satisface

$$f(a_1, \ldots, a_{ar(f)}) \, \rho \, f(a'_1, \ldots, a'_{ar(f)}),$$

pentru orice $f \in F$ de aritate $ar(f) > 0$ şi orice $a_1, a'_i, \ldots, a_{ar(f)}, a'_{ar(f)} \in A$ pentru care are loc $a_i \, \rho \, a'_i$, pentru orice $1 \leq i \leq ar(f)$.

Proprietatea din Definiţia 3.4.8 se mai numeşte *proprietatea de compatibilitate a relaţiei de echivalenţă cu operaţiile algebrei.*

Dacă (A, F) este o algebră, atunci $Con(A, F)$ va reprezenta mulţimea tuturor congruenţelor ei.

Exemplul 3.4.5. In cazul semigrupurilor, o congruenţă ρ pe un semigrup $(S, \cdot)$ satisface

$$(a \cdot b)\, \rho\, (a' \cdot b'),$$

pentru orice a, b, a', b' pentru care are loc $a\, \rho\, a'$ şi $b\, \rho\, b'$.

Se mai spune că ρ este o relaţie de echivalenţă *compabililă la stânga* şi *la dreapta* cu operatorul "$\cdot$". Aceasta pentru că relaţia de mai sus este echivalentă cu proprietatea

$$a\, \rho\, b \;\Rightarrow\; (\forall c \in S)((a \cdot c)\, \rho\, (b \cdot c) \;\wedge\; (c \cdot a)\, \rho\, (c \cdot b)).$$

Fie (A, F) o algebră şi $\rho \in Con(A, F)$. Mulţimea cât

$$A/\rho = \{[a]_\rho | a \in A\}$$

poate fi structurată ca o algebră de acelaşi tip cu (A, F) într-un mod foarte natural. Pentru fiecare $f \in F$ definim o nouă operaţie f_ρ prin

$$f_\rho([a_1]_\rho, \ldots, [a_{ar(f)}]_\rho) = [f(a_1, \ldots, a_{ar(f)})]_\rho,$$

pentru orice $a_1, \ldots, a_{ar(f)} \in A$. Dacă $ar(f) = 0$, atunci $f_\rho = [f]_\rho$.

Aceste operaţii nu depind de reprezentanţii de clasă datorită proprietăţii de compatibilitate cu operaţiile pe care o are congruenţa ρ. Mai exact, pentru orice $b_i \in [a_i]_\rho$ are loc

$$\begin{aligned} f_\rho([b_1]_\rho, \ldots, [b_{ar(f)}]_\rho) &= [f(b_1, \ldots, b_{ar(f)})]_\rho \\ &= [f(a_1, \ldots, a_{ar(f)})]_\rho \\ &= f_\rho([a_1]_\rho, \ldots, [a_{ar(f)}]_\rho). \end{aligned}$$

Algebra astfel obţinută, notată prin $(A/\rho, F/\rho)$, se numeşte *algebra cât indusă de* (A, F) *şi* ρ.

Propoziţia 3.4.6. Fie (A, F) o algebră şi $\rho \in Con(A, F)$. Atunci, $f : A \to A/\rho$ dată prin $f(a) = [a]_\rho$, pentru orice $a \in A$, este epimorfism.

Propoziţia 3.4.7. Dacă f este un homomorfism de la algebra (A, F) la algebra (A', F'), atunci $Ker(f)$ este congruenţă în (A, F).

Teorema 3.4.2. (Teorema de homomorfism)
Fie f un epimorfism de la algebra (A, F) la algebra (A', F'). Atunci, algebrele $(A/ker(f), F/Ker(f))$ şi (A', F') sunt izomorfe.

Demonstraţie Fie funcţia h de la $(A/ker(f), F/Ker(f))$ la (A', F') dată prin $h([a]_{ker(f)}) = f(a)$, pentru orice $a \in A$. Arătăm că h este izomorfism:

- *h este bine definită.* Dacă $a\, ker(f)\, b$, atunci $f(a) = f(b)$, ceea ce arată că definiţia funcţie h nu depinde de reprezentanţii de clasă aleşi;

- *f este homomorfism.* Pentru orice $g \in F$ şi $a_1, \ldots, a_{ar(g)} \in A$ au loc relaţiile:

$$h(g_{ker(f)}([a_1]_{ker(f)}, \ldots, [a_{ar(g)}]_{ker(f)})) =$$

$$\begin{aligned} &= h([g(a_1,\ldots,a_{ar(g)})]_{ker(f)}) \\ &= f(g(a_1,\ldots,a_{ar(g)})) \\ &= g'(f(a_1),\ldots,f(a_{ar(g)})) \\ &= g'(h([a_1]_{ker(f)}),\ldots,h([a_{ar(g)}]_{ker(f)})) \end{aligned}$$

care ne arată că h este homomorfism;

- h *este funcţie injectivă.* Dacă $f(a) = f(b)$, atunci $a\ ker(f)\ b$, ceea ce arată că h este injecţie;
- h *este funcţie surjectivă.* Pentru orice $b \in A'$ există $a \in A$ astfel încât $f(a) = b$. Ca urmare, $h([a]_{ker(f)}) = f(a) = b$, ce arată că h este surjecţie.

Deci, h este izomorfism. □

Teorema 3.4.2 poate fi adesea întâlnită şi sub denumirea de *prima teoremă de izomorfism*.

3.5 Algebre booleene

Un exemplu foarte important de algebră este cel de *algebră booleană* [14]. Aceste tipuri de algebre îşi au rădăcinile în studiile lui George Boole asupra operaţiile de reuniune, intersecţie şi complementară din teoria mulţimilor şi, asupra operaţiilor de conjuncţie, disjuncţie şi negaţie din logică. Algebrele booleene vin să extragă esenţa acestor operaţii şi să ofere un cadru general de studiu al proprietăţilor acestora.

Definiţia 3.5.1. O *algebră booleană* este o algebră $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$, unde $\vee, \wedge$ sunt operaţii binare pe A, $'$ este o operaţie unară pe A, iar 0 şi 1 sunt operaţii nulare pe A, ce satisfac:

(1) $\vee$ şi $\wedge$ sunt asociative şi comutative;

(2) $x \vee 0 = x$ şi $x \wedge 1 = x$, pentru orice $x \in A$;

(3) $\vee$ şi $\wedge$ sunt distributive una faţă de alta;

(4) $x \vee x' = 1$ şi $x \wedge x' = 0$, pentru orice $x \in A$. (complementariere)

Elementul x' din Definiţia 3.5.1 este numit *complementul* lui x.

Algebrele booleene fiind cazuri particulare de algebre, orice concept introdus pentru algebre se poate translata la algebre booleene. Este uşor de observat că $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$ este o algebră booleană dacă $(A, \vee, 0)$ şi $(A, \wedge, 1)$ sunt monoizi comutativi, operaţiile $\vee$ şi $\wedge$ sunt distributive una faţă de cealaltă şi are loc proprietatea de complementariere.

Exemplul 3.5.1.

(1) $(\mathcal{P}(A), \cup, \cap, ', \emptyset, A)$ este algebră booleană ($'$ fiind operaţia de complementariere a mulţimilor în raport cu A);

(2) Fie $B = \{0, 1\}$, $\vee$ şi $\wedge$ adunarea şi, respectiv, înmulţirea modulo 2, iar $'$ dată prin:

$$0' = 1 \text{ şi } 1' = 0.$$

Atunci, $\mathbf{B} = (B, \vee, \wedge, ', 0, 1)$ este algebră booleană.

0 poate fi interpretat ca valoarea de adevăr "fals", 1 ca valoarea de adevăr "adevărat", $\vee$ este disjuncţia, $\wedge$ este conjuncţia, iar $'$ este negaţia. Am obţinut astfel *algebra booleană a mulţimii valorilor de adevăr* B.

(3) Pe mulţimea B^n, unde B este ca la (2) iar $n \geq 1$, definim operaţiile $\vee$, $\wedge$ şi $'$ pe componente utilizând operaţiile de la (2). De exemplu,

$$(x_1, \ldots, x_n) \vee (y_1, \ldots, y_n) = (x_1 \vee y_1, \ldots, x_n \vee y_n).$$

Atunci $\mathbf{B}^n = (B^n, \vee, \wedge, ', \mathbf{0}, \mathbf{1})$, unde $\mathbf{0}$ şi $\mathbf{1}$ sunt n-upluri formate numai din 0 şi, respectiv, 1, este o algebră booleană.

Teorema 3.5.1. Fie $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$ o algebră booleană. Atunci, pentru orice $x, y \in A$, au loc următoarele proprietăţi:

(1) $x \vee x = x$ şi $x \wedge x = x$; (idempotenţă)

(2) $x \vee 1 = 1$ şi $x \wedge 0 = 0$;

(3) $(x \wedge y) \vee x = x$ şi $(x \vee y) \wedge x = x$; (absorpţie)

(4) $(x \vee y)' = x' \wedge y'$ şi $(x \wedge y)' = x' \vee y'$; (legile lui DeMorgan)

(5) $x \vee y = y$ dacă şi numai dacă $x \wedge y = x$.

Demonstraţie Pentru (1), (2), (3) şi (4) vom demonstra doar prima relaţie, cea de a doua obţinându-se prin dualizare.

(1) $x \vee x = (x \vee x) \wedge 1 = (x \vee x) \wedge (x \vee x') = x \vee (x \wedge x') = x \vee 0 = x$.

(2) $x \vee 1 = x \vee (x \vee x') = (x \vee x) \vee x' = x \vee x' = 1$.

(3) $(x \wedge y) \vee x = (x \wedge y) \vee (x \wedge 1) = x \wedge (y \vee 1) = x \wedge 1 = x$.

(4) Vom arăta întâi că dacă $w \vee z = 1$ şi $w \wedge z = 0$ atunci $z = w'$. In adevăr,

$$\begin{aligned} z &= z \vee 0 \\ &= z \vee (w \wedge w') \\ &= (z \vee w) \wedge (z \vee w') \\ &= 1 \wedge (w' \vee z) \\ &= (w' \vee w) \wedge (w' \vee z) \\ &= w' \vee (w \wedge z) \\ &= w' \vee 0 \\ &= w'. \end{aligned}$$

Ca urmare a acestui rezultat intermediar, pentru a demonstra prima lege a lui DeMorgan este suficient să arătăm că au loc relaţiile $(x \vee y) \vee (x' \wedge y') = 1$ şi $(x \vee y) \wedge (x' \wedge y') = 0$.

Avem,

$$\begin{aligned}(x \vee y) \vee (x' \wedge y') &= ((x \vee y) \vee x') \wedge ((x \vee y) \vee y') \\ &= (y \vee (x \vee x')) \wedge (x \vee (y \vee y')) \\ &= (y \vee 1) \wedge (x \vee 1) \\ &= 1 \wedge 1 \\ &= 1.\end{aligned}$$

Similar se arată şi cea de a doua relaţie şi, deci, prima parte de la (4) este demonstrată.

(5) Dacă $x \vee y = y$ atunci, utilizând absorbţia, obţinem

$$x = x \wedge (x \vee y) = x \wedge y.$$

Implicaţia în sens invers se obţine prin dualizare. □

Fie $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$ o algebră booleană. Definim relaţia $\leq \subseteq A \times A$ prin

$$x \leq y \Leftrightarrow x \vee y = y,$$

pentru orice $x, y \in A$. Relaţiile $<$, $\geq$ şi $>$ se definesc în mod uzual. Aceste relaţii vor fi numite *relaţiile induse* de algebra $\mathbf{A}$.

Teorema 3.5.2. Fie $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$ o algebră booleană şi $\leq$ relaţia indusă de $\mathbf{A}$. Atunci, au loc următoarele proprietăţi:

(1) $\leq$ este relaţie de ordine parţială pe A;

(2) $x \wedge y \leq x \leq x \vee y$, pentru orice $x, y \in A$;

(3) $0 \leq x \leq 1$, pentru orice $x \in A$.

Demonstraţie (1) şi (3) necesită verificări triviale, iar (2) se obţine de la proprietatea de absorbţie. □

Corolarul 3.5.1. Fie $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$ o algebră booleană. Atunci, $(A, \leq)$, unde $\leq$ este ordinea parţială indusă de $\mathbf{A}$, este latice pentru care 0 este cel mai mic element şi 1 este cel mai mare element.

Definiţia 3.5.2. Fie $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$ o algebră booleană. Un element $x \in A$ este numit *atom* dacă nu poate fi scris în forma $x = y \vee z$ cu y şi z ambele diferite de x şi 0.

Propoziţia 3.5.1. Fie $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$ o algebră booleană. Un element x diferit de 0 este atom dacă şi numai dacă nu există $y \in A$ astefel încât $0 < y < x$.

Demonstraţie Presupunem, prin contradicţie, că ar exista un atom x şi un element y astfel încât $0 < y < x$. Atunci,

$$x = x \wedge 1 = (y \vee x) \wedge (y \vee y') = y \vee (x \wedge y').$$

Deoarece x este atom, unul din elementele y sau $x \wedge y'$ trebuie să coincidă cu x. Cum am presupus că $y < x$, urmează că $x \wedge y' = x$. Dar atunci putem scrie

$$y = x \wedge y = (x \wedge y') \wedge y = x \wedge (y' \wedge y) = x \wedge 0 = 0,$$

ceea ce constituie o contradicţie.

Reciproc, dacă presupunem că nu există nici un element y cu $0 < y < x$ dar x nu este atom, atunci x poate fi scris $x = u \vee v$ cu u şi v ambele diferite de 0. Deoarece $u \leq u \vee v = x$ urmează $u < x$, ceea ce contrazice ipoteza. □

Exemplul 3.5.2.

(1) Pentru algebra boolenă $(\mathcal{P}(A), \cup, \cap, ', \emptyset, A)$ din Exemplul 3.5.1(1), în cazul în care A este finită, atomii sunt exact mulţimile de forma $\{a\}$ cu $a \in A$.

(2) Algebra $\mathbf{B}$ din Exemplul 3.5.1(2) are ca atom doar pe 1.

(3) Atomii algebrei $\mathbf{B}^n$ din Exemplul 3.5.1(3) sunt toate n-uplele ce conţin doar un 1 şi în rest numai 0.

In exemplul anterior, pentru oricare din cele trei algebre considerate, observăm că orice element poate fi scris ca o $\vee$-combinaţie de atomi. In plus, scrierea este unică exceptând ordinea în care sunt combinaţi atomii prin $\vee$. Mai exact, dacă considerăm algebra $(\mathcal{P}(A), \cup, \cap, ', \emptyset, A)$, cu A finită, atunci orice submulţime nevidă $B \subseteq A$ poate fi scrisă ca $\cup$-combinaţie de exact toţi atomii $\{a\}$ cu $a \in B$. De fapt, această observaţie va fi idee de demonstraţie a următoarei teoreme importante.

Teorema 3.5.3. Fie $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$ o algebră booleană finită. Atunci, orice element $x \in A$ diferit de 0 poate fi scris în mod unic (exceptând ordinea termenilor) în forma

$$x = a_1 \vee \cdots \vee a_k,$$

unde $a_1, \ldots a_k$ sunt atomi.

Demonstraţie Vom arăta întâi că orice element $x \in A$ diferit de 0 poate fi scris ca o $\vee$-combinaţie de atomi.

Presupunem, prin contradicţie, că există elemente $x \in A$ diferite de 0 ce nu pot fi scrise în această formă, şi fie S mulţimea acestora. Este clar că S nu conţine atomi şi, atunci, orice element x din S poate fi scris în forma $x = y \vee z$ cu $0 < y < x$ şi $0 < z < x$. Considerând un astfel de x cu o astfel de scriere, constatăm că cel puţin unul din elementele y sau z este în S; fie acesta y. Are loc $y < x$ şi, repetând acest procedeu cu y, deducem că există o secvenţă

$$x = x_0 > y = x_1 > \cdots$$

de elemente din S. Deoarece A este finită, rezultă că nu toate elementele din acest şir sunt diferite două câte două şi, deci, vor exista două numere naturale k şi m cu $k < m$ astfel încât $x_k = x_m$; contradicţie cu $x_k > x_m$. Ca urmare, orice element diferit de 0 din A poate fi scris ca o $\vee$-combinaţie de atomi.

Să ne ocupăm acum de unicitatea scrierii. Pentru acesta trebuie să remarcăm că este suficient să arătăm că orice element $x \in A$ poate fi scris ca o $\vee$-combinaţie a tuturor atomilor a cu $a \leq x$. In adevăr, dacă

$$x = \vee\{a \in A | a \text{ este atom şi } a \leq x\}$$

şi x ar avea şi o altă scriere, $x = b_1 \vee \cdots \vee b_k$ cu $b_1, \ldots, b_k$ atomi, atunci $b_i \leq x$ şi, ca urmare,

$$b_i \in \{a \in A | a \text{ este atom şi } a \leq x\}$$

pentru orice i. Pe de altă parte, dacă a este atom şi $a \leq x$ atunci

$$0 \neq a = a \wedge x = a \wedge (b_1 \vee \cdots \vee b_k) = (a \wedge b_1) \vee \cdots \vee (a \wedge b_k).$$

Cel puţin unul din $a \wedge b_i$ trebuie să fie diferit de 0 şi, deci, $a \wedge b_i = a = b_i$ pentru cel puţin un i. Aceasta ne arată că a este unul din atomii b_i şi, deci,

$$\{b_1, \ldots, b_k\} = \{a \in A | a \text{ este atom şi } a \leq x\}.$$

Deci, ceea ce rămâne de arătat este că orice $x \in A$ diferit de 0 poate fi scris în forma

$$x = \vee\{a \in A | a \text{ este atom şi } a \leq x\}.$$

Elementul 1 poate fi scris ca $\vee$-combinaţie de atomi, şi fie $1 = a_1 \vee \cdots \vee a_n$ o astfel de $\vee$-combinaţie. Deoarece $1 = 1 \vee y$ pentru orice $y \in A$, urmează că

$$1 = a_1 \vee \cdots \vee a_n = \vee\{a \in A | a \text{ este atom şi } a \leq 1\}.$$

Dacă considerăm acum un element $x \in A$ diferit de 0, atunci

$$x = x \wedge 1 = x \wedge (a_1 \vee \cdots \vee a_n) = (x \wedge a_1) \vee \cdots \vee (x \wedge a_n).$$

Deoarece $0 \leq x \wedge a_i \leq a_i$ şi a_i este atom, Propoziţia 3.5.1 ne spune că $x \wedge a_i = a_i$ dacă $a_i \leq x$, sau $x \wedge a_i = 0$, în caz contrar. Dar aceasta conduce la

$$x = \vee\{a \in A | a \text{ este atom şi } a \leq x\},$$

încheind demonstraţia teoremei. □

Izomorfismul de algebre booleene este izomorfism de algebre. Următoarea teoremă ne spune că orice algebră booleană finită este complet determinată, până la un izomorfism, de numărul de atomii ai ei.

Teorema 3.5.4. Orice două algebre booleene finite cu acelaşi număr de atomi sunt izomorfe.

Demonstraţie Fie $\mathbf{A}_1 = (A_1, \vee_1, \wedge_1, ', 0_1, 1_1)$ şi $\mathbf{A}_2 = (A_2, \vee_2, \wedge_2, '', 0_2, 1_2)$ două algebre booleene cu atomii $a_1, \ldots, a_n$ şi, respectiv, $b_1, \ldots, b_n$.

Considerăm funcţia $f : A_1 \to A_2$ dată prin $f(0_1) = 0_2$, $f(1_1) = 1_2$ şi $f(a_i) = b_i$, pentru orice i. Extindem f la un unic homomorfism de la $\mathbf{A}_1$ la $\mathbf{A}_2$. Teorema 3.5.3 conduce atunci cu uşurinţă la faptul că acest homomorfism este funcţie bijectivă. □

Următoarele două rezultate urmează direct de la această teoremă.

Corolarul 3.5.2. Orice algebră booleană finită cu n atomi este izomorfă cu algebra booleană a mulţimii tuturor părţilor unei mulţimi cu n elemente.

Corolarul 3.5.3. Algebra booleană $\mathbf{B}^n$ din Exemplul 3.5.1(3) este izomorfă cu algebra părţilor mulţimii $\{1, \ldots, n\}$.

Definiţia 3.5.3. O latice $M = (A; \vee, \wedge)$ este numită *complementată* dacă are un cel mai mic element 0 şi se poate defini o operaţie unară $'$ ce satisface:

(1) $(a')' = a$,

(2) $(a \vee b)' = a' \wedge b'$,

(3) $a \wedge a' = 0$,

pentru orice $a, b \in A$.

Am văzut cum de la o algebră booleană se poate ajunge la o latice. In baza Definiţiei 3.5.3, putem spune că algebrele booleene sunt exact latici distributive complementate.

Capitolul 4

Elemente de teoria numerelor cu aplicaţii în criptografie

In acest capitol vom prezenta câteva elemente de bază de teoria numerelor, necesare înţelegerii corecte a conceptelor ce vor urma. Pentru detalii, acolo unde este cazul, cititorul interesat este îndrumat către monografii standard, cum ar fi [169, 168, 73, 154], sau către [185] unde se poate găsi o colecţie de algoritmi de teoria numerelor, împreună cu studiile de complexitate aferente.

4.1 Divizibilitate. Numere prime

Notăm prin $|a|$ *modulul* numărului $a \in \mathbf{Z}$. Adică,

$$|a| = \begin{cases} a, & \text{dacă } a \geq 0 \\ -a, & \text{altfel.} \end{cases}$$

Teorema 4.1.1. (Teorema împărţirii cu rest)
Pentru orice două numere întregi a şi b cu $b \neq 0$, există $q, r \in \mathbf{Z}$ astfel încât $a = bq + r$ şi $0 \leq r < |b|$. In plus, q şi r sunt unicele cu aceste proprietăţi.

Demonstraţie Considerăm întâi cazul $b > 0$. Fie A mulţimea

$$A = \{a - bq | q \in \mathbf{Z}\} \cap \mathbf{N}.$$

A este nevidă deoarece, dacă $a < 0$ atunci $a - ba \in A$, iar dacă $a \geq 0$ atunci $a \in A$. Fiind submulţime de numere naturale, A va avea un cel mai mic element; fie acesta r. Atunci, r se poate scrie $r = a - bq$, unde $q \in \mathbf{Z}$. Vom arăta că q şi r astfel determinate satisfac teorema. Prin definiţie, $r \geq 0$. Vom arată că $r < b$. Dacă presupunem prin contradicţie că $r \geq b$, atunci numărul $r - b$ este în A deoarece $r - b \geq 0$ şi el se poate scrie în forma $r - b = a - b(q + 1)$, ceea ce va contrazice alegerea lui r. Deci, $0 \leq r < b$. Unicitatea numerelor q şi r se obţine cu uşurinţă, prin contradicţie. In adevăr, să presupunem că există $q, r, q', r' \in \mathbf{Z}$ astfel încât $a = bq + r$, $a = bq' + r'$, $0 \leq r < b$ şi $0 \leq r' < b$. Dacă $q = q'$ ($r = r'$) atunci urmează imediat că $r = r'$

($q = q'$). Ca urmare, presupunem că $q \neq q'$ şi $r \neq r'$. Fie, de exemplu, $q < q'$. Atunci, relaţia $bq + r = bq' + r'$ conduce la

$$r' = r - b(q' - q).$$

Cum $r < b$ şi $q' - q > 0$, obţinem $r' < 0$; contradicţie. Deci, $q = q'$ şi $r = r'$.

Cazul $b < 0$ se obţine din precedentul astfel. In primul rând, observăm că $-b > 0$ şi, atunci, există unice q' şi r' cu $a = (-b)q' + r'$ şi $0 \leq r' < (-b)$. Atunci, alegând $q = -q'$ şi $r = r'$ deducem că $a = bq + r$ şi $0 \leq r < |b|$. Unicitatea numerelor q şi r se obţine ca în cazul precedent. □

Numerele q şi r din Teorema împărţirii cu rest se numesc *câtul* şi, respectiv, *restul* împărţirii lui a la b. Ele se mai notează prin $a \; div \; b$ şi, respectiv, $a \; mod \; b$.

Definiţia 4.1.1. Relaţia binară $| \subseteq \mathbf{Z} \times \mathbf{Z}$ dată prin

$$a|b \;\; \Leftrightarrow \;\; (\exists c \in \mathbf{Z})(b = ac),$$

pentru orice $a, b \in \mathbf{Z}$, se numeşte *relaţia de divizibilitate* pe $\mathbf{Z}$.

Dacă $a|b$ atunci vom spune că a *divide* b sau că a *este divizor al lui* b sau că b *se divide prin* a sau că b *este multiplu al lui* a. Dacă a nu divide b atunci vom mai scrie $a \not| b$.

Observăm că dacă $a \neq 0$, atunci $a|b$ dacă şi numai dacă $b \; mod \; a = 0$. Dacă $a|b$ şi $|a|$ este diferit atât de 1 cât şi de b, atunci vom spune că a este *divizor propriu* al lui b. Direct de la definiţie obţinem următoarea propoziţie.

Propoziţia 4.1.1. Fie $a, b, c \in \mathbf{Z}$. Atunci, au loc următoarele proprietăţi:

(1) 0 divide doar 0;

(2) a divide 0 şi a;

(3) 1 divide a;

(4) $a|b$ dacă şi numai dacă $a| - b$;

(5) dacă $a|b$ şi $b|c$, atunci $a|c$;

(6) dacă $a|b + c$ şi $a|b$, atunci $a|c$;

(7) dacă $a|b$, atunci $ac|bc$. Reciproc, dacă $c \neq 0$ şi $ac|bc$, atunci $a|b$;

(8) dacă $a|b$ şi $a|c$, atunci $a|\beta b + \gamma c$, pentru orice $\beta, \gamma \in \mathbf{Z}$;

(9) dacă $a|b$ şi $b \neq 0$, atunci $|a| \leq |b|$. Dacă în plus a este divizor propriu al lui b, atunci $1 < |a| < |b|$.

Definiţia 4.1.2. Un număr natural $p \geq 2$ este numit *prim* dacă singurii lui divizori pozitivi sunt 1 şi p.

Altfel spus, numerele prime sunt numere $n \geq 2$ ce nu au divizori proprii. Numerele $n \geq 2$ ce au divizori proprii sunt numite *compuse* sau *compozite*.

Definiţia 4.1.3. Fie $a_1, \ldots, a_m \in \mathbf{Z}$, unde $m \geq 2$. Spunem că $a_1, \ldots, a_m$ sunt *prime între ele* sau *relativ prime* sau *coprime* dacă singurii divizori comuni ai acestor numere sunt 1 şi -1.

Vom nota $(a_1, \ldots, a_m) = 1$ pentru a specifica faptul că $a_1, \ldots, a_m$ sunt relativ prime (această notaţie va fi justificată în secţiunea următoare). Observăm că $(0, 1) = 1$. De asemenea, orice două numere dintre care unul este par şi celălalt impar, sunt prime între ele.

Următoarea teoremă este crucială în stabilirea multor proprietăţi în care intervine conceptul de numere relativ prime.

Teorema 4.1.2. Fie $m \geq 2$ şi $a_1, \ldots, a_m \in \mathbf{Z}$. Atunci, $(a_1, \ldots, a_m) = 1$ dacă şi numai dacă există $\alpha_1, \ldots, \alpha_m \in \mathbf{Z}$ astfel încât $\alpha_1 a_1 + \cdots + \alpha_m a_m = 1$.

Demonstraţie Fie $a_1, \ldots, a_m \in \mathbf{Z}$, unde $m \geq 2$.

Dacă presupunem că există $\alpha_1, \ldots, \alpha_m \in \mathbf{Z}$ astfel încât $\alpha_1 a_1 + \cdots + \alpha_m a_m = 1$, atunci $a_1, \ldots, a_m$ nu pot avea un divizor comun d diferit de 1 şi -1 deoarece, atunci, d ar divide suma $\alpha_1 a_1 + \cdots + \alpha_m a_m$ şi, deci şi pe 1. Ca urmare, $a_1, \ldots, a_m$ sunt relativ prime.

Reciproc, presupunem că $a_1, \ldots, a_m$ sunt relativ prime. Considerăm mulţimea

$$A = \{\alpha_1 a_1 + \cdots + \alpha_m a_m | \alpha_1, \ldots, \alpha_m \in \mathbf{Z}\} \cap \mathbf{N}.$$

Această mulţime este nevidă şi conţine elemente diferite de 0 (aceasta rezultă cu uşurinţă considerând, de exemplu, $\alpha_i = a_i$ pentru orice i şi remarcând că nu toate numerele a_i pot fi 0). Ca urmare, A va avea un cel mai mic element diferit de 0, fie acesta $d = \alpha_1 a_1 + \cdots + \alpha_m a_m$. Vom arăta că $d | a_i$ pentru orice i, ceea ce va implica $d = 1$, încheind demonstraţia teoremei.

Fie $1 \leq i \leq m$. In baza teoremei împărţirii cu rest, există unice q_i şi r_i astfel încât

$$a_i = dq_i + r_i \text{ şi } 0 \leq r_i < d.$$

Atunci,

$$\begin{aligned} r_i &= a_i - dq_i \\ &= a_i - (\alpha_1 a_1 + \cdots + \alpha_m a_m) q_i \\ &= (1 - q_i \alpha_i) a_i + \textstyle\sum_{j \neq i} (-q_i \alpha_j) a_j \\ &\geq 0, \end{aligned}$$

ceea ce arată că $r_i \in A$. Conform alegerii lui d şi a faptului că $r_i < d$, urmează că $r_i = 0$. Aceasta conduce însă la $d | a_i$. Ca urmare, $d = 1$. □

Corolarul 4.1.1. Fie $a_1, \ldots, a_m, b \in \mathbf{Z}$, unde $m \geq 2$. Dacă $(b, a_i) = 1$ pentru orice $1 \leq i \leq m$, atunci $(b, a_1 \cdots a_m) = 1$.

Demonstraţie Vom demonstra corolarul pentru $m = 2$, cazul general obţinându-se prin simplă inducţie.

Conform Teoremei 4.1.2, există α_1, α_2, β_1 şi β_2 astfel încât $\alpha_1 a_1 + \beta_1 b = 1$ şi $\alpha_2 a_2 + \beta_2 b = 1$. Atunci,

$$\begin{aligned} 1 &= (\alpha_1 a_1 + \beta_1 b)(\alpha_2 a_2 + \beta_2 b) \\ &= \alpha_1\alpha_2 a_1 a_2 + b(\alpha_1 a_1 \beta_2 + \alpha_2 a_2 \beta_1 + \beta_1\beta_2 b), \end{aligned}$$

ceea ce arată că $(b, a_1 a_2) = 1$. □

Corolarul 4.1.2. Fie $a_1, \ldots, a_m, b \in \mathbf{Z}$, unde $m \geq 2$. Dacă numerele $a_1, \ldots, a_m$ sunt prime între ele două câte două şi fiecare din ele divide b, atunci produsul lor divide b.

Demonstraţie Ca şi în cazul corolarului precedent vom face demonstraţia doar pentru $m = 2$.

Deoarece $(a_1, a_2) = 1$, există α_1 şi α_2 astfel încât $\alpha_1 a_1 + \alpha_2 a_2 = 1$, iar de la $a_1 | b$ şi $a_2 | b$ urmează că există β_1 şi β_2 astfel încât $b = a_1\beta_1 = a_2\beta_2$.

Atunci,

$$\begin{aligned} b &= a_1\beta_1 \\ &= a_1\beta_1(\alpha_1 a_1 + \alpha_2 a_2) \\ &= a_1\beta_1\alpha_1 a_1 + a_1 a_2 \alpha_2\beta_1 \\ &= a_2\beta_2\alpha_1 a_1 + a_1 a_2 \alpha_2\beta_1 \\ &= a_1 a_2(\alpha_1\beta_2 + \alpha_2\beta_1), \end{aligned}$$

ceea ce arată că $a_1 a_2 | b$.

Completarea inducţiei se face prin utilizarea Corolarului 4.1.1. □

Corolarul 4.1.3. Fie $a_1, \ldots, a_m, b \in \mathbf{Z}$, unde $m \geq 2$. Dacă b este prim cu a_1 şi divide produsul $a_1 \cdots a_m$, atunci b divide produsul $a_2 \cdots a_m$.

Demonstraţie Deoarece $(b, a_1) = 1$ urmează că există α şi β astfel încât $\alpha a_1 + \beta b = 1$, iar de la $b | a_1 \cdots a_m$ urmează că există γ astfel încât $a_1 \cdots a_m = b\gamma$. Atunci,

$$\begin{aligned} a_2 \cdots a_m &= 1 \cdot a_2 \cdots a_m \\ &= (\alpha a_1 + \beta b) a_2 \cdots a_m \\ &= \alpha a_1 \cdots a_m + \beta b a_2 \cdots a_m \\ &= \alpha b\gamma + \beta b a_2 \cdots a_m \\ &= b(\alpha\gamma + \beta a_2 \cdots a_m), \end{aligned}$$

ceea ce arată că $b | a_2 \cdots a_m$. □

Corolarul 4.1.4. Fie $a_1, \ldots, a_m, p \in \mathbf{Z}$, unde $m \geq 2$. Dacă p este prim şi divide produsul $a_1 \cdots a_m$, atunci există i astfel încât p divide a_i.

Demonstraţie Presupunem, prin contradicţie, că $p \not| a_i$, pentru orice i. Atunci, p este prim cu oricare din numerele a_i şi, deci, Corolarul 4.1.1 conduce la $(p, a_1 \cdots a_m) = 1$, ceea ce contrazice $p | a_1 \cdots a_m$. □

Fie $n \geq 2$ un număr natural. Numim *descompunere* a lui n orice secvenţă finită de numere naturale

$$n_1, \ldots, n_k \quad (k \geq 1)$$

astfel încât $n = n_1 \cdots n_k$.

Descompunerea unui număr natural $n \geq 2$ nu este în mod necesar unică. Simpla permutare a termenilor secvenţei face, în general, ca descompunerea să nu fie unică. Insă, astfel de permutări sunt irelevante şi, ca urmare, prin descompunere a numărului natural $n \geq 2$ vom înţelege orice secvenţă de perechi de numere naturale

$$(n_1, e_1), \ldots, (n_k, e_k) \quad (k \geq 1)$$

astfel încât:

- $2 \leq n_1 < \cdots < n_k$;
- $e_i > 0$, pentru orice $1 \leq i \leq k$;
- $n = n_1^{e_1} \cdots n_k^{e_k}$.

Convenim ca descompunerea de mai sus a numărului n să fie notată simplificat prin $\prod_{i=1}^{k} n_i^{e_i}$ (sau prin $\prod n_i^{e_i}$, dacă menţionarea numărului k este irelevantă sau se subânţelege din context).

Cu această nouă definiţie a descompunerii putem vorbi de descompuneri distincte ale aceluiaşi număr ca fiind descompuneri pentru care secvenţele corespunzătoare nu coincid. De exemplu, $20 = 4 \cdot 5$ şi $20 = 2^2 \cdot 5$ sunt descompuneri distincte ale lui 20. Dacă în descompunerea de mai sus numerele $n_1, \ldots, n_k$ sunt prime, atunci descompunerea lui n va fi numită *descompunere în factori primi*. Are loc:

Teorema 4.1.3. (Teorema fundamentală a aritmeticii)
Orice număr natural $n \geq 2$ poate fi descompus, în mod unic, în factori primi (unicitatea fiind înţeleasă aşa cum a fost specificat mai sus).

Demonstraţie Existenţa unei descompuneri în factori primi a oricărui număr natural $n \geq 2$ se obţine cu uşurinţă prin inducţie, luând în calcul cele două posibilităţi asupra lui n: n este prim, sau n nu este prim. In cel de-al doilea caz, n se descompune în produsul a două numere $n = n_1 n_2$ cu proprietatea $2 \leq n_1, n_2 < n$. Se aplică apoi ipoteza inductivă.

Pentru unicitate vom face apel din nou la inducţie şi, în plus, la Corolarul 4.1.4. Să presupunem că n admite două descompuneri în factori primi, $n = p_1^{e_1} \cdots p_s^{e_s}$ şi $n = q_1^{g_1} \cdots q_t^{g_t}$. Dacă $\sum_{i=1}^{s} e_i = \sum_{i=1}^{t} g_i = 1$, atunci obţinem imediat că $p_1 = q_1$. Altfel, dacă de exemplu $\sum_{i=1}^{t} g_i > 1$, relaţia $p_1 | n = q_1^{g_1} \cdots q_t^{g_t}$ conduce la existenţa unui i astfel încât $p_1 | q_i$ (Corolarul 4.1.4). Dar aceasta este posibil numai dacă $p_1 = q_i$. Simplificând cele două descompuneri ale lui n, prima prin p_1 şi a doua prin q_i, obţinem un nou număr $n' < n$ şi două descompuneri ale lui, pentru care putem aplica ipoteza inductivă. □

Aşa cum probabil este uşor de bănuit, există o infinitate de numere prime.

Teorema 4.1.4. Există o infinitate de numere prime.

Demonstraţie Presupunem, prin contradicţie, că există doar un număr finit de numere prime, fie acestea $p_1, \ldots, p_n$ ($n \geq 1$). Fie $a = p_1 \cdots p_n + 1$. Numărul a este strict mai mare decât oricare din cele n numere prime $p_1, \ldots, p_n$. Atunci, în baza Teoremei 4.1.3, el este divizibil prin unul din aceste numerele. Să presupunem că $a = p_i d$, unde $1 \leq i \leq n$ şi $d \geq 2$. Atunci,

$$1 = a - p_1 \cdots p_n = p_i d - p_1 \cdots p_n = p_i(d - \prod_{j \neq i} p_j),$$

ceea ce arată că p_i divide 1; contradicţie. □

Fie p_n al n-lea număr prim, pentru orice $n \geq 1$ ($p_1 = 2$). Deoarece nu se cunoaşte o formulă de determinare efectivă a numărului p_n, studiul distribuţiei numerelor prime joacă un rol foarte important în teoria numerelor. Prin *distribuţia* numerelor prime înţelegem, intuitiv, modul în care aceste numere sunt repartizate pe axa numerelor naturale. In principal, studiul distribuţiei numerelor prime se face prin intermediul funcţiei π definită pentru orice număr natural $n \geq 2$ prin

$$\pi(n) = |\{p \in \mathbf{N} | p \leq n \ \wedge \ p \text{ prim}\}|.$$

Următoarea teoremă, "intuită" de Gauss în 1801 [1] dar demonstrată abia în 1896 de matematicianul francez Jacques Hadamard şi, independent, de matematicianul belgian Charles-Jean de la Vallée-Poussin [2], estimează această funcţie prin intermediul funcţiei $\frac{n}{ln\, n}$ definită pentru orice număr natural $n \geq 2$ (ln denotă funcţia logaritm natural). Demonstraţia ei depăşeşte cu mult cadrul lucrării noastre. Pentru detalii cititorul este îndrumat către [169].

Teorema 4.1.5. (Teorema numerelor prime)

$$\lim_{n \to \infty} \frac{\pi(n)}{\frac{n}{ln\, n}} = 1.$$

Vom mai scrie $\pi(n) \sim \frac{n}{ln\, n}$ şi vom spune că funcţiile $\pi(n)$ şi $\frac{n}{ln\, n}$ sunt *echivalente asimptotic* [3]. Tabelul de mai jos prezintă câteva valori ale funcţiei π.

n	10^1	10^2	10^3	10^4	10^5	10^6	10^7	10^9
$\pi(n)$	4	25	168	1229	9592	78496	664579	50847478

Figura 4.1: Câteva valori ale funcţiei π

Corolarul 4.1.5. $\lim_{n \to \infty} \frac{p_n}{n\, ln\, n} = 1$.

[1] Menţionată în cartea sa "Disquisitiones Arithmeticae" publicată în 1801 ([60] este o traducere revizuită a acesteia în limba engleză).

[2] O demonstraţie mai simplă a fost propusă de Landau în 1903.

[3] A nu se confunda cu notaţia "$A \sim B$" utilizată pentru a desemna echipotenţa mulţimilor A şi B.

Acest corolar ne spune că putem aproxima p_n prin $n\,ln\,n$, pentru n suficient de mare [4].

Teorema 4.1.5 este de importanţă uriaşă în studiul numerelor prime oferind o aproximare asimptotică pentru $\pi(n)$. Ulterior, Rosser şi Schoenfeld [156] au rafinat acest rezultat obţinând aproximări mai precise. Următoarele două teoreme sunt datorate acestora.

Teorema 4.1.6. Pentru orice număr natural $n \geq 67$ are loc:

$$\frac{n}{ln\,n\,-\,\frac{1}{2}} < \pi(n) < \frac{n}{ln\,n\,-\,\frac{3}{2}}$$

Teorema 4.1.7. Pentru orice număr natural $n \geq 17$ are loc

$$\pi(n) > \frac{n}{ln\,n},$$

şi pentru orice $n \geq 2$ are loc

$$\pi(n) < 1.25506\frac{n}{ln\,n}$$

Cititorul poate compara cele două aproximări şi utiliza, de la caz la caz, pe cea mai bună. Ca o simplă aplicaţie a acestor rezultate, ne propunem să estimăm numărul de numere prime cu 100 de cifre. Pornind de la observaţia că 10^{100} şi 10^{99} nu sunt numere prime, putem estima numărul de numere prime cu 100 de cifre prin

$$\begin{aligned}
\pi(10^{100}) - \pi(10^{99}) &\approx \frac{10^{100}}{100\,ln\,10} - \frac{10^{99}}{99\,ln\,10} \\
&= \frac{10^{99}}{ln\,10}(\frac{1}{10} - \frac{1}{99}) \\
&> 0.39 \cdot 10^{98} \\
&\approx 4 \cdot 10^{97}
\end{aligned}$$

(utilizând $2.30 < ln\,10 < 2.31$).

Pentru a avea o imagine asupra acestui număr, îl putem compara pe acesta cu numărul de atomi din "universul vizibil", număr estimat de fizicieni ca fiind între 10^{79} şi 10^{81}.

4.2 Cel mai mare divizor comun

Lema 4.2.1. Fie $a_1, \ldots, a_m$ numere întregi nu toate 0, unde $m \geq 2$. Atunci, există cel mai mare număr natural d cu proprietatea $d|a_i$, pentru orice $1 \leq i \leq m$.

Demonstraţie Fie D_i mulţimea tuturor divizorilor numărului a_i, $1 \leq i \leq m$. Atunci, mulţimea $\bigcap_{i=1}^{m} D_i$ este nevidă (conţine cel puţin pe 1) şi finită (conform

[4] In 1939, J.B. Rosser [155] a stabilit inegalitatea $p_n > n\,ln\,n$, pentru orice $n \geq 1$.

ipotezei, cel puţin o mulţime D_i este finită). Cel mai mare element al acestei mulţimi satisface lema. □

Numărul d din Lema 4.2.1 se numeşte *cel mai mare divizor comun* al numerelor $a_1, \ldots, a_m$. El se mai notează prin $cmmdc(a_1, \ldots, a_n)$ sau, atunci când nu există pericol de confuzie, prin $(a_1, \ldots, a_n)$. Putem spune că numerele $a_1, \ldots, a_m$ sunt relativ prime dacă şi numai dacă $(a_1, \ldots, a_m) = 1$, ceea ce justifică notaţia adoptată în secţiunea anterioară.

Propoziţia 4.2.1. Fie $a_1, \ldots, a_n$ numere întregi nu toate 0, unde $m \geq 2$. Atunci:

(1) $(0, a_1, \ldots, a_m) = (a_1, \ldots, a_m)$;

(2) $(0, a_1) = |a_1|$, cu condiţia $a_1 \neq 0$;

(3) $(a_1, a_2) = (a_2, a_1 \ mod \ a_2)$, cu condiţia $a_2 \neq 0$.

Demonstraţie (1) şi (2) urmează imediat de la definiţii.

Pentru (3), dacă scriem $a_1 = a_2 q + r$ conform teoremei împărţirii cu rest, unde $0 \leq r < a_2$, atunci observăm că orice divizor comun al numerelor a_1 şi a_2 este divizor comun al numerelor a_2 şi r, şi reciproc. Ca urmare, $(a_1, a_2) = (a_2, r)$. □

Teorema 4.2.1. (Forma liniară a cmmdc)
Fie $a_1, \ldots, a_m$ numere întregi nu toate 0, unde $m \geq 2$. Atunci, există numerele întregi $\alpha_1, \ldots, \alpha_m$ astfel încât

$$(a_1, \ldots, a_m) = \alpha_1 a_1 + \cdots + \alpha_m a_m.$$

Demonstraţie Dacă $d = (a_1, \ldots, a_m)$, atunci există $a'_1, \ldots, a'_m$ astfel încât $a_i = da'_i$, pentru orice i. In plus, $(a'_1, \ldots, a'_m) = 1$. Afirmaţia din teoremă se obţine atunci cu uşurinţă de la Teorema 4.1.2. □

Corolarul 4.2.1. Fie $a_1, \ldots, a_m$ numere întregi nu toate 0, unde $m \geq 2$. Atunci, un număr natural d este cel mai mare divizor comun al numerelor $a_1, \ldots, a_m$ dacă şi numai dacă au loc următoarele proprietăţi:

(i) $d|a_i$, pentru orice $1 \leq i \leq m$;

(ii) $(\forall d' \in \mathbf{N})((\forall 1 \leq i \leq m)(d'|a_i) \ \Rightarrow \ d'|d)$.

Demonstraţie Dacă $d = (a_1, \ldots, a_m)$, atunci are loc (i). In plus, d se poate scrie în forma $d = \alpha_1 a_1 + \cdots + \alpha_m a_m$, unde $\alpha_1, \ldots, \alpha_m \in \mathbf{Z}$. Atunci, orice divizor d' a numerelor a_i va fi divizor şi al lui d. Deci, are loc (ii).

Reciproc, dacă d este un număr natural ce satisface (i) şi (ii), orice alt divizor comun d' al numerelor a_i va satisface $d' \leq d$ (de la (ii) şi Propoziţia 4.1.1(9)). Deci, $d = (a_1, \ldots, a_m)$. □

Corolarul 4.2.1 ne spune că proprietatea de a fi maximal în raport cu divizibilitatea este aceeaşi cu cea de a fi maximal în raport cu ordinea uzuală pe numere naturale, ambele considerate între divizorii comuni ai oricăror numere $a_1, \ldots, a_m$ nu toate 0.

Se poate formula şi o reciprocă a Teoremei 4.2.1, astfel:

"dacă există numerele întregi $\alpha_1, \ldots, \alpha_m$ astfel încât

$$\alpha_1 a_1 + \cdots + \alpha_m a_m | a_i$$

pentru orice i, atunci $(a_1, \ldots, a_m) = \alpha_1 a_1 + \cdots + \alpha_m a_m$."

In adevăr, orice divizor comun al numerelor $a_1, \ldots, a_m$ este divizor al sumei $\alpha_1 a_1 + \cdots + \alpha_m a_m$. Cum această sumă este divizor al numerelor $a_1, \ldots, a_m$, urmează că ea este cel mai mare divizor comun al lor (în baza Corolarului 4.2.1).

Corolarul 4.2.2. Fie $a_1, \ldots, a_m$ numere întregi nu toate 0, unde $m \geq 2$. Atunci, pentru orice $b \in \mathbf{Z}$, ecuaţia

$$a_1 x_1 + \cdots + a_m x_m = b,$$

în necunoscutele $x_1, \ldots, x_m$, are soluţii în $\mathbf{Z}$ dacă şi numai dacă $(a_1, \ldots, a_m)$ divide b.

Demonstraţie Dacă ecuaţia $a_1 x_1 + \cdots + a_m x_m = b$ are soluţii în $\mathbf{Z}$, fie $\alpha_1, \ldots, \alpha_m$ o astfel de soluţie, atunci orice divizor comun al numerelor $a_1, \ldots, a_m$ va fi divizor al numărului $a_1\alpha_1 + \cdots + a_m\alpha_m$ şi, deci, al lui b. Ca urmare, $(a_1, \ldots, a_m)|b$.

Reciproc, dacă $d = (a_1, \ldots, a_m)|b$ atunci există numerele k şi $\alpha_1, \ldots, \alpha_m$ astfel încât $b = kd$ şi $d = a_1\alpha_1 + \cdots + a_m\alpha_m$. Este clar atunci că $x_i = k\alpha_i$, pentru orice $1 \leq i \leq m$, este soluţie a ecuaţiei $a_1 x_1 + \cdots + a_m x_m = b$. □

Lema 4.2.2. Fie $a_1, \ldots, a_m$ numere întregi nenule, unde $m \geq 2$. Atunci, există cel mai mic număr natural nenul b cu proprietatea $a_i|b$, pentru orice $1 \leq i \leq m$.

Demonstraţie Similară Lemei 4.2.1. □

Numărul b din Lema 4.2.2 este multiplu comun al numerelor $a_1, \ldots, a_m$ şi, exceptând pe 0 care este şi el multiplu comun al acestor numere, b este cel mai mic număr natural cu această proprietate. Dacă unul din numerele $a_1, \ldots, a_m$ este 0, atunci 0 este unicul multiplu comun al acestora (deoarece 0 divide doar pe 0).

Deci, definim *cel mai mic multiplu comun* al numerelor $a_1, \ldots, a_m$ ca fiind numărul b din Lema 4.2.2, dacă aceste numere sunt nenule, şi 0, altfel. Cel mai mic multiplu comun al numerelor $a_1, \ldots, a_m$ se notează prin $cmmmc(a_1, \ldots, a_n)$ sau, atunci când nu există pericol de confuzie, prin $[a_1, \ldots, a_n]$.

Teorema 4.2.2. Fie $a_1, \ldots, a_m$ numere întregi, unde $m \geq 2$. Atunci, un număr natural b este cel mai mic multiplu comun al numerelor $a_1, \ldots, a_m$ dacă şi numai dacă au loc următoarele proprietăţi:

(i) $a_i|b$, pentru orice $1 \leq i \leq m$;

(ii) $(\forall b' \in \mathbf{N})((\forall 1 \leq i \leq m)(a_i|b') \Rightarrow b|b')$.

Demonstraţie Teorema se verifică cu uşurinţă dacă cel puţin unul din numerele $a_1, \ldots, a_m$ este 0. Să presupunem în continuare că toate aceste numere sunt diferite de 0.

Fie $b = [a_1, \ldots, a_m]$. Atunci $b > 0$ şi are loc (i). Fie b' un multiplu (comun) al numerelor a_i, $1 \leq i \leq m$. Deoarece b este cel mai mic multiplu comun al acestor numere, are loc $b \leq b'$, iar de la Teorema împărţirii cu rest deducem că există unice numerele q şi r astfel încât $b' = bq + r$ şi $0 \leq r < b$. Atunci $r = b' - bq$, de unde obţinem că r este un multiplu pozitiv al numerelor a_i, $1 \leq i \leq m$. Deoarece b este cel mai mic multiplu nenul al acestor numere, obţinem $r = 0$, ceea ce ne arată că $b|b'$. Deci, are loc (ii).

Reciproc, presupunem că b este un număr natural ce satisface (i) şi (ii). De la (ii) şi Propoziţia 4.1.1(9) urmează că b este cel mai mic număr natural ce satisface (i). Deci, $b = [a_1, \ldots, a_m]$. □

Teorema 4.2.3. Fie a şi b două numere naturale nu ambele 0. Atunci, are loc $ab = (a, b)[a, b]$.

Demonstraţie Dacă a sau b este 0, atunci teorema este trivial satisfcăcută. Să presupunem că a şi b sunt nenule. Fie $d = (a, b)$. Atunci, există a_1 şi b_1 astfel încât $a = da_1$, $b = db_1$ şi $(a_1, b_1) = 1$. Vom arăta că $[a, b] = da_1b_1$ ceea ce va încheia demonstraţia. Pentru aceasta este suficient de arătat că $da_1b_1|[a, b]$. Observăm că $a_1|[a, b]$ şi $b_1|[a, b]$ de unde, în baza faptului că $(a_1, b_1) = 1$ urmează $a_1b_1|[a, b]$. Similar, $d|[a, b]$ şi $a_1b_1|[a, b]$ combinate cu faptul că $(d, a_1b_1) = 1$ conduc la $da_1b_1|[a, b]$. □

Ca o consecinţă imediată a acestei teoreme obţinem:

Corolarul 4.2.3. Cel mai mic multiplu comun a două numere naturale relativ prime este egal cu produsul numerelor.

Propoziţia 4.2.2. *Fie $a_1, \ldots, a_m, b \in \mathbf{Z}$, unde $m \geq 2$, astfel încât $a_i|b$, pentru orice $1 \leq i \leq m$. Atunci, $[a_1, \ldots, a_m]|b$.*

Demonstraţie Vom face demonstraţia pentru cazul $m = 2$ (cazul general obţinându-se în mod similar).

Dacă $a_1 = 0$ sau $a_2 = 0$, atunci b trebuie să fie 0, iar afirmaţia din propoziţie urmează imediat. Să presupunem că a_1 şi a_2 sunt nenule. Fie $d = (a_1, a_2)$, $a_1 = da_1'$ şi $a_2 = da_2'$. Deoarece $d|b$, $a_1|b$ şi $a_2|b$ urmează că există b' astfel încât $b = db'$, $a_1'|b'$ şi $a_2'|b'$. Cum $(a_1', a_2') = 1$, deducem că are loc $a_1'a_2'|b'$, de unde urmează

$$[a, b] = da_1'a_2'|db' = b,$$

ceea ce încheie demonstraţia. □

Ne vom ocupa acum de determinarea algoritmică a celui mai mare divizor comun a două numere. Fără a restrânge generalitatea, putem considera numai numere naturale dintre care cel puţin unul nenul. Fie deci $a \geq b \geq 0$:

- dacă $a = b$ sau $b = 0$ atunci $(a, b) = a$;
- dacă $a > b > 0$, observăm că determinarea lui (a, b) se poate face luând în calcul doar divizorii (pozitivi ai) lui b. Dacă scriem $a = bq + r$, unde $0 \leq r < b$, atunci $(a, b) = (b, r)$ în baza Propoziţiei 4.2.1(3). Ca urmare, a determina (a, b) se reduce la a determina (b, r). Acest procedeu poate fi continuat până la ultimul rest diferit de 0, care va fi (a, b).

Ceea ce am descris poartă denumirea de *algoritmul lui Euclid* [5]. Mai exact, el constă în efectuarea împărţirilor succesive:

$$\begin{array}{lcll} r_{-1} & = & r_0q_1 + r_1, & 0 < r_1 < r_0 \\ r_0 & = & r_1q_2 + r_2, & 0 < r_2 < r_1 \\ \cdots & & & \\ r_{n-2} & = & r_{n-1}q_n + r_n, & 0 < r_n < r_{n-1} \\ r_{n-1} & = & r_nq_{n+1} + r_{n+1}, & r_{n+1} = 0, \end{array}$$

unde $r_{-1} = a$ şi $r_0 = b$. Atunci, conform celor menţionate mai sus, obţinem

$$\begin{array}{rcl} (a, b) & = & (r_{-1}, r_0) \\ & = & (r_0, r_1) \\ & \cdots & \\ & = & (r_{n-1}, r_n) \\ & = & (r_n, r_{n+1}) \\ & = & (r_n, 0) \\ & = & r_n. \end{array}$$

Să facem câteva observaţii importante asupra secvenţei de împărţiri de mai sus:

- numărul de împărţiri realizate de acest algoritm este $n + 1$;
- $q_i \geq 1$, pentru orice $1 \leq i \leq n$;
- $q_{n+1} \geq 2$ deoarece $r_n < r_{n-1}$.

Estimarea complexităţii algoritmului lui Euclid necesită estimarea lui n. In 1845, Gabriel Lamé a avut ideea de a compara resturile r_i cu termenii şirului lui Fibonacci $(F_i)_{i\geq 1}$ dat prin:

- $F_1 = 1 = F_2$;
- $F_n = F_{n-1} + F_{n-2}$, pentru orice $n \geq 3$.

Observăm că $r_n \geq 1 = F_2$ şi

$$r_{n-1} = r_nq_{n+1} \geq 2r_n \geq 2 = F_3.$$

Dacă presupunem că $r_{i+2} \geq F_{n-i}$ şi $r_{i+1} \geq F_{n-i+1}$, atunci

$$r_i = r_{i+1}q_{i+2} + r_{i+2} \geq r_{i+1} + r_{i+2} \geq F_{n-i+1} + F_{n-i} = F_{n-i+2},$$

[5] A fost prezentat de Euclid în volumul VI al lucrării lui, *Elements*.

pentru orice $i = n-2, \ldots, 0, -1$. De aici obţinem $b = r_0 \geq F_{n+2}$.

Fie $R = (1+\sqrt{5})/2$. Prin simplă inducţie matematică putem arăta că are loc $F_2 = R^0$ şi $F_{i+2} > R^i$, pentru orice $i \geq 1$. Obţinem atunci

$$b \geq F_{n+2} > R^n,$$

ceea ce conduce la

$$log_{10}\, b > n\, log_{10} R > \frac{n}{5},$$

ultima inegalitate urmând de la faptul că $log_{10} R = 0.208\cdots > 1/5$. Dacă numărul b necesită k cifre în scriere zecimală, atunci $b < 10^k$. Combinând cu inegalitatea de mai sus obţinem $n < 5k$, ceea ce conduce la $n+1 \leq 5k$. Am obţinut astfel:

Teorema 4.2.4. (G. Lamé, 1845)
Fie $a > b > 0$ numere naturale. Atunci, numărul de împărţiri necesare algoritmului lui Euclid pentru determinarea celui mai mare divizor comun al numerelor a şi b nu depăşeşte de 5 ori numărul de cifre din scrierea zecimală a lui b.

Utilizând acum inegalitatea $a \geq F_{n+3} > R^{n+1}$, obţinem

$$n+1 < log_R\, a,$$

ceea ce conduce la:

Teorema 4.2.5. Fie N un număr natural nenul. Atunci, pentru orice două numere naturale $a, b \leq N$, nu ambele nule, algoritmul lui Euclid aplicat acestor numere necesită cel mult $\lfloor log_R\, N \rfloor - 1$ paşi.

Estimarea de mai sus a numărului de paşi, atunci când avem de aplicat algoritmul lui Euclid, este suficient de bună prin aceea că aplicarea acestui algoritm numerelor F_{n+3} şi F_{n+2}, ce satisfac $a \geq F_{n+3}$ şi $b \geq F_{n+2}$, necesită de asemenea tot $n+1$ paşi de împărţire:

$$\begin{array}{lcll} F_{n+3} & = & 1 \cdot F_{n+2} + F_{n+1}, & 0 < F_{n+1} < F_{n+2} \\ F_{n+2} & = & 1 \cdot F_{n+1} + F_n, & 0 < F_n < F_{n+1} \\ \cdots & & & \\ F_4 & = & 1 \cdot F_3 + F_2, & 0 < F_2 < F_3 \\ F_3 & = & 2 \cdot F_2. & \end{array}$$

Aşa cum am văzut, cel mai mare divizor comun a două numere a şi b poate fi exprimat ca o combinţie liniară a acestora, $(a,b) = \alpha a + \beta b$. Există multe situaţii în care suntem interesaţi în a determina (algoritmic) o astfel de combinaţie liniară. Dacă analizăm secvenţa de împărţiri de mai sus, prin care se determină (a,b), constatăm următoarele:

$$\begin{array}{lcllcl} r_1 & = & a - bq_1 & = & 1 \cdot a + (-q_1) \cdot b & \\ r_2 & = & b - r_1 q_2 & = & (-q_2) \cdot a + (1 + q_1 q_2) \cdot b & \\ r_3 & = & r_1 - r_2 q_3 & = & (1 + q_2 q_3) \cdot a + (-q_1 - q_3 - q_1 q_2 q_3) \cdot b & \\ \cdots & & & & & \end{array}$$

Adică, odată cu determinarea restului (la un pas), putem determina şi combinaţia liniară a acestuia (în funcţie de a şi b). Ceea ce ne rămâne de făcut mai departe este de a găsi o metodă elegantă de exprimare a combinaţiei liniare a restului în baza combinaţiilor liniare de la paşii anteriori. Dacă fiecărui element x ce intervine în secvenţa de împărţiri de mai sus îi asociem un vector $V_x = (\alpha, \beta)$ ce furnizează combinaţia liniară (în funcţie de a şi b) a lui x, adică $x = \alpha a + \beta b$, atunci combinaţia liniară a resturilor se poate determina prin:

$$\begin{array}{rlcll} & & & & V_a = (1,0) \\ & & & & V_b = (0,1) \\ 1. & a & = & bq_1 + r_1 & V_{r_1} = V_a - q_1 V_b \\ 2. & b & = & r_1 q_2 + r_2 & V_{r_2} = V_b - q_2 V_{r_1} \\ 3. & r_1 & = & r_2 q_3 + r_3 & V_{r_3} = V_{r_1} - q_3 V_{r_2} \\ & \cdots & & & \\ n. & r_{n-2} & = & r_{n-1} q_n + r_n & V_{r_n} = V_{r_{n-2}} - q_n V_{r_{n-1}} \\ n+1. & r_{n-1} & = & r_n q_{n+1} & \end{array}$$

In acest mod putem determina atât (a, b) cât şi combinaţia liniară (de a şi b) a acestuia. Algoritmul pe care l-am obţinut se numeşte *algoritmul extins al lui Euclid*. Corectitudinea lui se demonstrează imediat în baza a ceea ce a fost menţionat mai sus, iar complexitatea acestuia este aceeaşi cu a algoritmului lui Euclid. Mai precis, la fiecare pas, pe lângă o împărţire se fac două înmulţiri (complexitatea unei înmulţiri fiind aceeaşi cu a unei împărţiri) şi două scăderi (complexitatea unei scăderi fiind liniară în raport cu lungimea maximă a reprezentării binare a operanzilor).

Posibilitatea determinării algoritmice a unei combinaţii liniare a celui mai mare divizor comun a două numere a şi b conduce la posibilitatea determinării algoritmice a unei soluţii a ecuaţiei $ax + by = c$, în ipoteza $(a, b)|c$ (dacă această relaţie nu este satisfăcută atunci ecuaţia nu are soluţii). In adevăr, fie α şi β astfel încât $\alpha a + \beta b = (a, b)$, şi fie c' astfel încât $c = (a, b)c'$. Atunci,

$$c'\alpha a + c'\beta b = (a, b)c' = c,$$

ceea ce ne arată că $x = c'\alpha$ şi $y = c'\beta$ constituie o soluţie a ecuaţiei $ax + by = c$.

Analizând algoritmul lui Euclid observăm că fracţia a/b poate fi scrisă:

$$\frac{a}{b} = q_1 + \frac{r_1}{b} = q_1 + \frac{1}{\frac{b}{r_1}} = q_1 + \frac{1}{q_2 + \frac{r_2}{r_1}} \cdots = q_1 + \frac{1}{q_2 + \frac{1}{q_3 + \cdots + \frac{1}{q_{k+1}}}}$$

Ultimul termen din acest şir de egalităţi poartă denumirea de *fracţie continuă asociată fracţiei* a/b, şi se mai notează prin $[q_1, \ldots, q_{k+1}]$ (a nu se confunda cu cel mai mic multiplu comun). Ca urmare, algoritmul lui Euclid conduce direct la următorul rezultat:

Teorema 4.2.6. Orice număr raţional admite o reprezentare ca fracţie continuă.

Ceea ce trebuie să remarcăm este că reprezentarea unui număr raţional ca fracţie continuă nu este unică. In adevăr, este uşor de vazut că are loc

$$[q_1, \ldots, q_n] = \begin{cases} [q_1, \ldots, q_{n-1}, q_n - 1, 1], & \text{dacă } q_n > 1, \\ [q_1, \ldots, q_{n-1} + 1], & \text{dacă } q_n = 1 \end{cases}$$

ceea ce ne arată că orice număr raţional are măcar două reprezentări ca fracţie continuă. Insă, reprezentarea numerelor iraţionale ca fracţie continuă este unică [73, 83].

Fie $[q_1, \ldots, q_n]$ o fracţie continuă ce reprezintă numărul raţional a/b. Fracţiile continue $[q_1, \ldots, q_i]$, unde $1 \leq i \leq n$, se mai numesc *convergentele* fracţiei continue $[q_1, \ldots, q_n]$. Dacă notăm prin a_i/b_i numărul raţional ce are reprezentarea $[q_1, \ldots, q_i]$ ca fracţie continuă, atunci următoarele proprietăţi sunt imediate.

Propoziţia 4.2.3. Fie $[q_1, \ldots, q_n]$ o fracţie continuă ce reprezintă fracţia a/b.

(1) Numerele raţionale a_i/b_i verifică relaţiile de recurenţă:

- $a_1 = q_1$ şi $b_1 = 1$;
- $a_2 = q_2 q_1 + 1$ şi $b_2 = q_2$;
- $a_i = q_i a_{i-1} + a_{i-2}$ şi $b_i = q_i b_{i-1} + b_{i-2}$,

pentru orice $3 \leq i \leq n$;

(2) Au loc relaţiile:

- $a_i b_{i-1} - a_{i-1} b_i = (-1)^{i-1}$, pentru orice $2 \leq i \leq n$;
- $\frac{a_i}{b_i} - \frac{a_{i-1}}{b_{i-1}} = \frac{(-1)^{i-1}}{b_i b_{i-1}}$, pentru orice $2 \leq i \leq n$;
- $a_i b_{i-2} - a_{i-2} b_i = (-1)^i a_i$, pentru orice $3 \leq i \leq n$;

(3) Pentru orice $3 \leq i \leq n$, are loc $b_i \geq b_{i-1} + 1$. Ca urmare, $b_n \geq n$ dacă $n \geq 3$;

(4) Pentru orice i astfel încât $2i + 1 \leq n$ au loc relaţiile:

- $\frac{a_{2i-1}}{b_{2i-1}} > \frac{a_{2i+1}}{b_{2i+1}}$;
- $\frac{a_{2i-2}}{b_{2i-2}} < \frac{a_{2i}}{b_{2i}}$;

(5) Orice convergentă a fracţiei continue $[q_1, \ldots, q_n]$ este ireductibilă.

4.3 Congruenţe

Fie m un număr întreg. Definim pe $\mathbf{Z}$ relaţia binară $\equiv_m$, numită *relaţia de congruenţă modulo* m sau *congruenţa modulo* m, prin:

$$a \equiv_m b \quad \Leftrightarrow \quad m|(a - b),$$

pentru orice $a, b \in \mathbf{Z}$. Dacă $a \equiv_m b$ atunci vom spune că a şi b sunt *congruente modulo* m, şi vom mai nota aceasta prin $a \equiv b \, mod \, m$.

Ne vom referi adesea la aceste congruenţe ca fiind *congruenţe modulare*. Ele apar frecvent în viaţa de zi cu zi. De exemplu, determinarea zilei săptămânii (luni, marţi etc.) ce va fi pe o anumită dată a anului face apel la împărţirea zilelor în grupe de câte 7 şi considerarea restului. Numărarea obiectelor unei mulţimi face iarăşi apel la împărţirea acestora în grupe, cel mai adesea de câte 10, şi apoi numărarea acestora (care, la rândul lor, pot fi numărate prin repetarea procedeului de împărţire în grupe). Problemele legate de congruenţe au fascinat omenirea de sute de ani. In secolul al 4-lea, autorul chinez Sun Tzu Suan Ching formula următoarea problema:

> "Avem un număr de obiecte, dar nu ştim câte. Dacă le numărăm câte 3, atunci ne rămân 2. Dacă le numărăm câte 5, atunci ne rămân 3. Dacă le numărăm câte 7, atunci ne rămân 2. Câte obiecte sunt ?".

Următoarele proprietăţi pot fi obţinute cu uşurinţă de la definiţia congruenţelor modulare şi de la proprietăţile relaţiei de divizibilitate.

Propoziţia 4.3.1. Fie a, b, c, d, m şi m' numere întregi şi $f : \mathbf{Z} \to \mathbf{Z}$ o funcţie polinomială cu coeficienţi întregi. Atunci, au loc următoarele proprietăţi:

(1) $\equiv_m$ este relaţie de echivalenţă pe $\mathbf{Z}$;

(2) $a \equiv_m b$ dacă şi numai dacă $a \; mod \; m = b \; mod \; m$;

(3) dacă $a \equiv_m b$, atunci $(a, m) = (b, m)$;

(4) (reguli de combinare)
dacă $a \equiv_m b$ şi $c \equiv_m d$, atunci $a + c \equiv_m b + d$, $a - c \equiv_m b - d$, $ac \equiv_m bd$ şi $f(a) \equiv_m f(b)$;

(5) (reguli de simplificare)

(a) dacă $ac \equiv_{mc} bc$ şi $c \neq 0$, atunci $a \equiv_m b$;

(b) dacă $ac \equiv_m bc$ şi $d = (m, c)$, atunci $a \equiv_{m/d} b$;

(c) dacă $ac \equiv_m bc$ şi $(m, c) = 1$, atunci $a \equiv_m b$;

(6) (reguli de descompunere şi compunere)

(a) dacă $a \equiv_{mm'} b$, atunci $a \equiv_m b$ şi $a \equiv_{m'} b$;

(b) dacă $a \equiv_m b$ şi $a \equiv_{m'} b$, atunci $a \equiv_{[m,m']} b$;

(c) dacă $a \equiv_m b$ şi $a \equiv_{m'} b$, iar m şi m' sunt prime între ele, atunci $a \equiv_{mm'} b$.

Demonstraţie Vom demonstra doar (6b). Presupunem că au loc relaţiile $a \equiv_m b$ şi $a \equiv_{m'} b$. Prima relaţie conduce la $m|(a - b)$, a doua la $m'|(a - b)$, de unde, în baza Propoziţiei 4.2.2, urmează $[m, m']|(a - b)$. □

Vom nota prin $\mathbf{Z}_m$ mulţimea claselor de echivalenţă induse de $\equiv_m$ (clasa de echivalenţă a lui $a \in \mathbf{Z}$ fiind notată prin $[a]_m$). Vom face în cele ce urmează câteva observaţii importante asupra acestor clase de echivalenţă:

- deoarece un număr întreg se divide la m dacă şi numai dacă se divide la $-m$, deducem că relaţiile de congruenţă modulo m şi $-m$ coincid. Ca urmare, putem considera numai relaţii de congruenţă modulo m pentru care $m \geq 0$;
- în cazul $m = 0$, $a \equiv b \, mod \, m$ dacă şi numai dacă $a = b$. Deci, orice element din $\mathbf{Z}$ induce o clasă de echivalenţă formată doar din el;
- în cazul $m \geq 1$, mulţimea $\mathbf{Z}_m$ are m elemente. In adevăr, oricare două numere distincte dintre numerele $0, \ldots, m-1$ nu sunt congruente modulo m deoarece diferenţa lor este diferită de 0 şi strict mai mică, în valoare absolută, decât m. Ca urmare, numerele $0, \ldots, m-1$ sunt în clase de echivalenţă diferite. In plus, în fiecare clasă de echivalenţă indusă de un element $n \in \mathbf{Z}$ se găseşte unul din cele m elemente de mai sus (în baza teoremei împărţirii cu rest).

Aşadar, relaţia de congruenţă modulo m împarte mulţimea $\mathbf{Z}$ în m clase de echivalenţă pentru care putem alege numerele $0, \ldots, m-1$ drept reprezentanţi de clasă, acestea fiind exact resturile posibile ale împărţirii numerelor întregi la m. Din acest motiv, clasele de echivalenţă induse de $\equiv_m$ mai sunt numite şi *clase de echivalenţă modulo* m sau *clase de resturi modulo* m. Uneori ele mai sunt notate prin $\mathbf{0}, \ldots, \mathbf{m-1}$.

Pe mulţimea $\mathbf{Z}_m$ introducem următoarele operaţii:

- $\oplus$, operaţie binară dată prin $[a]_m \oplus [b]_m = [a+b]_m$, pentru orice $a, b \in \mathbf{Z}$;
- $[0]_m$, operaţie 0-ară;
- $\ominus$, operaţie unară dată prin $\ominus[a]_m = [-a]_m$, pentru orice $a \in \mathbf{Z}$;
- $\otimes$, operaţie binară dată prin $[a]_m \otimes [b]_m = [ab]_m$, pentru orice $a, b \in \mathbf{Z}$;
- $[1]_m$, operaţie 0-ară.

Cu acestea, structura $(\mathbf{Z}_m, \oplus, \ominus, [0]_m)$ devine grup ciclic comutativ, în timp ce structura $(\mathbf{Z}_m, \oplus, \ominus, [0]_m, \otimes, [1]_m)$, inel comutativ cu unitate. Este de remarcat că pentru $m = 1$ acest inel este *trivial* în sensul că are doar un singur element, $[0]_1 = [1]_1$, care este element neutru atât pentru operaţia notată aditiv cât şi pentru operaţia notată multiplicativ.

Scăderea în inelul $\mathbf{Z}_m$ se defineşte prin

$$[a]_m \oplus (\ominus[b]_m),$$

notată simplificat $[a]_m \ominus [b]_m$, pentru orice $a, b \in \mathbf{Z}$. Ca urmare,

$$[a]_m \ominus [b]_m = [a-b]_m.$$

Conform teoremei împărţirii cu rest, orice număr întreg a se poate scrie în forma $a = qm + r$, unde $q, r \in \mathbf{Z}$ şi $0 \leq r < m$. Determinarea lui r va fi numită *reducerea modulo* m a lui a sau, în general, *reducerea modulară* a lui a.

Inelul $\mathbf{Z}_0$ este izomorf cu inelul $\mathbf{Z}$ şi, ca urmare, vom evita cazul $m = 0$ orientând studiile cu precădere asupra lui $\mathbf{Z}_m$ cu $m \geq 1$.

Considerând mulţimea $\mathbf{Z}'_m = \{0, \ldots, m-1\}$ înzestrată cu operaţiile

- $+'$, dată prin $a +' b = (a + b)\ mod\ m$, pentru orice $a, b \in \mathbf{Z}'_m$;
- 0, ca operaţie 0-ară;
- $-'$, dată prin $-'a = m - a\ mod\ m$, pentru orice $a \in \mathbf{Z}'_m$;
- $\cdot'$, dată prin $a \cdot' b = a \cdot b\ mod\ m$, pentru orice $a, b \in \mathbf{Z}'_m$;
- 1, ca operaţie 0-ară (0 şi 1 vor coincide în cazul $m = 1$),

constatăm că aceasta devine inel comutativ cu unitate, izomorf cu inelul $\mathbf{Z}_m$. Din acest motiv putem identifica inelul $\mathbf{Z}_m$ cu $\mathbf{Z}'_m$. Ca urmare, vom prefera să renotăm $\mathbf{Z}_m = \{0, \ldots, m-1\}$ şi operaţiile $\oplus$, $\ominus$ şi $\otimes$ prin $+$, $-$ şi, respectiv, $\cdot$ (aceasta din urmă fiind omisă atunci când nu există pericol de confuzie). Atragem însă atenţia că aceste operaţii, privite în $\mathbf{Z}_m$, sunt echivalente cu corespondentele lor în $\mathbf{Z}$ la care se adaugă şi reducerea modulară (a se vedea definiţiile operaţiilor $+'$, $-'$ şi $\cdot'$).

Menţionăm că nu orice element $a \in \mathbf{Z}_m$ are un invers multiplicativ [6] (de exemplu, $2 \in \mathbf{Z}_6$) dar, atunci când există el este unic. De asemenea, $\mathbf{Z}_m$ poate avea divizori ai lui 0 [7] (de exemplu, $2 \cdot 3 = 0$ în $\mathbf{Z}_6$).

Să vedem ce condiţii trebuie să satisfacă un element $a \in \mathbf{Z}_m$ pentru a avea un invers multiplicativ. Observăm că au loc echivalenţele

$$\begin{aligned}(\exists x \in \mathbf{Z}_m)(ax \equiv 1\ mod\ m) &\Leftrightarrow (\exists x \in \mathbf{Z}_m)(m|ax-1)\\ &\Leftrightarrow (\exists x, y \in \mathbf{Z})(ax - my = 1)\\ &\Leftrightarrow (a, m) = 1\end{aligned}$$

(ultima echivalenţă urmează de la Corolarul 4.2.2).

Ca urmare, elementele din $\mathbf{Z}_m$ care admit inverşi multiplicativi sunt exact acele elemente care sunt prime cu m. Fie $\mathbf{Z}_m^*$ mulţimea acestor elemente. Dacă notăm prin a^{-1} inversul multiplicativ al lui $a \in \mathbf{Z}_m^*$, atunci $(\mathbf{Z}_m^*, \cdot, ^{-1}, 1)$ devine grup comutativ, numit *grupul unităţilor* inelului $\mathbf{Z}_m$. Este uşor de văzut că $\mathbf{Z}_1^* = \{0\}$, caz în care $1 = 0$ (clasele de echivalenţă modulo 1 induse de 0 şi 1 coincid) [8].

Algoritmul extins al lui Euclid ne permite determinarea inversului multiplicativ modulo m al lui a. In adevăr, dacă $(a, m) = 1$ atunci, cu ajutorul algoritmului extins al lui Euclid, putem determina α şi β astfel încât $\alpha a + \beta m = 1$. De aici urmează cu uşurinţă că $\alpha\ mod\ m$ este inversul multiplicativ modulo m al lui a.

De exemplu, grupul unităţilor inelului $\mathbf{Z}_1$ coincide cu $\mathbf{Z}_1$, iar grupul unităţilor inelului $\mathbf{Z}_{26}$ are 12 elemente; acestea şi inversele lor sunt următoarele:

$$1^{-1} = 1,\ 3^{-1} = 9,\ 5^{-1} = 21,\ 7^{-1} = 15,\ 11^{-1} = 19,\ 17^{-1} = 23,\ 25^{-1} = 25.$$

[6] Invers relativ la operaţia de înmulţire.

[7] Elemente diferite de 0 al căror produs este 0.

[8] In multe tratate de teoria numerelor, $\mathbf{Z}_m^*$ se introduce ca fiind mulţimea numerelor strict pozitive ce nu depăşesc m şi care sunt prime cu m. In această variantă, $\mathbf{Z}_1^*$ este mulţimea formată doar din 1 (fiind aceeaşi cu $\mathbf{Z}_2^*$); pentru $m > 1$, această definiţie produce aceeaşi mulţime $\mathbf{Z}_m^*$ ca şi definiţia mai sus adoptată. Cum diferenţa dintre aceste două abordări diferă doar din punct de vedere a mulţimii $\mathbf{Z}_1^*$, vom prefera să mergem pe varianta deja adoptată prin care $\mathbf{Z}_m^*$ este grupul unităţilor inelului $\mathbf{Z}_m$, pentru orice $m \geq 1$.

4.4 Funcţia lui Euler

Reamintim că $\mathbf{Z}_m^* = \{a \in \mathbf{Z}_m | (a, m) = 1\}$, pentru orice $m \geq 1$. Funcţia ϕ ce asociază fiecărui număr $m \geq 1$ cardinalul mulţimii $\mathbf{Z}_m^*$ este numită *funcţia lui Euler*. Vom fi interesaţi în cele ce urmează de determinarea unei formule de evaluare a acestei funcţii. Observăm întâi că $\phi(1) = 1$ şi $\phi(p) = p - 1$, pentru orice număr prim p.

Teorema 4.4.1. Fie $m, m' \geq 1$ numere prime între ele şi $f : \mathbf{Z}_m \times \mathbf{Z}_{m'} \to \mathbf{Z}_{mm'}$ dată prin

$$f(a, a') = (ma' + m'a) \; mod \; mm',$$

pentru orice $a \in \mathbf{Z}_m$ şi $a' \in \mathbf{Z}_{m'}$. Atunci:

(1) funcţia f este bijecţie;

(2) restricţia funcţiei f la $\mathbf{Z}_m^* \times \mathbf{Z}_{m'}^*$ stabileşte o bijecţie între această mulţime şi $\mathbf{Z}_{mm'}^*$.

Demonstraţie (1) Mulţimile $\mathbf{Z}_m \times \mathbf{Z}_{m'}$ şi $\mathbf{Z}_{mm'}$ are acelaşi număr de elemente, şi anume mm'. Ca urmare, este suficient de arătat că f este funcţie injectivă. Fie deci $(a, a'), (b, b') \in \mathbf{Z}_m \times \mathbf{Z}_{m'}$. Presupunem că are loc $f(a, a') = f(b, b')$. Atunci,

$$ma' + m'a \equiv_{mm'} mb' + m'b,$$

de unde obţinem

$$m(a' - b') \equiv_{mm'} m'(b - a).$$

In baza Propoziţiei 4.3.1(6a) deducem

$$m(a' - b') \equiv_m m'(b - a)$$

şi

$$m(a' - b') \equiv_{m'} m'(b - a).$$

Prima relaţie combinată cu $(m, m') = 1$ conduce la $a \equiv_m b$. Cum $a, b \in \mathbf{Z}_m$, urmează $a = b$. Similar, a doua relaţie conduce la $a' = b'$. Deci, f este injectivă.

(2) In baza rezultatului de la (1) este suficient de arătat că are loc:

(a) dacă $a \in \mathbf{Z}_m^*$ şi $a' \in \mathbf{Z}_{m'}^*$ atunci $(ma' + m'a) \; mod \; mm'$ este în $\mathbf{Z}_{mm'}^*$;

(b) orice element din $\mathbf{Z}_{mm'}^*$ este de forma $(ma' + m'a) \; mod \; mm'$, unde $a \in \mathbf{Z}_m^*$ şi $a' \in \mathbf{Z}_{m'}^*$.

Vom demonstra (a) prin contradicţie. Adică, vom presupune că $ma' + m'a \; mod \; mm'$ şi mm' nu sunt relativ prime. Deci, ele vor avea un divizor comun $d > 1$. Este clar că d este divizor comun şi pentru numerele $ma' + m'a$ şi mm'.

Cum d divide mm' şi $(m, m') = 1$, putem presupune că $d|m$ sau $d|m'$, dar nu ambele. Presupunem că $d|m$ (celălalt caz este similar acestuia). Atunci, $(d, m') = 1$. Cum $d|ma' + m'a$, obţinem $d|m'a$ care combinată cu $(d, m') = 1$ conduce la $d|a$. Dar atunci, $(a, m) > 1$; contradicţie. Deci, $ma' + m'a \; mod \; mm' \in \mathbf{Z}_{mm'}^*$.

Demonstrăm acum (b). Fie $b \in \mathbf{Z}^*_{mm'}$. De la (1) urmează că există $a \in \mathbf{Z}_m$ şi $a' \in \mathbf{Z}_{m'}$ astfel încât $b = ma' + m'a \ mod \ mm'$. Vom arăta că $(m, a) = 1$ şi $(m', a') = 1$. Presupunem, prin contradicţie, că $(m, a) > 1$. Fie $d = (m, a)$. Atunci, d este divizor comun pentru $ma' + m'a$ şi mm', ceea ce ne arată că d este divizor comun pentru b şi mm' contrazicând $(b, mm') = 1$.

Similar se raţionează în cazul $(m', a') > 1$. Deci, (b) este demonstrată. □

Corolarul 4.4.1. Au loc următoarele proprietăţi:

(1) $\phi(ab) = \phi(a)\phi(b)$, pentru orice $a, b \geq 1$ prime între ele;

(2) dacă $a \geq 2$ este un număr natural a cărui descompunere în factori primi este $a = \prod p_i^{e_i}$, atunci
$$\phi(a) = \prod (p_i^{e_i} - p_i^{e_i - 1}).$$

Demonstraţie (1) Dacă $(a, b) = 1$, atunci $\mathbf{Z}^*_{ab}$ şi $\mathbf{Z}^*_a \times \mathbf{Z}^*_b$ sunt izomorfe (conform Teoremei 4.4.1(2)). Deci, cele două mulţimi au acelaşi cardinal. Adică, $\phi(ab) = \phi(a)\phi(b)$.

(2) In baza proprietăţii de la (1) şi a descompunerii în factori primi a oricărui număr natural $a \geq 2$, este suficient de arătat că are loc $\phi(p^e) = p^e - p^{e-1}$, pentru orice număr prim p şi orice număr natural $e \geq 1$.

Fie p şi e ca mai sus. Numerele mai mici decât p^e ce nu sunt prime cu p^e sunt exact multiplii lui p. Aceştia sunt

$$1 \cdot p, 2 \cdot p, \ldots, (p^{e-1} - 1) \cdot p.$$

Ca urmare,
$$\phi(p^e) = p^e - 1 - (p^{e-1} - 1) = p^e - p^{e-1},$$
ceea ce încheie demonstraţia. □

Teorema 4.4.2. (Teorema lui Euler)
Fie $m \geq 1$. Atunci, $a^{\phi(m)} \equiv 1 \ mod \ m$, pentru orice $a \in \mathbf{Z}^*_m$.

Demonstraţie Fie $a_1, \ldots, a_{\phi(m)}$ o enumerare a elementelor mulţimii $\mathbf{Z}^*_m$. Fie $a \in \mathbf{Z}^*_m$. Atunci, $aa_1, \ldots, aa_{\phi(m)}$ este, de asemenea, o enumerare a elementelor mulţimii $\mathbf{Z}^*_m$ (pentru orice $i \neq j$, aa_i şi aa_j nu sunt congruente modulo m). Ca urmare,
$$a_1 \cdots a_{\phi(m)} = (aa_1) \cdots (aa_{\phi(m)}) = a^{\phi(m)} a_1 \cdots a_{\phi(m)}.$$
Deoarece orice element din $\mathbf{Z}^*_m$ are un invers multiplicativ, relaţia de mai sus conduce la $a^{\phi(m)} \equiv 1 \ mod \ m$. □

Corolarul 4.4.2. Fie $m \geq 1$. Atunci $a^{\phi(m)} \equiv 1 \ mod \ m$, pentru orice $a \in \mathbf{Z}$ cu $(a, m) = 1$.

Demonstraţie Dacă $a \in \mathbf{Z}^*_m$, atunci corolarul urmează direct de la Teorema 4.4.2. Altfel, se utilizează Teorema împărţirii cu rest şi se aplică restului Teorema 4.4.2 (restul şi m sunt prime între ele). □

Corolarul 4.4.3. (Teorema lui Fermat)
Dacă p este un număr prim, atunci $a^{p-1} \equiv 1 \; mod \; p$, pentru orice $a \in \mathbf{Z}$ cu $p \not| a$.

Demonstraţie Dacă $p \not| a$ atunci $(a, p) = 1$. Corolarul urmează atunci de la Corolarul 4.4.2 şi faptul că $\phi(p) = p - 1$. □

Corolarul 4.4.3 poate fi formulat echivalent astfel.

Corolarul 4.4.4. (Teorema lui Fermat)
Dacă p este un număr prim, atunci $a^p \equiv a \; mod \; p$, pentru orice $a \in \mathbf{Z}$.

Demonstraţie Dacă $p|a$, atunci $a^p \equiv_m 0 \equiv_m a$. Altfel, $a^{p-1} \equiv 1 \; mod \; p$ care combinată cu $a \equiv a \; mod \; m$ conduce la $a^p \equiv a \; mod \; p$. □

Dacă presupunem că p este prim şi $p \not| a$, atunci relaţia $a^p \equiv a \; mod \; p$ conduce, în baza Propoziţiei 4.3.1(5c), la $a^{p-1} \equiv 1 \; mod \; p$. Deci, Corolarul 4.4.4 implică Corolarul 4.4.3. Cum în demonstrarea Corolarului 4.4.4 s-a utilizat Corolarul 4.4.3, deducem că afirmaţiile din cele două corolare sunt echivalente.

Teorema 4.4.3. Fie $m \geq 1$. Atunci, $\sum_{d|m} \phi(d) = m$ (d este subânţeles ca fiind divizor pozitiv al lui m deoarece funcţia ϕ este definită numai pentru numere strict pozitive).

Demonstraţie Fie $A = \{1, \ldots, m\}$ şi $A_d = \{a \in A | (a, m) = d\}$, pentru orice $1 \leq d \leq m$. Este clar că $A_d \cap A_{d'} = \emptyset$, pentru orice $d \neq d'$, şi $A = \bigcup_{d=1}^{m} A_d$. Ca urmare,

$$m = |A| = \sum_{d=1}^{m} |A_d|.$$

Fie $1 \leq d \leq m$. Dacă $d \not| m$, atunci $A_d = \emptyset$. Dacă $d|m$ atunci $A_d \neq \emptyset$ şi

$$\begin{aligned} |A_d| &= |\{a \in A | (a, m) = d\}| \\ &= |\{a' | 1 \leq a' \leq m/d, \; (a', m/d) = 1\}| \\ &= \phi(m/d). \end{aligned}$$

Atunci,

$$m = |A| = \sum_{d=1}^{m} |A_d| = \sum_{d|m} |A_d| = \sum_{d|m} \phi(m/d) = \sum_{d|m} \phi(d),$$

ceea ce încheie demonstraţia teoremei. □

4.5 Rădăcini primitive

Fie $m \geq 1$. Am văzut în Secţiunea 4.3 că un element $a \in \mathbf{Z}_m$ are un invers multiplicativ dacă şi numai dacă $(a, m) = 1$. Atunci, $\mathbf{Z}_m^* = \{a \in \mathbf{Z}_m | (a, m) = 1\}$ în raport cu înmulţirea claselor de resturi devine grup comutativ. Deoarece în $\mathbf{Z}_m$ există $\phi(m)$ numere prime cu m (Secţiunea 4.4), ordinul grupului $\mathbf{Z}_m^*$ este $\phi(m)$.

Vom nota prin $ord_m(a)$ ordinul elementului $a \in \mathbf{Z}_m^*$ şi ne vom referi la el ca fiind *ordinul lui a modulo m* [9]. In cazul $m = 1$, $\mathbf{Z}_1^* = \{0\}$ şi $ord_1(0) = 1$.

Cititorul poate demonstra cu uşurinţă următoarele proprietăţi.

Propoziţia 4.5.1. Fie $m \geq 1$ and $a \in \mathbf{Z}_m^*$. Atunci, au loc următoarele proprietăţi:

(1) $ord_m(a) = min\{k \geq 1 | a^k \equiv 1 \ mod \ m\}$;

(2) Dacă $a^k \equiv 1 \ mod \ m$, atunci $ord_m(a)|k$. In particular, $ord_m(a)|\phi(m)$;

(3) $ord_m(a) = \phi(m)$ dacă şi numai dacă $a^{\phi(m)/q} \not\equiv 1 \ mod \ m$, pentru orice factor prim q al lui $\phi(m)$;

(4) $a^k \equiv a^l \ mod \ m$ dacă şi numai dacă $k \equiv l \ mod \ ord_m(a)$;

(5) elementele $a^0 \ mod \ m, \ a^1 \ mod \ m, \ldots, a^{ord_m(a)-1} \ mod \ m$ sunt distincte două câte două;

(6) $ord_m(a^k) = ord_m(a)/(k, ord_m(a))$, pentru orice $k \geq 1$;

(7) dacă $ord_m(a) = d_1 d_2$, atunci $ord_m(a^{d_1}) = d_2$.

Corolarul 4.5.1. Fie $m \geq 1$ and $a, b \in \mathbf{Z}_m^*$. Dacă $ord_m(a)$ si $ord_m(b)$ sunt prime între ele, atunci $ord_m(ab \ mod \ m) = ord_m(a) ord_m(b)$.

$\mathbf{Z}_m$ cu adunarea formează grup ciclic, pentru orice $m \geq 1$. Nu acelaşi lucru se poate afirma despre $\mathbf{Z}_m^*$ cu înmulţirea. Atunci când $\mathbf{Z}_m^*$ este grup ciclic (în raport cu înmulţirea), generatorii lui mai sunt numiţi *rădăcini primitive modulo m* [10]. Este clar că $a \in \mathbf{Z}_m^*$ este rădăcină primitivă modulo m dacă şi numai dacă $ord_m(a) = \phi(m)$. 0 este (singura) rădăcină primitivă modulo 1.

Cititorul poate verifica cu uşurintă că are loc:

Propoziţia 4.5.2. Fie $m \geq 1$ şi $a \in \mathbf{Z}_m^*$. Atunci, au loc următoarele proprietăţi:

(1) a este rădăcină primitivă modulo m dacă şi numai dacă $ord_m(a) = \phi(m)$;

(2) a este rădăcină primitivă modulo m dacă şi numai dacă are loc

$$(\forall q)(q \text{ factor prim } q \text{ al lui } \phi(m) \ \Rightarrow \ a^{\phi(m)/q} \not\equiv 1 \ mod \ m);$$

(3) dacă a este rădăcină primitivă modulo m atunci, pentru orice $k \geq 1$, a^k este rădăcină primitivă modulo m dacă şi numai dacă $(k, \phi(m)) = 1$;

(4) dacă există rădăcini primitive modulo m, atunci există numărul acestora este $\phi(\phi(m))$.

Determinarea valorilor lui m pentru care grupul $\mathbf{Z}_m^*$ este ciclic a fost rezolvată de către Gauss. Rezultatul nu este trivial şi îl prezentăm fără demonstraţie.

Teorema 4.5.1. Grupul $\mathbf{Z}_m^*$ este ciclic dacă şi numai dacă $m$este de forma 1, 2, 4, p^k sau $2p^k$, unde p este număr prim iar $k \geq 1$.

[9] Notaţia $ord_m(a)$ a fost introdusă de Gauss în *Disquisitiones Arithmeticae* publicată în 1801 [60].

[10] Termenul de *rădăcină primitivă* a fost introdus de Euler în 1773. Demonstraţia lui, conform căreia există rădăcini primitive modulo p, pentru orice număr prim p, s-a dovedit a fi eronată. Gauss a furnizat mai multe demonstraţii corecte acestui rezultat.

4.6 Problema logaritmului discret

Fie G un grup ciclic finit şi a un generator al său. Atunci,

$$G = \{a^0 = e, a^1, \ldots, a^{|G|-1}\},$$

unde $a^i \neq a^j$ pentru orice $0 \leq i, j < |G|$ cu $i \neq j$.

Dacă $b \in G$, atunci există $k < |G|$ astfel încât $b = a^k$. Numărul k se numeşte *indexul* sau *logaritmul discret al lui b în grupul G relativ la a*. Dacă $G = \mathbf{Z}_m^*$, atunci k se mai numeşte şi *indexul* sau *logaritmul discret al lui b modulo m relativ la a*.

Determinarea algoritmică a indexului k poartă denumirea de *problema logaritmului discret*. In informatică, această problemă se enunţă astfel:

Problema logaritmului discret
Instanţă: un grup ciclic finit G, un generator a al său şi $b \in G$;
Intrebare: determinaţi $k < |G|$ astfel încât $b = a^k$.

Soluţia cea mai la îndemână de rezolvare a acestei probleme este de a enumera elementele grupului G, în forma

$$a^0, a^1, a^2, \ldots, a^{|G|-1}$$

până se găseşte a^k ce este b. Această soluţie necesită, în cazul cel mai nefavorabil, parcurgerea a $|G| - 1$ paşi în care, exceptând primii doi paşi, restul paşilor necesită o înmulţire cu a şi o comparaţie a rezultatului cu b.

Dacă $|G|$ este foarte mare, de exemplu $|G| \geq 10^{100}$, atunci această soluţie nu poate fi pusă în practică.

Un algoritm de complexitate $\mathcal{O}(\sqrt{|G|})$ a fost propus de Daniel Shanks în [166]. Ideea acestuia se bazează pe următoarele. Fie k cu $0 \leq k < |G|$ şi $n = \lfloor\sqrt{|G|}\rfloor$. Atunci, k poate fi scris în forma

$$k = qn + r,$$

unde $0 \leq q, r < n$ (folosind teorema împărţirii cu rest şi observând că $q < n$). Ca urmare, determinarea numărului k cu proprietatea $b = a^k$ se poate reduce la determinarea a două numere q şi r ce satisfac $0 \leq q, r < n$ şi $b = a^{nq+r}$. Dacă rescriem egalitatea $b = a^{nq+r}$ în forma $a^r = b(a^{-n})^q$, atunci problema noastră se reduce la a găsi două numere q şi r ce satisfac $0 \leq q, r < n$ şi $a^r = b(a^{-n})^q$. Determinarea acestor numere se poate face astfel:

- se calculează valorile $a^0, a^1, \ldots, a^{n-1}$ şi se stochează într-o listă L;
- se calculează succesiv $b, ba^{-n}, b(a^{-n})^2, \ldots, b(a^{-n})^{n-1}$ şi, pentru fiecare valoare calculată se verifică dacă aceasta se găseşte în lista L. Apartenenţa unei astfel de valori la lista L determină automat r şi q, întrerupând totodată şi procesul de generare.

Algoritmul sugerat este următorul.

Algoritmul Baby-step Giant-step
`input`: un grup ciclic finit G, a un generator al său şi $b \in G$;
`output`un număr k astfel încât $0 \leq k < |G|$ şi $b = a^k$;
`begin`
1. $n := \sqrt{|G|}$;
2. `calculează` o listă $L := \{(r, a^r) | 0 \leq r < n\}$ şi `sorteaz-o` după a doua componentă;
3. `calculează` a^{-n} (folosind eventual a^{n-1} din lista L);
4. $c := b$;
5. `for` $q := 0$ `to` $n - 1$ `do`
6. `if` $(\exists r)((r, c) \in L)$
7. `then begin` $k := qn + r$; `quit end`
8. `else` $c := ca^{-n}$;

`end`.

Algoritmul de mai sus necesită $\mathcal{O}(n)$ spaţiu pentru memorarea listei L. Calculul acesteia necesită $\mathcal{O}(n)$ înmulţiri. Complexitatea sortării ei este $\mathcal{O}(n \log n)$ (ea poate fi sortată o dată cu generarea fiecărei noi perechi ce urmează a fi adaugată listei). Precalculând lista L, pasul 3 necesită o înmulţire şi o determinare de invers, iar pasul 5 necesită $\mathcal{O}(n)$ înmulţiri şi $\mathcal{O}(n)$ căutări în tabelă.

Denumirea algoritmului provide de la faptul că determinarea elementelor listei se face prin înmulţiri cu a (*baby-steps*), iar reactualizarea lui c se face prin înmulţiri cu a^{-n} (*giant-steps*).

Incheiem secţiunea prin remarca că nu se cunoaşte nici o soluţie polinomială (în raport cu $\log |G|$) de rezolvare a acestei probleme. Problema este considerată intractabilă pentru valori mari ale lui $|G|$ (adică, algoritmii existenţi nu pot fi utilizaţi în practică), ceea ce o face destul de potrivită pentru dezvoltarea de tehnici criptografice bazate pe ea. De exemplu, semnătura digitală ElGamal îşi bazează securitatea pe această problemă (cine doreşte să atace cu succes semnătura ElGamal, este pus în situaţia rezolvării unei instanţe a acestei probleme.).

4.7 Ecuaţii congruenţiale

Ecuaţiile congruenţiale, similare ecuaţiilor clasice, joacă un rol deosebit de important în aritmetică, structura grupurilor ciclice, criptografie etc. Vom prezenta mai jos câteva elemente de bază asupra acestora.

Definiţia 4.7.1. Fie $f(x) = a_n x^n + \cdots a_0$ un polinom cu coeficienţi întregi şi $m \geq 2$.

(1) Spunem că f are *gradul* n *modulo* m dacă $a_n \not\equiv 0 \; mod \; m$.

(2) Spunem că $c \in \mathbf{Z}$ este *rădăcină modulo* m *a lui* f sau că este *soluţie* a ecuaţiei $f(x) \equiv 0 \; mod \; m$ dacă $f(c) \equiv 0 \; mod \; m$.

Ecuaţiile de forma $f(x) \equiv 0 \; mod \; m$ vor fi numite *ecuaţii congruenţiale*. Dacă gradul polinomului f este n modulo m, atunci vom mai spune că ecuaţia congruenţială este de *grad* n sau *ordin* n. Ecuaţiile congruenţiale de grad 1 vor mai fi numite şi *ecuaţii congruenţiale liniare*.

O primă ecuaţie congruenţială a fost implicit întâlnită în Teorema 4.4.2,

$$x^{\phi(m)} - 1 \equiv 0 \; mod \; m.$$

Conform Teoremei 4.4.2, această ecuaţie are $\phi(m)$ soluţii distincte (ne-congruente) modulo m (vom spune mai simplu că această ecuaţie are $\phi(m)$ soluţii modulo m).

Să ne întreptăm atenţia spre ecuaţii congruenţiale liniare $ax + b \equiv 0 \; mod \; m$ în necunoscuta x, unde $a, b \in \mathbf{Z}$. Conform definiţiei congruenţei modulo m, a determina o soluţie a acestei ecuaţii revine la a determina un cuplu (x, y) de numere întregi astfel încât

$$ax + (-m)y = -b.$$

Apelând la Corolarul 4.2.2, această ultimă ecuaţie admite soluţie (în x şi y) dacă şi numai dacă $(a, m)|b$. Am obţinut astfel următorul rezultat important.

Teorema 4.7.1. Fie $a, b, m \in \mathbf{Z}$ cu $m \geq 2$. Atunci, ecuaţia $ax \equiv b \, mod \, m$ are soluţii în $\mathbf{Z}$ dacă şi numai dacă $(a, m)|b$. In plus, dacă această ecuaţie are soluţii, atunci ea are exact (a, m) soluţii în $\mathbf{Z}_m$, ce sunt de forma

$$(x_0 + im/(a, m)) \; mod \; m,$$

unde x_0 este o soluţie (arbitrară dar fixată) a acestei ecuaţii şi $0 \leq i < (a, m)$.

Demonstraţie Conform observaţiei de mai sus, ne rămâne de demonstrat doar partea a doua a acestei teoreme.

Presupunem că ecuaţia $ax \equiv b \; mod \; m$ are soluţii şi, fie x_0 o soluţiei a ei. Vom arăta că numerele din teoremă sunt soluţii, distincte două câte două, ale ecuaţiei şi, reciproc, orice soluţie a ecuaţiei este de forma menţionată în teoremă.

Fie d cel mai mare divizor comun al numerelor a şi m. Prin simplă verificare deducem că pentru orice $i \in \{0, \ldots, d-1\}$,

$$(x_0 + im/d) \; mod \; m$$

este soluţie a acestei ecuaţii şi, orice două astfel de soluţii sunt distincte.

Fie $c \in \mathbf{Z}_m$ o soluţie a ecuaţiei $ax \equiv b \; mod \; m$. Relaţiile $ac \equiv b \; mod \; m$ şi $ax_0 \equiv b \; mod \; m$ conduc la $ac \equiv ax_0 \; mod \; m$, iar în baza Propoziţiei 4.3.1(5b) obţinem $c \equiv x_0 \; mod \; m/d$. Ca urmare, există un număr $i \geq 0$ astfel încât $c = (x_0 + im/d) \; mod \; m$. Deoarece

$$c = (x_0 + im/d) \; mod \; m = (x_0 + (i \; mod \; m)m/d) \; mod \; m,$$

deducem că numărul i poate fi ales satisfăcând $0 \leq i \leq d-1$. Deci, orice soluţie din $\mathbf{Z}_m$ a ecuaţiei $ax \equiv b \, mod \, m$ este de forma

$$(x_0 + im/(a, m)) \; mod \; m,$$

unde $0 \leq i < d$. □

Corolarul 4.7.1. Fie $a, m \in \mathbf{Z}$ cu $a \neq 0$ şi $m \geq 2$. Atunci, următoarele afirmaţii sunt echivalente:

(1) ecuaţia $ax \equiv 1 \; mod \; m$ are soluţie în $\mathbf{Z}_m$, şi în acest caz ea este unică în $\mathbf{Z}_m$;

(2) $(a, m) = 1$;

(3) a admite un invers multiplicativ în $\mathbf{Z}_m$, ce este unic în $\mathbf{Z}_m$.

Soluţiile modulo m ale ecuaţiilor $ax \equiv b \; mod \; m$, atunci când $(a, m)|b$, se obţin astfel. Fie $d = (a, m)$ şi $\alpha, \beta \in \mathbf{Z}$ astfel încât $a\alpha + m\beta = d$ (α şi β determinate cu ajutorul algoritmului extins al lui Euclid). Considerând $b = db'$, relaţia de mai sus conduce la $a\alpha b' + m\beta b' = db'$, de unde deducem că are loc $a\alpha b' \equiv b \; mod \; m$. Deci, $\alpha b' \; mod \; m$ este soluţie în $\mathbf{Z}_m$ a ecuaţiei $ax \equiv b \; mod \; m$. Celelalte soluţii în $\mathbf{Z}_m$ se obţin pe baza relaţiei din Teorema 4.7.1.

Teorema 4.7.2. (Teorema lui Lagrange)
Orice ecuaţie congruenţială de grad n modulo p, unde p este un număr prim, are cel mult n soluţii (ne-congruente) modulo p.

Demonstraţie Fie p un număr prim. Vom face demonstraţia prin inducţie matematică după gradul ecuaţiei.

Fie $f(x) = a_1x + a_0$ un polinom de gradul 1 modulo p. Ca urmare, $(a_1, p) = 1$ şi, atunci, Teorema 4.4.1 ne spune că ecuaţia $f(x) \equiv 0 \; mod \; p$ are exact o soluţie modulo p.

Presupunem afirmaţia din teoremă adevărată pentru ecuaţii congruenţiale modulo p de grad $n \geq 1$, şi fie $f(x) = a_{n+1}x^{n+1} + a_nx^n + \cdots + a_0$ un polinom de gradul $n + 1$ modulo p. Presupunem, prin contradicţie, că ecuaţia $f(x) \equiv 0 \; mod \; p$ are cel puţin $n + 2$ soluţii (ne-congruente) modulo p, şi fie $c_0, \ldots, c_{n+1}$ soluţii distincte (ne-congruente) modulo p ale acesteia. Atunci,

$$f(x) - f(c_0) = (x - c_0)g(x),$$

unde $g(x)$ este un polinom de grad cel mult n cu proprietatea că coeficientul lui x^n în g este a_{n+1}. Cum $a_{n+1} \not\equiv 0 \; mod \; p$, deducem că g are gradul n modulo p.

Arătăm acum că $c_1, \ldots, c_{n+1}$ sunt rădăcini (ne-congruente) modulo p ale lui g, ceea ce va constitui o contradicţie cu ipoteza inductivă şi, deci, afirmaţia noastră va fi falsă. Fie $1 \leq i \leq n + 1$. Deoarece $f(c_i) \equiv 0 \; mod \; p$, obţinem

$$f(c_i) - f(c_0) = (c_i - c_0)g(c_i) \equiv 0 \; mod \; p.$$

c_i şi c_0 nu sunt congruente modulo p şi, atunci, relaţia de mai sus conduce la $g(c_i) \equiv 0 \; mod \; p$. Deci, c_i este rădăcină modulo p a lui g. Cum $c_1, \ldots, c_{n+1}$ nu sunt congruente modulo p (două câte două), deducem că g ar avea cel puţin $n + 1$ rădăcini (ne-congruente) modulo p. Ca urmare afirmaţia făcută este falsă şi, deci, teorema este demonstrată. □

In Teorema 4.7.2, dacă nu se care ca p să fie număr prim, atunci concluzia acesteia ar putea să nu fie adevărată (a se vedea Exemplul 4.8.1(2)).

Corolarul 4.7.2. Fie p un număr prim şi d un divizor al lui $p-1$. Atunci, ecuaţia $x^d - 1 \equiv 0 \; mod \; p$ are d soluţii (ne-congruente) modulo p.

Demonstraţie Fie $p-1 = de$. Atunci,

$$x^{p-1} - 1 = (x^d - 1)(x^{d(e-1} + \cdots + x^d + 1) = (x^d - 1)g(x).$$

Conform teoremei lui Euler, $x^{p-1}-1$ are exact $p-1$ rădăcini (ne-congruente) modulo p, iar relaţia de mai sus ne spune că orice rădăcină a acestui polinom este rădăcină pentru $x^d - 1$ sau $g(x)$.

Polinomul $g(x)$ are cel mult $d(e-1)$ rădăcini (ne-congruente) modulo p (Teorema 4.7.2). Combinând cu relaţia de mai sus, deducem că $x^d - 1$ trebuie să aibă cel puţin $(p-1) - d(e-1) = d$ rădăcini (ne-congruente) modulo p. Aplicând Teorema 4.7.2 polinomului $x^d - 1$, deducem că el are cel mult d rădăcini (ne-congruente) modulo p. Ca urmare, acest polinom trebuie să aibă exact d rădăcini (ne-congruente) modulo p. □

4.8 Teorema chineză a resturilor

Problema lui Sun Tzu Suan Ching, menţionată la începutul Secţiunii 4.3, poate fi formalizată astfel: determinaţi $x \in \mathbf{Z}$ astfel încât acesta să verifice simultan congruenţele

$$\begin{cases} x \equiv 2 \; mod \; 3 \\ x \equiv 3 \; mod \; 5 \\ x \equiv 2 \; mod \; 7 \end{cases}$$

Ca urmare, suntem conduşi la rezolvarea unor sisteme de ecuaţii congruenţiale liniare. Menţionăm încă de la început că astfel de sisteme pot să aibă sau să nu aibă soluţie. De exemplu, sistemul de mai sus admite soluţia $x = 23$, dar sistemul

$$\begin{cases} x \equiv 0 \; mod \; 3 \\ x \equiv 1 \; mod \; 6 \end{cases}$$

nu are soluţii (altfel, ar exista $\alpha, \beta \in \mathbf{Z}$ astfel încât $3\alpha = x$ şi $6\beta = x - 1$, ceea ce ar conduce la $3\alpha - 6\beta = 1$ care nu admite soluţii în α şi β deoarece $3 \not| 1$).

Să considerăm cazul unui sistem cu trei ecuaţii,

$$\begin{cases} x \equiv b_1 \; mod \; m_1 \\ x \equiv b_2 \; mod \; m_2 \\ x \equiv b_3 \; mod \; m_3, \end{cases}$$

şi să analizăm posibilităţile de a obţine soluţii pentru acesta.

In primul rând, putem porni de la ideea determinării unei soluţii x_i pentru fiecare ecuaţie şi de a combina aceste soluţii în una singură x astfel încât, prin reducere modulo m_i, x să devină soluţie pentru a i-a ecuaţie. O variantă naturală de a combina x_1, x_2 şi x_2 ar fi prin

$$x = m_2 m_3 x_1 + m_1 m_3 x_2 + m_1 m_2 x_3.$$

Atunci, x este soluţie a primei ecuaţii dacă şi numai dacă

$$m_2 m_3 x_1 \equiv b_1 \; mod \; m_1.$$

Ca urmare, x_1 ales la început nu trebuie să fie soluţie a primei ecuaţii a sistemului ci a ecuaţiei

$$m_2 m_3 x \equiv b_1 \; mod \; m_1,$$

ceea ce este posibil dacă şi numai dacă $(m_2 m_3, m_1)|b_1$.

In mod similar, x_2 trebuie să fie soluţie a ecuaţiei

$$m_1 m_3 x \equiv b_2 \; mod \; m_2,$$

ceea ce este posibil dacă şi numai dacă $(m_1 m_3, m_2)|b_2$, iar x_3 trebuie să fie soluţie a ecuaţiei

$$m_1 m_2 x \equiv b_3 \; mod \; m_3,$$

ceea ce este posibil dacă şi numai dacă $(m_1 m_2, m_3)|b_3$.

Aceste observaţii conduc la următorul rezultat foarte important.

Teorema 4.8.1. (Teorema chineză a resturilor)
Fie $k \geq 1$ un număr natural şi $m_1, \ldots, m_k$ numere întregi prime între ele două câte două. Atunci, pentru orice $b_1, \ldots, b_k \in \mathbf{Z}$, sistemul de ecuaţii

$$\begin{cases} x \equiv b_1 \; mod \; m_1 \\ \cdots \\ x \equiv b_k \; mod \; m_k \end{cases}$$

admite o unică soluţie modulo $m_1 \cdots m_k$.

Demonstraţie Fie $m = m_1 \cdots m_k$. Fie $c_i = m/m_i$, pentru orice $1 \leq i \leq k$. Deoarece m_i este relativ prim cu oricare m_j, $j \neq i$, deducem că m_i este prim şi cu c_i. In baza Teoremei 4.7.2, ecuaţia

$$c_i x \equiv b_i \; mod \; m_i$$

admite soluţii, pentru orice $1 \leq i \leq k$; fie x_i o astfel de soluţie.

Este imediat de verificat că $x = c_1 x_1 + \cdots + c_k x_k$ este soluţie a sistemului, iar $x \; mod \; m$ este soluţie în $\mathbf{Z}_m$.

Vom arăta că $y \equiv x \; mod \; m$, pentru orice altă soluţie y a sistemului. Fie y o soluţie a sistemului. Deoarece $y \equiv b_i \; mod \; m_i$ şi $x \equiv b_i \; mod \; m_i$ pentru orice $1 \leq i \leq k$, deducem $y \equiv x \; mod \; m_i$ pentru orice $1 \leq i \leq k$, care combinate cu $(m_i, m_j) = 1$ pentru orice $i \neq j$ conduc la $y \equiv x \; mod \; m$. □

Unica soluţie $x \in \mathbf{Z}_m$ a sistemului din Teorema chineză a resturilor poate fi efectiv determinată utilizând, de exemplu, algoritmul extins al lui Euclid.

Teorema chineză a resturilor poate fi generalizată natural în următoarele două variante.

Corolarul 4.8.1. Fie $k \geq 2$ un număr natural şi $a_1, \ldots, a_k, m_1, \ldots, m_k$ numere întregi astfel încât $(a_i, m_i) = 1$ şi $(m_i, m_j) = 1$, pentru orice $1 \leq i, j \leq k$ cu $i \neq j$. Atunci, pentru orice $b_1, \ldots, b_k \in \mathbf{Z}$, sistemul de ecuaţii

$$\begin{cases} a_1 x \equiv b_1 \ mod \ m_1 \\ \cdots \\ a_k x \equiv b_k \ mod \ m_k \end{cases}$$

admite o unică soluţie modulo $m_1 \cdots m_k$.

Demonstraţie Deoarece $(a_i, m_i) = 1$ pentru orice $1 \leq i \leq k$, deducem că sistemul

$$\begin{cases} a_1 x \equiv b_1 \ mod \ m_1 \\ \cdots \\ a_k x \equiv b_k \ mod \ m_k \end{cases}$$

este echivalent cu sistemul

$$\begin{cases} x \equiv a_1^{-1} b_1 \ mod \ m_1 \\ \cdots \\ x \equiv a_k^{-1} b_k \ mod \ m_k \end{cases}$$

ce admite soluţie unică modulo $m_1 \cdots m_k$ în baza Teoremei chineze a resturilor. □

Teorema 4.8.2. Fie $k \geq 1$ un număr natural şi $b_1, \ldots, b_k, m_1, \ldots, m_k$ numere întregi. Atunci, sistemul de ecuaţii

$$\begin{cases} x \equiv b_1 \ mod \ m_1 \\ \cdots \\ x \equiv b_k \ mod \ m_k \end{cases}$$

admite soluţii dacă şi numai dacă $b_i \equiv b_j \ mod \ (m_i, m_j)$, pentru orice $1 \leq i, j \leq k$ cu $i \neq j$. In plus, dacă acest sistem admite soluţii, atunci aceasta este unică modulo $[m_1, \ldots, m_k]$.

Demonstraţie Vom demonstra teorema pentru cazul particular $k = 2$.

Dacă sistemul de mai sus admite o soluţie, fie aceasta $x = a$, atunci

$$a \equiv b_1 \ mod \ m_1 \ \text{ şi } \ a \equiv b_2 \ mod \ m_2.$$

De aici urmează

$$a \equiv b_1 \ mod \ (m_1, m_2) \ \text{ şi } \ a \equiv b_2 \ mod \ (m_1, m_2),$$

ceea ce conduce la $b_1 \equiv b_2 \ mod \ (m_1, m_2)$.

Reciproc, presupunem că $(m_1, m_2) | b_1 - b_2$. Orice soluţie a primei ecuaţii este de forma $x = b_1 + m_1 y$, unde $y \in \mathbf{Z}$. Cerinţa ca o astfel de soluţie să verifice şi cea de a doua ecuaţie conduce la problema existenţei unui $y \in \mathbf{Z}$ astfel încât

$$b_1 + m_1 y \equiv b_2 \ mod \ m_2$$

sau, altfel spus, la existenţa unei soluţii (în y) a ecuaţiei

$$m_1 y \equiv b_1 - b_2 \, mod \, m_2.$$

Cum $(m_1, m_2)|b_1 - b_2$, deducem că această ecuaţie admite soluţii (în y). Ca urmare, sistemul admite soluţii.

Să presupunem acum că sistemul admite soluţii şi să arătăm că orice două soluţii sunt congruente modulo $[m_1, m_2]$. Fie deci a şi a' două soluţii. Din faptul că acestea trebuie să verifice prima ecuaţie obţinem

$$a \equiv a' \, mod \, m_1.$$

Similar, $a \equiv a' \, mod \, m_2$. Aceste două relaţii conduc la $a \equiv a' \, mod \, [m_1, m_2]$, în baza Propoziţiei 4.3.1(6b). □

O aplicaţie importantă a teoremei chineze a resturilor constă în determinarea numărului de soluţii ale ecuaţiilor de forma

$$f(x) \equiv 0 \, mod \, m_1 \cdots m_k,$$

unde f este un polinom cu coeficienţi întregi, iar $m_1, \ldots, m_k$ sunt numere naturale prime între ele două câte două.

Teorema 4.8.3. Fie f un polinom cu coeficienţi întregi şi $m_1, \ldots, m_k$ numere naturale prime între ele două câte două, unde $k \geq 2$. Atunci, un număr $a \in \mathbf{Z}$ este soluţie a ecuaţiei

$$f(x) \equiv 0 \, mod \, m_1 \cdots m_k,$$

dacă şi numai dacă a este soluţie a fiecărei ecuaţii

$$f(x) \equiv 0 \, mod \, m_i,$$

$1 \leq i \leq k$. In plus, numărul soluţiilor în $\mathbf{Z}_{m_1 \cdots m_k}$ ale ecuaţiei

$$f(x) \equiv 0 \, mod \, m_1 \cdots m_k$$

este egal cu produsul numerelor de soluţii în $\mathbf{Z}_{m_i}$ ale ecuaţiilor

$$f(x) \equiv 0 \, mod \, m_i,$$

unde $1 \leq i \leq k$.

Demonstraţie Este clar că $a \in \mathbf{Z}$ este soluţie a ecuaţiei

$$f(x) \equiv 0 \, mod \, m_1 \cdots m_k$$

dacă şi numai dacă este soluţie a ecuaţiilor

$$f(x) \equiv 0 \, mod \, m_i,$$

$1 \leq i \leq k$.

Fie a_i soluţie în $\mathbf{Z}_{m_i}$ a ecuaţiei

$$f(x) \equiv 0 \; mod \; m_i,$$

$1 \leq i \leq k$. Sistemul

$$\begin{cases} x \equiv a_1 \; mod \; m_1 \\ \cdots \\ x \equiv a_k \; mod \; m_k \end{cases}$$

admite o unică soluţie în $\mathbf{Z}_{m_1 \cdots m_k}$ (conform Teoremei chineze a resturilor). Mai mult, se verifică cu uşurinţă că aceasta este soluţie a ecuaţiei

$$f(x) \equiv 0 \; mod \; m_1 \cdots m_k.$$

Ca urmare, orice k-uplu de soluţii $(a_1, \ldots, a_k) \in \mathbf{Z}_{m_1} \times \cdots \times \mathbf{Z}_{m_k}$ a ecuaţiilor

$$f(x) \equiv 0 \; mod \; m_i,$$

$1 \leq i \leq k$, conduce la o unică soluţie în $\mathbf{Z}_{m_1 \cdots m_k}$ a ecuaţiei

$$f(x) \equiv 0 \; mod \; m_1 \cdots m_k,$$

şi reciproc, pentru orice soluţie a a acestei ecuaţii, $a \; mod \; m_i$ este soluţie în $\mathbf{Z}_{m_i}$ a ecuaţiei

$$f(x) \equiv 0 \; mod \; m_i,$$

$i \leq i \leq k$. Aceasta încheie demonstraţia celei de a doua părţi a teoremei. □

Exemplul 4.8.1.

(1) Fie p un număr prim. Ecuaţia $x^2 \equiv 1 \; mod \; p$ are exact 2 soluţii în $\mathbf{Z}_p$, şi anume $x = 1$ şi $x = p - 1$. In adevăr,

$$\begin{array}{rcl} x^2 \equiv 1 \; mod \; p & \Leftrightarrow & p | x^2 - 1 \\ & \Leftrightarrow & p | (x-1)(x+1) \\ & \Leftrightarrow & p | x - 1 \text{ sau } p | x + 1 \\ & \Leftrightarrow & x \equiv 1 \; mod \; p \text{ sau } x \equiv -1 \; mod \; p. \end{array}$$

(2) Fie $p_1, \ldots, p_k$ numere prime distincte, $k \geq 2$. Atunci, ecuaţia

$$x^2 \equiv 1 \; mod \; p_1 \cdots p_k$$

are exact 2^k soluţii în $\mathbf{Z}_{p_1 \cdots p_k}$.

4.9 Complexitatea operaţiilor

4.9.1 Ordine de mărime

Analiza eficienţei algoritmilor nu este întotdeauna un lucru simplu, fiind adesea foarte dificil de determinat timpul exact de execuţie al unui algoritm. In astfel de situaţii suntem forţaţi în determinarea unei aproximări a timpului de execuţie şi, frecvent, putem determina doar aproximări asimptotice.

In această secţiune vom prezenta principalele *ordine de mărime* prin intermediul cărora vom putea discuta despre comportarea asimptotică a algoritmilor. Reamintim că prin $\mathbf{R}_+$ ($\mathbf{R}_+^*$) s-a notat mulţimea numerelor reale pozitive (strict pozitive).

Fie g o funcţie de la $\mathbf{N}$ la $\mathbf{R}_+$. Considerăm următoarele mulţimi:

$$\begin{aligned}
\mathcal{O}(g) &= \{f : \mathbf{N} \to \mathbf{R}_+ | (\exists c \in \mathbf{R}_+^*)(\exists n_0 \in \mathbf{N})(\forall n \geq n_0)(f(n) \leq cg(n))\} \\
\Omega(g) &= \{f : \mathbf{N} \to \mathbf{R}_+ | (\exists c \in \mathbf{R}_+^*)(\exists n_0 \in \mathbf{N})(\forall n \geq n_0)(cg(n) \leq f(n))\} \\
\Theta(g) &= \{f : \mathbf{N} \to \mathbf{R}_+ | (\exists c_1, c_2 \in \mathbf{R}_+^*)(\exists n_0 \in \mathbf{N})(\forall n \geq n_0) \\
&\qquad\qquad (c_1 g(n) \leq f(n) \leq c_2 g(n))\} \\
o(g) &= \{f : \mathbf{N} \to \mathbf{R}_+ | (\forall c \in \mathbf{R}_+^*)(\exists n_0 \in \mathbf{N})(\forall n \geq n_0)(f(n) \leq cg(n))\}
\end{aligned}$$

Definiţia 4.9.1.1. Fie f şi g funcţii de la $\mathbf{N}$ la $\mathbf{R}_+$, şi $X \in \{\mathcal{O}, \Omega, \Theta, o\}$. Spunem că *$f$ este de ordinul X al lui g*, şi notăm $f(n) = X(g(n))$, dacă $f \in X(g)$.

Notaţia "$\mathcal{O}$" a fost introdusă de Paul Bachmann în 1894 [4] fiind apoi popularizată intens de Edmund Landau [99, 100], în timp ce notaţia "o" îi este datorată lui Landau [99] [11].

Intuitiv, "$f(n) = \mathcal{O}(g(n))$" înseamnă că f nu creşte mai repede, din punct de vedere asimptotic, decât g (eventual multiplicată printr-o constantă). Atragem atenţia asupra notaţiei "$f(n) = \mathcal{O}(g(n))$"; ea nu trebuie gândită ca o egalitate, ci ca apartenenţa funcţiei f la mulţimea $\mathcal{O}(g)$. Cititorul s-ar putea arăta nedumerit, şi pe bună dreptate, de adoptarea a încă unei notaţii (cea prin "=") atâta timp cât notaţia prin "$\in$" este clară şi corectă din punct de vedere formal. Adoptarea acestei noi notaţii este datorată faptului că aceasta este încetăţenită în rândul matematicienilor şi informaticienilor.

Evident, o notaţie de genul $f(n) \neq \mathcal{O}(g(n)$ înseamnă că f nu este de ordinul $\mathcal{O}$ al lui g.

Următoarea propoziţie, a cărei demonstraţie urmează cu uşurinţă de la definiţii, prezintă câteva din proprietăţile de bază ale ordinelor de mărime.

Propoziţia 4.9.1.1. Fie f, g, h şi k funcţii de la $\mathbf{N}$ la $\mathbf{R}_+$. Atunci, au loc următoarele proprietăţi:

(1) $f(n) = \mathcal{O}(f(n))$;

[11] Toate aceste notaţii pot fi considerate într-un cadru mai general, cel al funcţiilor definite pe $\mathbf{R}$ cu valori în $\mathbf{R}$. Pentru necesităţile noastre, varianta deja considerată este suficientă.

(2) dacă $f(n) = \mathcal{O}(g(n))$ şi $g(n) = \mathcal{O}(h(n))$, atunci $f(n) = \mathcal{O}(h(n))$;

(3) $f(n) = \mathcal{O}(g(n))$ dacă şi numai dacă $g(n) = \Omega(f(n))$;

(4) $f(n)) = \Theta(g(n))$ dacă şi numai dacă $f(n) = \mathcal{O}(g(n))$ şi $f(n) = \Omega(g(n))$;

(5) dacă $f(n) = \mathcal{O}(h(n))$ şi $g(n) = \mathcal{O}(k(n))$, atunci $(f \cdot g)(n) = \mathcal{O}(h(n)k(n))$ şi $(f + g)(n) = \mathcal{O}(max\{h(n), k(n)\})$;

(6) dacă există $n_0 \in \mathbf{N}$ astfel încât $g(n) \neq 0$ pentru orice $n \geq n_0$, atunci $f(n) = o(g(n))$ dacă şi numai dacă $lim_{n\to\infty}\frac{f(n)}{g(n)} = 0$.

Următoarele inegalităţi sunt foarte utile în stabilirea ordinelor de mărime pentru diverse funcţii:

- (Formula lui Stirling)

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n+1}} \leq n! \leq \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n}},$$

  pentru orice $n \geq 1$;

- pentru orice constante reale ϵ şi c astfel încât $0 < \epsilon < 1 < c$ are loc

$$1 < ln\, ln\, n < ln\, n < e^{\sqrt{(ln\, n)(ln\, ln\, n)}} < n^\epsilon < n^c < n^{ln\, n} < c^n < n^n < c^{c^n}$$

  (fiecare inegalitate este satisfăcută pentru orice $n \geq n_0$, unde n_0 este ales convenabil pentru fiecare inegalitate).

Astfel, putem obţine:

- dacă f este un polinom de grad k cu coeficienţi reali dar astfel încât coeficientul termenului de grad maxim este pozitiv iar funcţia polinomială asociată (notată tot prin f) ia valori reale pozitive, atunci $f(n) = \Theta(n^k)$;
- pentru orice constantă reală $c > 1$, $log_c\, n = \Theta(log\, n)$ (log reprezintă funcţia logaritm în baza 2);
- pentru orice număr real ϵ cu $0 < \epsilon < 1$, $log\, n = \mathcal{O}(n^\epsilon)$;
- pentru orice număr natural $k \geq 1$, $log^k\, n = \mathcal{O}(n)$;
- $n! = \Omega(2^n)$ şi $n! = o(n^n)$;
- $log(n!) = \Theta(nlog\, n)$ [12];

[12] De fapt, observăm că $lim_{n\to\infty}\frac{n!}{\left(\frac{n}{e}\right)^n\sqrt{2\pi n}} = 1$, ceea ce ne arată că $\left(\frac{n}{e}\right)^n\sqrt{2\pi n}$ constituie o şi mai bună aproximare asimptotică pentru $n!$. Astfel, este interesant de remarcat că, pentru $n = 100$, $e^{\frac{1}{1201}} \approx 1.00083299$ şi $e^{\frac{1}{1200}} \approx 1.00083368$. Ca urmare, formula lui Stirling oferă o margine superioară pentru aproximarea lui $n!$ mai mare de $(1 + 10^{-6})$ ori decât marginea inferioară.

- dacă $f : \mathbf{N} \to \mathbf{R}_+$ este o funcţie astfel încât există $n_0 \in \mathbf{N}$ cu proprietatea $f(n) \geq 1$ pentru orice $n \geq n_0$, atunci

$$\frac{1}{2}2^{\lceil log_2 f(n) \rceil} \leq f(n) \leq 2^{\lceil log_2 f(n) \rceil}$$

pentru orice $n \geq n_0$, ceea ce conduce la $f(n) = \Theta(2^{\lceil log_2 f(n) \rceil})$.

Atragem însă atenţia aupra unor situaţii de genul $4^n \neq \mathcal{O}(2^n)$ [13].

Dacă $\mathcal{A}$ şi $\mathcal{B}$ sunt mulţimi de funcţii ca cele definite mai sus ($\mathcal{O}(g)$ etc.) iar f este o funcţie de la $\mathbf{N}$ la $\mathbf{R}_+$, atunci vom nota:

- $f + \mathcal{A} = \{f + g | g \in \mathcal{A}\}$;
- $\mathcal{A} + \mathcal{B} = \{f + g | f \in \mathcal{A},\ g \in \mathcal{B}\}$;
- $f\mathcal{A} = \{f \cdot g | g \in \mathcal{A}\}$. Dacă f este funcţia constantă c, atunci vom scrie $c\mathcal{A}$ în loc de $f\mathcal{A}$;
- $\mathcal{A}\mathcal{B} = \{fg | f \in \mathcal{A},\ g \in \mathcal{B}\}$;
- $\mathcal{O}(\mathcal{A}) = \bigcup_{f \in \mathcal{A}} \mathcal{O}(f)$.

Egalitatea $\mathcal{A} = \mathcal{B}$ va fi înţeleasă prin incluziune (pentru orice funcţie f, dacă $f \in \mathcal{A}$ atunci $f \in \mathcal{B}$). Această ultimă convenţie este în strânsă legătură cu convenţia deja adoptată ($f(n) = \mathcal{O}(g(n))$, de exemplu). Astfel, $\mathcal{O}(f(n)) = \mathcal{O}(g(n))$ ne spune că orice funcţie ce este de ordinul $\mathcal{O}$ al lui f este şi de ordinul $\mathcal{O}$ al lui g (dar nu în mod necesar şi invers), iar $f(n) + \mathcal{O}(g(n)) = \mathcal{O}(h(n))$ ne spune că suma dintre f şi o funcţie de ordinul $\mathcal{O}$ al lui g este o funcţie de ordinul $\mathcal{O}$ al lui h. De exemplu,

$$\frac{1}{3}n^3 + \mathcal{O}(n^2) = \mathcal{O}(n^3).$$

Putem accepta şi notaţii de genul $f(n) = g(n) + \mathcal{O}(h(n))$ pentru a specifica faptul că f este suma dintre g şi o funcţie de ordinul $\mathcal{O}$ al lui h (sau, altfel spus, f este un element al mulţimii $g + \mathcal{O}(h)$).

Următoarea propoziţie urmează cu uşurinţă de la definiţii (dar atragem încă o dată atenţia asupra faptului că egalitatea dintre mulţimile noastre de funcţii desemnează de fapt incluziune de la stânga la dreapta).

Propoziţia 4.9.1.2. Fie f şi g funcţii de la $\mathbf{N}$ la $\mathbf{R}_+$ şi $c \in \mathbf{R}_+$. Atunci, au loc următoarele proprietăţi:

(1) $\mathcal{O}(f(n)) + \mathcal{O}(g(n)) = \mathcal{O}(f(n) + g(n))$;

(2) $c\mathcal{O}(f(n)) = \mathcal{O}(f(n))$;

(3) $\mathcal{O}(\mathcal{O}(f(n))) = \mathcal{O}(f(n))$;

(4) $\mathcal{O}(f(n))\mathcal{O}(g(n)) = \mathcal{O}(f(n)g(n))$;

(5) $\mathcal{O}(f(n)g(n)) = f(n)\mathcal{O}(g(n))$.

[13] Dacă presupunem, prin contradicţie, că $4^n = \mathcal{O}(2^n)$, atunci există o constantă reală $c > 0$ şi un număr natural n_0 astfel încât $4^n \leq c2^n$, pentru orice $n \geq n_0$. De aici urmează $2^n \leq c$, pentru orice $n \geq n_0$, ceea ce este fals.

4.9.2 Timpul de execuţie al unui algoritm

Estimarea "timpului" necesar execuţiei unui algoritm se realizează în raport cu o anumită unitate de măsură a datelor de intrare. Cel mai adesea se ia în considerare "lungimea" reprezentării datelor de intrare (a operanzilor) într-o bază b [14] [15]. Menţionăm întâi că orice număr natural n poate fi *reprezentat în baza* $b \geq 2$ în forma

$$n = n_{k-1}b^{k-1} + \cdots + n_0,$$

unde $0 \leq n_i < b$ pentru orice $0 \leq i < k$, şi $n_{k-1} \neq 0$. Această reprezentare este complet determinată de secvenţa de numere

$$(n_{k-1}, \ldots, n_0)_b,$$

motiv pentru care această secvenţă va fi numită reprezentarea în baza b a lui n. Numărul $k > 0$ este numit *lungimea reprezentării* lui n în baza b (sau *lungimea* lui n, atunci când baza b este subânţeleasă din context), n_{k-1} se numeşte *cifra cea mai semnificativă*, iar n_0 *cifra cea mai puţin semnificativă* a reprezentării lui n în baza b. Atunci când $k = 1$ spunem că n este *număr în precizie simplă*; altfel, n este un *număr în precizie multiplă*.

Relaţia dintre n şi k este dată prin

$$k = \lfloor \log_b n \rfloor + 1$$

(cu convenţia $\log_b 0 = 0$). Ca urmare, funcţia $f(n)$ ce furnizează lungimea reprezentării în baza b a lui n satisface relaţia $f(n) = \Theta(\log_b n)$.

Atunci când $b \leq 10$, notaţia în baza b va fi simplificată la $(n_{k-1} \cdots n_0)_b$. Putem face această simplificare şi pentru $10 < b \leq 35$ dacă folosim, de exemplu, alfabetul englez şi renotăm numerele 10, 11 etc. prin A, B etc., în această ordine. Astfel, $(1A2)_{16}$ constituie aceeaşi reprezentare în baza 16 ca şi secvenţa $(1, 10, 2)_{16}$. Atunci când vom lucra cu baza 2 sau 10 vom folosi o nouă simplificare, obţinută prin eliminarea parantezelor şi a indicelui bazei. Cifrele utilizate pentru scrierea unui număr în baza 2 sunt numite *cifre binare* sau *biţi*. *Secvenţele binare* sunt secvenţe de biţi. Uneori este convenabil să completăm la stânga cu zerouri reprezentarea în baza b a lui n. Ne vom referi la şirurile obţinute ca fiind tot reprezentarea în baza b a lui n (această convenţie are caracter pur tehnic).

Presupunem că cititorul este familiarizat cu operaţiile uzuale de adunare, scădere, înmulţire şi împărţire cu numere scrise într-o bază b. Acestea se realizează prin repetarea de un număr finit de ori a următoarelor operaţiilor considerate primitive:

- compararea a două cifre ale bazei;

[14] O *bază de numeraţie* este un număr natural $b \geq 2$. Numerele i, cu $0 \leq i < b$, sunt numite cifrele bazei b.

[15] Reprezentarea internă a datelor în calculator se face utilizând codificarea binară a acestora. Operaţiile cu date sunt astfel convertite în operaţii cu şiruri binare (deplasare la stânga sau dreapta cu o poziţie, adunare de şiruri binare etc.). Ca urmare, complexitatea executării de către calculator a unor operaţii cu anumite tipuri de date se rezumă la complexitatea realizării unor operaţii cu şiruri binare, aceasta din urmă fiind "măsurată" în raport cu lungimea şirurilor.

- adunare, scădere şi înmulţire a 2 cifre ale bazei, luând în considerare şi transportul, având drept răspuns o cifră a bazei şi un transport;
- împărţire a unui număr format din două cifre ale bazei la o cifră a bazei, având drept răspuns un cât şi un rest (ambele fiind cifre ale bazei).

Regulile de bază folosite în evaluarea complexităţii unui algoritm sunt următoarele:

- complexitatea unei structuri repetitive (de tip *for*, *while*, *until*) este dată de complexitatea testului la care se adaugă complexitatea maximă a corpului respectivei structuri, şi înmulţind totul cu numărul de repetări a acestuia;
- complexitatea unei structuri secvenţiale este complexitatea maximă a componentelor structurii;
- complexitatea structurii *if-then-else* (*if-then*) este dată de complexitatea testului la care se adaugă complexitatea maximă a ramurilor (a ramurii).

Vom trece acum în revistă complexitatea realizării operaţiilor de bază cum ar fi adunarea, scăderea, înmulţirea etc. Vom urma cu precădere [185] unde cititorul poate găsi detalii complete asupra algoritmilor discutaţi sumar mai jos şi a complexităţii acestora.

Adunarea şi scăderea a două numere cu reprezentare binară pe cel mult k biţi pot fi realizate în complexitate timp $\mathcal{O}(k)$.

Inmulţirea, făcută "şcolăreşte", poate fi realizată în complexitate timp $\mathcal{O}(k^2)$. Dacă însă se utilizează algoritmul Karatsuba, complexitatea timp devine $\mathcal{O}(k^{\log 3})$, iar dacă se utilizează transformata Fourier discretă, complexitatea scade dramatic la $\mathcal{O}(k \log k)$. Cum însă înmulţirea realizată prin transformata Fourier discretă este eficientă doar pentru valori mari ale lui k (de obicei, cel puţin 1000), în practică, algoritmul Karatsuba se dovedeşte de preferat.

Impărţirea realizată prin algoritmul uzual de împărţire necesită complexitate timp $\mathcal{O}(k^2)$. Există şi o variantă recursivă a împărţirii, bazată oarecum pe ideea ce stă la baza algoritmului Karatsuba, ce necesită doar $\mathcal{O}(k^{\log 3} + k \log k)$.

O analiză simplistă a algoritmului lui Euclid ne arată că complexitatea acestuia nu depăşeşte $\mathcal{O}(k^3)$. Insă, o analiză atentă conduce la un rezultat mult mai bun. Impărţirea lui r_i la r_{i+1} cu obţinerea câtului q_{i+2} şi a restului r_{i+2}, utilizând notaţiile din Secţiunea 4.2, se poate realiza în $\mathcal{O}((\log r_{i+1})(\log q_{i+2}))$. Atunci,

$$\begin{array}{rcl} \sum_{i=-1}^{n-1}(\log\ r_{i+1})(\log\ q_{i+2}) & \leq & (\log\ b)\sum_{i=-1}^{n-1}\log\ q_{i+2} \\ & = & (\log\ b)(\log\ q_1 \cdots q_{n+1}) \end{array}$$

Nu este greu de văzut că

$$q_1 \cdots q_{n+1} \leq a,$$

ceea ce conduce la complexitatea timp $\mathcal{O}((\log\ a)(\log\ b))$. Deci, dacă numerele se reprezintă pe cel mult k biţi, atunci complexitatea algoritmului (extins al) lui Euclid este $\mathcal{O}(k^2)$. Există şi alţi algoritmi de calcul al celui mai mare divizor comun a două numere. Cea mai eficientă soluţie este de complexitate $\mathcal{O}(k^2/\log k)$.

Exponenţierea modulară, adică calculul lui $a^n \bmod m$, necesită complexitate timp $\mathcal{O}(k^3)$, presupunând că a, n şi m se reprezintă pe cel mult k biţi. Există multe soluţii pentru această problemă, depinzând de diverse particularităţi ale exponentului sau modulului, soluţii de complexitate mult mai bună decât cea menţionată mai sus. Toate acestea pot fi găsite în [185].

Atunci când pentru o problemă nu se cunoaşte nici un algoritm de complexitate timp polinomială iar algoritmii existenţi sunt inpracticabili pentru valori rezonabile ale datelor de intrare, vom spune că problema este *dificilă* sau *grea* sau *intractabilă*. De exemplu, factorizarea numerelor este o problemă dificilă. Pentru a înţelege ce înseamnă aceasta aducem la cunoştinţa cititorului următorul rezultat. Pe data de 9 mai 2005, o echipă a Agenţiei Federale Germane pentru securitatea informaţiei, compusă din F. Bahr, M. Boehm, J. Franke şi T. Kleinjung, a anunţat factorizarea unui număr de 200 de cifre, număr cunoscut sub denumirea de RSA-200. Acest număr face parte dintr-o selecţie de numere propuse de compania americană *RSA Security*, numere ce au exact 2 factori primi şi care sunt considerate ca fiind dificil de factorizat ("pietre de încercare" pentru problema facorizării). Echipa germană a utilizat pentru factorizarea acestui număr o reţea de calculatoare ce au lucrat în paralel. Timpul CPU necesar factorizării acestui număr folosind un procesor AMD Opteron la 2.2 GHz ar fi fost de aproximativ 55 de ani (a se vedea rubrica "challenges" la `http://www.rsasecurity.com/rsalabs//wiki/RSA-200`). Oricum, echipa germană a început lucrul la sfârşitul anului 2003 şi factorizarea s-a încheiat cu succes în mai 2005.

4.10 Aplicaţii: partajarea secretelor

Vom arăta în această secţiune cum, Teorema chineză a resturilor, în principal, poate fi utilizată în definirea unei scheme de partajare a secretelor.

Presupunem că o activitate ce impune păstrarea anumitor secrete (de exemplu, deservirea unei filiale a unei bănci) necesită un număr n de persoane (angajaţi), şi o anumită subramură a acestei activităţi poate fi realizată de oricare k persoane din cele n, dar nu mai puţin de k (de exemplu, deschiderea seifului băncii). Ca urmare, vom presupune că are loc $2 \leq k \leq n$. Presupunem în continuare că realizarea acestei subactivităţi necesită cunoaşterea unei anumite parole (de acces). Această parolă trebuie "partajată" între cele n persoane în aşa fel încât:

– oricare k persoane din cele n să poată reface parola (în mod unic şi în timp "eficient");

– orice grup de mai puţin de k persoane din cele n să nu poată reface parola (unic) în timp "eficient".

Cum putem alege o astfel de parolă, şi cum o putem partaja astfel încât să fie satisfăcute deziderateleе de mai sus ?

Inainte de a da un răspuns problemei de mai sus facem menţiunea că parola poate fi întotdeauna codificată numeric astfel că o putem presupune a fi un număr natural

S. Fie $2 \leq k \leq n$ numere naturale. O *schemă de partajare a secretelor* constă dintr-un $(n+1)$-uplu $(S; I_1, \ldots, I_n)$ de numere naturale astfel încât:

- cunoaşterea a cel puţin k numere distincte $I_{i_1}, \ldots, I_{i_k}$ conduce la determinarea "eficientă" a lui S;
- cunoaşterea a mai puţin de k numere dintre $I_1, \ldots, I_n$ nu permite determinarea "uşoară" a lui S (schema este rezistentă la atacuri de coaliţie de dimensiune cel mult $(k-1)$).

S se numeşte *secretul* sau *parola* schemei, iar I_j, *parole* sau *secrete parţiale*.

Aşa cum constatăm, o soluţie a problemei descrise la începutul secţiunii ar putea consta în construcţia unei scheme de partajare a secretelor. Vom prezenta în cele ce urmează o variantă de construcţie a unei astfel de scheme, propusă de Maurice Mignotte în 1983 ([130]). Ea se bazează pe ceea ce numim azi (k,n)-*secvenţe Mignotte*, ce sunt secvenţe

$$m_1 < \cdots < m_n$$

de n numere naturale relativ prime două câte două, cu proprietatea suplimentară

$$m_1 \cdots m_k > m_{n-k+2} \cdots m_n$$

(produsul celor mai mici k numere îl depşeşte pe cel al celor mai mari $(k-1)$ numere).

De exemplu, secvenţa de numere prime

$$11, 13, 17, 19, 23, 29, 31, 37$$

constituie o $(4,8)$-secvenţă Mignotte.

Teorema 4.10.1. ([130]) Pentru orice $2 \leq k \leq n$ şi orice număr natural γ, există şi se poate construi efectiv o (k,n)-secvenţă Mignotte

$$m_1 < \cdots < m_n$$

astfel încât $(\alpha - \beta)/\beta \geq \gamma$, unde $\alpha = m_1 \cdots m_k$ şi $\beta = m_{n-k+2} \cdots m_n$.

Această teoremă ne spune că putem determina secvenţe Mignotte astfel încât diferenţa dintre α şi β, ponderată prin β, să fie oricât de mare (folosind notaţiile din teoremă).

Fie $m_1, \ldots, m_n$ o (k,n)-secvenţă Mignotte. Alegem arbitrar S cu proprietatea $\beta < S < \alpha$ şi determinăm I_i prin

$$I_i = S \bmod m_i,$$

pentru orice $1 \leq i \leq n$.

Vom arăta acum că $(n+1)$-uplul $(S; I_1, \ldots, I_n)$ este o (k,n)-schemă de partajare a secretelor.

Fie $I_{i_1}, \ldots, I_{i_k}$ secrete parţiale distincte. In baza Teoremei chineze a resturilor, sistemul

$$(*) \begin{cases} x & \equiv & I_{i_1} \ mod \ m_{i_1} \\ & \cdots & \\ x & \equiv & I_{i_k} \ mod \ m_{i_k} \end{cases}$$

admite o unică soluţie modulo $m_{i_1} \cdots m_{i_k}$. Mai mult, această soluţie este chiar S (ea este soluţie din modul în care s-au definit numerele I_i, iar din modul de alegere a lui ei avem $S < m_1 \cdots m_k \leq m_{i_1} \cdots m_{i_k}$). Determinarea lui S, pornind de la $I_{i_1}, \ldots, I_{i_k}$ se face în timp polinomial.

Pe de altă parte, oricare $(k-1)$ secrete parţiale $I_{i_1}, \ldots, I_{i_{k-1}}$ conduc, în baza Teoremei chineze a resturilor, la o unică soluţie modulo $m_{i_1} \cdots m_{i_{k-1}}$ a sistemului

$$(**) \begin{cases} x & \equiv & I_{i_1} \ mod \ m_{i_1} \\ & \cdots & \\ x & \equiv & I_{i_{k-1}} \ mod \ m_{i_{k-1}} \end{cases}$$

S este soluţie a sistemului dar, deoarece $S > m_{n-k+2} \cdots m_n \geq m_{i_1} \cdots m_{i_{k-1}}$, S nu este în $\mathbf{Z}_{m_{i_1} \cdots m_{i_{k-1}}}$. Singura metodă de determinare a ei, cunoscând doar $I_{i_1}, \ldots, I_{i_{k-1}}$ constă în:

- determinarea unicei soluţii din $\mathbf{Z}_{m_{i_1} \cdots m_{i_{k-1}}}$ a sistemului $(**)$, fie aceasta S';
- determinarea elementelor din intervalul (β, α) ce sunt congruente cu S' modulo $m_{i_1} \cdots m_{i_{k-1}}$, şi verificarea fiecăruia ca o posibilă parolă.

Intre β şi α există aproximativ $(\alpha - \beta)/\beta$ numere congruente cu S' (numerele dintre β şi α, în ordine crescătoare, furnizează prin împărţire la β resturile $0, \ldots, \beta - 1$, ciclic). In baza Teoremei 4.10.1, secvenţa Mignotte poate fi aleasă astfel încât numărul $(\alpha - \beta)/\beta$ să fie oricât de mare, ceea ce va asigura securitate schemei (de exemplu, alegerea schemei Mignotte astfel încât $(\alpha - \beta)/\beta > 10^{100}$ asigură securitate schemei).

4.11 Aplicaţii: criptografie cu chei publice

In această secţiune vom puncta câteva din aplicaţiile majore ale teoriei grupurilor, şi în special a grupurilor ciclice, în criptografie. Prezentarea noastră va urma [187].

4.11.1 Introducere în criptografie

Criptosistem. Meoda fundamentală de securizare a comunicaţiei între două părţi A şi B faţă de o terţă parte C constă în stabilirea, de comun acord de către A şi B, a unui *sistem de criptare* (*cifrare*) a mesajelor. Un astfel de sistem trebuie să ofere posibilitatea criptării mesajelor ce urmează a fi transmise şi, totodată, a decriptării "corecte" a mesajelor (criptate) recepţionate. De exemplu, A şi B pot conveni ca fiecare literă a mesajului să fie înlocuită ciclic cu litera aflată la distanţa k de litera în cauză (considerând alfabetul ordonat în mod uzual). Astfel, pentru $k = 2$, litera a

s-ar înlocui cu c, litera b cu d etc. Vom spune în acest caz că *cheia de criptare* este $k = 2$; ea poate fi schimbată la fiecare nouă comunicare (cu condiţia de a putea fi stabilită de A şi B în deplină securitate).

Mesajele originale, ce urmează a fi criptate şi transmise, se construiesc pe baza unei mulţimi finite de elemente "atomice", mulţime numită *alfabet*. Elementele acestei mulţimi pot fi literele şi cifrele caracteristice unei anumite limbi (engleză, japoneză, chineză etc.), secvenţe formate din anumite litere şi/sau cifre (gândite ca entităţi indivizibile), logograme sau diverse simboluri speciale reprezentând cuvinte sau fraze. Mesajele formate pe baza unui astfel de alfabet vor fi numite *texte iniţiale* sau *texte sursă* sau *texte clare* sau *plaintexte* [16]. Ele sunt secvenţe (înşiruiri) de elemente ale alfabetului. Similar, putem presupune că textele produse prin criptarea textelor sursă sunt construite pe baza unui alt alfabet (nu neapărat identic cu cel peste care se construiesc textele sursă). Textele rezultate prin criptarea textelor sursă vor fi numite *texte criptate* sau *texte cifrate* sau *criptotexte* [17].

Definiţia 4.11.1. Un *sistem de criptare* sau *criptosistem* este un 5-uplu

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}),$$

unde:

(i) $\mathcal{P}$ este o mulţime finită şi nevidă numită *alfabetul plaintextelor*. Elementele ei sunt numite *simboluri/caractere plaintext*;

(ii) $\mathcal{C}$ este o mulţime finită şi nevidă numită *alfabetul criptotextelor*. Elementele ei sunt numite *simboluri/caractere criptotext*;

(iii) $\mathcal{K}$ este o mulţime finită şi nevidă numită *spaţiul cheilor de criptare*. Elementele ei sunt numite *chei* (de criptare);

(iv) $\mathcal{E}$ şi $\mathcal{D}$ sunt două mulţimi de funcţii,

$$\mathcal{E} = \{e_K : \mathcal{P} \to \mathcal{C} | K \in \mathcal{K}\}$$

şi

$$\mathcal{D} = \{d_K : \mathcal{C} \to \mathcal{P} | K \in \mathcal{K}\},$$

astfel încât pentru orice $K \in \mathcal{K}$ şi $x \in \mathcal{P}$ are loc $d_K(e_K(x)) = x$.

Plaintextele (textele ce urmează a fi criptate) se construiesc pe baza alfabetului $\mathcal{P}$, iar *criptotextele* (textele ce rezultă prin criptare) sunt elemente ale mulţimii tuturor textelor construite pe baza alfabetului $\mathcal{C}$. Pentru fiecare cheie $K \in \mathcal{K}$, funcţia e_K (d_K) se numeşte *funcţia/regula de criptare* (*decriptare*). Observăm că funcţia d_K este invers la stânga al funcţiei e_K, iar e_K este invers la dreapta al funcţiei d_K. Ca urmare, e_K este funcţie injectivă, iar d_K, surjectivă. Injectivitatea regulilor de criptare conduce imediat la $|\mathcal{P}| \leq |\mathcal{C}|$.

[16] Terminologia de "plaintext" este de provenienţă englezească şi preferăm să o utilizăm datorită uşurinţei în exprimare.

[17] Terminologia de "criptotext" provine de la cuvântul englezesc "cryptotext".

Moduri de operare. Criptarea unui plaintext $x_1 \cdots x_n$, unde $x_i \in \mathcal{P}$ pentru orice $1 \leq i \leq n$, se poate realiza în una din următoarele două variante numite *moduri de operare*:

- *Modul de operare cu cheie fixă.* Se alege o cheie K şi se criptează fiecare caracter x_i prin e_K. Adică, criptotextul va fi

$$e_K(x_1) \cdots e_K(x_n);$$

- *Modul de operare cu cheie variabilă.* Pentru fiecare i se determină o cheie K_i şi se criptează x_i prin e_{K_i}. Adică, criptotextul va fi

$$e_{K_1}(x_1) \cdots e_{K_n}(x_n).$$

  Determinarea cheilor K_i se face de obicei printr-un aşa numit *generator de chei*, pornindu-se de la o cheie iniţială, un anumit şir de iniţializare şi, eventual, plaintextele x_j şi criptotextele asociate lor, pentru orice $j < i$.

Criptosistemele cărora li se asociază un generator de chei şi, deci, care se utilizează în modul de operare cu cheie variabilă, se mai numesc şi *criptosisteme cu chei şir*.

Criptosistemele dezvoltate până în prezent pot fi împărţite în două clase fundamentale:

- *criptosisteme simetrice,* caracterizate prin aceea că regula de decriptare poate fi determinată "uşor" pornind de la regula de criptare, şi invers;

- *criptosisteme asimetrice* sau *cu chei publice*, opuse celor simetrice. Intr-un astfel de criptosistem, cunoaşterea unei reguli e_K nu permite, prin metodele cunoscute la momentul actual, determinarea în mod eficient a regulii d_K.

Adesea, în cadrul criptosistemelor simetrice, alfabetul plaintextelor este o mulţime de secvenţe de lungime $m \geq 1$ formate din elemente ale unei alte mulţimi P. Adică, $\mathcal{P} \subseteq P^m$. Dacă utilizatorul îşi formează plaintextele din elemente ale mulţimii P, atunci criptarea cu un altfel de criptosistem se realizează prin împărţirea plaintextului sursă în *blocuri* de lungime m (eventual ultimul bloc este completat astfel încât să aibă lungimea m), criptându-se apoi fiecare bloc în parte. Din acest motiv, astfel de criptosisteme mai sunt numite şi *criptosisteme bloc*. Argumentul fundamental care stă la baza considerării acestora rezidă din faptul că, pentru ele, regulile de criptare fac apel intensiv la elementele constitutive ale blocului. Marea majoritate a criptosistemelor simetrice ce au utilitate practică relevantă sunt criptosisteme bloc.

Interacţiunea dintre utilizatorii legali şi cei ilegali ai unui criptosistem poate fi reprezentată schematic ca în Figura 4.2, unde:

– A şi B sunt *utilizatori legali* ce au convenit a comunica utilizând un criptosistem a priori stabilit între ei, şi cu o anumită cheie de criptare K;

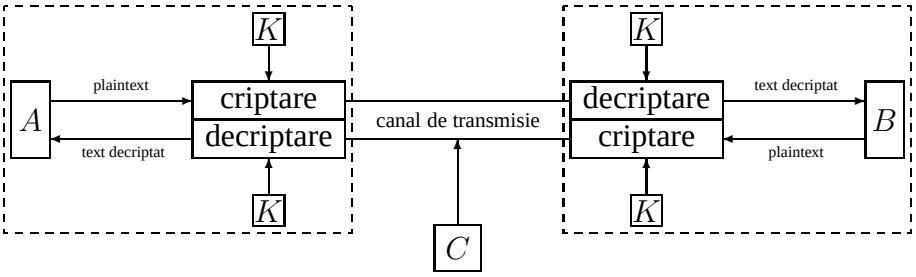

Figura 4.2: Interacţiunea dintre utilizatorii legali şi ilegali ai unui criptosistem

– C este *utilizator ilegal* ce încearcă a captura criptotexte ce se transmit între A şi B în ideea determinării plaintextelor corespunzătoare (sau, în cel mai bun caz, a cheii de criptare), sau ce încearcă a altera mesajele ce se transmit între A şi B.

Construcţia criptosistemelor trebuie realizată având în vedere următoarele două deziderate fundamentale:

(1) pentru orice cheie K, funcţiile e_K şi d_K să poată fi calculate "eficient" [18] (cunoscând cheia K);

(2) determinarea cheii de criptare pe baza unor criptotexte cunoscute (eventual şi prin cunoaşterea plaintextelor corespunzătoare) să fie "imposibilă" (în sens opus sensului de la (1), de calcul eficient).

Criptanaliza este ştiinţa ce se ocupă cu studiul tehnicilor (metodelor) prin care se poate reface un plaintext pornind de la unul sau mai multe criptotexte, fără a cunoaşte, a priori, cheia de criptare. Fiecare tehnică în parte este numită *tehnică de criptanaliză* sau *tehnică de atac* sau *atac*. Dacă un criptosistem poate fi atacat cu succes printr-o anumită tehnică, atunci vom spune că el poate fi *spart* prin respectiva tehnică. Persoanele ce se ocupă cu studiul şi practicarea tehnicilor de atac se numesc *criptanalişti*. Cuplul celor două ştiinţe, criptografie şi criptanaliză, este întâlnit adesea sub terminologia de *criptologie*.

O presupunere fundamentală în criptanaliză, enunţată pentru prima dată de D.A. Kerckhoffs în secolul 19 (a se vedea [90]), constă în aceea că securitatea oferită de un criptosistem trebuie să se bazeze numai şi numai pe cheia de criptare (nu şi pe anumite detalii constructive ale acestora deoarece, mai devreme sau mai târziu, acestea pot deveni cunoscute criptanaliştilor). Altfel spus, Kerckhoffs porneşte de la ideea că, în procesul de criptanaliză, criptanaliştii au toate detaliile constructive ale criptosistemului, exceptând cheia utilizată la momentul respectiv. Chiar dacă în situaţii reale lucrurile nu stau în acest fel, presupunerea lui Kerckhoffs este importantă prin aceea că un criptosistem rezistent la atacuri ce iau în considerare această presupunere va fi rezistent şi la atacuri pentru care această presupunere nu este satisfăcută.

Există, în general, 4 tipuri de atacuri:

[18] Prin *calculul eficient* (*uşor*) al unei funcţii vom înţelege existenţa unui algoritm determinist de complexitate timp polinomială de calcul al respectivei funcţii.

1. *Atac de criptotext*. Criptanalistul are la dispoziţie un număr de criptotexte (obţinute cu aceeaşi cheie) şi, scopul lui este de a determina plaintextele corespunzătoare, sau cheia de criptare (în cel mai bun caz) sau un algoritm de determinare a unui (nou) plaintext pornind de la criptotextul asociat (fără a cunoaşte cheia de criptare). Schematic, aceasta poate fi exprimată ca mai jos:

   Date iniţiale: $y_1 = e_K(x_1), \ldots, y_i = e_K(x_i)$;

   Cerinţe: $x_1, \ldots, x_i$ sau K sau un algoritm de determinare a lui x_{i+1} pornind de la $y_{i+1} = e_K(x_{i+1})$.

2. *Atac de plaintext cunoscut*. Criptanalistul are la dispoziţie un număr de criptotexte (obţinute cu aceeaşi cheie) dar şi plaintextele corespunzătoare. Scopul lui este de a determina cheia de criptare sau un algoritm de determinare a unui (nou) plaintext pornind de la criptotextul asociat (fără a cunoaşte cheia de criptare). Schematic, aceasta poate fi exprimată ca mai jos:

   Date iniţiale: $(x_1, y_1 = e_K(x_1)), \ldots, (x_i, y_i = e_K(x_i))$;

   Cerinţe: K sau un algoritm de determinare a lui x_{i+1} pornind de la $y_{i+1} = e_K(x_{i+1})$.

3. *Atac de plaintext ales*. Criptanalistul are posibilitatea de a determina criptotextele unor plaintexte alese de el. Scopul lui este de a determina cheia de criptare sau un algoritm de determinare a unui (nou) plaintext pornind de la criptotextul asociat (fără a cunoaşte cheia de criptare). Schematic, aceasta poate fi exprimată ca mai jos:

   Date iniţiale: $(x_1, y_1 = e_K(x_1)), \ldots, (x_i, y_i = e_K(x_i))$, unde $x_1, \ldots, x_i$ sunt alese de criptanalist;

   Cerinţe: K sau un algoritm de determinare a lui x_{i+1} pornind de la $y_{i+1} = e_K(x_{i+1})$.

4. *Atac adaptiv de plaintext ales*. Acesta este un caz particular al metodei de atac de plaintext ales. Criptanalistul are posibilitatea de a cunoaşte criptotextele asociate plaintextelor alese de el şi, mai mult, are posibilitatea de a determina noi perechi (plaintext,criptotext) cu plaintextul ales în funcţie de concluziile obţinute prin analiza perechilor (plaintext,criptotext) anterioare.

Există şi alte metode de atac, mai puţin frecvente sau eficiente. Unul dintre acestea este *atacul de criptotext ales* în care se presupune că criptanalistul are posibilitatea de a determina plaintextele corespunzătoare unor criptotexte alese de el, iar un altul constă în enumerarea şi verificarea tuturor cheilor posibile. Acest ultim atac, întâlnit sub denumirea de *atac brute-force* sau *atac prin căutare exhaustivă a cheilor* va fi abreviat de noi prin EKS [19].

[19] De la Exhaustive Key Search.

Corespondenţa literă–cifră. Marea majoritate a criptosistemelor ce sunt utilizate astăzi în practică sunt construite utilizând aparatul algebric, uzând de o anumită corespondenţă bijectivă între setul de caractere necesar construirii plaintextelor şi criptotextelor şi o anumită submulţime de numere naturale (o astfel de corespondenţă ar putea fi, de exemplu, codul EBCDIC sau ASCII cu numerele privite în sistemul zecimal). O astfel de "codificare" a setului de caractere utilizat nu trebuie gândită ca o metodă de creştere a securităţii criptosistemelor; ea nu este altceva decât un pas intermediar în aplicarea unor metode criptografice bazate, în mod esenţial, pe teoria numerelor.

Fără a aduce nici un prejudiciu teoriei generale, se utilizează un set redus de caractere ca cel din din Figura 4.3, împreună cu *corespondenţa literă–cifră* asociată.

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

Figura 4.3: Corespondenţa literă-cifră

Astfel, se înlocuiesc 'ă' prin 'a', 'â' prin 'a', 'î' prin 'i', 'ş' prin 's' şi 'ţ' prin 't'. De asemenea, nu se face distincţie între literele mari şi mici, spaţiile sunt inserate între caractere doar pentru claritate, nu se folosesc semne de punctuaţie, caractere matematice sau de orice alt tip. Aceasta se bazează pe faptul că orice limbaj este dependent de context în mare măsură, sensul unei propoziţii fiind clar chiar prin convenţiile făcute mai sus.

Dacă $x = x_1 \cdots x_n$ este un text de lungime $n \geq 1$ peste alfabetul considerat mai sus, atunci prin *deplasarea/shiftarea* lui x cu k poziţii, unde $0 \leq k \leq 25$, înţelegem textul obţinut din x prin înlocuirea fiecărei litere x_i cu litera de pe poziţia $(k_i + k)\ mod\ 26$, unde k_i este poziţia literei x_i, pentru orice i.

Criptosisteme cu chei publice. O caracteristică esenţială a criptosistemelor cu chei simetrice este aceea că, cheia de criptare trebuie să fie secretă (cunoscută numai de utilizatorii legali ai criptosistemului). Această cheie furnizează atât regula de criptare cât şi cea de decriptare, fiecare din aceste reguli fiind algoritmic calculabile în timp "eficient" (un astfel de criptosistem va fi prezentat în Secţiunea 4.11.2). Transmiterea cheii între utilizatorii legali trebuie realizată folosind un canal de transmisie sigur ceea ce, în practică, poate constitui o problemă majoră (interceptarea cheii, de către un utilizator ilegal, compromite criptosistemul în totalitate).

Opus ideii de cheie secretă se află conceptul de *cheie publică*. Astfel de chei, aşa cum ne spune şi numele lor, sunt publice (pot fi cunoscute de oricine); cunoaşterea lor asigură determinarea eficientă a regulilor de criptare dar, fără informaţii suplimentare (uzual numite *trape secrete*), regulile de decriptare sunt "foarte dificil" de determinat. Criptosistemele bazate pe astfel de chei se numesc *criptosisteme cu chei publice*. Ideea realizării unor astfel de criptosisteme a fost prezentată în 1976, de

către W. Diffie şi M.E. Hellman [42], la o Conferinţă Naţională de Informatică în SUA (a se vedea şi [43]) [20]. Primul criptosistem de acest tip a fost propus de Merkle şi Hellman [117, 118, 76] [21]. Multe alte criptosisteme au fost apoi propuse şi, multe dintre ele, s-au dovedit a fi nesigure. Altele asigură securitate dar sunt inutilizabile din punct de vedere practic, iar altele ori se bazează pe chei extrem de lungi ori criptotextul produs este foarte mare în raport cu plaintextul. Puţine dintre ele s-au dovedit a fi atât sigure cât şi practic utilizabile. Dintre acestea menţionăm criptosistemul RSA, criptosistemul Rabin, criptosistemul ElGamal, criptosistemul Chor-Rivest (din clasa criptosistemelor bazate pe problema rucsacului), criptosistemul McEliece şi criptosisteme bazate pe curbe eliptice. Primele trei pot fi utilizate şi în construcţia de semnături digitale (a se vedea Secţiunea 4.11.3).

Criptosistemele cu chei publice sunt predispuse atacului de plaintext cunoscut deoarece cheia este publică. Astfel, oricine interceptează un criptotext y poate încerca diverse plaintexte x pentru a obţine $e_K(x) = y$. Perechile (x, y) astfel obţinute pot atunci constitui baza unui atac de plaintext cunoscut. Ca urmare, cu cât spaţiul plaintextelor este mai mic, cu atât mai mult criptosistemul este nesigur. De fapt, discuţia de mai sus ne spune că criptosistemele cu chei publice nu pot asigura securitate necondiţionată. Deci, ceea ce putem cere de la astfel de criptosisteme este doar securitatea algoritmică (ceea ce nu este puţin dacă această securitate algoritmică este "bine aleasă" adică, bazată pe probleme a căror rezolvare este algoritmic "foarte dificilă" la momentul actual).

4.11.2 Criptosistemul RSA

Criptosistemul RSA a fost propus de Ronald Rivest, Adi Shamir şi Leonard Adleman în 1977 [151], luându- si denumirea de la numele celor care l-au propus. Dintre toate criptosistemele cu chei publice propuse până în prezent, RSA s-a dovedit a fi cel mai uşor de implementat şi utilizat. El se bazează pe "dificultatea" factorizării numerelor foarte mari (a se vedea discuţia de la sfârşitul Secţiunii 4.9).

4.11.2.1 Descrierea criptosistemului

Criptosistemul RSA foloseşte aritmetica modulară în $\mathbf{Z}_n$, unde n este produsul a două numere prime distincte p şi q.

Descrierea criptosistemului

- fie p şi q două numere prime distincte şi fie $n = pq$;
- $\mathcal{P} = \mathcal{C} = \mathbf{Z}_n$;
- $\mathcal{K} = \{(n, p, q, e, d) | e \in \mathbf{Z}^*_{\phi(n)} \ \wedge \ ed \equiv 1 \ mod \ \phi(n)\}$;

[20] Independent, această idee apare şi la R. Merkle dar, datorită unor probleme editoriale, prima lucrare a acestuia apare abia în 1978 [116].

[21] Şi în acest caz, lucrarea lui Merkle şi Hellman apare în 1978, ceea ce face ca criptosistemul RSA, care a fost publicat în 1977, să fie considerat de unii autori ca fiind primul astfel de criptosistem.

- pentru orice cheie $K = (n, p, q, e, d) \in \mathcal{K}$ şi $x, y \in \mathbf{Z}_n$,

$$e_K(x) = x^e \ mod \ n \ \text{ şi } \ d_K(y) = y^d \ mod \ n;$$

- pentru orice cheie $K = (n, p, q, e, d) \in \mathcal{K}$, (n, e) este cheia publică, iar (p, q, d), cea secretă.

Arătăm că pentru orice cheie $K = (n, p, q, e, d)$, d_K este invers la stânga a funcţie e_K, adică $x^{ed} \equiv x \ mod \ n$, pentru orice $x \in \mathbf{Z}_n$. Avem de analizat două cazuri:

(1) $x \in \mathbf{Z}_n^*$. Atunci, $x^{ed} \equiv x \ mod \ n$ deoarece $ed \equiv 1 \ mod \ \phi(n)$;

(2) $x \in \mathbf{Z}_n - \mathbf{Z}_n^*$. Fie $t \in \mathbf{Z}$ astfel încât $ed - 1 = t\phi(n)$. Dacă $x = 0$, atunci are loc $x^{ed} \equiv x \ mod \ n$. Presupunem deci că $x \neq 0$. Atunci, deoarece x şi n nu sunt prime între ele, $(x, n) \geq 2$. Cum $n = pq$ şi $x < n$, deducem că $(x, n) = p$ sau $(x, n) = q$. Să presupunem că $(x, n) = p$ (celălalt caz este complet similar acestuia). Atunci, q nu va divide x şi, în baza teoremei lui Fermat, deducem că are loc

$$x^{q-1} \equiv 1 \ mod \ q$$

de la care urmează $x^{t\phi(n)} \equiv 1 \ mod \ q$. Combinând aceasta cu $x \equiv x \ mod \ p$ şi cu faptul că p şi q sunt prime între ele, deducem

$$x^{t\phi(n)+1} \equiv x \ mod \ pq.$$

Adică, $x^{ed} \equiv x \ mod \ n$.

Ca urmare, structura definită mai sus este un criptosistem.

Presupunem acum că două persoane A şi B doresc să schimbe mesaje criptându-le cu RSA cu parametrii p, q şi n (p şi q cunoscuţi doar de A şi B, iar n public). Pentru aceasta, A alege o pereche de numere (e_A, d_A) ca în algoritm, face public e_A (cheia lui publică) şi reţine secret d_A (cheia lui secretă). Similar, B alege (e_B, d_B), face public e_B şi reţine secret d_B. Presupunem acum că A doreşte a cripta şi transmite un mesaj α către B. Acest mesaj se codifică numeric (folosind, de exemplu, corespondenţa deja adoptată) după care, şirul numeric obţinut se împarte în blocuri de cifre, de dimensiune egală, astfel încât fiecare bloc să poată fi considerat ca un număr din $\mathbf{Z}_n$ (eventual, ultimul bloc se completează cu zerouri la dreapta pentru a avea aceeaşi lungime ca şi celelate blocuri). Fie $x = x_1 \cdots x_m$ această descompunere. Se calculează apoi

$$y_i = x_i^e \ mod \ n,$$

pentru orice $1 \leq i \leq m$. Secvenţa $y = y_1 \cdots y_m$ reprezintă codul numeric al criptotextului.

Decriptarea secvenţei y cere determinarea subsecvenţelor $y_1, \ldots, y_m$ (ca mai sus). Aceasta poate constitui o problemă serioasă pentru B ca urmare a faptului că aceste

subsecvenţe pot avea lungimi diferite iar B nu va şti cum să le separe. Această problemă poate fi depăşită dacă, înainte de a fi trimise, secvenţele $y_1, \ldots, y_m$ sunt completate la stânga cu zero-uri astfel încât toate să aibă aceeaşi lungime ca şi blocurile x_i. Fie $y'_1, \ldots, y'_m$ aceste noi blocuri. Atunci, decriptarea decurge calculând

$$x_i = (y'_i)^d \; mod \; n,$$

pentru orice $1 \leq i \leq m$.

Exemplul 4.11.2.1. Fie $p = 101$ şi $q = 113$ două numere prime. Atunci,

$$n = 11413 \;\text{ şi }\; \phi(n) = 11200.$$

Deoarece $\phi(n) = 2^6 5^2 7$, urmează că e poate fi ales ca fiind orice număr mai mic strict decât n ce nu este divizibil prin 2, 5 sau 7. Fie $e = 3533$. Atunci, cu ajutorul algoritmului lui Euclid determinăm $d = e^{-1} \; mod \; 11200 = 6597$.

Numerele $n = 11413$ şi $e = 3533$ sunt publice; p, q şi d sunt secrete. Criptarea textului 9726 produce criptotextul 5761 deoarece

$$9726^{3533} \; mod \; 11413 = 5761.$$

Decriptarea lui 5761 conduce, evident, la 9726.

Vom discuta în cele ce urmează câteva aspecte legate de implementarea criptosistemului RSA. Utilizarea acestui criptosistem în practică necesită parcurgerea următorilor paşi:

- alegerea a două numere prime, p şi q (este necesar ca aceste numere să fie mari, de cel puţin 100 de cifre fiecare, pentru a asigura securitate criptosistemului);
- alegerea unui număr $e \in \mathbf{Z}^*_{\phi(n)}$ şi determinarea inversului modulo $\phi(n)$ al acestuia, fie el d.

Este recomandabil ca numerele p şi q să aibă lungimi apropiate (dacă nu chiar egale). Alegerea lui e constituie un alt punct cheie atât pentru asigurarea securităţii criptosistemului dar cât şi pentru eficienţa implementării. Se constată că valori mici ale lui e asigură criptare rapidă şi, totodată, dă posibilitatea unui factor mare de decriptare d, ceea ce măreşte securitatea criptosistemului (asupra acestui aspect vom reveni). Cele mai utilizate valori pentru e sunt 3, 17 şi 65537 ($2^{16} + 1$) (reprezentarea binară a lui 65537 are doar doi biţi 1, ceea ce ne spune că se vor face doar 16 ridicări la puterea a 2-a şi o înmulţire).

Algoritmii ce implementează regulile e_K şi d_K sunt eficienţi, de complexitate $\mathcal{O}((\log n)^3)$.

Generarea numerelor prime p şi q, de 100 de cifre să spunem, ar ridica următoarea problemă care, aparent, ar fi cât se poate de serioasă: nu epuizăm, într-o perioadă de utilizare a unui astfel de criptosistem, toate numerele prime de 100 de cifre? Răspunsul este cât se poate de simplu dacă facem apel la Teorema numerelor prime. In baza ei putem spune că există proximativ $4 \cdot 10^{97}$ numere prime cu 100 de cifre,

număr ce depăşeşte numărul de atomi din universul vizibil (a se vedea Secţiunea 4.1). Ca urmare, nu trebuie să ne fie teamă că am "epuiza" toate numerele prime într-o perioadă, fie ea oricât de lungă, de utilizare a criptosistemului RSA.

Testarea primalităţii unui număr p se poate face prin algorithmi probabilişti de complexitate $\mathcal{O}((\log p)^3)$ (de exemplu, algorithmul Miller-Rabin). Dejavantajul unui algoritm probabilist constă în aceea că el furnizează răspuns corect dacă numărul este prim, dar poate furniza răspuns eronat dacă numărul nu este prim (adică, el ne poate spune că numărul este prim, chiar dacă acesta nu este). Iterarea unui algoritm probabilist peste o aceeaşi intrare (acelaşi număr pentru care se testează primalitatea) de un număr de ori şi obţinerea de fiecare dată a unui răspuns de genul "numărul este prim" face ca probabilitatea de răspuns eronat să scadă dramatic. De exemplu, în cazul algoritmului Miller-Rabin, iterarea de 100 de ori a acestuia poate conduce la o probabilitate de răspuns eronat de cel mult $1/4^{100}$. In 2002, Agrawal, Kayal şi Saxena au arătat că problema primalităţii poate fi rezolvată prin algoritmi determinişti de complexitate timp polinomială (lucrarea acestora a fost publicată în 2004 [1]). Algoritmii determinişti de complexitate timp polinomială cunoscuţi la momentul actual au complexitate destul de ridicată, $\mathcal{O}((\log p)^{10.5})$, ceea ce face ca tot algoritmii probabilişti să fie preferaţi deocamdată.

Generarea unui număr prim se face în două etape: se generează aleator un număr, după care se verifică primalitatea acestuia. O problemă ce se ridică relativ la generarea aleatoare a numerelor este următoarea: câte numere mari trebuiesc generate aleator pentru a depista cel puţin un număr prim? Pentru a răspunde acestei întrebări reamintim că Teorema numerelor prime afirmă că pentru orice număr natural m există aproximativ $m/ln(m)$ numere prime mai mici decât m. Aceasta înseamnă că, dacă p a fost generat aleator, probabilitatea ca el să fie prim este $1/ln(p)$. In cazul în care p este un număr de 512 biţi, atunci probabilitatea de mai sus este aproximativ $1/177$. Adică, în medie, generarea a 177 de numere mari va include şi un număr prim. Dacă însă cerem încă de la început ca numerele generate să fie impare, atunci probabilitatea de mai sus se dublează.

O implementare a acestei metode pe o maşină SPARC II a condus la depistarea unor numere prime de 256 de biţi în 2.8 secunde, a unor numere prime de 512 de biţi în 24 secunde, a unor numere prime de 768 de biţi în 2 minute, şi a unor numere prime de 1024 de biţi în 5.1 minute [162].

4.11.2.2 Criptanaliză RSA

Trebuie să remarcăm că dacă se cunoaşte descompunerea lui n în factori primi, $n = pq$, criptosistemul RSA este compromis (în ipoteza că această descompunere este cunoscută de o persoană neautorizată). In adevăr, cunoscând p şi q se poate determina uşor $\phi(n)$ şi, atunci, cunoaşterea cheii publice e conduce la determinarea imediată a cheii secrete d (în timp $\mathcal{O}((\log n)^3)$). Calcularea lui $\phi(n)$ în timp polinomial determinist cunoscând doar n, are aceleaşi consecinţe. Calculul lui $\phi(n)$ în timp eficient face apel tot la factorizarea lui n. Ca urmare, putem spune că criptosistemul RSA îşi bazează securitatea pe problema factorizării pentru care, la momentul actual, nu se cunosc metode eficiente (dacă numerele prime p şi q sunt alese aleator

şi suficient de mari).

Vom prezenta în continuare 3 modalităţi de a ataca criptosistemul RSA, fără a cunoaşte factorizarea lui n. Fiecare din aceste atacuri speculează utilizare "greşită" a criptosistemului sau alegere nepotrivită a parametrilor.

Atacul lui Davida a fost propus în [36], el exploatând utilizare cu greşeli a criptosistemului.

Presupunem că A trimite un mesaj x lui B folosind cheia publică e_B. Mai presupunem că C intercepteză mesajul transmis, $y = x^{e_B} \ mod \ n$, şi multiplică y cu z, unde z este ales astfel încât există $z^{-1} \ mod \ n$, şi trimite $y' = yz^{e_B} \ mod \ n$ lui B. Decriptând acest mesaj, B va găsi $x' = y'^{d_B} \ mod \ n$ care, foarte probabil, este un plaintext fără înţeles. Ca urmare, el va renunţa la acest mesaj. Dacă C va putea obţine x', atunci el va putea determina x prin

$$x = x'z^{-1} \ mod \ n.$$

Atacul lui Lenstra a fost propus în [102], el exploatând, ca şi atacul lui Davida, utilizare cu greşeli a criptosistemului.

Decriptarea unui mesaj criptat cu RSA poate fi realizată eficient dacă, dat $y = x^e \ mod \ n$, se calculează

$$x_p = y^{d \ mod \ p-1} \ mod \ p$$

şi

$$x_q = y^{d \ mod \ q-1} \ mod \ q.$$

De la acestea, x se obţine cu ajutorul Teoremei chineze a resturilor ca fiind unica soluţie modulo $n = pq$ a sistemului

$$\begin{cases} x & \equiv & x_p \ mod \ p \\ x & \equiv & x_q \ mod \ q \end{cases}$$

Să presupunem că x_p a fost calculat corect dar x_q a fost calculat greşit. Ca urmare, rezolvarea sistemului de mai sus va conduce la un x' diferit de x şi, privit ca plaintext, fără înţeles. Dacă acest x' este obţinut de C, atunci se poate determina în timp eficient p prin

$$p = ((x'^{e_B} - y) \ mod \ n, n).$$

Ca urmare, criptosistemul este complet neutralizat de C.

Atac de exponent secret mic. Atacul pe care îl prezentăm aici este datorat lui M. Wiener [196]. El exploatează alegerea unui parametru e care produce o cheie secretă d "prea mică".

Teorema 4.11.2.1. Fie p şi q numere prime astfel încât $q < p < 2q$, $n = pq$, şi $e, d \in \mathbf{Z}^*_{\phi(n)}$ astfel încât $ed \equiv 1 \ mod \ \phi(n)$. Dacă $d < 1/3\sqrt[4]{n}$, atunci d se poate determina în timp polinomial determinist în raport cu $log \, n$ cunoscând doar n şi e.

Demonstraţie Relaţia $e, d \in \mathbf{Z}^*_{\phi(n)}$ conduce la existenţa unui număr natural k astfel încât $ed - 1 = k\phi(n)$ sau, echivalent,

$$\left|\frac{e}{\phi(n)} - \frac{k}{d}\right| = \frac{1}{d\phi(n)} \qquad (1)$$

Mai mult, $ed - k\phi(n) = 1$ ne spune că $(k, d) = 1$, deci fracţia k/d este ireductibilă.

Ca urmare, determinarea lui d poate fi redusă la determinarea unor fracţii ireductibile k/d care să verifice (1). Vom arăta că orice fracţie ireductibilă k/d ce verifică (1) va veriica şi

$$\left|\frac{e}{n} - \frac{k}{d}\right| < \frac{1}{2d^2} \qquad (2)$$

In plus, determinarea fracţiilor k/d ce verifică (2) se poate face în timp polinomial în raport cu $log\, n$.

Să presupunem deci că k/d verifică (1). Stabilim întâi următoarele relaţii:

$$\begin{aligned} n - \phi(n) &= p + q - 1 \\ &< 3q - 1 && \text{(de la } p < 2q\text{)} \\ &< 3q \\ &< 3\sqrt{n} && \text{(de la } q < p \text{ şi } n = pq\text{)} \end{aligned}$$

şi

$$\begin{aligned} k\phi(n) &= ed - 1 \\ &< ed \\ &< d\phi(n) && \text{(de la } e < \phi(n)\text{)} \\ &< 1/3\sqrt[4]{n}\,\phi(n) && \text{(conform ipotezei)} \end{aligned}$$

care conduce la $k < 1/3\sqrt[4]{n}$.

Acum, obţinem:

$$\begin{aligned} |e/n - k/d| &= |ed - kn|/(nd) \\ &= |1 + k\phi(n) - kn|/(nd) \\ &= |k(n - \phi(n)) - 1|/(nd) \\ &< (k(n - \phi(n)))/(nd) \\ &< (3k\sqrt{n})(nd) \\ &< (3k)(d\sqrt{n}) \\ &< (\sqrt[4]{n})(d\sqrt{n}) \\ &= 1/(d\sqrt[4]{n}) \\ &< 1/(3d^2) && \text{(de la } \sqrt[4]{n} > 3d\text{)} \\ &< 1/(2d^2) \end{aligned}$$

Deci, k/d verifică (2).

Fracţiile k/d ce verifică (2) sunt exact convergentele fracţiei continue asociate lui e/n [73]. Acestea se obţin pe baza câturilor succesive ale împărţirii lui e la n prin algoritmul lui Euclid (a se vedea Secţiunea 4.2). Există $log\, n$ astfel de convergente ce pot fi determinate în timp polinomial în raport cu $log\, n$. Verificarea faptului că una din aceste convergente este exact fracţia k/d ce verifică (1) se face tot în timp

polinomial în raport cu $log\, n$. Deci, d se poate determina în timp polinomial în raport cu $log\, n$. □

Cu valori de 512 biţi pentru p şi q, va rezulta n de 1024 biţi. Ca urmare, pentru a contracara atacul lui Wiener, d trebuie să fie de cel puţin 256 biţi.

Cele prezentate de noi în această secţiune nu au dorit altceva decât să scoată în evidenţă subtilitatea unor astfel de atacuri şi, de ce nu, să atragă interesul cititorului spre acest domeniu extrem de util şi interesant.

La momentul actual nu se cunoaşte nici o metodă viabilă (practică) de atac împotriva criptosistemului RSA. Utilizarea unor algoritmi cuatici sau moleculari pentru rezolvarea problemei factorizării ar putea constitui o adevărată ameninţare asupra acestui criptosistem. Din nefericire pentru cercetători şi din fericire pentru susţinătorii criptosistemului RSA, algoritmii cuatici şi moleculari sunt, la momentul actual, doar pe hârtie. Punerea lor în practică se loveşte de probleme dificile de inginerie genetică sau cuantică.

4.11.3 Semnături digitale

O altă aplicaţie importantă a problemelor algoritmice ce apar în cadrul grupului $\mathbf{Z}_m^*$ o constituie semnătura digitală.

4.11.3.1 Introducere

Semnătura are scopul de a certifica originalitatea datelor care se transmit între diverse părţi. Frecvent, semnăm cecuri, scrisori, contracte; originalitatea semnăturii noastre conferă textului în cauză caracterul de original. In general, o semnătură trebuie să satisfacă următoarele cerinţe:

– să fie autentică (produsul original al semnatarului);

– să fie nefalsificabilă;

– să nu poate fi reutilizată (odată folosită pentru un document, să nu poată fi transferată pe alt document prin diverse tehnici);

– să nu poată fi repudiată (renegată) de semnatarul ei.

In realitate, nici una din cerinţele de mai sus nu este complet satisfăcută. Există diverse metode de verificare a originalităţii semnăturii (analize grafologice etc.) dar care nu conferă garanţii complete.

Necesitatea utilizării semnăturilor este cât se poate de clară pentru oricine dintre noi. Să presupunem că A şi B sunt două părţi ce doresc să comunice, schimbând între ele diverse mesaje (documente, acte, scrisori etc.). Există cel puţin două aspecte fundamentale care trebuiesc luate în consideraţie:

1. dacă B primeşte un mesaj x de la A, atunci B trebuie să aibă garanţia că x este, în adevăr, mesajul original trimis de A (acest mesaj nu a fost schimbat pe parcurs de o terţă persoană C sau, chiar de către A – A comunică lui B că îi va trimite mesajul x dar, în fapt, el trimite un alt mesaj invocând, la nevoie, posibilitatea alterării mesajului pe canalul de transmisie);

2. dacă A trimite mesajul x către B, atunci A trebuie să aibă garanţia că mesajul x este cel recepţionat de B (mesajul nu a fost schimbat pe parcurs de o terţă persoană C sau, chiar de către B).

Intr-un astfel de context, utilizarea semnăturii (personale) constituie soluţia problemei. A va semna suplimentar mesajul x prin determinarea, după o procedură de semnare secretă (personală), a entităţii $sig(x)$ (semnătură grafică, amprentă digitală etc.). Când B recepţionează cuplul $(x, sig(x))$, el va verifica, cu o procedură publică $ver(x, sig(x))$, autenticitatea semnăturii $sig(x)$ asupra mesajului x. Interpunerea lui C între A şi B, fără a cunoaşte procedura de semnare, trebuie să fie ineficientă. Discuţia purtată până acum poate fi bine descrisă prin intermediul Figurii 4.4.

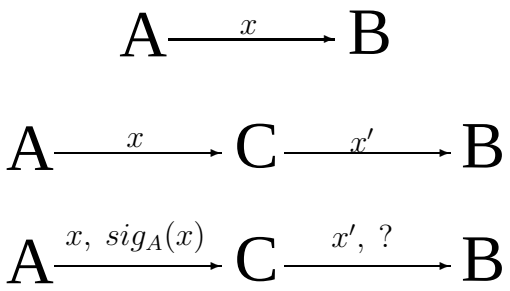

Figura 4.4: Schema de semnare cu interpunerea unei terţe părţi

In 1976, Diffie şi Hellman [43], odată cu lansarea ideii de cheie publică, propune utilizarea semnăturilor şi în cazul transmisiei datelor prin intermediul calculatorului (mesaje, programe etc.). Cerinţele pe care trebuie să le satisfacă o semnătură, cât şi precauţiile (măsurile de siguranţă) pe care trebuie să şi le ia părţile implicate în comunicare, discutate mai sus, trebuie să rămână valabile şi în acest caz. Spre deosebire de semnătura uzuală a fiecăruia, ce este în linii mari aceeaşi pe orice document, semnătura digitală variază de la document la document.

Definiţia 4.11.3.1. O *schemă de semnare*, sau *semnătură digitală*, este un 5-uplu

$$Sig = (\mathcal{P}, \mathcal{S}, \mathcal{K}, \mathcal{A}_s, \mathcal{A}_v),$$

unde:

(1) $\mathcal{P}$ este o mulţime finită şi nevidă ale cărei elemente sunt numite *mesaje*;

(2) $\mathcal{S}$ este o mulţime finită şi nevidă ale cărei elemente sunt numite *semnături*;

(3) $\mathcal{K}$ este o mulţime finită şi nevidă ale cărei elemente sunt numite *chei de semnare*;

(4) $\mathcal{A}_s = \{sig_K : \mathcal{P} \to \mathcal{S} | K \in \mathcal{K}\}$ şi $\mathcal{A}_v = \{ver_K : \mathcal{P} \times \mathcal{S} \to \{0,1\} | K \in \mathcal{K}\}$ sunt mulţimi $\mathcal{K}$-indexate ale căror elemente sunt numite *reguli/algoritmi de semnare* şi, respectiv, *reguli/algoritmi de verificare*, astfel încât

$$ver_K(x,y) = 1 \quad \Leftrightarrow \quad sig_K(x) = y,$$

pentru orice $x \in \mathcal{P}$, $y \in \mathcal{S}$ şi $K \in \mathcal{K}$.

Utilizarea unei scheme de semnare de către două părţi A şi B decurge astfel:

- *semnare.* A decide asupra mesajului pe care îl are de transmis lui B, fie acesta x, alege o cheie K de semnare, semnează obţinând $sig_K(x)$, şi transmite lui B cuplul $(x, sig_K(x))$;
- *verificare.* B primeşte un cuplu $(x, sig_K(x))$ şi acceptă semnătura lui A dacă şi numai dacă $ver_K(x, sig_K(x)) = 1$.

O cerinţă naturală asupra schemelor de semnare digitală constă în aceea că semnarea şi verificarea trebuie să se facă "uşor" (în timp polinomial determinist). Algoritmii de semnare trebuie să fie secreţi, iar cei de verificare, publici. Interceptarea unui mesaj şi a unei semnături asociate nu trebuie să permită determinarea algoritmului de semnare şi nici falsificarea semnăturii prin alte mijloace (semnarea corectă a altor mesaje fără a cunoaşte algoritmul de semnare).

Semnăturile digitale nu pot asigura securitate necondiţionată pentru simplul motiv că, verificarea tuturor posibilităţilor (ce sunt în număr finit) conduce la determinarea algoritmului de semnare utilizat.

Semnăturile digitale pot fi utilizate în conjuncţie cu metodele de criptare cu chei publice astfel. Presupunem că A doreşte să transmită lui B un mesaj x. A îşi alege un algoritm (propriu) de semnare sig_A, semnează mesajul, fie $y = sig_A(x)$, şi criptează cuplul (x, y) folosind un algoritm de criptare cu chei publice e_B (propriu lui B, deci pentru care B cunoaşte regula de decriptare d_B). Rezultatul, $z = e_B(x, y)$, este trimis lui B (Figura 4.5(a)).

Evident, A poate cripta întâi pe x, $e_B(x)$, şi apoi semna rezultatul (Figura 4.5(b)). Dar, în acest caz, un utilizator ilegal C care interceptează cuplul

$$(e_B(x), sig_A(e_B(x)))$$

poate înlocui semnătura lui A, $sig_A(e_B(x))$, fără a cunoaşte mesajul original x, prin semnătura lui proprie $sig_C(e_B(x))$, şi trimite lui B cuplul

$$(e_B(x), sig_C(e_B(x)))$$

(Figura 4.5(c)). In urma recepţionării acestui cuplu, B va considera mesajul ca venind din partea lui C şi nu a lui A (mesajul poartă semnătura lui C). O astfel de situaţie poartă denumirea de *impersonificare*. Pentru a o evita, se recomandă întâi semnarea mesajului şi apoi criptarea cuplului (mesaj, semnătură).

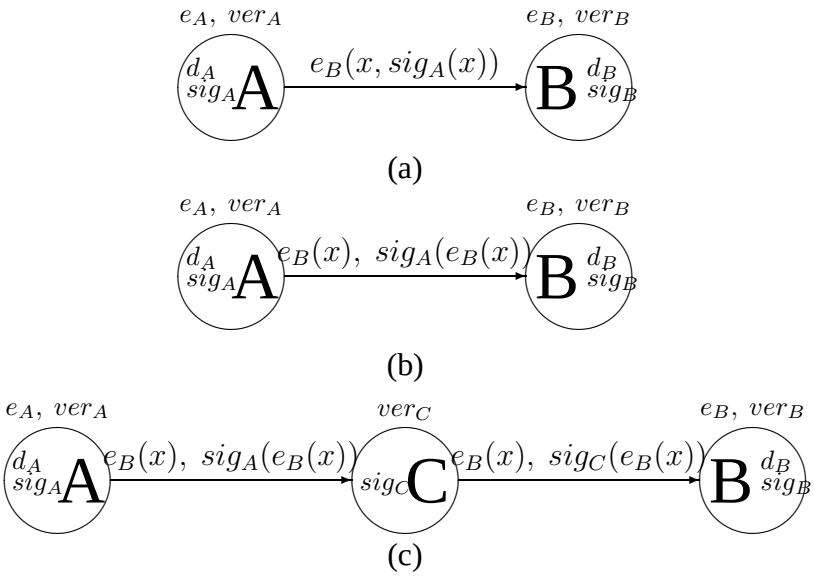

Figura 4.5: Semnăre digitală în conjuncţie cu criptare cu chei publice

Adesea, cheia K utilizată pentru semnare conţine un parametru k, numit *parametru de securitate*, care, la schimbarea cheii, se schimbă doar el. In acest caz este de preferat de păstrat restul cheii drept cheie şi de gândit parametrul de securitate ca fiind un parametru auxiliar. Altfel spus, este de preferat ca procedura de semnare să fie o funcţie de 2 variabile, $sig_K(x, k)$. Evident, aceasta nu modifică cu absolut nimic strategia generală.

4.11.3.2 Semnătura ElGamal

Semnătura ElGamal a fost propusă în 1985 [47]. Această schemă este nedeterministă în sensul că pentru un mesaj x pot exista mai multe semnături valide; algoritmul de verificare trebuie să fie capabil de a accepta oricare din acestea ca semnături autentice.

Ideea de bază în cadrul acestei semnături este de a semna un mesaj $x \in \mathbf{Z}_p^*$, unde p este un număr prim, printr-o pereche $(\gamma, \delta) \in \mathbf{Z}_p^* \times \mathbf{Z}_{p-1}$ astfel încât x este combinaţie liniară de γ şi δ cu doi parametri secreţi a şi k,

$$x = (a\gamma + k\delta)\ mod\ (p-1).$$

Această combinaţie este aleasă modulo $p-1$ deoarece verificarea semnăturii se va face prin intermediul echivalenţei

$$u \equiv v\ mod\ (p-1) \ \Leftrightarrow\ \alpha^u \equiv \alpha^v\ mod\ p,$$

pentru orice rădăcină primitivă α a lui $\mathbf{Z}_p^*$.

Descrierea semnăturii:

- fie p un număr prim şi α o rădăcină primitivă modulo p;

- $\mathcal{P} = \mathbf{Z}_p^*$;
- $\mathcal{S} = \mathbf{Z}_p^* \times \mathbf{Z}_{p-1}$;
- $\mathcal{K} = \{(p, \alpha, a, \beta) | a \in \mathbf{Z}_{p-1},\ \beta = \alpha^a \ mod\ p\}$;
- pentru orice $K = (p, \alpha, a, \beta)$ şi $k \in \mathbf{Z}_{p-1}^*$, mesajul $x \in \mathbf{Z}_p^*$ este semnat prin

$$sig_K(x, k) = (\gamma, \delta),$$

unde

$$\gamma = \alpha^k \ mod\ p \quad \text{şi} \quad \delta = (x - a\gamma)k^{-1} \ mod\ (p-1)$$

(k^{-1} este determinat modulo $p-1$), iar verificarea semnăturii (γ, δ) pentru mesajul x se face prin

$$ver_K(x, (\gamma, \delta)) = 1 \quad \Leftrightarrow \quad \beta^\gamma \gamma^\delta \equiv \alpha^x \ mod\ p.$$

Numerele p, α şi β sunt publice (unui grup de utilizatori), iar a este secret (particular fiecărui utilizator în parte). In plus, semnarea unui mesaj x presupune alegerea unui parametru de securitate k, ceea ce întăreşte securitatea schemei fără a afecta eficienţa verificării (observăm că algoritmul de verificare nu depinde de k).

Schema descrisă mai sus este o schemă de semnare. In adevăr, are loc următorul şir de echivalenţe

$$\begin{aligned} \delta = (x - a\gamma)k^{-1} \ mod\ p - 1 \quad &\Leftrightarrow \quad k\delta \equiv (x - a\gamma) \ mod\ p - 1 \\ &\Leftrightarrow \quad x \equiv (a\gamma + k\delta) \ mod\ p - 1 \\ &\Leftrightarrow \quad \alpha^x \equiv \alpha^{a\gamma + k\delta} \ mod\ p \\ &\Leftrightarrow \quad \alpha^x \equiv \beta^\gamma \gamma^\delta \ mod\ p. \end{aligned}$$

Exemplul 4.11.3.1. Fie $p = 467$, $\alpha = 2$ şi $a = 127$. Atunci,

$$\begin{aligned} \beta &= \alpha^a \ mod\ p \\ &= 2^{127} \ mod\ 467 \\ &= 132. \end{aligned}$$

Presupunem că se doreşte a se semna mesajul $x = 100$ folosind parametrul de securitate $k = 213$ ($k \in \mathbf{Z}_{466}^*$ şi $k^{-1} = 431$). Numerele γ şi δ vor fi date prin:

$$\gamma = 2^{213} \ mod\ 467 = 29,$$

şi

$$\delta = (100 - 127 \cdot 29) \cdot 431 \ mod 466 = 51.$$

Ca urmare, $sig_K(x, k) = (29, 51)$.

Verificarea se face calculând

$$132^{29} \cdot 29^{51} \ mod\ 467$$

şi

$$2^{100} \ mod\ 467$$

şi constatând că ele sunt congruente modulo 467.

Algoritmii de semnare şi verificare pentru semnătura ElGamal au complexitatea $\mathcal{O}((\log p)^3)$.

Semnătura ElGamal îşi bazează securitatea pe intractabilitatea problemei logaritmului discret. Ca urmare, p şi α trebuie aleşi astfel încât să fie îndeplinit acest deziderat (p trebuie să fie număr prim suficient de mare, în general de cel puţin 1024 biţi, iar α să fie rădăcină primitivă modulo p generată aleator).

Chiar dacă p şi α sunt aleşi ca mai sus, există situaţii care compromit semnătura, aaş cum se va vedea mai jos. Menţionăm însă că nici una din situaţiile descrise mai jos nu reprezintă un pericol real asupra semnăturii dacă aceasta este utilizată "cu grijă".

Determinarea semnăturii pentru un mesaj dat. Presupunem că, dat mesajul x, dorim să construim o semnătură validă pentru el, (γ, δ), fără a cunoaşte a. Presupunem că, printr-o anume metodă, am determinat γ. Atunci, urmează să determinăm δ. Aceasta se reduce la rezolvarea ecuaţiei

$$\gamma^\delta \equiv \alpha^x \beta^{-\gamma} \; mod \; p,$$

care constituie subiectul problemei logaritmului discret.

Dacă presupunem că, printr-o anume metodă, am determinat δ, determinarea lui γ se reduce la rezolvarea ecuaţiei

$$\beta^\gamma \gamma^\delta \equiv \alpha^x \; mod \; p.$$

Pentru rezolvarea acetei ecuaţii, care nu "pare" a fi problema logaritmului discret, nu se cunoaşte în prezent nici o metodă polinomială deterministă.

Determinarea mesajului pentru o semnătură dată Dacă presupunem că am ales o pereche (γ, δ) ca semnătură, determinarea unui mesaj x pentru care această pereche să fie semnătură se poate face prin rezolvarea ecuaţiei

$$\beta^\gamma \gamma^\delta \equiv \alpha^x \; mod \; p,$$

care este o instanţă a problemei logaritmului discret.

Determinarea simultană a semnăturii şi a mesajului Există posibilitatea determinării "simultane" a 3 numere γ, δ şi x astfel încât (γ, δ) să fie semnătură pentru x. In adevăr, pentru orice i şi j cu $0 \leq i, j \leq p-2$ şi $(j, p-1) = 1$, numerele

$$\begin{aligned} \gamma &= \alpha^i \beta^j \; mod \; p \\ \delta &= -\gamma j^{-1} \; mod \; (p-1) \\ x &= -\gamma i j^{-1} \; mod \; (p-1), \end{aligned}$$

unde j^{-1} este determinat modulo $p-1$, verifică

$$\begin{aligned} \beta^\gamma \gamma^\delta &\equiv \beta^{\alpha^i \beta^i} (\alpha^i \beta^j)^{-\alpha^i \beta^j j^{-1}} \; mod \; p \\ &\equiv \beta^{\alpha^i \beta^i} \alpha^{-ij^{-1} \alpha^i \beta^j} \beta^{-\alpha^i \beta^j} \; mod \; p \\ &\equiv \alpha^{-ij^{-1} \alpha^i \beta^j} \; mod \; p \\ &\equiv \alpha^{-\gamma i j^{-1}} \; mod \; p \\ &\equiv \alpha^x \; mod \; p, \end{aligned}$$

ceea ce arată că (γ, δ) este semnătură pentru x.

Falsificarea semnăturii cunoscând o semnătură Dacă se poate intercepta un cuplu $(x, (\gamma, \delta))$, unde (γ, δ) este semnătură a lui x, atunci se pot semna (valid) şi alte mesaje. In adevăr, pentru orice h, i şi j cu $0 \leq h, i, j \leq p-2$ şi $(h\gamma - j\delta, p-1) = 1$, numerele

$$\begin{aligned} \gamma' &= \gamma^h \alpha^i \beta^j \ mod\, p \\ \delta' &= \delta\gamma'(h\gamma - j\delta)^{-1} \ mod\, (p-1) \\ x' &= \gamma'(hx + i\delta)(h\gamma - j\delta)^{-1} \ mod\, (p-1), \end{aligned}$$

unde $(h\gamma - j\delta)^{-1}$ este determinat modulo $p-1$, verifică

$$\beta^{\gamma'}(\gamma')^{\delta'} \equiv \alpha^{x'} \ mod\, p,$$

ceea ce arată că (γ', δ') este semnătură pentru x'.

După cum putem constata, nici una din metodele de mai sus nu constituie un atac serios la adresa semnăturii ElGamal. De altfel, nu se cunoaşte nici un atac serios la adresa acestei semnături, exceptând unele "neglijenţe" de protocol pe care le vom semnala în continuare.

Cunoaşterea numărului secret k Presupunem că s-a interceptat o pereche de numere $(x, (\gamma, \delta))$, unde (γ, δ) este semnătura lui x. Atunci, cunoaşterea lui k conduce la determinarea imediată a lui a prin relaţia

$$a = (x - k\gamma)\delta^{-1} \ mod\, (p-1),$$

ceea ce compromite semnătura.

Utilizarea aceluiaşi k pentru a semna mesaje diferite Dacă se utilizează acelaşi număr k pentru a semna două mesaje distincte x_1 şi x_2, atunci semnăturile sunt de forma (γ, δ_1) şi (γ, δ_2). Ca urmare,

$$\beta^{\gamma}\gamma^{\delta_1} \equiv \alpha^{x_1} \ mod\, p$$

şi

$$\beta^{\gamma}\gamma^{\delta_2} \equiv \alpha^{x_2} \ mod\, p,$$

ceea ce conduce la

$$\alpha^{x_1 - x_2} \equiv \gamma^{\delta_2 - \delta_1} \ mod\, p.$$

Deoarece $\gamma = \alpha^k \ mod\, p$, relaţia de mai sus conduce la

$$\alpha^{x_1 - x_2} \equiv \alpha^{k(\delta_2 - \delta_1)} \ mod\, p,$$

care este echivalentă cu

$$k(\delta_2 - \delta_1) \equiv x_1 - x_2 \ mod\, (p-1).$$

Aceasta este o ecuaţie modulară în necunoscuta k, ce admite soluţie (k a fost utilizat pentru a semna x_1 şi x_2). Conform Teoremei 4.7.1, soluţiile în $\mathbf{Z}_{p-1}$ ale acestei ecuaţii sunt

$$(k_0 + i(p-1)/d) \; mod \; p - 1,$$

unde k_0 este o soluţie arbitrară a ei, $d = (\delta_2 - \delta_1, p-1)$ şi $0 \leq i < d$.

Determinarea unei soluţii k_0 se poate face cu algoritmul extins al lui Euclid, aşa cum a fost discutat în Secţiunea 4.2. Valoarea reală a lui k se determină testând relaţia $\gamma \equiv \alpha^k \; mod \; p$.

4.11.3.3 Semnătura DSS

Semnătura DSS (Digital Signature Standard) a fost propusă în august 1991 de NIST ca metodă standard de schemă de semnare [48]. După 3 ani de dispute şi critici asupra ei, în mai 1994 semnătura DSS a fost adoptată şi publicată [141]. Ea este, în esenţă, o variaţie a schemei de semnare ElGamal, variaţie generată pe baza următoarei observaţii. Spre deosebire de un criptosistem, o semnătură digitală trebuie să asigure securitate pentru o perioadă îndelungată de timp (semnătura pe un document important trebuie să îşi păstreze intacte calităţile pentru ani şi ani de zile). Deoarece securitatea schemei de semnare ElGamal este bazată în mod direct pe problema logaritmului discret, numărul prim p trebuie ales suficient de mare (512 biţi sau, aşa cum se sugerează, chiar 1024 de biţi). Dacă p este ales de 512 biţi, semnătura ve avea 1024 de biţi, ceea ce este considerat neconvenabil din punct de vedere practic (de exemplu, pentru SmartCard-uri se preferă semnături mult mai scurte). Schema de semnare DSS produce semnături de 320 de biţi fără a compromite securitatea acesteia. Idea de bază constă în utilizarea unui număr prim de 512 biţi (sau chiar 1024 de biţi) şi a unui factor prim q de 160 de biţi al lui $p-1$. Calculele se vor realiza în subgrupul $\mathbf{Z}_q$ al lui $\mathbf{Z}_p^*$, utilizând un element $\alpha \in \mathbf{Z}_p^*$ de ordin q. Atunci, $x \in \mathbf{Z}_p^*$ va fi semnat prin $(\gamma, \delta) \in \mathbf{Z}_q \times \mathbf{Z}_q$ astfel încât x este o combinaţie liniară de γ şi δ cu doi parametri secreţi $a \in \mathbf{Z}_q$ şi $k \in \mathbf{Z}_q^*$,

$$x = (-a\gamma + k\delta) \; mod \; q.$$

Această combinaţie este aleasă modulo q deoarece verificarea semnăturii se va face prin intermediul echivalenţei

$$u \equiv v \; mod \; q \quad \Leftrightarrow \quad \alpha^u \equiv \alpha^v \; mod \; p,$$

pentru orice element $\alpha \in \mathbf{Z}_p^*$ de ordin q. Se poate vedea astfel analogia cu schema ElGamal, înlocuind rădăcina primitivă α (element de ordin $p-1$) printr-un element de ordin q.

Securitatea schemei este bazată pe problema logaritmului discret în $\mathbf{Z}_q$, problemă care, la momentul actual, este intractabilă.

Descrierea semnăturii:

- se aleg numerele p, q şi α astfel încât p este prim, problema logaritmului discret în $\mathbf{Z}_p^*$ este intractabilă, q este un factor prin al lui $p-1$, iar $\alpha \in \mathbf{Z}_p^*$ este un element de ordin q (ca urmare, $\alpha \; mod \; q$ este rădăcină primitivă în $\mathbf{Z}_q^*$);

- $\mathcal{P} = \mathbf{Z}_p^*$;
- $\mathcal{S} = \mathbf{Z}_q \times \mathbf{Z}_q$;
- $\mathcal{K} = \{(p, q, \alpha, a, \beta) | a \in \mathbf{Z}_q \ \wedge \ \beta = \alpha^a \ mod \ p\}$;
- pentru orice cheie $K = (p, q, \alpha, a, \beta)$ şi $k \in \mathbf{Z}_q^*$,

$$sig_K(x, k) = (\gamma, \delta),$$

unde

$$\gamma = (\alpha^k \ mod \ p) \ mod \ q \quad \text{şi} \quad \delta = (x + a\gamma)k^{-1} \ mod \ q,$$

iar

$$ver_K(x, (\gamma, \delta)) = 1 \quad \Leftrightarrow \quad (\alpha^{e_1}\beta^{e_2} \ mod \ p) \ mod \ q = \gamma,$$

unde

$$e_1 = x\delta^{-1} \ mod \ q \quad \text{şi} \quad e_2 = \gamma\delta^{-1} \ mod \ q,$$

pentru orice $x \in \mathbf{Z}_p^*$ (k^{-1} şi δ^{-1} sunt calculate modulo q).

Numerele p, q, α şi β sunt publice (unui grup de utilizatori), iar a este secret (particular fiecărui utilizator în parte).

Verificarea faptului că schema descrisă mai sus este, în adevăr, o schemă de semnare decurge astfel. Relaţia $k\delta \equiv (x + a\gamma) \ mod \ q$ este echivalentă, succesiv, cu:

$$\begin{aligned} k\delta \equiv (x + a\gamma) \ mod \ q \quad &\Leftrightarrow \quad k \equiv (x + a\gamma)\delta^{-1} \ mod \ q \\ &\Leftrightarrow \quad \alpha^k \equiv \alpha^{(x+a\gamma)\delta^{-1}} \ mod \ p \\ &\Leftrightarrow \quad \alpha^k \equiv \alpha^{x\delta^{-1}}(\alpha^a)^{\gamma\delta^{-1}} \ mod \ p \\ &\Leftrightarrow \quad \alpha^k \equiv \alpha^{x\delta^{-1}}\beta^{\gamma\delta^{-1}} \ mod \ p \\ &\Leftrightarrow \quad \alpha^k \equiv \alpha^{e_1}\beta^{e_2} \ mod \ p \end{aligned}$$

(pentru cea de a doua echivalenţă s-a utilizat faptul că α are ordinul q în $\mathbf{Z}_p^*$). De aici urmează $y = (\alpha^{e_1}\beta^{e_2} \ mod \ p) \ mod \ q$.

Exemplul 4.11.3.2. Fie $q = 101$, $p = 78q + 1 = 7879$ şi 3 un element primitiv în $\mathbf{Z}_{7879}$. Considerăm $\alpha = 3^{78} \ mod \ 7879 = 170$ şi $a = 75$. Atunci,

$$\beta = \alpha^a \ mod \ p = 4567.$$

Presupunem că se doreşte a se semna mesajul $x = 1234$ folosind parametrul de securitate $k = 50$ ($k \in \mathbf{Z}_{75}$ şi $k^{-1} = 99$). Numerele γ şi δ vor fi date prin:

$$\gamma = (170^{50} \ mod \ 7879) \ mod \ 101 = 94,$$

şi

$$\delta = (1234 + 75 \cdot 94) \cdot 99 \ mod \ 101 = 97.$$

Ca urmare, $sig_K(x, k) = (94, 97)$.

Algoritmii de semnare şi verificare au complexitate $\mathcal{O}((log\, p)^3)$. Se recomandă ca p şi q să satisfacă

$$2^{l-1} < p < 2^l, \quad 2^{159} < q < 2^{160}$$

cu $512 \leq l \leq 1024$ multiplu de 64 [50].

Numărul α se poate alege pornind de la un element primitiv α_0 în $\mathbf{Z}_p$, prin

$$\alpha = \alpha_0^{\frac{p-1}{q}}\ mod\ p.$$

Prin modul de alegere al lui α, numerele β şi γ vor fi, de asemenea, rădăcini de ordin q ale lui 1 modulo p. Necesitatea calculului lui δ^{-1} cere îndeplinirea relaţiei $\delta \not\equiv 0\ mod\ q$ (probabilitatea de a se obţine un δ cu $\delta \equiv 0\ mod\ q$ este de 2^{-160}).

Capitolul 5

Completitudine în teoria mulţimilor parţial ordonate

5.1 Completitudine

Noţiunile de completitudine joacă un rol deosebit de important în teoria mulţimilor parţial ordonate şi a aplicaţiilor acesteia. In general, o mpo este numită completă dacă orice submulţime a ei ce satisface o proprietate dată admite infimum şi/sau supremum. Prima noţiune de completitudine a fost introdusă de Birkhoff în 1933 [8] prin considerarea conceptului de latice completă (mpo pentru care orice submulţime nevidă admite infimum şi supremum). Ulterior, apar şi alte tipuri de completitudine; acestea, împreună cu câteva proprietăţi de bază ale lor, vor fi discutate în capitolul de faţă.

Multe studii în informatica teoretică asupra "recursivităţii" fac apel la studiul supremului unor tipuri de lanţuri [20, 103, 153, 191]. Este natural atunci a introduce un concept de completitudine prin lanţuri. Meritul unui studiu profund al acestui tip de completitudine aparţine lui Markowski [119, 120, 121].

Definiţia 5.1.1. Spunem că o mpo $M = (A; \leq)$ este *completă prin lanţuri* dacă orice lanţ în M admite supremum.

Observaţia 5.1.1.

(1) Definiţia 5.1.1 poate fi reformulată echivalent prin: M este mpo completă prin lanţuri dacă au loc următoarele proprietăţi:

- M are cel mai mic element;
- orice lanţ nevid $L \subseteq A$ admite suprem.

Această reformulare este bazată pe faptul că existenţa supremului lanţului vid este echivalentă cu existenţa celui mai mic element $\perp_A$. De multe ori vom prefera să folosim această variantă deoarece, în demonstraţii, apare frecvent necesitatea clasificării lanţurilor în vide şi nevide.

(2) Unii autori cer în Definiţia 5.1.1 doar existenţa supremului lanţurilor nevide, iar ceea ce am numit noi mpo completă ei numesc *mpo completă pointată*. Atunci când nu vom cere existenţa celui mai mic element ne vom referi la astfel de mpo ca fiind *mpo slab complete* (deci, în astfel de mpo este asigurat supremum oricărui lanţ nevid; cel mai mic element poate să existe sau nu).

Exemplul 5.1.1.

(1) Orice mpo care are cel mai mic element şi pentru care orice lanţ este finit, este completă. In particular, ordinalii finiţi şi mpo plate sunt mpo complete.

(2) $(\mathcal{P}(A); \subseteq)$, unde A este o mulţime arbitrară, este mpo completă.

(3) Mulţimea $\mathbf{N}$ a numerelor naturale împreună cu ordinea uzuală nu este mpo completă deoarece lanţul $\mathbf{N}$ nu admite supremum.

(4) Fie A şi B două mulţimi şi $(A \rightsquigarrow B)$ mulţimea tuturor funcţiilor parţiale pe A cu valori în B. Considerăm pe $(A \rightsquigarrow B)$ relaţia binară $\leq$ dată prin

$$f \leq g \Leftrightarrow Dom(f) \subseteq Dom(g) \wedge (\forall x \in Dom(f))(f(x) = g(x)),$$

pentru orice $f, g \in (A \rightsquigarrow B)$. Atunci, $((A \rightsquigarrow B); \leq)$ este mpo completă. In adevăr, dacă L este un lanţ în $(A \rightsquigarrow B)$, funcţia $\bigcup L$ (existenţa acesteia este asigurată de faptul că $(A \rightsquigarrow B)$ este sistem de funcţii compatibile două câte două (Secţiunea 1.1.4)) este supremul acestui lanţ.

Lema 5.1.1. Fie M şi M' două mpo izomorfe. Atunci, M este completă dacă şi numai dacă M' este completă.

Demonstraţie Fie $M = (A; \leq)$ şi $M' = (A'; \leq')$ două mpo şi $f : A \to A'$ un izomorfism între ele. Presupunem că M este completă. Atunci,

– $f(\perp_M)$ este cel mai mic element al mpo M';

– pentru orice lanţ L' în M', $f^{-1}(L')$ este lanţ în M. Atunci, $sup_M(f^{-1}(L'))$ există şi $sup_{M'}(L') = f(sup_M(f^{-1}(L')))$.

Deci, M' este completă.

Un raţionament similar, realizat prin prisma funcţiei f^{-1}, ne arată că dacă M' este completă atunci M este completă. □

Fie A o mulţime şi $(B, \leq)$ o mpo. Pe mulţimea $(A \to B)$ a tuturor funcţiilor de la A la B definim relaţia binară $\leq_{(A \to B)}$ prin:

$$f \leq_{(A \to B)} g \quad \Leftrightarrow \quad (\forall a \in A)(f(a) \leq g(a)),$$

pentru orice $f, g : A \to B$. Este uşor de verificat că $\leq_{(A \to B)}$ este ordine parţială pe mulţimea $(A \to B)$.

Fie $S \subseteq (A \to B)$, $a \in A$ şi $S(a) = \{f(a) | f \in S\}$. Dacă $S = \emptyset$, atunci $S(a) = \emptyset$. Următoarea lemă, ce are un caracter tehnic, va fi intens utilizată în multe din demonstraţiile ce urmează.

Lema 5.1.2. Fie A o mulţime şi $(B;\leq)$ o mpo. Atunci, pentru orice submulţime nevidă $S \subseteq (A \to B)$ are loc

$$\exists sup(S) \quad \Leftrightarrow \quad (\forall a \in A)(\exists sup(S(a))).$$

In plus, dacă există $sup(S)$ atunci are loc

$$(\forall a \in A)(((sup(S))(a) = sup(S(a)))$$

(supremul mulţimii S este considerat în raport cu $\leq_{(A\to B)}$, iar cel al mulţimii $S(a)$ în raport cu $\leq$).

Demonstraţie Verificarea primei părţi a lemei constituie un simplu exerciţiu lăsat în seama cititorului.

Presupunem că există $sup(S)$, fie acesta $f : A \to B$. Este clar că pentru orice $a \in A$, $f(a)$ este majorant pentru $S(a)$. Dacă presupunem, prin contradicţie, că există $a \in A$ astfel încât $f(a)$ nu este cel mai mic majorant pentru $S(a)$, atunci $sup(S(a)) < f(a)$ (există $sup(S(a))$ conform ipotezei şi primei părţi a lemei). Considerăm funcţia $f' : A \to B$ dată prin

$$f'(x) = \begin{cases} f(x), & x \neq a \\ b, & x = a \end{cases}$$

pentru orice $x \in A$. Atunci, este uşor de verificat că f' este majorant pentru S şi $f' <_{(A\to B)} f$, ceea ce contrazice faptul că f este cel mai mic majorant al mulţimii S.

Deci, există $sup(S(a))$ şi acesta este $f(a) = (sup(S))(a)$, pentru orice $a \in A$.

Reciproc, presupunem că există $sup(S(a))$ pentru orice $a \in A$. Considerăm funcţia $f : A \to B$ dată prin $f(a) = sup(S(a))$, pentru orice $a \in A$. Este uşor de văzut că f este cel mai mic majorant pentru S. Ca urmare, există supremum mulţimii S şi, în plus, $(sup(S))(a) = sup(S(a))$ pentru orice $a \in A$. □

Teorema 5.1.1. Fie A o mulţime nevidă şi $(B;\leq)$ o mpo. Dacă $(B;\leq)$ este mpo completă, atunci $((A \to B);\leq_{(A\to B)})$ este mpo completă.

Demonstraţie Cel mai mic element al mpo $((A \to B);\leq_{(A\to B)})$ este funcţia $\bot_{(A\to B)} : A \to B$ dată prin $\bot_{(A\to B)}(a) = \bot_B$ pentru orice $a \in A$.

Fie $L \subseteq (A \to B)$ un lanţ nevid. Atunci, pentru orice $a \in A$, $L(a)$ este lanţ nevid în B. Deoarece $(B;\leq)$ este mpo completă, există $sup(L(a))$ pentru orice $a \in A$. Atunci, Lema 5.1.2 asigură existenţa supremului lanţului L. Ca urmare, $((A \to B);\leq_{(A\to B)})$ este mpo completă. □

Exemplul 5.1.2. Mulţimea parţial ordonată $((\mathbf{N}_\bot \to \mathbf{N}_\bot);\leq_{(\mathbf{N}_\bot\to\mathbf{N}_\bot)})$ este completă şi, ca urmare, orice lanţ va admite supremum. Să considerăm lanţul de funcţii $L = \{f_i | i \geq 0\} \subseteq (\mathbf{N}_\bot \to \mathbf{N}_\bot)$ dat ca în diagrama de mai jos:

L	$\bot$	0	1	2	3	$\cdots$
f_0	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\cdots$
f_1	$\bot$	1	$\bot$	$\bot$	$\bot$	$\cdots$
f_2	$\bot$	1	$1!$	$\bot$	$\bot$	$\cdots$
f_3	$\bot$	1	$1!$	$2!$	$\bot$	$\cdots$
$\cdots$	$\cdots$					

Acest lanţ admite supremum, ce poate fi determinat utilizând Lema 5.1.2 ca fiind:

$$sup(L)(x) = \begin{cases} 1, & x = 0 \\ x!, & x \in \mathbf{N} \\ \perp, & x = \perp, \end{cases}$$

pentru orice $x \in \mathbf{N}_\perp$. Observăm că lanţul L constituie o aproximare a funcţiei factorial: f_0 aproximează funcţia factorial pentru $\perp$, f_1 aproximează funcţia factorial pentru $\perp$ şi 0 etc. Aceste aproximări sunt înţelese în sensul

$$f_0 \leq_{\mathbf{N}_\perp \to \mathbf{N}_\perp} f_1 \leq_{\mathbf{N}_\perp \to \mathbf{N}_\perp} f_2 \leq_{\mathbf{N}_\perp \to \mathbf{N}_\perp} \cdots \leq_{\mathbf{N}_\perp \to \mathbf{N}_\perp} sup(L)$$

Exemplul 5.1.3. Exemplul 5.1.2 poate fi generalizat în mod natural. Orice funcţie $f : \mathbf{N}_\perp \to \mathbf{N}_\perp$ poate fi aproximată printr-un lanţ L de funcţii definite ca în diagrama de mai jos:

L	$\perp$	0	1	2	$\cdots$
f_0	$f(\perp)$	$\perp$	$\perp$	$\perp$	$\cdots$
f_1	$f(\perp)$	$f(0)$	$\perp$	$\perp$	$\cdots$
f_2	$f(\perp)$	$f(0)$	$f(1)$	$\perp$	$\cdots$
$\cdots$	$\cdots$				

Operaţii cu mpo complete. Vom studia acum modul în care se păstrează proprietatea de completitudine prin trecere la intersecţie, reuniune, sume şi produse ordonate (aşa cum au fost definite în Secţiunea 3.2.4).

Intersecţia unei familii $((A_i; \leq_i) | i \in I)$ de mpo complete poate să nu fie mpo completă din două motive:

1. prin intersecţie nu se păstrează cel mai mic element. De exemplu, să considerăm $\mathbf{Z}_1 = \mathbf{Z}_- \cup \{\perp_1\}$ şi $\mathbf{Z}_2 = \mathbf{Z}_- \cup \{\perp_2\}$, unde $\mathbf{Z}_-$ este mulţimea numerelor întregi negative, iar $\perp_1$ şi $\perp_2$ sunt două elemente diferite între ele şi diferite de orice număr întreg. Aceste două mulţimi cu ordinea naturală pe $\mathbf{Z}_-$ extinsă prin considerarea elementelor $\perp_1$ şi respectiv $\perp_2$ ca fiind cele mai mici elemente, conduc la mpo complete. Intersecţia lor este $\mathbf{Z}_-$ ce nu mai are cel mai mic element;

2. prin intersecţie nu se mai asigură supremum unor lanţuri (intersecţia dualelor mulţimilor parţial ordonate de la punctul anterior ne arată aceasta).

Reuniunea unei familii disjuncte $((A_i; \leq_i) | i \in I)$ de mpo complete poate să nu fie mpo completă dintr-un singur motiv: nu se asigură existenţa celui mai mic element. Insă, $\perp\!\!\!\perp \oplus \bigcup_{i \in I}(A_i; \leq_i)$ este mpo completă (considerând $\perp$ un nou element). Dacă $(A_i; \leq_i)$ nu este o familie disjunctă (dar elementele ei sunt mpo complete), $\perp\!\!\!\perp \oplus \uplus_{i \in I}(A_i; \leq_i)$ este mpo completă.

Vom analiza acum cazul sumei ordonate.

Propoziţia 5.1.1. Fie $\mathcal{I} = (I; \leq)$ o mto completă şi $((A_1; \leq_i) | i \in I)$ o familie de mpo complete disjuncte. Atunci, suma ordonată $\sum^o_{i \in I}(A_i; \leq_i)$ este mpo completă.

Demonstraţie Fie $(A;\leq') = \sum^{o}_{i\in I}(A_i;\leq_i)$. Cel mai mic element al acestei mpo este cel mai mic element al mpo $(A_{i_0};\leq_{i_0})$, unde i_0 este cel mai mic element al mto $\mathcal{I}$ (existenţa acestuia este asigurată de faptul că $\mathcal{I}$ este completă).

Fie $L \subseteq A$ un lanţ şi $K = \{i \in I | \exists a \in L : \ a \in A_i\}$. Mulţimea K este lanţ în I. Deoarece $\mathcal{I}$ este completă, va exista $k = sup(K)$. Considerăm acum sublanţul $L' = L \cap A_k$ al lanţului L. Deoacere $(A_k;\leq_k)$ este completă, există $sup_{A_k}(L')$ şi este uşor de arătat că $sup_A(L) = sup_{A_k}(L')$. □

Există cazuri în care disjunctivitatea familiei $((A_i;\leq_i)|i \in I)$ nu este asigurată. Suma ordonată disjunctă (Secţiunea 3.2.4) rezolvă această problemă dar, cel mai mic element al ei poate să nu existe chiar dacă fiecare mpo din familie are un cel mai mic element. Remedierea acestei situaţii se pot face astfel. Fie $\{(A_1;\leq_i)|i \in I\}$ o familie de mpo. Atunci, *suma separată* a acestei familii, notată $\sum^{s}_{i\in I}(A_i;\leq_i)$, este mpo

$$\sum_{i\in I}^{s}(A_i;\leq_i) = (A;\leq'),$$

unde $A = (\bigcup_{i\in I} A_i \times \{i\}) \cup \{\perp\}$, $\perp$ este un nou element iar $\leq'$ este ordinea parţială

$$x \leq' y \ \Leftrightarrow \text{ are loc una din următoarele două proprietăţi:}$$

1. există $i \in I$ şi $a, b \in A_i$ astfel încât $x = (a, i)$, $y = (b, i)$ şi $a \leq_i b$, sau
2. $x = \perp$ şi $y \in A$,

pentru orice $x, y \in A$.

Suma separată este oarecum similară sumei disjuncte, diferenţa constând în faptul că se introduce un cel mai mic element. Este cât se poate de clar că sumă separată de mpo complete este mpo completă.

Prin sumă separată sunt păstrate posibilele elemente minimale ale mpo în cauză. Următoarea variantă de sumă elimină aceste elemente.

Fie $\{(A_1;\leq_i)|i \in I\}$ o familie de mpo. Atunci, *suma de fuziune* a acestei familii, notată $\sum^{f}_{i\in I}(A_i;\leq_i)$, este mpo

$$\sum_{i\in I}^{f}(A_i;\leq_i) = (A;\leq'),$$

unde $A = (\bigcup_{i\in I}(A_i - \{\perp_{A_i}\}) \times \{i\}) \cup \{\perp\}$, $\perp$ este un nou element iar $\leq'$ este ordinea parţială

$$x \leq' y \ \Leftrightarrow \text{ are loc una din următoarele două proprietăţi:}$$

1. există $i \in I$ şi $a, b \in A_i - \{\perp_{A_i}\}$ astfel încât $x = (a, i)$, $y = (b, i)$ şi $a \leq_i b$, sau
2. $x = \perp$ şi $y \in A$,

pentru orice $x, y \in A$.

Suma de fuziune poate fi considerată un caz particular de sumă separată (mpo în cauză nu au cel mai mic element). Ca urmare, sumă de fuziune de mpo complete este mpo completă.

In cazul produsului ordonat (direct) al unei familii indexate de mpo remarcăm încă de la început că acesta nu face apel la nici un fel de ordine (parţială sau totală) pe mulţimea indexilor familiei. Primul nostru rezultat nu este altceva decât o simplă generalizate a Lemei 5.1.2.

Lema 5.1.3. Fie $((A_i; \leq_i)|i \in I)$ o familie nevidă de mpo. Atunci, pentru orice submulţime nevidă $S \subseteq \prod_{i \in I} A_i$ are loc

$$\exists sup(S) \quad \Leftrightarrow \quad (\forall i \in I)(\exists sup(S(i))).$$

In plus, dacă există $sup(S)$, atunci are loc

$$(\forall i \in I)(((sup(S))(i) = sup(S(i))).$$

Demonstraţie Similară Lemei 5.1.2. □

Dacă în Lema 5.1.3 alegem $I = A$, $A_i = B$ şi $\leq_i = \leq$ pentru orice $i \in I$, atunci obţinem rezultatul din Lema 5.1.2.

Propoziţia 5.1.2. Fie $((A_i; \leq_i)|i \in I)$ o familie nevidă de mpo complete. Atunci, $\prod_{i \in I}(A_i; \leq_i)$ este mpo completă.

Demonstraţie Cel mai mic element al mpo $\prod_{i \in I}(A_i; \leq_i)$ este funcţia $\perp_{\prod_{i \in I} A_i}$ dată prin $\perp_{\prod_{i \in I} A_i}(i) = \perp_{A_i}$, pentru orice $i \in I$.

Restul demonstraţiei decurge similar demonstraţiei Teoremei 5.1.1. □

Corolarul 5.1.1. Fie $n \geq 1$ un număr natural şi $(A_i; \leq_i)$ mpo complete, unde $1 \leq i \leq n$. Atunci, $\times_{i=1}^{n}(A_i; \leq i)$ este mpo completă.

Demonstraţie Este uşor de verificat că $\times_{i=1}^{n}(A_i; \leq_i)$ şi $\prod_{i \in \{1,\ldots,n\}}(A_i; \leq_i)$ sunt mpo izomorfe. Atunci, Propoziţia 5.1.2 şi Lema 5.1.1 conduc direct la faptul că $\times_{i=1}^{n}(A_i; \leq i)$ este mpo completă. □

Exemplul 5.1.4. Produs direct sau cartezian de mpo plate este mpo completă.

Submulţimi parţial ordonate complete. Vom discuta acum conceptul de submulţime parţial ordonată completă care, menţionăm încă de la început, nu trebuie confundat cu cel de sub-mpo care este şi completă. Vom începe cu o observaţie care va justifica introducerea conceptului.

Observaţia 5.1.2. Fie $M = (A; \leq)$ o mpo completă şi $M' = (A'; \leq')$ o sub-mpo a lui M.

(1) Dacă M' este completă, atunci pentru orice lanţ nevid L în M' există $sup_M(L)$ şi $sup_M(L) \leq sup_{M'}(L)$. Această inegalitate poate fi strictă. Drept exemplu să considerăm $A = \mathcal{P}(\mathbf{N})$ şi $A' = \{B \in A | B \text{ finită}\} \cup \{\mathbf{N}\}$. Atunci, în raport cu incluziunea, A şi A' sunt mulţimi parţial ordonate complete. In plus, A' este submulţime parţial ordonată a lui A . Fie lanţul

$$L = \{\{0\}, \{0,2\}, \{0,2,4\}, \ldots\}$$

în A'. Observăm că $sup_{M'}(L) = \mathbf{N}$, $sup_M(L) = \{n \in \mathbf{N} | n \text{ este par}\}$ şi $sup_M(L) \subset sup_{M'}(L)$.

(2) Dacă M' are cel mai mic element şi $sup_M(L) \in M'$, pentru orice lanţ nevid L în M', atunci M' este completă. In acest caz are loc $sup_M(L) = sup_{M'}(L)$, pentru orice lanţ nevid L în M'.

Definiţia 5.1.2. Fie $M = (A; \leq)$ o mpo completă şi $M' = (A'; \leq')$ o sub-mpo a lui M. Spunem că M' este *submulţime parţial ordonată completă* (abreviat, *sub-mpo completă*) a lui M dacă pentru orice lanţ nevid L în M' are loc $sup_M(L) \in A'$.

Observaţia 5.1.3.

(1) Definiţia 5.1.2 este în concordanţă cu Observaţia 5.1.1(1) în sensul că am putea înlocui cerinţa din această definiţie prin

(1') $\perp_M \in A'$;

(2') $(\forall L \subseteq A')(L \text{ lanţ nevid în } M' \Rightarrow sup_M(L) \in A')$.

Aceasta se bazează pe faptul că $sup_M(\emptyset) = \perp_M$, care conform cerinţei (1') trebuie să fie în A'. Dorim însă să subliniem că dacă la (1') s-ar cere "M' are cel mai mic element", atunci această concordanţă nu s-ar mai păstra deoarece ar fi posibil ca M' să aibă un cel mai mic element diferit de $\perp_M$. Ca urmare, putem spune că M' este sub-mpo completă a lui M dacă M' păstrează atât cel mai mic element al lui M cât şi supremum lanţurilor.

(2) Cerinţa din Definiţia 5.1.2 poate fi modificată echivalent la: pentru orice lanţ $L \subseteq A'$ există $sup_{M'}(L)$ şi are loc $sup_{M'}(L) = sup_M(L)$.

(3) Orice sub-mpo completă a unei mpo complete este mpo completă.

Propoziţia 5.1.3. Intersecţia unei familii (nevide) de sub-mpo complete ale unei mpo complete este sub-mpo completă.

Demonstraţie Fie $\mathcal{M}$ o familie nevidă de sub-mpo complete a mpo complete $M = (A; \leq)$.

Fiecare sub-mpo din $\mathcal{M}$ păstrează cel mai mic element al mpo M. Ca urmare, acesta se va găsi şi în $\bigcap \mathcal{M}$. Similar, orice sub-mpo din $\mathcal{M}$ păstrează supremul lanţurilor calculat în M. Ca urmare, supremul oricărui lanţ nevid din $\mathcal{M}$, calculat în M, va fi în $\mathcal{M}$. □

Reamintim că mulţimea tuturor funcţiilor monotone de la o mpo $M = (A; \leq)$ la o mpo $M' = (A'; \leq')$ este notată prin $(M \rightarrow_m M')$ sau $(A \rightarrow_m A')$, atunci când nu există pericol de confuzie. Este clar că $(A \rightarrow_m A') \subseteq (A \rightarrow A')$.

Fie $\leq_{(A \rightarrow_m A')} = \leq_{(A \rightarrow A')} |_{(A \rightarrow_m A')}$. Cu această relaţie, $(A \rightarrow_m A')$ devine sub-mpo a mulţimii parţial ordonate $(A \rightarrow A')$. Vom arăta că $(A \rightarrow_m A')$ este chiar sub-mpo completă.

Teorema 5.1.2. Fie $M = (A, \leq)$ şi $M' = (A', \leq')$ două mpo. Dacă M' este completă, atunci $(A \rightarrow_m A')$ este sub-mpo completă a mpo complete $(A \rightarrow A')$.

Demonstraţie Funcţia $\perp_{(A \rightarrow A')}$ este monotonă, ceea ce ne spune că este element al mpo $(A \rightarrow_m A')$.

Fie $L \subseteq (A \rightarrow_m A')$ un lanţ nevid de funcţii monotone. Acest lanţ admite supremum în $(A \rightarrow A')$. Vom arăta că acest supremum este funcţie monotonă. Fie $a, b \in A$ cu $a \leq b$. Vom arăta că are loc

$$(sup_{(A \rightarrow A')}(L))(a) \leq' (sup_{(A \rightarrow A')}(L))(b).$$

Conform Lemei 5.1.2, este suficient de arătat că are loc

$$sup_{(A \rightarrow A')}(L(a)) \leq' sup_{(A \rightarrow A')}(L(b)),$$

iar conform Propoziţiei 3.2.3.5 este suficient de arătat că $L(b)$ este cofinal în $L(a)$. In primul rând remarcăm că $L(a)$ şi $L(b)$ sunt lanţuri nevide în M' deoarece L este lanţ nevid de funcţii. Pe de altă parte, orice element din $L(a)$ este de forma $f(a)$ cu $f \in L$. Cum f este funcţie monotonă şi $a \leq b$, deducem că are loc $f(a) \leq' f(b)$, iar $f(b)$ este element al lanţului $L(b)$. Deci, $L(b)$ este cofinală în $L(a)$. Ca urmare, $sup_{(A \rightarrow A')}(L) \in (A \rightarrow_m A')$.

Deci, $(A \rightarrow_m A')$ este sub-mpo completă a mpo complete $(A \rightarrow A')$. □

Observaţia 5.1.4. Teorema 5.1.2 poate fi reformulată echivalent astfel.

Fie $M = (A, \leq)$ şi $M' = (A', \leq')$ două mpo. Dacă $M' = (A', \leq')$ este completă, atunci supremum oricărui lanţ nevid de funcţii monotone de la M la M' este funcţie monotonă.

5.2 Teoria de punct fix a mulţimilor parţial ordonate

5.2.1 Funcţii continue

In analiza matematică, o funcţie f este continuă dacă este compatibilă cu limita şirurilor numerice în sensul că, pentru orice şir convergent $(a_n)_{n \geq 0}$ are loc

$$f(\lim_{n \rightarrow \infty} a_n) = \lim_{n \rightarrow \infty} f(a_n).$$

In cazul mulţimilor parţial ordonate, limita este supremum sau infimum unei submulţimi (submulţimile pot fi arbitrare, dirijate, lanţuri etc.). Ca urmare, este natural să

extindem conceptul de continuitate de mai sus la funcţii definite pe mpo prin a cere compatibilitatea acestora cu supremum sau infimum, de exemplu în sensul

$$f(sup(S)) = sup(f(S)),$$

pentru orice submulţime S ce satisface o anumită proprietate dată a priori (de exemplu, cea de lanţ). Ca o astfel de definiţie să funcţioneze, domeniul pe care este definită funcţia în cauză trebuie să aibă proprietatea că pentru orice submulţime S ca mai sus, $sup(S)$ este un element al domeniului. Ca urmare, domeniul trebuie să fie o mpo completă într-un sens bine precizat (prin submulţimi, prin mulţimi dirijate, prin lanţuri etc.).

Conceptul de funcţie continuă definită pe mpo complete (prin lanţuri) este unul din cele mai studiate concepte de continuitate. Acesta va fi cel pe care îl vom prezenta şi noi în cele ce urmează.

Definiţia 5.2.1.1. Fie $M = (A; \leq)$ şi $M' = (A'; \leq')$ mpo complete şi $f : A \to A'$ o funcţie. Spunem că f este *continuă* dacă pentru orice lanţ nevid L în M există $sup(f(L))$ şi $f(sup(L)) = sup(f(L))$.

Observaţia 5.2.1.1.

(1) Cerinţa "există $sup(f(L))$" din Definiţia 5.2.1.1 este necesară deoarece în mpo complete este asigurat supremul lanţurilor dar definiţia nu garantează că $f(L)$ este lanţ.

(2) In Definiţia 5.2.1.1 se cere ca lanţul L să fie nevid. Dacă s-ar da posibilitatea ca relaţia din definiţie să fie satisfăcută şi de lanţul vid, atunci s-ar obţine

$$f(\perp_M) = f(sup(\emptyset)) = sup(f(\emptyset)) = sup(\emptyset) = \perp_{M'},$$

deci s-ar păstra şi cel mai mic element al mulţimilor parţial ordonate. In cazul în care o funcţie f satisface şi această condiţie suplimentară ea este numită *funcţie continuă strictă* sau *funcţie continuă în sens strict*.

Funcţiile monotone păstrează lanţurile şi, ca urmare, este de aşteptat să existe o legătură destul de strânsă între monotonie şi continuitate.

Teorema 5.2.1.1. (Continuitate şi monotonie)
Fie $M = (A; \leq)$ şi $M' = (A'; \leq')$ mpo complete şi $f : A \to A'$ o funcţie. Atunci, f este continuă dacă şi numai dacă:

(1) f este monotonă;

(2) pentru orice lanţ nevid $L \subseteq A$, $f(sup(L)) \leq' sup(f(L))$.

Demonstraţie Să presupunem că f este continuă. Fie $a, b \in A$ cu $a \leq b$. Considerăm lanţul $L = \{a, b\}$ al cărui suprem este b. Continuitatea funcţiei f conduce atunci la

$$f(b) = f(sup(L)) = sup(f(L)) = sup(\{f(a), f(b)\}),$$

de unde urmează $f(a) \leq' f(b)$. Deci, f este monotonă.

Inegalitatea de la (2) urmează direct de la definiţia continuităţii.

Reciproc, presupunem că sunt îndeplinite condiţiile (1) şi (2) din enunţul teoremei. Fie $L \subseteq A$ un lanţ nevid. Monotonia funcţiei f conduce la faptul că $f(L)$ este lanţ (în M'), iar completitudinea mulţimii parţial ordonate M' conduce la existenţa supremului acestui lanţ. In plus, $f(sup(L))$ este majorant al lanţului $f(L)$ deoarece $sup(L)$ este majorant al lanţului L şi f este monotonă. Cum $sup(f(L))$ este cel mai mic majorant al lanţului $f(L)$, obţinem

$$sup(f(L)) \leq' f(sup(L)),$$

care combinată cu inegalitatea de la (2) conduce la $f(sup(L)) = sup(f(L))$. Deci, f este continuă. □

Observaţia 5.2.1.2. Punctul (2) al Teoremei 5.2.1.1 poate fi reformulat echivalent prin:

(2') pentru orice lanţ nevid $L \subseteq A$, $f(sup(L)) = sup(f(L))$.

Corolarul 5.2.1.1. Fie $M = (A; \leq)$ şi $M' = (A'; \leq')$ mpo complete şi $f : A \to A'$ o funcţie. Dacă M are numai lanţuri finite, atunci f este continuă dacă şi numai dacă este monotonă.

Demonstraţie Ceea ce avem de arătat este că monotonia funcţiei f implică continuitatea acesteia, în ipoteza în care M are numai lanţuri finite.

Fie $L \subseteq A$ un lanţ nevid. Conform ipotezei, L este finit şi, deci, are cel mai mare element, fie acesta a. Atunci, monotonia funcţiei f conduce la

$$sup(f(L)) \leq' f(a) = f(sup(L)).$$

Deci, conform Teoremei 5.2.1.1, f este continuă. □

Observaţia 5.2.1.3. In general, nu orice funcţie monotonă este continuă. Să considerăm, spre exemplu, funcţia $\varphi : (\mathbf{N}_\perp \to \mathbf{N}_\perp) \to \mathbf{N}_\perp$ dată prin

$$\varphi(f) = \begin{cases} 1, & (\forall n \in \mathbf{N})(f(n) \neq \perp) \\ \perp, & \text{altfel,} \end{cases}$$

pentru orice $f \in (\mathbf{N}_\perp \to \mathbf{N}_\perp)$. Dacă $\varphi(f) = 1$ atunci vom spune că f este *totală*; altfel, spunem că f este *parţială*.

Este uşor de văzut că φ este funcţie monotonă. In adevăr, dacă considerăm două funcţii $f, g \in (\mathbf{N}_\perp \to \mathbf{N}_\perp)$ astfel încât $f \leq_{(\mathbf{N}_\perp \to \mathbf{N}_\perp)} g$, atunci:

– dacă f este parţială, $\varphi(f) = \perp \leq \varphi(g)$ ($\leq$ este ordinea parţială pe $\mathbf{N}_\perp$);

– dacă f este totală atunci g este totală şi, deci, $\varphi(f) = \varphi(g)$.

Fie acum lanţul de funcţii L din diagrama de mai jos.

L	$\perp$	0	1	2	3	$\cdots$
f_0	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\cdots$
f_1	$\perp$	0	$\perp$	$\perp$	$\perp$	$\cdots$
f_2	$\perp$	0	0	$\perp$	$\perp$	$\cdots$
$\cdots$	$\cdots$					
$sup(L)$	$\perp$	0	0	0	0	$\cdots$

Supremum acestui lanţ este funcţia ce ia valoarea 0 pentru orice număr natural, şi $\perp$ pentru $\perp$. Ca urmare, $\varphi(sup(L)) = 1$. Pe de altă parte, $\varphi(f_i) = \perp$, pentru orice $i \geq 0$. Deci,

$$\varphi(sup(L)) = 1 \neq \perp = sup(\varphi(L)),$$

ceea ce ne arată că φ nu este continuă.

Ca urmare, "ne-continuitatea" funcţiei φ rezidă din faptul că există lanţuri de funcţii parţiale al căror supremum este funcţie totală. Dacă unei funcţii parţiale îi este asociată o anumită valoare (prin φ), iar unei funcţii totale o altă valoare, atunci φ nu poate păstra supremum unor astfel de lanţuri.

Incheiem observaţia prin remarca că dacă înlocuim "$\perp$" din definiţia funcţiei φ prin 0, atunci φ nu este nici monotonă şi nici continuă.

Exemplul 5.2.1.1.

(1) Dacă $M_i = (A_i; \leq_i)$ sunt mpo plate, $1 \leq i \leq n$, şi $M = (A; \leq)$ este o mpo completă, atunci orice funcţie monotonă $f : A_1 \times \cdots \times A_n \to A$ este continuă. In adevăr, mpo plate sunt complete şi produs cartezian de mpo complete este mpo completă. In plus, $\times_{i=1}^{n}(A_i; \leq_i)$ are numai lanţuri finite.

(2) Orice funcţie constantă (definită pe o mpo completă şi cu valori într-o mpo completă) este continuă.

(3) Funcţia identitate (definită pe o mpo completă şi cu valori în aceeaşi mpo) este continuă.

(4) Funcţiile proiecţie $pr_i : A_1 \times \cdots \times A_n \to A_i$, unde $1 \leq i \leq n$ şi $(A_j, \leq_j)$ sunt mpo complete pentru orice $1 \leq j \leq n$, sunt funcţii continue.

Teorema 5.2.1.2. Compunere de funcţii continue este funcţie continuă.

Demonstraţie Fie $M_1 = (A_1; \leq_1)$, $M_2 = (A_2; \leq_2)$ şi $M_3 = (A_3; \leq_3)$ mpo complete, iar $f : A_1 \to A_2$ şi $g : A_2 \to A_3$ funcţii continue.

Fie $L \subseteq A_1$ un lanţ nevid. Deoarece f este continuă, $f(L) \subseteq A_2$ este lanţ nevid, există $sup(f(L))$ şi $f(sup(L)) = sup(f(L))$.

$f(L)$ fiind lanţ şi g fiind continuă, există $sup(g(f(L)))$ şi

$$g(sup(f(L))) = sup(g(f(L))).$$

Conbinând această relaţie cu cea de mai sus obţinem:

$$g(f(sup(L))) = g(sup(f(L))) = sup(g(f(L))),$$

ceea ce arată că $g \circ f$ este funcţie continuă. □

Continuitatea funcţiilor de tipul $f : A \to A_1 \times \cdots \times A_n$**.** Studiul continuităţii acestui tip de funcţii face apel direct la funcţiile proiecţie care sunt continue.

Teorema 5.2.1.3. Fie $M = (A; \leq)$ şi $M_i = (A_i; \leq_i)$ mpo complete, unde $n \geq 2$ şi $1 \leq i \leq n$, şi $f : A \to A_1 \times \cdots \times A_n$ o funcţie. Atunci, f este continuă dacă şi numai dacă $pr_i \circ f$ sunt funcţii continue, pentru orice $1 \leq i \leq n$.

Demonstraţie Să presupunem că f este continuă. Atunci, cum funcţia proiecţie pr_i este continuă şi compunere de funcţii continue este funcţie continuă, rezultă că $pr_i \circ f$ este funcţie continuă, pentru orice i.

Reciproc, presupunem că pentru orice i, $pr_i \circ f$ este funcţie continuă. Vom arăta întâi că f este monotonă.

Fie $a, b \in$A cu $a \leq b$. Atunci, pentru orice i, $pr_i(f(a)) \leq_i pr_i(f(b))$ (deoarece $pr_i \circ f$ este monotonă). Ca urmare, conform definiţiei relaţiei de ordine parţială $\leq'$ pe $M_1 \times \cdots \times M_n$, are loc $f(a) \leq' f(b)$. Deci, f este monotonă.

Fie acum $L \subseteq A$ un lanţ nevid. Pentru a arăta că

$$f(sup(L)) \leq' sup(f(L))$$

avem de arătat că

$$pr_i(f(sup(L))) \leq_i pr_i(sup(f(L))),$$

pentru orice i. Are loc:

$$\begin{array}{rcll} pr_i(f(sup(L))) & = & sup(pr_i(f(L))) & (pr_i \circ f \text{ este continuă}) \\ & = & pr_i(sup(f(L))) & (pr_i \text{ este continuă şi } f(L) \text{ este lanţ}) \end{array}$$

pentru orice i. Deci, f este continuă. □

Corolarul 5.2.1.2. Fie $M = (A; \leq)$ şi $M_i = (A_i; \leq_i)$ mpo complete, unde $n \geq 2$ şi $1 \leq i \leq n$, şi $f_i : A_i \to A_i'$ funcţii continue. Atunci, funcţia $f : A \to A_1 \times \cdots \times A_n$ dată prin $f(a) = (f_1(a), \ldots, f_n(a))$, pentru orice $a \in A$, este continuă.

Demonstraţie De la Teorema 5.2.1.3 şi relaţia $pr_i \circ f = f_i$, pentru orice i. □

In mod uzual, funcţia din Corolarul 5.2.1.2 se notează prin $(f_1, \ldots, f_n)$.

Corolarul 5.2.1.3. Fie $M = (A; \leq)$ şi $M_i = (A_i; \leq_i)$ mpo complete şi $f_i : A_i \to A_i'$ funcţii continue, unde $n \geq 2$ şi $1 \leq i \leq n$. Atunci, funcţia

$$f : A_1 \times \cdots \times A_n \to A_1' \times \cdots \times A_n'$$

dată prin $f(a_1, \ldots, a_n) = (f_1(a_1), \ldots, f_n(a_n))$, pentru orice $a_i \in A_i$, $1 \leq i \leq n$, este continuă.

Demonstraţie Putem scrie:

$$f(a_1, \ldots, a_n) = ((f_1 \circ pr_1)(a_1, \ldots, a_n), \ldots, (f_n \circ pr_n)(a_1, \ldots, a_n)),$$

pentru orice $(a_1, \ldots, a_n) \in A_1 \times \cdots \times A_n$.

Atunci, corolarul urmează de la Teorema 5.2.1.3 şi Corolarul 5.2.1.2. □

Corolarul 5.2.1.4. Fie $M = (A; \leq)$, $M_i = (A_i; \leq_i)$ şi $M' = (A'; \leq')$ mpo complete, unde $n \geq 2$ şi $1 \leq i \leq n$, iar $f_i : A \to A_i$ şi $f : A_1 \times \cdots \times A_n \to A'$ funcţii continue. Atunci, funcţia $f \circ (f_1, \ldots, f_n)$ este continuă.

Demonstraţie De la Corolarul 5.2.1.2 şi Teorema 5.2.1.3. □

Continuitatea funcţiilor de tipul $f : A_1 \times \cdots \times A_n \to A$. Studiul continuităţii acestui tip de funcţii face apel la conceptul de *funcţie Curry* [1].

Definiţia 5.2.1.2. Fie $f : A_1 \times \cdots \times A_n \to A$ o funcţie, unde $n \geq 2$. *Funcţia Curry asociată funcţiei* f este funcţia $f^c : A_1 \times \cdots \times A_{n-1} \to (A_n \to A)$ dată prin

$$f^c(a_1, \ldots, a_{n-1})(a_n) = f(a_1, \ldots, a_n),$$

pentru orice $(a_1, \ldots, a_{n-1}) \in A_1 \times \cdots \times A_{n-1}$ şi $a_n \in A_n$.

Vom conveni ca funcţiile $f^c(a_1, \ldots, a_{n-1}) : A_n \to A$ definite ca în Definiţia 5.2.1.2, pentru orice $(a_1, \ldots, a_{n-1}) \in A_1 \times \cdots \times A_{n-1}$, să fie numite tot *funcţii Curry asociate funcţiei* f.

Teorema 5.2.1.4. Fie $M = (A; \leq)$ şi $M_i = (A_i; \leq_i)$ mpo, unde $n \geq 2$ şi $1 \leq i \leq n$, şi $f : A_1 \times \cdots \times A_n \to A$ o funcţie. Atunci, f este monotonă dacă şi numai dacă funcţiile Curry asociate funcţiei f sunt monotone.

Demonstraţie Vom face demonstraţia pentru cazul $n = 2$, raţionamentul putând fi extins pentru $n \geq 2$ arbitrar.

Presupunem că f este funcţie monotonă. Fie $a_1 \in A_1$. Vom arăta că $f^c(a_1)$ este funcţie monotonă. Fie $a_2, a_2' \in A_2$ cu $a_2 \leq_2 a_2'$. Atunci,

$$\begin{array}{rcll} f^c(a_1)(a_2) & = & f(a_1, a_2) & \\ & \leq & f(a_1, a_2') & ((a_1, a_2) \leq' (a_1, a_2') \text{ şi } f \text{ monotonă}) \\ & = & f^c(a_1)(a_2'), & \end{array}$$

ceea ce ne arată că $f^c(a_1)$ este monotonă ($\leq'$ este ordinea parţială pe $A_1 \times A_2$).

Vom arăta acum că f^c este funcţie monotonă. Fie $a_1, a_1' \in A_1$ cu $a_1 \leq_1 a_1'$. Trebuie să arătăm că $f^c(a_1) \leq_{(A_2 \to A)} f^c(a_1')$. Fie $a_2 \in A_2$. Atunci,

$$\begin{array}{rcll} f^c(a_1)(a_2) & = & f(a_1, a_2) & \\ & \leq & f(a_1', a_2) & ((a_1, a_2) \leq' (a_1, a_2') \text{ şi } f \text{ monotonă}) \\ & = & f^c(a_1')(a_2), & \end{array}$$

ceea ce ne arată că f^c este monotonă.

Reciproc, presupunem că funcţiile Curry asociate funcţiei f sunt monotone. Fie $(a_1, a_2), (a_1', a_2') \in A_1 \times A_2$ cu $(a_1, a_2) \leq' (a_1', a_2')$. Atunci,

$$\begin{array}{rcll} f(a_1, a_2) & = & f^c(a_1)(a_2) & \\ & \leq & f^c(a_1)(a_2') & (f^c(a_1) \text{ monotonă}) \\ & \leq & f^c(a_1')(a_2') & (f^c \text{ monotonă}) \\ & = & f(a_1', a_2'), & \end{array}$$

ceea ce ne arată că f este monotonă. □

[1] Denumirea acestor funcţii provine de la numele logicianului american Haskell B. Curry (1900-1982). Aşa cum menţionează logicianul rus Moses Schönfinkel în [163], aceste tipuri de funcţii au fost utilizate de Gottlob Frege (1848–1925) cu mult înaintea lui Haskell Curry. Unii autori atribuie lui Schönfinkel utilizarea pentru prima dată a acestor tipuri de funcţii.

Teorema 5.2.1.5. Fie $M = (A; \leq)$ şi $(M_i = (A_i; \leq_i)$ mpo complete, unde $n \geq 2$ şi $1 \leq i \leq n$, şi $f : A_1 \times \cdots \times A_n \to A$ o funcţie. Atunci, f este continuă dacă şi numai dacă funcţiile Curry asociate funcţiei f sunt continue.

Demonstraţie Ca şi în cazul Teoremei 5.2.1.4, vom face demonstraţia doar pentru $n = 2$.

Presupunem că f este funcţie continuă. Atunci, f este monotonă şi, conform Teoremei 5.2.1.4, funcţiile Curry asociate sunt monotone.

Fie $a_1 \in A_1$ şi $L \subseteq A_2$ un lanţ nevid. Vom arăta că este satisfăcută relaţia

$$f^c(a_1)(sup(L)) = sup(f^c(a_1)(L)).$$

Are loc:

$$\begin{array}{rcll} f^c(a_1)(sup(L)) & = & f(a_1, sup(L)) & \\ & = & f(sup(\{a_1\}), sup(L)) & \\ & = & f(sup(\{a_1\} \times L)) & (\{a_1\} \times L \text{ lanţ}) \\ & = & sup(f(\{a_1\} \times L)) & (f \text{ continuă}) \\ & = & sup(f^c(a_1)(L)), & \end{array}$$

ceea ce arată că $f^c(a_1)$ este funcţie continuă.

Fie acum $L \subseteq A_1$ un lanţ nevid. Vom arăta că $f^c(sup(L)) = sup(f^c(L))$. Fie $a_2 \in A_2$. Atunci,

$$\begin{array}{rcll} f^c(sup(L))(a_2) & = & f(sup(L), a_2) & \\ & = & f(sup(L), sup(\{a_2\})) & \\ & = & f(sup(L \times \{a_2\})) & (L \times \{a_2\} \text{ lanţ}) \\ & = & sup(f(L \times \{a_2\})) & (f \text{ continuă}) \\ & = & sup(f^c(L)(a_2)) & \\ & = & sup(f^c(L))(a_2), & (\text{Lema } 5.1.2) \end{array}$$

ceea ce arată că f^c este funcţie continuă.

Reciproc, presupunem că funcţiile Curry asociate funcţiei f sunt continue. Deci, ele sunt şi monotone ceea ce conduce la faptul că f este monotonă.

Fie $L \subseteq A_1 \times A_2$ un lanţ nevid. Vom arăta că are loc $f(sup(L)) \leq' sup(f(L))$. Considerăm $L_1 = \{a|(\exists b)((a, b) \in L)\}$ şi $L_2 = \{b|(\exists a)((a, b) \in L)\}$. Este clar că L_1 şi L_2 sunt lanţuri nevide în M_1 şi, respectiv, M_2, şi $sup(L) = (sup(L_1), sup(L_2))$. Atunci,

$$\begin{array}{rcl} f(sup(L)) & = & f(sup(L_1), sup(L_2)) \\ & = & f^c(sup(L_1))(sup(L_2)) \\ & = & sup(f^c(sup(L_1))(L_2)) \\ & = & sup(\{f^c(sup(L_1))(a_2)|a_2 \in L_2\}) \\ & = & sup(\{sup(f^c(L_1)(a_2))|a_2 \in A_2\}) \\ & = & sup(\{sup(\{f^c(a_1)(a_2)|a_1 \in L_1\})|a_2 \in L_2\}) \\ & = & sup(\{sup(\{f(a_1, a_2)|a_1 \in L_1\})|a_2 \in L_2\}) \\ & = & sup(\{f(a_1, a_2)|a_1 \in L_1,\ a_2 \in L_2\}) \end{array}$$

(pentru ultima egalitate se verifică cu uşurinţă că sunt îndeplinite ipotezele Propoziţiei 3.2.1.1).

Vom arăta că $f(L)$ este cofinală în $f(L_1 \times L_2)$, ceea ce va conduce la faptul că $sup(f(L_1 \times L_2)) \leq sup(f(L))$ care va stabili continuitatea funcţiei f.

Fie $(a_1, a_2) \in L_1 \times L_2$. Atunci, există a_1' şi a_2' astfel încât $(a_1, a_2'), (a_1', a_2) \in L$. Fie $\leq'$ ordinea parţială pe $A_1 \times A_2$. Dacă $(a_1, a_2') \leq' (a_1', a_2)$ atunci $a_1 \leq_1 a_1'$, ceea ce conduce la $f(a_1, a_2) \leq f(a_1', a_2)$, iar dacă $(a_1', a_2) \leq' (a_1, a_2')$ atunci $a_2 \leq_2 a_2'$, ceea ce conduce la $f(a_1, a_2) \leq f(a_1, a_2')$.

Deci, $f(L)$ este cofinală în $f(L_1 \times L_2)$ şi demonstraţia este astfel încheiată. □

Prin $[M_1 \to M_2]$, sau $[A_1 \to A_2]$ atunci când M_1 şi M_2 sunt subînţelese din context, vom nota mulţimea tuturor funcţiilor continue de la mpo completă $M_1 = (A_1; \leq_1)$ la mpo completă $M_2 = (A_2; \leq_2)$.

Corolarul 5.2.1.5. Fie $M_1 = (A_1; \leq_1)$ şi $M_2 = (A_2; \leq_2)$ mpo complete. Atunci, funcţia $\psi : [A_1 \to A_2] \times A_1 \to A_2$ dată prin $\psi(f, a) = f(a)$, pentru orice $a \in A_1$ şi $f \in [A_1 \to A_2]$, este continuă.

Demonstraţie Funcţia ψ^c este funcţia identitate, iar pentru $f \in [A_1 \to A_2]$, $\psi^c(f)$ este funcţia f. Ca urmare, funcţiile Curry asociate funcţiei ψ sunt continue, ceea ce conduce la faptul că ψ este continuă. □

Teorema 5.2.1.6. Fie $M_1 = (A_1; \leq_1)$ şi $M_2 = (A_2; \leq_2)$ mpo complete. Atunci, $[A_1 \to A_2]$ este sub-mpo completă a mpo complete $(A_1 \to_m A_2)$.

Demonstraţie Cel mai mic element al mulţimii $(A_1 \to_m A_2)$ este şi cel mai mic element al mulţimii $[A_1 \to A_2]$.

Fie $L \subseteq [A_1 \to A_2]$ un lanţ nevid. Deoarece orice funcţie continuă este monotonă, lanţul L admite supremum în $(A_1 \to_m A_2)$. Vom arăta că acest supremum este funcţie continuă (el fiind funcţie monotonă).

Fie $K \subseteq A_1$ un lanţ nevid. Atunci,

$$\begin{aligned}
(sup_{(A_1 \to_m A_2)}(L))(sup(K)) &= sup(L(sup(K))) \\
&= sup(\{f(sup(K)) | f \in L\}) \\
&= sup(\{sup(f(K)) | f \in L\}) \\
&= sup(\{sup(\{f(a) | a \in K\}) | f \in L\}) \\
&= sup(\{f(a) | a \in K,\ f \in L\}) \\
&= sup((sup_{(A_1 \to_m A_2)}(L))(K))
\end{aligned}$$

ceea ce arată că $sup_{(A_1 \to_m A_2)}(L)$ este funcţie continuă (prima egalitate urmează de la Lema 5.1.2, a treia egalitate de la faptul că f este continuă, iar ultima egalitate de la Corolarul 3.2.3.1). □

Observaţia 5.2.1.4. Teorema 5.2.1.6 poate fi reformulată echivalent astfel:

> "Fie $M = (A, \leq)$ şi $M' = (A', \leq')$ două mpo. Dacă M şi M' sunt complete, atunci supremum oricărui lanţ nevid de funcţii continue de la A la A' este funcţie continuă.".

5.2.2 Puncte fixe şi inducţie de punct fix

Stabilirea existenţei punctelor fixe ale unei funcţii cât şi determinarea acestora este de importanţă uriaşă în matematică şi informatică. De exemplu, semantica denotaţională a structurilor repetitive se bazează pe determinarea celui mai mic punct fix a unei funcţii continue (detalii asupra aplicaţiilor teoriei punctelor fixe vor fi date în Secţiunea 5.3).

Definiţia 5.2.2.1. Fie A o mulţime nevidă şi $f : A \to A$ o funcţie. Se numeşte *punct fix* al funcţiei f orice element $a \in A$ cu proprietatea $f(a) = a$.

In această secţiune vom studia existenţa punctelor fixe pentru funcţii monotone şi continue definite pe mpo complete în unul din sensurile deja studiate. Inainte de aceasta facem observaţia că o funcţie poate să nu aibă nici un punct fix, poate avea un număr finit de puncte fixe sau chiar o infinitate de puncte fixe. In plus, dacă funcţia este definită pe o mpo, atunci putem discuta despre puncte fixe minimale sau cel mai mic punct fix al ei (atunci când acesta există).

Vom începe printr-un exemplu care să ne ajute la formarea unei imagini asupra modului de lucru cu puncte fixe.

Exemplul 5.2.2.1. [2] Fie A o mulţime şi $f : \mathcal{P}(A) \to \mathcal{P}(A)$ o funcţie monotonă. Să considerăm o submulţime $X \subseteq A$ cu proprietatea $X \subseteq f(X)$ (mulţimea vidă satisface această proprietate). Dacă aplicăm f acestei submulţimi obţinem $f(X)$ ce este inclusă în $f^2(X)$ în baza monotoniei funcţiei f. Repetând procedeul obţinem

$$f^2(X) \subseteq f^3(X).$$

Intuitiv, continuând acest procedeu oricât de mult, mulţimea limită Y care s-ar obţine nu s-ar mai modifica prin aplicarea funcţiei f, adică ea ar satisface $f(Y) = Y$. Această mulţime limită nu este alta decât

$$Y = sup(\{X, f(X), f^2(X), \ldots\})$$

(şi ea există deoarece este limita unui lanţ într-o mpo completă, în acest caz limita fiind reuniunea lanţului). Este uşor de văzut că, în adevăr, $f(Y) = Y$ deoarece:

$$\begin{array}{rcl} Y & = & sup(\{X, f(X), f^2(X), \ldots\}) \\ & = & sup(\{f(X), f^2(X), f^3(X), \ldots\}) \\ & = & \bigcup_{i\geq 1} f^i(X) \\ & = & f(\bigcup_{i\geq 0} f^i(X)) \\ & = & f(sup(\{X, f(X), f^2(X), \ldots\})) \\ & = & f(Y). \end{array}$$

Ca urmare, pornind de la o submulţime X a mulţimii A ce satisface $X \subseteq f(X)$ am putut pune în evidenţă atât un punct fix cât şi modul de determinare al acestuia.

[2] Rezultatul ce face subiectul acestui exemplu a fost descoperit de Knaster şi Tarski în 1927 (conform celor menţionate de Tarski în [179]). In [91] sunt prezentate un număr de aplicaţii ale acestui rezultat.

Dar dacă dorim să determinăm cel mai mic punct fix al funcţiei f? Intuitiv, pentru determinarea acestuia, ar trebui să pornim cu $\emptyset$.

In adevăr, $Y_0 = sup(\{\emptyset, f(\emptyset), f^2(\emptyset), \ldots\})$ este punct fix al funcţiei f şi, dacă Z este un alt punct fix al acesteia atunci relaţiile

- $\emptyset \subseteq Z$;
- $f(\emptyset) \subseteq f(Z) = Z$, pe baza monotoniei funcţiei f şi a faptului că $f(Z) = Z$;
- $f(f(\emptyset)) \subseteq f(Z) = Z$ etc.

conduc la

$$Y_0 = sup(\{\emptyset, f(\emptyset), f^2(\emptyset), \ldots\}) \subseteq Z.$$

Deci, Y_0 este cel mai mic punct fix al funcţiei f.

Exemplul 5.2.2.2. Să analizăm acum ce elemente de bază s-au folosit în obţinerea rezultatelor din exemplul anterior:

- în primul rând, s-a folosit faptul că există supremum lanţurilor. De fapt, mulţimea parţial ordonată $(\mathcal{P}(A); \subseteq)$ este completă;
- în al doilea rând, s-a utilizat faptul că $sup(L) = sup(L - \{X\})$, unde X este primul element al lanţului L. Această proprietate este însă satisfăcută în orice mpo de orice lanţ ce are cel puţin 2 elemente (eliminarea celui mai mic element al lanţului, atunci când există, nu modifică supremum lanţului);
- cea de a treia proprietate utilizată este $sup(f(L)) = f(sup(L))$, pentru orice lanţ L. Această proprietate nu ne spune altceva decât că f este, de fapt, o funcţie continuă;
- ca o ultimă proprietate, în determinarea celui mai mic punct fix s-a pornit de la $\emptyset$ care este cel mai mic element al mpo $(\mathcal{P}(A); \subseteq)$. Existenţa celui mai mic element este însă garantată în orice mpo completă.

Ca urmare, rezultatul din Exemplul 5.2.2.1 poate fi generalizat la a arăta că orice funcţie continuă definită pe o mpo completă are un cel mai mic punct fix. Să intrăm puţin în detalii.

Fie $M = (A; \leq)$ o mpo completă şi $f : A \to A$ o funcţie continuă. Prin inducţie după $i \geq 0$ se arată cu uşurinţă că are loc

$$f^i(\perp_A) \leq f^{i+1}(\perp_A),$$

pentru orice $i \geq 0$, ceea ce conduce la faptul că $L = \{f^i(\perp_A) | i \geq 0\}$ este lanţ în M. Completitudinea mpo M asigură existenţa supremului acestui lanţ. Arătăm că $sup(L)$ este punct fix pentru f. Are loc:

$$f(sup(L)) = sup(f(L)) = sup(\{f^i(\perp_A) | i \geq 1\}) = sup(L).$$

Deci, $sup(L)$ este punct fix pentru f.

Dacă $a \in A$ este punct fix pentru f, atunci prin inducţie după $i \geq 0$ se obţine $f^i(\perp_A) \leq a$ pentru orice $i \geq 0$, ceea ce arată că $sup(L) \leq a$. Deci, $sup(L)$ este cel mai mic punct fix al funcţiei f.

Fie $M = (A; \leq)$ o mpo completă. Notăm prin μ_M, sau μ dacă M se subînţelege din context, funcţia $\mu : [A \to A] \to A$ dată prin

$$\mu(f) = \text{cel mai mic punct fix al funcţiei } f,$$

pentru orice funcţie continuă $f : A \to A$. Funcţia μ_M se mai numeşte şi *funcţia de punct fix* asociată mpo M.

Teorema 5.2.2.1. Funcţia de punct fix asociată unei mpo complete este continuă.

Demonstraţie Fie $M = (A; \leq)$ o mpo completă. Vom arăta că μ este supremum unui lanţ de funcţii continue, ceea ce va conduce la faptul că μ este continuă (a se vedea Observaţia 5.2.1.4).

Fie $F_i : [A \to A] \to A$ funcţia dată prin $F_i(f) = f^i(\bot_A)$, pentru orice funcţie continuă $f : A \to A$ şi $i \geq 0$. Arătăm prin inducţie după $i \geq 0$ că F_i este funcţie continuă, pentru orice $i \geq 0$.

Funcţia F_0 este funcţia constantă $\bot_A$ şi, deci, este continuă. Dacă presupunem că F_i este continuă, unde $i \geq 0$, atunci

$$F_{i+1}(f) = f^{i+1}(\bot_A) = f(F_i(f)) = \psi(id(f), F_i(f)) = (\psi \circ (id, F_i))(f),$$

pentru orice $f \in [A \to A]$, ceea ce ne arată că $F_{i+1} = \psi \circ (id, F_i)$, unde id este funcţia identitate. Deci, F_{i+1} este continuă fiind compunere de funcţii continue.

Mulţimea $L = \{F_i | i \geq 0\}$ este lanţ de funcţii. In adevăr, pentru orice $0 \leq i \leq j$ are loc

$$F_i(f) = f^i(\bot_A) \leq f^j(\bot_A) = F_j(f),$$

pentru orice $f \in [A \to A]$.

Supremum lanţului L este dat prin

$$\begin{array}{rcll} (sup(L))(f) & = & sup(L(f)) & \text{(Lema 5.1.2)} \\ & = & sup(\{F_i(f) | i \geq 0\}) & \\ & = & sup(\{f^i(\bot_A) | i \geq 0\}) & \\ & = & \mu(f), & \end{array}$$

pentru orice $f \in [A \to A]$, ceea ce arată că $sup(L) = \mu$. Deci, μ este funcţie continuă. □

Dacă o funcţie continuă are o anumită proprietate, putem concluziona că cel mai mic punct fix al ei are respectiva proprietate ? Următoarea teoremă, datorată lui Park, ne furnizează un exemplu de proprietate ce poate fi transferată de la o funcţie continuă la cel mai mic punct fix al ei.

Teorema 5.2.2.2. (Teorema lui Park)
Fie $M = (A; \leq)$ o mpo completă şi $f : A \to A$ o funcţie continuă. Dacă există $x \in A$ astfel încât $f(x) \leq x$, atunci $\mu(f) \leq x$.

Demonstraţie Fie $x \in A$ astfel încât $f(x) \leq x$. Monotonia funcţiei f conduce la

$$f(\bot_A) \leq f(x) \leq x.$$

Inductiv, obţinem $f^i(\bot_A) \leq x$, pentru orice $i \geq 0$. Ca urmare,

$$sup(\{f^i(x) | i \geq 0\}) \leq x,$$

ceea ce ne arată că $\mu(f) \leq x$. □

Definiţia 5.2.2.2. Fie $M = (A; \leq)$ o mpo completă şi P un predicat pe A. Spunem că P este *admisibil* dacă are loc

$$(\forall L \subseteq A \text{ lanţ nevid})((\forall a \in L)(P(a)) \;\Rightarrow\; P(sup(L))).$$

Următoarea teoremă, datorată lui Dana Scott [62] şi numită *Principiul inducţiei de punct fix*, este o simplă combinaţie dintre inducţia matematică şi conceptul de predicat admisibil.

Teorema 5.2.2.3. (Principiul inducţiei de punct fix)
Fie $M = (A; \leq)$ o mpo completă şi $f : A \to A$ o funcţie continuă. Dacă P este predicat pe A astfel încât:

(1) P este admisibil;

(2) $P(\bot_A)$;

(3) $(\forall i \geq 0)(P(f^i(\bot_A)) \;\Rightarrow\; P(f^{i+1}(\bot_A)))$,

atunci $P(\mu(f))$.

Demonstraţie In baza proprietăţilor de la (2) şi (3), prin inducţie matematică, obţinem că are loc $P(f^i(\bot_A))$, pentru orice $i \geq 0$. Cum P este admisibil, deducem că are loc $P(sup(\{f^i(\bot_A) | i \geq 0\}))$, adică $P(\mu(f))$. □

Evident că suntem interesaţi de existenţa predicatelor admisibile. Determinarea unor astfel de predicate nu este un lucru simplu. Vom prezenta mai jos câteva construcţii ce conduc la astfel de predicate. Ele sunt bazate pe disjuncţie, conjuncţie şi inegalitate (egalitate) de funcţii continue. Vom începe întâi cu o observaţie importantă.

Observaţia 5.2.2.1.

(1) Nu orice predicat este admisibil. Fie, de exemplu, $P : (\mathbf{N}_\bot \to \mathbf{N}_\bot) \to \{0, 1\}$ dat prin

$$P(x) = \begin{cases} 0, & (\forall n \in \mathbf{N})(f(n) \neq \bot) \\ 1, & \text{altfel.} \end{cases}$$

şi lanţul L din tabelul de mai jos

L	$\bot$	0	1	2	3	$\cdots$
f_0	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\cdots$
f_1	$\bot$	0	$\bot$	$\bot$	$\bot$	$\cdots$
f_2	$\bot$	0	0	$\bot$	$\bot$	$\cdots$
f_3	$\bot$	0	0	0	$\bot$	$\cdots$
$\cdots$	$\cdots$					$\cdots$
$sup(L)$	$\bot$	0	0	0		$\cdots$

$P(f_i) = 1$ pentru orice $i \geq 0$, dar $P(sup(L)) = 0$. Deci, P nu este admisibil.

(2) Dacă P este admisibil atunci nu rezultă, în general, că $\neg P$ este admisibil. De exemplu, dacă considerăm predicatul Q obţinut din predicatul P de la (1) prin înlocuirea lui 0 cu 1 şi a lui 1 cu 0, atunci constatăm că Q este admisibil (supremum unui lanţ de funcţii totale pe $\mathbf{N}$ este o funcţie totală, conform definiţiei ordinii parţiale pe funcţii şi a ordinii parţiale pe $\mathbf{N}_\bot$). Insă, $\neg Q = P$ care nu este admisibil.

Teorema 5.2.2.4. Fie $M = (A; \leq)$ o mpo completă şi P, Q predicate pe A. Dacă P şi Q sunt admisibile, atunci $P \vee Q$ şi $P \wedge Q$ sunt admisibile.

Demonstraţie Admisibilitatea predicatului $P \wedge Q$ decurge imediat de la admisibilitatea predicatelor P şi Q. Dacă elementele unui lanţ nevid satisfac $P \wedge Q$, atunci ele satisfac atât P cât şi Q. Atunci, P şi Q sunt satisfăcute şi de supremum lanţului L, ceea ce ne arată că $P \wedge Q$ este admisibil.

Fie $L \subseteq A$ un lanţ nevid astfel încât $(P \vee Q)(a)$, pentru orice $a \in L$. Considerăm laţurile $L_1 = \{a \in L | P(a)\}$ şi $L_2 = \{a \in L | Q(a)\}$. Dacă unul din ele este vid, atunci celălalt este chiar L, iar admisibilitatea predicatelor P şi Q conduce la $(P \vee Q)(sup(L))$. Dacă ambele sunt nevide, atunci unul din ele este cofinal în celălalt (ceea ce poate fi arătat uşor prin contradicţie), ceea ce asigură că supremum lanţului L este dat de supremum acestuia. Atunci, admisibilitatea predicatelor P şi Q conduce la $(P \vee Q)(sup(L))$. Deci, $P \vee Q$ este predicat admisibil. □

Teorema 5.2.2.5. Fie $M = (A; \leq)$ şi $M' = (A'; \leq')$ mpo complete şi $f_i, g_i : A \to A'$ funcţii continue, unde $n \geq 1$ şi $1 \leq i \leq n$. Atunci, predicatul $P : A \to \{0, 1\}$ dat prin

$$P(a) \quad \Leftrightarrow \quad (\forall i)(f_i(a) \leq' g_i(a)),$$

pentru orice $a \in A$, este admisibil.

Demonstraţie Fie $L \subseteq A$ un lanţ nevid. Presupunem că are loc $P(a)$, pentru orice $a \in L$. Adică, $f_i(a) \leq' g_i(a)$, pentru orice $1 \leq i \leq n$.

Vom arăta că $f_i(sup(L)) \leq' g_i(sup(L))$, pentru orice $1 \leq i \leq n$, ceea ce va conduce la $P(sup(L))$, adică P este admisibil. Cum funcţiile f_i şi g_i sunt continue, relaţia de mai sus se reduce la a arăta că are loc

$$sup(f_i(L)) \leq' sup(g_i(L)),$$

pentru orice $1 \leq i \leq n$. Aceasta însă urmează de la ipotezele teoremei care ne spun că $g_i(L)$ este cofinală în $f_i(L)$, pentru orice i.

Deci, P este admisibil. □

Corolarul 5.2.2.1. Fie $M = (A; \leq)$ şi $M' = (A'; \leq')$ mpo complete şi $f, g : A \to A$ funcţii continue. Atunci, predicatul P dat prin

$$P(a) \quad \Leftrightarrow \quad f(a) = g(a),$$

pentru orice $a \in A$, este admisibil.

Demonstraţie Considerăm predicatele P_1 şi P_2 date prin

$$P_1(a) \quad \Leftrightarrow \quad f(a) \leq' g(a)$$

şi

$$P_2(a) \quad \Leftrightarrow \quad g(a) \leq' f(a),$$

pentru orice $a \in A$. Conform Teoremei 5.2.2.5, P_1 şi P_2 sunt admisibile. Atunci, Teorema 5.2.2.4 conduce la faptul că $P_1 \wedge P_2$ este admisibil. Insă, $P_1 \wedge P_2 = P$. □

5.3 Aplicaţii: semantica denotaţională a programelor

Vom arăta în această secţiune cum putem utiliza aparatul mulţimilor parţial ordonate complete şi al funcţiilor continue pentru a descrie semantica limbajelor de programelor. Vom considera două clase de programe, *programe recursive* şi *programe while*, şi vom urma, în principal, ideile din [115, 174, 112]. Indicăm însă şi [140] pentru mai multe detalii asupra semanticii limbajelor de programare.

5.3.1 λ-notaţie

La începutul anilor 1930 multe din cercetările matematice canalizate pe introducerea unui concept de funcţie "efectiv calculabilă" au inceput să se finalizeze prin efortul conjugat al mai multor matematicieni de renume precum Church, Kleene, Turing, Gödel şi alţii. Rezultatul este ceea ce numim azi *funcţie recursivă*. Un prim pas în definirea acestora a fost făcut de Church prin introducerea λ-notaţiei şi a conceptului de funcţie λ-definibilă [27] (a se vedea şi [28]). Ulterior, λ-notaţia s-a dovedit o achiziţie inestimabilă în studiul limbajelor de programare şi a semanticii acestora, şi în special în cadrul limbajelor de programare funcţională.

Descrierea limbajului λ-notaţiei parcurge două mari etape: sintaxa şi semantica.

Sintaxa λ-notaţiei se bazează pe utilizarea simbolurilor auxiliare "{", "}", "(", ")", "[", "]", ",", "." şi "λ", şi pe conceptele de *tip*, *bază* şi *λ-term* pe care le descriem după cum urmează:

- **Tip.** Fie $\mathcal{T}_0$ o mulţime ale cărei elemente le numim *tipuri de bază*. *Tipurile peste* $\mathcal{T}_0$ se definesc inductiv prin:

- orice tip de bază este tip;
- dacă $\tau_1, \ldots, \tau_n, \tau$ sunt tipuri, atunci $(\tau_1, \ldots, \tau_n \to \tau)$ este tip.

Notăm mulţimea astfel definită prin $\mathcal{T}$.

- **Bază.** O *bază pentru λ-notaţie* este un sistem $\mathcal{B} = (\mathcal{T}_0, \mathcal{V}, \mathcal{F})$, unde:
  - $\mathcal{T}_0$ este o mulţime de tipuri de bază;
  - $\mathcal{V}$ este o mulţime de *variabile cu tip* (fiecare din ele având asociat un unic tip peste $\mathcal{T}_0$);
  - $\mathcal{F}$ este o mulţime de *simboluri funcţionale cu tip* (fiecare din ele având asociat un unic tip peste $\mathcal{T}_0$).

  In plus, vom presupunem următoarele:
  - mulţimile $\mathcal{V}$, $\mathcal{F}$ şi $\mathcal{T}$ sunt disjuncte două câte două;
  - pentru orice tip există oricâte variabile este nevoie.

  Simbolurile funcţionale al căror tip va fi din $\mathcal{T}_0$ vor fi numite şi *simboluri constante*. Variabilele vor fi notate prin $x, y, z, \ldots, p, q, r, \ldots, F, G, H, \ldots$, iar simbolurile funcţionale prin $f, g, h, \ldots$.

- **λ-term.** Fie $\mathcal{B}$ o bază pentru λ-notaţie. *λ-termii peste $\mathcal{B}$* se definesc prin inducţie simultană astfel:
  - dacă t este variabilă sau simbol funcţional de tip τ, atunci t este *λ-term de tip τ*;
  - (*aplicaţie*) dacă u este λ-term de tip $(\tau_1, \ldots, \tau_n \to \sigma)$, iar t_i sunt λ-termi de tip τ_i, $1 \leq i \leq n$, atunci $u(t_1, \ldots, t_n)$ este *λ-term de tip σ*;
  - (*abstracţie*) dacă u este λ-term de tip σ, iar x_i sunt variabile de tip τ_i, $1 \leq i \leq n$, atunci $[\lambda x_1, \ldots, x_n.u]$ este *λ-term de tip* $(\tau_1, \ldots, \tau_n \to \sigma)$.

Exemplul 5.3.1.1. Fie $\mathcal{T}_0 = \{nat, bool\}$. Atunci,

$$nat,\ bool,\ (nat, nat \to nat) \text{ şi } (nat, nat \to bool)$$

sunt tipuri peste $\mathcal{T}_0$. Presupunem că x şi y sunt variabile de tip nat, b este variabilă de tip $bool$, 2 şi 3 sunt constante de tip nat, iar $+$ şi $\cdot$ sunt simboluri funcţionale de tip $(nat, nat \to nat)$. Atunci,

$$\cdot(2, x),\ \cdot(3, y) \text{ şi } +(\cdot(2, x), \cdot(3, y))$$

sunt λ-termi de tip nat,

$$[\lambda x. \cdot (2, x)] \text{ şi } [\lambda y. \cdot (3, y)]$$

sunt λ-termi de tip $(nat \to nat)$, iar

$$[\lambda x, y. +(\cdot(2, x), \cdot(3, y))]$$

este λ-term de tip $(nat, nat \to nat)$.

Dată o funcţie $f : A_1 \to A_2$, $x \in A_1$ şi $d \in A_2$, vom nota prin $f[x/d]$ funcţia definită prin:

$$f[x/d](y) = \begin{cases} f(y), & \text{dacă } y \neq x \\ d, & \text{altfel,} \end{cases}$$

pentru orice $y \in A_1$.

Această notaţie poate fi extinsă în mod natural la $f[x_1/d_1] \cdots [x_n/d_n]$.

Lema 5.3.1.1. Fie A_1 o mulţime, $x_1, \ldots, x_n \in A_1$ unde $n \geq 1$, şi $(A_2; \leq_2)$ o mpo completă. Atunci, funcţia $\psi_{x_1,\ldots,x_n} : (A_1 \to A_2) \times A_2^n \to (A_1 \to A_2)$ dată prin

$$\psi_{x_1,\ldots,x_n}(f, (d_1, \ldots, d_n)) = f[x_1/d_1] \cdots [x_n/d_n],$$

pentru orice $f \in (A_1 \to A_2)$ şi $(d_1, \ldots, d_n) \in A_2^n$, este continuă.

Demonstraţie Vom demonstra lema pentru $n = 1$ (aceeaşi demonstraţie poate fi generalizată pentru $n > 1$).

Fie $L \subseteq (A_1 \to A_2) \times A_2$ un lanţ nevid. Supremum acestui lanţ există deoarece A_2 şi $(A_1 \to A_2)$ sunt mpo complete. Mai mult, dacă notăm

$$L_1 = \{f | (\exists d)((f, d) \in L)\}$$

şi

$$L_2 = \{d | (\exists f)((f, d) \in L)\},$$

atunci $sup(L) = (sup(L_1), sup(L_2))$.

Are loc

$$sup(\psi_{x_1}(L)) = sup(\{f[x_1/d_1] | (f, d_1) \in L\}),$$

iar în baza Lemei 5.1.2 obţinem

$$sup(\psi_{x_1}(L))(y) = \begin{cases} sup(L_1(y)), & \text{dacă } y \neq x \\ sup(L_2), & \text{altfel,} \end{cases}$$

pentru orice $y \in A_1$.

Pe de altă parte,

$$\psi_{x_1}(sup(L))(y) = sup(L_1)[x_1/sup(L_2)](y) = \begin{cases} sup(L_1)(y), & \text{dacă } y \neq x \\ sup(L_2), & \text{altfel,} \end{cases}$$

pentru orice $y \in A_1$.

In baza Lemei 5.1.2 obţinem $sup(\psi_{x_1}(L)) = \psi_{x_1}(sup(L))$, ceea ce stabileşte continuitatea funcţiei ψ_{x_1}. □

Semantica λ-notaţiei se bazează pe conceptele de *interpretare* a (elementelor) unei baze, *atribuire* şi *funcţie semantică*:

- **Interpretare a unei baze.** O *interpretare* a unei bazei $\mathcal{B}$ este un cuplu

$$\mathcal{I} = (((D_\tau; \leq_\tau) | \tau \in \mathcal{T}_0), \mathcal{I}_0),$$

unde:

– pentru orice $\tau \in \mathcal{T}_0$, $(D_\tau; \leq_\tau)$ este mpo completă numită *domeniul tipului* τ.

Pentru tipurile $\tau = (\tau_1, \ldots, \tau_n \to \sigma)$ ce nu sunt de bază domeniile se definesc inductiv prin

$$[D_{\tau_1} \times \cdots \times D_{\tau_n} \to D_\sigma];$$

– $\mathcal{I}_0 : \mathcal{F} \to \bigcup_{\tau \in \mathcal{T}} D_\tau$ este o funcţie de *interpretare iniţială* cu proprietatea că pentru orice $f \in \mathcal{F}$, dacă f este de tip τ atunci $\mathcal{I}_0(f) \in D_\tau$.

- **Atribuire.** Fie $\mathcal{B}$ o bază pentru o λ-notaţie şi $\mathcal{I}$ o interpretare a ei. O *atribuire* sau *asignare* pentru baza $\mathcal{B}$ sub interpretarea $\mathcal{I}$ este orice funcţie

$$\gamma : \mathcal{V} \to \bigcup_{\tau \in \mathcal{T}} D_\tau$$

astfel încât, pentru orice $x \in \mathcal{V}$, dacă x are tipul τ atunci $\gamma(x) \in D_\tau$.

Vom nota prin $\Gamma_{\mathcal{B},\mathcal{I}}$ mulţimea tuturor atribuirilor pentru baza $\mathcal{B}$ sub interpretarea $\mathcal{I}$. Atunci când $\mathcal{B}$ şi $\mathcal{I}$ sunt clare din context, notaţia $\Gamma_{\mathcal{B},\mathcal{I}}$ va fi simplificată la Γ.

- **Funcţia semantică a λ-termilor**. Fie $\mathcal{B}$ o bază, $\mathcal{I}$ o interpretare a bazei $\mathcal{B}$, şi t un λ-term de tip τ. Dacă t nu conţine variabile atunci, intuitiv, interpretând fiecare element din t obţinem un element din D_τ. Dacă însă t conţine variabile atunci, pentru fiecare atribuire a variabilelor obţinem o interpretare a λ-termului t. Deci, în acest caz, interpretarea lui t trebuie să fie o funcţie ce depinde de atribuiri. Ca urmare, *funcţia semantică a lui* t se defineşte ca fiind funcţia $\mathcal{I}(t) : \Gamma \to D_\tau$ dată prin:

  – dacă $t = x \in \mathcal{V}$, atunci $\mathcal{I}(t)(\gamma) = \gamma(x)$, pentru orice $\gamma \in \Gamma$;

  – dacă $t = f \in \mathcal{F}$, atunci $\mathcal{I}(t)(\gamma) = \mathcal{I}_0(f)$, pentru orice $\gamma \in \Gamma$;

  – dacă $t = u(t_1 \ldots, t_n)$, unde u este de tip $(\tau_1, \ldots, \tau_n \to \sigma)$ iar t_i sunt de tip τ_i, $1 \leq i \leq n$, atunci

$$\mathcal{I}(t)(\gamma) = \mathcal{I}(u)(\gamma)(\mathcal{I}(t_1)(\gamma), \ldots \mathcal{I}(t_n)(\gamma)),$$

  pentru orice $\gamma \in \Gamma$;

  – dacă $t = [\lambda x_1, \ldots, x_n.u]$ este de tip $\tau = (\tau_1, \ldots, \tau_n \to \sigma)$, unde u este de tip σ iar x_i sunt de tip τ_i, $1 \leq i \leq n$, atunci

$$\mathcal{I}(t)(\gamma) : D_{\tau_1} \times \cdots \times D_{\tau_n} \to D_\sigma$$

$$\mathcal{I}(t)(\gamma)(d_1, \ldots, d_n) = \mathcal{I}(u)(\gamma[x_1/d_1] \cdots [x_n/d_n]),$$

  pentru orice $\gamma \in \Gamma$ şi $(d_1, \ldots, d_n) \in D_{\tau_1} \times \cdots \times D_{\tau_n}$ (intuitiv, în acest caz, t denotă o funcţie de $x_1, \ldots, x_n$. Atunci, interpretarea lui va depinde de atribuirea doar a variabilelor diferite de $x_1, \ldots, x_n$).

In continuare vom stabili câteva proprietăţi foarte importante ale mulţimii Γ şi funcţiei semantice a λ-termilor.

Propoziţia 5.3.1.1. Fie $\mathcal{B}$ o bază şi $\mathcal{I}$ o interpretare a ei. Atunci, $\Gamma_{\mathcal{B},\mathcal{I}}$ cu ordinea parţială pe funcţii este mpo completă.

Demonstraţie Fie familia de mpo $((D_x; \leq_x)|x \in \mathcal{V})$, unde, pentru orice $x \in \mathcal{V}$, dacă tipul variabilei x este τ atunci $(D_x; \leq_x) = (D_\tau; \leq_\tau)$. Atunci, are loc

$$\Gamma_{\mathcal{B},\mathcal{I}} = \{\gamma : \mathcal{V} \to \bigcup_{x\in\mathcal{V}} D_x | (\forall x \in \mathcal{V})(\gamma(x) \in D_x)\} = \prod_{x\in\mathcal{V}} D_x,$$

iar ordinea parţială pe $\Gamma_{\mathcal{B},\mathcal{I}}$ este întocmai ordinea parţială pe $\prod_{x\in\mathcal{V}} D_x$.

Ca urmare, mpo indusă de $\Gamma_{\mathcal{B},\mathcal{I}}$ este exact produsul direct al familiei de mpo complete $((D_x; \leq_x)|x \in \mathcal{V})$ şi, deci, este mpo completă. □

Teorema 5.3.1.1. (Continuitatea funcţiei semantice a λ-termilor)
Fie $\mathcal{B}$ o bază şi $\mathcal{I}$ o interpretare a ei. Atunci, pentru orice tip τ şi λ-term t de tip τ are loc:

(1) $\mathcal{I}(t)(\gamma) \in D_\tau$, pentru orice $\gamma \in \Gamma$;

(2) $\mathcal{I}(t) : \Gamma \to D_\tau$ este funcţie continuă.

Demonstraţie Vom demonstra teorema prin inducţie structurală asupra λ-termului t.

Cazul 1: $t = x \in \mathcal{V}$. Atunci, $\mathcal{I}(t)(\gamma) = \gamma(x) \in D_\tau$, pentru orice atribuire γ. Deci, are loc (1). Pentru a demonstra (2) considerăm un lanţ nevid $L \subseteq \Gamma$ şi observăm că $\mathcal{I}(t)(L) = L(x)$ este lanţ nevid în D_τ. Cum $(D_\tau; \leq_\tau)$ este mpo completă, există $sup(L(x))$ şi, conform Lemei 5.1.2, are loc

$$sup(L(x)) = (sup(L))(x).$$

Deci, există $sup(\mathcal{I}(t)(L))$ şi

$$sup(\mathcal{I}(t)(L)) = sup(L(x)) = (sup(L))(x) = \mathcal{I}(t)(sup(L)),$$

ceea ce ne arată că $\mathcal{I}(t)$ este funcţie continuă.

Cazul 2: $t = f \in \mathcal{F}$. Atunci, $\mathcal{I}(t)(\gamma) = \mathcal{I}_0(f) \in D_\tau$, pentru orice atribuire γ. Continuitatea funcţiei $\mathcal{I}(t)$ urmează imediat de la faptul că aceasta este funcţie constantă.

Cazul 3: $t = u(t_1, \ldots, t_n)$, u este de tip $(\tau_1, \ldots, \tau_n \to \tau)$ şi t_i este de tip τ_i pentru orice $1 \leq i \leq n$. Presupunem că λ-termii $u, t_1, \ldots, t_n$ satisfac (1) şi (2) din teoremă. Atunci, pentru orice atribuire γ,

$$\mathcal{I}(t)(\gamma) = \mathcal{I}(u)(\gamma)(\mathcal{I}(t_1)(\gamma), \ldots, \mathcal{I}(t_n)(\gamma)),$$

care este un element din D_τ conform definiţiei funcţiei de interpretare. De asemenea, putem scrie

$$\mathcal{I}(t) = \psi \circ (\mathcal{I}(u), (\mathcal{I}(t_1), \ldots, \mathcal{I}(t_n))),$$

ceea ce ne arată că $\mathcal{I}(t)$ este funcţie continuă fiind compunere de funcţii continue (conform ipotezei, Corolarului 5.2.1.2 şi Corolarului 5.2.1.5).

Cazul 4: $t = [\lambda x_1, \ldots, x_n.u]$, u este de tip σ, x_i este de tip τ_i pentru orice $1 \leq i \leq n$, şi $\tau = (\tau_1, \ldots, \tau_n \to \sigma)$. Presupunem că λ-termul u satisface (1) şi (2) din teoremă. Fie γ o atribuire. Pentru a arăta că are loc $\mathcal{I}(t)(\gamma) \in D_\tau$ avem de arătat că $\mathcal{I}(t)(\gamma)$ este funcţie continuă de la $D_{\tau_1} \times \cdots \times D_{\tau_n}$ la D_σ. Ca urmare, avem de arătat că $\mathcal{I}(t)$ şi $\mathcal{I}(t)(\gamma)$ sunt funcţii continue. Forma acestor funcţii sugerează considerarea unei funcţii continue g ale cărei funcţii Curry asociate să fie exact aceste funcţii. Ca urmare, considerăm funcţia

$$g : \Gamma \times (D_{\tau_1} \times \cdots \times D_{\tau_n}) \to D_\sigma$$

dată prin

$$g(\gamma, (d_1, \ldots, d_n)) = \mathcal{I}(u)(\gamma[x_1/d_1] \cdots [x_n/d_n]),$$

pentru orice γ şi $d_i \in D_{\tau_i}$, $1 \leq i \leq n$.

Funcţia g este compunere de funcţii continue (conform ipotezei şi Lemei 5.3.1.1) deoarece ea poate fi scrisă în forma

$$g = \mathcal{I}(u) \circ \psi_{x_1,\ldots,x_n}.$$

Deci, g este continuă. Atunci, funcţiile Curry asociate funcţiei g sunt continue. Adică, funcţiile $\mathcal{I}(t)$ şi $\mathcal{I}(t)(\gamma)$, pentru orice atribuire γ, sunt funcţii continue. □

Definiţia 5.3.1.1. Fie $\mathcal{B}$ o bază şi $\mathcal{I}$ o interpretare a ei. Spunem că o variabilă x *apare liber* în λ-termul t dacă:

- $t = x \in \mathcal{V}$, sau
- $t = u(t_1, \ldots, t_n)$ şi x apare liber în unul din termii $u, t_1, \ldots, t_n$, sau
- $t = [\lambda x_1, \ldots, x_n.u]$, x este diferită de $x_1, \ldots, x_n$ şi apare liber în u.

Dacă x nu apare liber în t, atunci spunem că x este *mărginită* în t.

Observaţia 5.3.1.1. Atragem explicit atenţia asupra faptului că într-un λ-term de forma $t = u(t_1, \ldots, t_n)$, o variabilă x poate apare liber într-un λ-term t_i şi mărginită în u. In t, x va fi liberă. Un astfel de caz este următorul:

$$t = [\lambda x.f(3, x)](f(2, x)).$$

Dacă considerăm tipul nat cu interpretarea $D_{nat} = \mathbf{N}_\perp$, $\mathcal{I}_0(3), \mathcal{I}_0(2) \in \mathbf{N}$, x de tip nat şi $\mathcal{I}_0(f) : \mathbf{N}_\perp^2 \to \mathbf{N}_\perp$, atunci

$$\begin{aligned}
\mathcal{I}(t)(\gamma) &= \mathcal{I}([\lambda x.f(3, x)])(\gamma)(\mathcal{I}(f(2, x))(\gamma)) \\
&= \mathcal{I}(f(3, x))(\gamma[x/\mathcal{I}(f(2, x))(\gamma)]) \\
&= \mathcal{I}_0(f)(\mathcal{I}_0(3), \mathcal{I}(x))(\gamma[x/\mathcal{I}_0(f)(\mathcal{I}_0(2), \gamma(x))]) \\
&= \mathcal{I}_0(f)(\mathcal{I}_0(3), \mathcal{I}_0(f)(\mathcal{I}_0(2), \gamma(x))),
\end{aligned}$$

pentru orice atribuire γ.

In cazul în care $\mathcal{I}_0(2)$ ($\mathcal{I}_0(3)$) este numărul natural 2 (3), iar $\mathcal{I}_0(f)$ este adunarea numerelor naturale (extinsă natural), obţinem $\mathcal{I}(t)(\gamma) = 3 + (2 + \gamma(x))$.

Teorema 5.3.1.2. (Teorema de coincidenţă)
Fie $\mathcal{B}$ o bază, $\mathcal{I}$ o interpretare a ei şi t un λ-term. Atunci, pentru orice două atribuiri γ şi γ' ce satisfac $\gamma(x) = \gamma'(x)$ pentru orice variabilă x ce apare liber în t, are loc $\mathcal{I}(t)(\gamma) = \mathcal{I}(t)(\gamma')$.

Demonstraţie Vom demonstra teorema prin inducţie structurală asupra λ-termului t.

Cazul 1: $t = x \in \mathcal{V}$. Atunci, x este liberă în t şi, deci, pentru orice două atribuiri γ şi γ' ce satisfac $\gamma(x) = \gamma'(x)$ are loc

$$\mathcal{I}(t)(\gamma) = \gamma(x) = \gamma'(x) = \mathcal{I}(t)(\gamma').$$

Cazul 2: $t = f \in \mathcal{F}$. Atunci, pentru orice două atribuiri γ şi γ' are loc

$$\mathcal{I}(t)(\gamma) = \mathcal{I}_0(t) = \mathcal{I}(t)(\gamma').$$

Cazul 3: $t = u(t_1, \ldots, t_n)$, u este de tip $(\tau_1, \ldots, \tau_n \rightarrow \tau)$ şi t_i este de tip τ_i pentru orice $1 \leq i \leq n$. Presupunem că λ-termii $u, t_1, \ldots, t_n$ satisfac teorema. Fie γ şi γ' două atribuiri ce coincid pe variabilele libere din t. Atunci, ele vor coincide şi pe variabilele libere din $u, t_1, \ldots, t_n$, ceea ce conduce la

$$\begin{aligned} \mathcal{I}(t)(\gamma) &= \mathcal{I}(u)(\gamma)(\mathcal{I}(t_1)(\gamma), \ldots, \mathcal{I}(t_1)(\gamma)) \\ &= \mathcal{I}(u)(\gamma')(\mathcal{I}(t_1)(\gamma'), \ldots, \mathcal{I}(t_1)(\gamma')) \\ &= \mathcal{I}(t)(\gamma'). \end{aligned}$$

Cazul 4: $t = [\lambda x_1, \ldots, x_n.u]$, u este de tip σ, x_i este variabilă de tip τ_i pentru orice $1 \leq i \leq n$, şi $\tau = (\tau_1, \ldots, \tau_n \rightarrow \sigma)$. Presupunem că λ-termul u satisface teorema, şi fie γ şi γ' două atribuiri ce coincid pe variabilele libere din t. Atunci, $\gamma[x_1/d_1] \cdots [x_n/d_n]$ şi $\gamma'[x_1/d_1] \cdots [x_n/d_n]$ vor coincide şi pe variabilele libere din u, ceea ce conduce la

$$\begin{aligned} \mathcal{I}(t)(\gamma)(d_1, \ldots, d_n) &= \mathcal{I}(u)(\gamma[x_1/d_1] \cdots [x_n/d_n]) \\ &= \mathcal{I}(u)(\gamma'[x_1/d_1] \cdots [x_n/d_n]) \\ &= \mathcal{I}(t)(\gamma')(d_1, \ldots, d_n), \end{aligned}$$

pentru orice $(d_1, \ldots, d_n) \in D_{\tau_1} \times \cdots \times D_{\tau_n}$. □

Corolarul 5.3.1.1. Fie $\mathcal{B}$ o bază, $\mathcal{I}$ o interpretare a ei şi t un λ-term. Dacă t nu are variabile libere, atunci $\mathcal{I}(t)(\gamma) = \mathcal{I}(t)(\gamma')$, pentru orice două atribuiri γ şi γ'.

Demonstraţie Direct de la Teorema 5.3.1.2 □

5.3.2 Programe recursive

Programele recursive au fost introduse de McMarthy în 1963 [122]. Abordarea noastră va urma [115, 112].

Un program recursiv este o mulţime de *ecuaţii recursive*, fiecare astfel de ecuaţie fiind alcătuită din 2 termi, unul definind *antetul ecuaţiei*, iar celălalt, *corpul acesteia*.

Aşa cum vom vedea, termii ce intră în componenţa ecuaţiilor recursive sunt λ-termi peste o bază potrivit aleasă.

Vom considera o mulţime $\mathcal{T}_0 = \{b, d\}$ de *tipuri de bază*. b specifică un tip de bază boolean, iar d specifică un tip arbitrar de date.

Tipurile pentru programe recursive se definesc inductiv prin:

- orice tip de bază este tip;
- dacă $\beta_1, \ldots, \beta_s, \beta$ sunt tipuri de bază, atunci $(\beta_1, \ldots, \beta_s \to \beta)$ este tip.

O *bază* pentru construcţia de programe recursive este un triplet $\mathcal{B} = (\mathcal{T}_0, \mathcal{V}, \mathcal{F})$, unde:

- $\mathcal{T}_0$ este o mulţime de tipuri de bază (ca mai sus);
- $\mathcal{V}$ este o mulţime de *variabile*. Variabilele sunt împărţite în două clase:
  - variabile de tip d, notate prin $x, y, z, \ldots$ (eventual indexate);
  - variabile de tip $(\underbrace{d, \ldots, d}_{n \geq 1} \to d)$. Acestea vor fi numite *variabile funcţionale* şi vor fi notate prin $F, G, H, \ldots$ (eventual indexate).
- $\mathcal{F}$ este o mulţime de *simboluri funcţionale*. Acestea vor fi notate $f, g, h, \ldots$ (eventual indexate) şi vom presupune că printre ele se găsesc şi următoarele simboluri funcţionale:

simbol funcţional	tip
$false$	b
$true$	b
$\neg$	$(b \to b)$
$=$	$(d, d \to b)$
$\vee,\ \wedge,\ \to,\ \leftrightarrow$	$(b, b \to b)$
if_then_else	$(b, d, d \to d)$

Fie $\mathcal{B}$ o bază pentru construcţia de programe recursive. *Termii* utilizaţi în definirea programelor recursive sunt definiţi inductiv astfel:

- orice variabilă de tip d este term de tip d;
- orice simbol funcţional f de tip b sau d este term de tip b sau, respectiv, d;
- dacă f este simbol funcţional de tip $(\beta_1, \ldots, \beta_s \to \beta)$ iar t_i sunt termi de tip β_i, $1 \leq i \leq s$, atunci $f(t_1, \ldots, t_n)$ este term de tip β;
- dacă F este variabilă funcţională de tip $(\underbrace{d, \ldots, d}_{s \geq 1} \to d)$ iar t_i sunt termi de tip d, $1 \leq i \leq s$, atunci $F(t_1, \ldots, t_n)$ este term de tip d.

O *ecuaţie* sau *procedură recursivă* este o pereche de termi

$$(F(x_1, \ldots, x_s), t),$$

unde $x_1, \ldots, x_s$ sunt variabile distincte, iar t este un term de tip d ce poate conţine orice variabilă funcţională însă, ca variabile de tip d, el poate conţine doar $x_1, \ldots, x_s$. Termul $F(x_1, \ldots, x_s)$ se numeşte *antetul ecuaţiei*, iar termul t, *corpul* acesteia. Uzual, ecuaţia $(F(x_1, \ldots, x_s), t)$ se mai notează prin

$$F(x_1, \ldots, x_s) \Leftarrow t$$

($\Leftarrow$ fiind un simbol nou).

Un *program recursiv* este un cuplu (S, k), unde:

- S este o mulţime de ecuaţii recursive

  $$S = \{(F_1(x_{11}, \ldots, x_{1s_1}), t_1), \ldots, (F_n(x_{n1}, \ldots, x_{ns_n}), t_n)\},$$

  astfel încât pentru orice $1 \leq i \leq n$, t_i poate conţine ca variabile funcţionale doar $F_1, \ldots, F_n$;

- $1 \leq k \leq n$.

Uzual, sistemul (S, k) se notează prin

$$(S, k) \quad \begin{cases} F_1(x_{11}, \ldots, x_{1s_1}) & \Leftarrow & t_1 \\ \cdots & & \\ F_n(x_{n1}, \ldots, x_{ns_n}) & \Leftarrow & t_n \end{cases}$$

Variabila funcţională F_k se numeşte *variabila funcţională principală*. Atunci când programul recursiv este format doar dintr-o singură ecuaţie recursivă, îl vom nota mai simplu prin

$$(S) \;\; F(x_1, \ldots, x_s) \Leftarrow t$$

Aşa cum se poate constata, termii ce intervin în definirea programelor recursive sunt λ-termi peste o bază $\mathcal{B}$ potrivit aleasă (ce include toate elementele menţionate mai sus). O astfel de bază va fi numită *bază pentru programe recursive*.

Exemplul 5.3.2.1. Următoarele construcţii sunt programe recursive:

(1) $(S) \;\; F(x) \Leftarrow if\ x = 0\ then\ 1\ else\ x * F(x-1)$

(acest program, interpretat peste $\mathbf{N}_\perp$ aşa cum vom vedea în secţiunea următoare, calculează funcţia factorial).

(2) $(S, 1) \quad \begin{cases} F_1(x) & \Leftarrow & if\ x = 0\ then\ 0\ else\ F_2(x-1) \\ F_2(x) & \Leftarrow & if\ x = 0\ then\ 1\ else\ F_1(x-1) \end{cases}$

(acest program calculează restul împărţirii unui număr natural la 2 atunci când este interpretat peste $\mathbf{N}_\perp$, aşa cum se va arăta în secţiunea următoare).

5.3.3 Semantica denotaţională a programelor recursive

Fie $\mathcal{B}$ o bază pentru programe recursive şi

$$(S,k)\ \begin{cases} F_1(x_{11},\ldots,x_{1s_1}) & \Leftarrow\ t_1 \\ \cdots & \\ F_n(x_{n1},\ldots,x_{ns_n}) & \Leftarrow\ t_n \end{cases}$$

un program recursiv. Asociem fiecărei ecuaţii recursive

$$F_i(x_{i1},\ldots,x_{is_i})\ \Leftarrow\ t_i$$

un λ-term

$$T_i = [\lambda F_1,\ldots,F_n.[\lambda x_{i1},\ldots,x_{is_i}.t_i]]$$

al cărui tip este

$$\tau_i = ((\underbrace{d,\ldots,d}_{s_1}\to d),\ldots,(\underbrace{d,\ldots,d}_{s_n}\to d)\to(\underbrace{d,\ldots,d}_{s_i}\to d)),$$

unde $1\le i\le n$.

Fie $\mathcal{I}$ o interpretare a bazei $\mathcal{B}$. Vom presupune următoarele:

- $D_b = Bool_\perp$, unde $Bool=\{0,1\}$;
- $D_d = D_\perp$, unde D este un domeniu nevid arbitrar dar fixat;
- $\mathcal{I}_0(false)=0$ şi $\mathcal{I}_0(true)=1$;
- $\mathcal{I}_0(f)$ este extensia naturală a simbolului funcţional f, pentru orice

  $$f\in\{\neg,=,\vee,\wedge,\to,\leftrightarrow\}$$

  (a se vedea Secţiunea 3.2.4). In cazul $f = if\_then\_else$, $\mathcal{I}_0(f)$ este extensia din Exemplul 3.2.4.1(3).

Sub această interpretare, domeniul tipului τ_i va fi

$$D_{\tau_i} = [[D_\perp^{s_1}\to D_\perp]\times\ldots\times[D_\perp^{s_n}\to D_\perp]\to[D_\perp^{s_i}\to D_\perp]].$$

Vom nota prin D^* domeniul

$$D^* = [D_\perp^{s_1}\to D_\perp]\times\ldots\times[D_\perp^{s_n}\to D_\perp].$$

Definiţia 5.3.3.1. Fie (S,k) un program recursiv şi $\mathcal{I}$ o interpretare ca mai sus. *Funcţia semantică* a programului (S,k) sub interpretarea $\mathcal{I}$ este funcţia

$$\phi_{\mathcal{I}}(S,k): D^*\to D^*$$

dată prin

$$\phi_{\mathcal{I}}(S,k) = (\mathcal{I}(T_1)(\gamma),\ldots,\mathcal{I}(T_n)(\gamma)),$$

unde γ este o atribuire arbitrară.

Observaţia 5.3.3.1.

(1) Definiţia funcţiei semantice a unui program recursiv este consistentă în sensul că ea nu depinde de asignarea aleasă deoarece λ-termii $T_1, \ldots, T_n$ nu au variabile libere (a se vedea Corolarul 5.3.1.1).

(2) Funcţia semantică $\phi_{\mathcal{I}}(S,k)$ este continuă deoarece $pr_i \circ \phi_{\mathcal{I}}(S,k) = \mathcal{I}(T_i)(\gamma)$ este funcţie continuă, pentru orice $1 \leq i \leq n$ (Corolarul 5.2.1.2).

Definiţia 5.3.3.2. Fie (S,k) un program recursiv şi $\mathcal{I}$ o interpretare ca mai sus. *Semantica denotaţională* a programului (S,k) este funcţia parţială

$$\mathcal{M}_{\mathcal{I}}(S,k) : D_d^{s_k} \rightsquigarrow D_d$$

dată prin

$$\mathcal{M}_{\mathcal{I}}(S,k)(a) = \begin{cases} pr_k(\mu(\phi_{\mathcal{I}}(S,k)))(a), & \text{dacă această valoare nu este } \perp \\ \text{nedefinită}, & \text{altfel,} \end{cases}$$

pentru orice $a \in D_{\perp}^{s_k}$.

Vom încheia secţiunea printr-un exemplu de calcul a semanticii denotaţionale a unui program recursiv.

Exemplul 5.3.3.1. Fie programul recursiv

$$(S,1) \begin{cases} F_1(x) \Leftarrow if\ x = 0\ then\ 0\ else\ F_2(x-1) \\ F_2(x) \Leftarrow if\ x = 0\ then\ 1\ else\ F_1(x-1) \end{cases}$$

din Exemplul 5.3.2.1(2) interpretat peste $\mathbf{N}_{\perp}$ (adică, $D_{\perp} = \mathbf{N}_{\perp}$).

λ-termii asociaţi sunt

$$T_1 = [\lambda F_1, F_2.[\lambda x.if\ x = 0\ then\ 0\ else\ F_2(x-1)]]$$

şi

$$T_2 = [\lambda F_1, F_2.[\lambda x.if\ x = 0\ then\ 1\ else\ F_1(x-1)]].$$

Aceşti λ-termi au tipurile

$$\tau_1 = \tau_2 = ((nat \to nat), (nat \to nat) \to (nat \to nat)),$$

iar domeniile corespunzătoare sunt

$$D_{\tau_1} = D_{\tau_2} = [[\mathbf{N}_{\perp} \to \mathbf{N}_{\perp}] \times [\mathbf{N}_{\perp} \to \mathbf{N}_{\perp}] \to [\mathbf{N}_{\perp} \to \mathbf{N}_{\perp}]].$$

Funcţia semantică a programului este

$$\phi_{\mathcal{I}}(S,1) = (\mathcal{I}(T_1)(\gamma), \mathcal{I}(T_n)(\gamma)),$$

unde γ este o atribuire arbitrară dar fixată.

Acum, va trebui să calculăm cel mai mic punct fix al funcţiei semantice a programului, ceea ce se reduce la calculul supremului lanţului

$$L = \{\phi_{\mathcal{I}}(S,1)^n(\perp_{(\mathbf{N}_{\perp} \to \mathbf{N}_{\perp})}, \perp_{(\mathbf{N}_{\perp} \to \mathbf{N}_{\perp})}) | n \geq 0\},$$

unde $\perp_{(\mathbf{N}_{\perp} \to \mathbf{N}_{\perp})}$ este cel mai mic element al mpo complete $[\mathbf{N}_{\perp} \to \mathbf{N}_{\perp}]$, adică funcţia ce returnează $\perp$ pentru orice $x \in \mathbf{N}_{\perp}$.

Pentru a putea lucra uşor cu aceste funcţii vom face următoarele notaţii:

- f_0 va desemna funcţia $\perp_{(\mathbf{N}_\perp \to \mathbf{N}_\perp)}$;
- $\phi_{\mathcal{I}}(S,1)^n(f_0,g_0) = (f_n,g_n)$, pentru orice $n \geq 0$, unde $g_0 = f_0$. In plus, $f_n = \mathcal{I}(T_1)(\gamma)(f_{n-1},g_{n-1})$ şi $g_n = \mathcal{I}(T_2)(\gamma)(f_{n-1},g_{n-1})$, pentru orice $n \geq 1$.

Pentru determinarea perechilor (f_n,g_n) vom face câteva iteraţii până când vom putem intui forma acestora. Are loc:

$$\begin{aligned} f_1(k) &= \mathcal{I}(T_1)(\gamma)(f_0,g_0)(k) \\ &= \mathcal{I}(T_1)(\gamma[F_1/f_0][F_2/g_0][x/k]) \\ &= if\ k=0\ then\ 0\ else\ g_0(k-1) \\ &= if\ k=0\ then\ 0\ else\ \perp \end{aligned}$$

$$\begin{aligned} g_1(k) &= \mathcal{I}(T_2)(\gamma)(f_0,g_0)(k) \\ &= \mathcal{I}(T_2)(\gamma[F_1/f_0][F_2/g_0][x/k]) \\ &= if\ k=0\ then\ 1\ else\ f_0(k-1) \\ &= if\ k=0\ then\ 1\ else\ \perp \end{aligned}$$

$$\begin{aligned} f_2(k) &= \mathcal{I}(T_1)(\gamma)(f_1,g_1)(k) \\ &= \mathcal{I}(T_1)(\gamma[F_1/f_1][F_2/g_1][x/k]) \\ &= if\ k=0\ then\ 0\ else\ g_1(k-1) \\ &= if\ k=0\ then\ 0\ else\ if\ k-1=0\ then\ 1\ else\ \perp \\ &= if\ k=0\ then\ 0\ else\ if\ k=1\ then\ 1\ else\ \perp \end{aligned}$$

$$\begin{aligned} g_2(k) &= \mathcal{I}(T_2)(\gamma)(f_1,g_1)(k) \\ &= \mathcal{I}(T_2)(\gamma[F_1/f_1][F_2/g_1][x/k]) \\ &= if\ k=0\ then\ 1\ else\ f_1(k-1) \\ &= if\ k=0\ then\ 1\ else\ if\ k-1=0\ then\ 0\ else\ \perp \\ &= if\ k=0\ then\ 1\ else\ if\ k=1\ then\ 0\ else\ \perp \end{aligned}$$

$$\begin{aligned} f_3(k) &= \mathcal{I}(T_1)(\gamma)(f_2,g_2)(k) \\ &= \mathcal{I}(T_1)(\gamma[F_1/f_2][F_2/g_2][x/k]) \\ &= if\ k=0\ then\ 0\ else\ g_2(k-1) \\ &= if\ k=0\ then\ 0\ else\ if\ k-1=0\ then\ 1\ else \\ &\qquad if\ k-1=1\ then\ 0\ else\ \perp \\ &= if\ k=0\ then\ 0\ else\ if\ k=1\ then\ 1\ else \\ &\qquad if\ k=2\ then\ 0\ else\ \perp \end{aligned}$$

$$\begin{aligned} g_3(k) &= \mathcal{I}(T_2)(\gamma)(f_2,g_2)(k) \\ &= \mathcal{I}(T_2)(\gamma[F_1/f_2][F_2/g_2][x/k]) \\ &= if\ k=0\ then\ 1\ else\ f_2(k-1) \\ &= if\ k=0\ then\ 1\ else\ if\ k-1=0\ then\ 0\ else \\ &\qquad if\ k-1=1\ then\ 1\ else\ \perp \\ &= if\ k=0\ then\ 1\ else\ if\ k=1\ then\ 0\ else \\ &\qquad if\ k=2\ then\ 1\ else\ \perp \end{aligned}$$

pentru orice $k \in \mathbf{N}_\perp$. In acest moment putem presupune că are loc

$$f_n(k) = \begin{cases} 0, & \text{dacă } k < n \text{ este par} \\ 1, & \text{dacă } k < n \text{ este impar} \\ \perp, & \text{altfel} \end{cases}$$

şi

$$g_n(k) = \begin{cases} 1, & \text{dacă } k < n \text{ este par} \\ 0, & \text{dacă } k < n \text{ este impar} \\ \perp, & \text{altfel} \end{cases}$$

pentru orice $n \geq 1$ şi $k \in \mathbf{N}_\perp$. Presupunerea noastră se dovedeşte a fi corectă, ceea ce poate fi demonstrat cu uşurinţă prin inducţie matematică.

Acum, supremum lanţului L se obţine imediat ca fiind

$$sup(L) = sup(\{(f_n, g_n) | n \geq 0\}) = (f^*, g^*),$$

unde

$$f^*(k) = \begin{cases} 0, & \text{dacă } k \in \mathbf{N} \text{ este par} \\ 1, & \text{dacă } k \in \mathbf{N} \text{ este impar} \\ \perp, & \text{dacă } k = \perp \end{cases}$$

şi

$$g^*(k) = \begin{cases} 1, & \text{dacă } k \in \mathbf{N} \text{ este par} \\ 0, & \text{dacă } k \in \mathbf{N} \text{ este impar} \\ \perp, & \text{dacă } k = \perp \end{cases}$$

pentru orice $k \in \mathbf{N}_\perp$. Ca urmare, semantica denotaţională a programului $(S, 1)$ este

$$\mathcal{M}_\mathcal{I}(S, 1)(k) = \begin{cases} 0, & \text{dacă } k \in \mathbf{N} \text{ este par} \\ 1, & \text{dacă } k \in \mathbf{N} \text{ este impar} \\ \text{nedefinită}, & \text{dacă } k = \perp, \end{cases}$$

pentru orice $k \in \mathbf{N}_\perp$.

5.3.4 Programe while

O altă clasă importantă de programe, de natură imperativă, este cea a *programelor while*. Diferenţa majoră între acestea şi programele recursive constă în aceea că programele while folosesc o structură specială pentru iteraţii, numită *while_do*.

O *bază* pentru construcţia programelor while este definită ca un triplet $\mathcal{B} = (\mathcal{V}, \mathcal{F}, \mathcal{P})$ format din 3 mulţimi disjuncte între ele, unde:

- $\mathcal{V}$ este o mulţime de *variabile*;
- $\mathcal{F}$ este o mulţime de *simboluri funcţionale,* fiecare având asociată o aritate. Simbolurile funcţionale de aritate 0 sunt numite şi *constante funcţionale*;
- $\mathcal{P}$ este o mulţime de *simboluri predicative*, fiecare având asociată o aritate. Simbolurile predicative de aritate 0 sunt numite şi *constante propoziţionale*.

Termii peste o bază $\mathcal{B}$ se definesc inductiv prin:

- orice variabilă sau constantă (simbol funcţional de aritate 0) este term;
- dacă $t_1, \ldots, t_n$ sunt termi şi f este simbol funcţional de aritate n, unde $n \geq 1$, atunci $f(t_1, \ldots, t_n)$ este term.

Expresiile logice peste o bază $\mathcal{B}$ sunt formule fără cuantificatori ale logicii cu predicate peste $\mathcal{B}$. Pentru definirea acestora vom utiliza *simbolurile logice* "$true$", "$false$", "$=$", "$\neg$", "$\vee$", "$\wedge$", "$\Rightarrow$" şi "$\Leftrightarrow$", şi *simbolurile auxiliare* "$($", "$)$", "$,$" şi "$.$". Toate aceste simboluri sunt presupune distincte între ele şi distincte de elementele mulţimilor bazei $\mathcal{B}$. Atunci, expresiile logice se definesc inductiv prin:

- simbolurile $true$ şi $false$ sunt expresii logice;
- constantele propoziţionale sunt expresii logice;
- dacă t_1 şi t_2 sunt termi, atunci $(t_1 = t_2)$ este expresie logică;
- dacă $t_1, \ldots, t_n$ sunt termi şi P este simbol predicativ n-ar, unde $n \geq 1$, atunci $P(t_1, \ldots, t_n)$ este expresie logică;
- dacă e_1 şi e_2 sunt expresii logice, atunci $(\neg e_1)$, $(e_1 \vee e_2)$, $(e_1 \wedge e_2)$, $(e_1 \Rightarrow e_2)$ şi $(e_1 \Leftrightarrow e_2)$ sunt expresii logice.

Acum, *programele while* peste o bază $\mathcal{B}$ se definesc inductiv, utilizând *simbolurile auxiliare* "$:=$", "$;$", "if", "$then$", "$else$", "$while$", şi "do", astfel:

- dacă $x \in \mathcal{V}$ şi t este term peste $\mathcal{B}$, atunci $x := t$ este program while peste $\mathcal{B}$;
- dacă S_1 şi S_2 sunt programe while peste $\mathcal{B}$ şi e este o expresie logică peste $\mathcal{B}$, atunci $S_1; S_2$, $if\ e\ then\ S_1\ else\ S_2$ şi $while\ e\ do\ S_1$ sunt programe while peste $\mathcal{B}$.

Simbolurile "if", "$then$" şi "$else$", luate împreună, nu trebuiesc confundate cu simbolul funcţional if_then_else utilizat în cadrul programelor recursive. Structura indusă de aceste simboluri va fi interpretată oarecum similar structurii if_then_else de la programe recursive.

Atunci când S_2 este obţinut prin intermediul constructorului ";" ($S_2 = S_2'; S_2''$), vom scrie $if\ e\ then\ S_1\ else\ (S_2)$ în loc de $if\ e\ then\ S_1\ else\ S_2$. Aceasta pentru a delimita zona de acţiune a lui if_then_else. De exemplu, structura

$$if\ e\ then\ S_1\ else\ S_2'; S_2''$$

poate fi interpretată ca fiind programul $if\ e\ then\ S_1\ else\ S_2'$ urmat de S_2'', sau $if\ e\ then\ S_1\ else\ S_2$ unde $S_2 = S_2'; S_2''$. Prin convenţia adoptată eliminăm această situaţie ambiguă. O altă metodă de eliminare a acestei ambiguităţi se poate face prin considerarea unui nou simbol, $endif$, şi utilizarea structurii

$$if\ e\ then\ S_1\ else\ S_2\ endif.$$

O discuţie similară are loc pentru $while\ e\ do\ S_1$.

Trebuie să remarcăm că programele while nu sunt liber inductiv definite deoarece structura $S_1; S_2; S_3$ are cel puţin două construcţii inductive diferite (şi astfel de cazuri sunt de fapt singurele posibile ce fac ca definiţia programelor while să nu fie liber

inductivă). O astfel de ambiguitate nu mai poate fi eliminată chiar aşa de simplu cum am făcut mai sus. In general, construcţiile ce nu sunt liber inductiv definite pot crea probleme relativ la definiţia (recursivă) a unei funcţii semantice a acestora. In cazul programelor while vom arăta că, cu toate că definiţia acestora nu este liber inductivă, funcţia semantică poate fi definită în mod consistent.

Exemplul 5.3.4.1. Următoarele construcţii sunt programe while peste o bază potrivit aleasă:

1. $while\ x > 0\ do\ x := x - 1$.
2. $y := 1;\ while\ \neg(x = 1)\ do\ (y := y * x;\ x := x - 1)$. Acest program while calculează funcţia factorial atunci când este interpretat peste numere naturale.
3. $z := 0;\ while\ y \leq x\ do\ (z := z + 1;\ x := x - y)$.

5.3.5 Semantica denotaţională a programelor while

Fie $\mathcal{B}$ o bază pentru programe while. O *interpretare* pentru baza $\mathcal{B}$ este o pereche $\mathcal{I} = (D, \mathcal{I}_0)$ formată dintr-un domeniu nevid D şi o funcţie de *interpretare iniţială* $\mathcal{I}_0$ ce satisface:

- $\mathcal{I}_0(f)$ este funcţie de la D^n la D, pentru orice $f \in \mathcal{F}$ de aritate $n \geq 0$;
- $\mathcal{I}_0(P)$ este funcţie de la D^n la $Bool$, pentru orice $P \in \mathcal{P}$ de aritate $n \geq 0$, unde $Bool = \{0, 1\}$ este o mulţime ce conţine două elemente distincte.

O *atribuire* sau *asignare* a bazei $\mathcal{B}$ sub o interpretare $\mathcal{I}$ este o funcţie $\gamma : \mathcal{V} \to D$. In teoria programării imperative astfel de funcţii se mai numesc şi *stări*. O stare furnizează deci valorile variabilelor la un moment dat. Insă, trebuie remarcat că nu se cere ca stările să fie, toate, accesibile de la *starea iniţială* a programului. Vom nota prin $\Gamma_{\mathcal{B},\mathcal{I}}$ mulţimea tuturor atribuirilor bazei $\mathcal{B}$ sub interpretarea $\mathcal{I}$. Notaţia va fi simplificată la Γ atunci când $\mathcal{B}$ şi $\mathcal{I}$ sunt clare din context. Este bine de avut continuu în vedere că γ reprezintă o stare, iar un program while nu face altceva decât să transforme o stare într-o altă stare.

Semantica termilor şi *expresiilor logice* ce ajută la construcţia programelor while se defineşte recursiv ca o funcţie $\mathcal{I}$ de la mulţimea termilor şi expresiilor logice la mulţimea $(\Gamma \to D)$, astfel:

- $\mathcal{I}(t)(\gamma) = \mathcal{I}_0(t)$, dacă $t \in \mathcal{F}$ este constantă;
- $\mathcal{I}(t)(\gamma) = \gamma(t)$, dacă $t \in \mathcal{V}$ este variabilă;
- $\mathcal{I}(f(t_1, \ldots, t_n))(\gamma) = \mathcal{I}_0(f)(\mathcal{I}(t_1)(\gamma), \ldots, \mathcal{I}(t_n)(\gamma))$;
- $\mathcal{I}(true)(\gamma) = 1$ şi $\mathcal{I}(false)(\gamma) = 0$;
- $\mathcal{I}(p)(\gamma) = \mathcal{I}_0(p)$, pentru orice constantă propoziţională p;

- $\mathcal{I}(t_1 = t_2)(\gamma) = \begin{cases} 1, & \text{dacă } \mathcal{I}(t_1)(\gamma) = \mathcal{I}(t_2)(\gamma) \\ 0, & \text{altfel} \end{cases}$

  (egalitatea $\mathcal{I}(t_1)(\gamma) = \mathcal{I}(t_2)(\gamma)$ este identitate de elemente în D);

- $\mathcal{I}(P(t_1, \ldots, t_n))(\gamma) = \mathcal{I}_0(P)(\mathcal{I}(t_1)(\gamma), \ldots, \mathcal{I}(t_2)(\gamma))$;

- $\mathcal{I}(\neg e)(\gamma) = \begin{cases} 1, & \text{dacă } \mathcal{I}(e)(\gamma) = 0 \\ 0, & \text{altfel} \end{cases}$

  (egalitatea $\mathcal{I}(e)(\gamma) = 0$ este identitate de elemente în $Bool$).

  In mod similar se defineşte $\mathcal{I}$ pentru celelalte expresii logice;

pentru orice $\gamma \in \Gamma$.

Cu aceste elemente pregătitoare putem introduce semantica denotaţională a programelor while. Inainte de aceasta trebuie să remarcăm că mulţimea atribuirilor (stărilor) nu este mpo completă deoarece nu avem definită nici o relaţie de ordine parţială pe D care să transforme D într-o mpo completă. Există două moduri de a transforma Γ într-o mpo completă. Un mod este de a transforma Γ într-o mpo plată $\Gamma_\perp$, iar altul este de a transforma D într-o mpo plată $D_\perp$ (faptul că $D_\perp$ este completă asigură că mulţimea tuturor funcţiilor de la Γ la $D_\perp$ este completă, în raport cu ordinea parţială pe funcţii indusă de ordinea parţială pe $D_\perp$). Vom adopta prima variantă deoarece prin cea de a doua se introduc multe stări suplimentare, pe când prin prima metodă se introduce doar o singură stare suplimentară, şi anume $\perp$. Cazul $\gamma = \perp$ va trebui tratat separat deoarece termii şi expresiile logice sunt interpretate ca funcţii definite pe Γ şi nu pe $\Gamma_\perp$.

Definiţia 5.3.5.1. Fie S un program while peste o bază $\mathcal{B}$ şi $\mathcal{I}$ o interpretare a bazei $\mathcal{B}$. *Funcţia semantică* a programului S sub interpretarea $\mathcal{I}$ este funcţia

$$\phi_{\mathcal{I}}(S) : \Gamma_\perp \to \Gamma_\perp$$

dată prin:

- $\phi_{\mathcal{I}}(S)(\gamma) = \begin{cases} \gamma[x/\mathcal{I}(t)(\gamma)], & \text{dacă } \gamma \neq \perp \\ \perp, & \text{dacă } \gamma = \perp, \end{cases}$

  pentru orice $\gamma \in \Gamma_\perp$, dacă S este programul $x := t$;

- $\phi_{\mathcal{I}}(S) = \phi_{\mathcal{I}}(S_2) \circ \phi_{\mathcal{I}}(S_1)$, dacă S este programul $S_1; S_2$;

- $\phi_{\mathcal{I}}(S)(\gamma) = \begin{cases} \phi_{\mathcal{I}}(S_1)(\gamma), & \text{dacă } \mathcal{I}(e)(\gamma) = 1 \text{ şi } \gamma \neq \perp \\ \phi_{\mathcal{I}}(S_2)(\gamma), & \text{dacă } \mathcal{I}(e)(\gamma) = 0 \text{ şi } \gamma \neq \perp \\ \perp, & \text{dacă } \gamma = \perp, \end{cases}$

  pentru orice $\gamma \in \Gamma_\perp$, dacă S este programul $if\ e\ then\ S_1\ else\ S_2$;

- $\phi_{\mathcal{I}}(S) = \mu(F)$, dacă S este programul $while\ e\ do\ S_1$, unde F este funcţia

$$F : [\Gamma_\perp \to \Gamma_\perp] \to [\Gamma_\perp \to \Gamma_\perp]$$

dată prin

$$F(f)(\gamma) = \begin{cases} (f \circ \phi_{\mathcal{I}}(S_1))(\gamma)), & \text{dacă } \mathcal{I}(e)(\gamma) = 1 \text{ şi } \gamma \neq \perp \\ \gamma, & \text{dacă } \mathcal{I}(e)(\gamma) = 0 \text{ şi } \gamma \neq \perp \\ \perp, & \text{dacă } \gamma = \perp, \end{cases}$$

pentru orice $f \in [\Gamma_\perp \to \Gamma_\perp]$ şi $\gamma \in \Gamma_\perp$.

Definiţia funcţiei semantice pentru structura $while\ e\ do\ S$ se bazează pe observaţia că, dacă scriem informal

$$while\ e\ do\ S \ = \ \begin{cases} (while\ e\ do\ S) \circ S, & \text{dacă } e \text{ este adevărată} \\ id, & \text{dacă } e \text{ este falsă} \end{cases}$$

şi notăm prin F funcţia din membrul drept al egalităţii, unde id este funcţia identitate, atunci $while\ e\ do\ S$ verifică proprietatea

$$F(while\ e\ do\ S) = while\ e\ do\ S.$$

Cu alte cuvinte, $while\ e\ do\ S$ este punct fix al funcţiei F.

Deoarece definiţia programelor while nu este liberă, va trebui să arătăm că, în adevăr, $\phi_{\mathcal{I}}(S)$ este funcţie. Concomitent vom arăta că aceasta este şi continuă.

Teorema 5.3.5.1. Pentru orice program recursiv S, $\phi_{\mathcal{I}}(S)$ este funcţie continuă.

Demonstraţie Vom face demonstraţia prin inducţie structurală asupra programului S.

Cazul 1: S este de forma $x := t$. Este clar că $\phi_{\mathcal{I}}(S)$ este funcţie. Ea este şi continuă deoarece este o extensie naturală (a se vedea Secţiunea 3.2.4).

Cazul 2: S este de forma $S_1; S_2$. Presupunem că $\phi_{\mathcal{I}}(S_1)$ şi $\phi_{\mathcal{I}}(S_2)$ sunt funcţii continue. Cum compunerea de funcţii continue conduce la funcţii continue, deducem că $\phi_{\mathcal{I}}(S)$ este funcţie continuă.

Cazul 3: S este de forma $if\ e\ then\ S_1\ else\ S_2$. Presupunem că $\phi_{\mathcal{I}}(S_1)$ şi $\phi_{\mathcal{I}}(S_2)$ sunt funcţii continue. Este clar că $\phi_{\mathcal{I}}(S)$ este funcţie. Ea este şi continuă deoarece poate fi considerată ca o compunere a funcţiilor $\mathcal{I}(S_1)$ şi $\phi_{\mathcal{I}}(S_2)$ cu extensia naturală la $\Gamma_\perp$ a funcţiei if_then_else definită pe Γ (Secţiunea 3.2.4).

Cazul 4: S este de forma $while\ e\ do\ S_1$. Presupunem că $\phi_{\mathcal{I}}(S_1)$ este funcţie continuă. Va fi suficient să arătăm că F este o funcţie continuă. Atunci, ea va avea un cel mai mic punct fix care va fi o funcţie continuă deoarece este element al mulţimii $[\Gamma_\perp \to \Gamma_\perp]$. Ca urmare, aceasta va demonstra atât faptul că $\phi_{\mathcal{I}}$ este bine-definită, cât şi faptul că este funcţie continuă.

Vom arăta că F este continuă în mod direct. Fie $L \subseteq [\Gamma_\perp \to \Gamma_\perp]$ un lanţ nevid de funcţii. Cum $[\Gamma_\perp \to \Gamma_\perp]$ este mpo completă, există $sup(L)$. Va trebui să arătăm că are loc $F(sup(L))(\gamma) = sup(F(L))(\gamma)$, pentru orice $\gamma \in \Gamma_\perp$. Vom lua în considerare 3 cazuri (şi vom utiliza intens Lema 5.1.2):

- $\gamma = \bot$. Atunci, $F(sup(L))(\gamma) = \bot$ (conform definiţiei funcţiei F) şi

$$sup(F(L))(\gamma) = sup(F(L)(\gamma)) = sup(\{\bot\}) = \bot,$$

stabilind astfel egalitatea $F(sup(L))(\gamma) = sup(F(L))(\gamma)$.

- $\mathcal{I}(e)(\gamma) = 0$ şi $\gamma \neq \bot$. Atunci, $F(sup(L))(\gamma) = \gamma$ (conform definiţiei funcţiei F) şi

$$sup(F(L))(\gamma) = sup(F(L)(\gamma)) = sup(\{\gamma\}) = \gamma,$$

stabilind astfel egalitatea $F(sup(L))(\gamma) = sup(F(L))(\gamma)$.

- $\mathcal{I}(e)(\gamma) = 1$ şi $\gamma \neq \bot$. Atunci, $F(sup(L))(\gamma) = sup(L)(\phi_{\mathcal{I}}(S_1)(\gamma))$ (conform definiţiei funcţiei F). Cum $\phi_{\mathcal{I}}(S_1)$ este funcţie continuă,

$$F(L) = \{f \circ \phi_{\mathcal{I}}(S_1) | f \in L\}$$

este lanţ de funcţii continue în $[\Gamma_\bot \to \Gamma_\bot]$, al cărui suprem este $sup(F(L)) = sup(L) \circ \phi_{\mathcal{I}}(S_1)$ (ceea ce este uşor de văzut). Atunci,

$$sup(F(L))(\gamma) = (sup(L) \circ \phi_{\mathcal{I}}(S_1))(\gamma)) = sup(L)(\phi_{\mathcal{I}}(S_1)(\gamma)),$$

stabilind astfel egalitatea $F(sup(L))(\gamma) = sup(F(L))(\gamma)$.

Aceasta încheie demonstraţia teoremei [3]. □

Semantica denotaţională a programelor while se obţine ca şi în cazul programelor recursive.

Definiţia 5.3.5.2. Fie S un program while peste o bază $\mathcal{B}$ şi $\mathcal{I}$ o interpretare a bazei $\mathcal{B}$. *Semantica denotaţională* a programului S este funcţia parţială

$$\mathcal{M}_{\mathcal{I}}(S) : \Gamma \rightsquigarrow \Gamma$$

dată prin

$$\mathcal{M}_{\mathcal{I}}(S)(\gamma) = \begin{cases} \phi_{\mathcal{I}}(S)(\gamma), & \text{dacă această valoare nu este } \bot \\ \text{nedefinită}, & \text{altfel,} \end{cases}$$

pentru orice $\gamma \in \Gamma$.

[3] Utilizând λ-notaţia se poate da o altă demonstraţie faptului că F este continuă, arătând că F este interpretarea unui λ-term T ca cel de mai jos [112]:

$$T = [\lambda f.[\lambda\gamma.if\ \gamma = \gamma\ then\ if\ E\ then\ T\ else\ \gamma\ else\ t]]$$

In cadrul acestui λ-term interpretarea lui T_1 trebuie să fie $\phi_{\mathcal{I}}(S_1)$ iar interpretarea expresiei e trebuie să fie extensia naturală a funcţiei $\mathcal{I}(e)$ (reamintim că $\mathcal{I}(e)$ este definită pe Γ şi nu pe $\Gamma_\bot$). In plus, if_then_else din cadrul acestui λ-term T trebuie interpretat ca în Secţiunea 3.2.4.

Deşi această soluţie ar părea mai simplă, ea ridică multe probleme relativ la readaptarea funcţiilor la domeniul $\Gamma_\bot$. Din punctul nostru de vedere, demonstraţia deja adoptată este de preferat.

Incheiem secţiune printr-un exemplu de calcul a semanticii denotaţionale a unui program while.

Exemplul 5.3.5.1. Fie programul while S dat prin

$$y := 1;\ while\ \neg(x = 1)\ do\ (y := y * x;\ x := x - 1)$$

şi interpretat peste $D = \mathbf{N}$ cu interpretarea uzuală a operatorilor $\neg$, $*$ şi $-$ (mai mult, vom nota $\mathcal{I}_0(*)$ tot prin $*$, $\mathcal{I}_0(-)$ tot prin $-$ şi $\mathcal{I}_0(n)$ tot prin n, pentru orice $n \geq 0$).

Fie γ o atribuire diferită de $\perp$. Atunci,

$$\begin{aligned}
\phi_{\mathcal{I}}(S)(\gamma) &= \phi_{\mathcal{I}}(while\ \neg(x = 1)\ do\ (y := y * x;\ x := x - 1))(\phi_{\mathcal{I}}(y := 1)(\gamma)) \\
&= \phi_{\mathcal{I}}(while\ \neg(x = 1)\ do\ (y := y * x;\ x := x - 1))(\gamma[y/\mathcal{I}_0(1)]) \\
&= \phi_{\mathcal{I}}(while\ \neg(x = 1)\ do\ (y := y * x;\ x := x - 1))(\gamma[y/1]) \\
&= \mu(F)(\gamma[y/1]),
\end{aligned}$$

unde $F(f)(\gamma')$ este dată prin

- $F(f)(\gamma') = f(\phi_{\mathcal{I}}(y := y * x;\ x := x - 1)(\gamma'))$, dacă $\mathcal{I}(\neg(x = 1))(\gamma') = 1$ şi $\gamma' \neq \perp$;
- $F(f)(\gamma') = \gamma'$, dacă $\mathcal{I}(\neg(x = 1))(\gamma') = 0$ şi $\gamma' \neq \perp$;
- $F(f)(\gamma') = \perp$, dacă $\gamma' = \perp$,

pentru orice $f \in [\Gamma_\perp \to \Gamma_\perp]$ şi $\gamma' \in \Gamma_\perp$.

Dacă explicităm şi mai mult funcţia F obţinem că aceasta este dată prin

- $F(f)(\gamma') = f(\gamma'[y/\gamma'(y) * \gamma'(x)][x/(\gamma'[y/\gamma'(y) * \gamma'(x)](x) - 1)])$, dacă $\gamma'(x) \neq 1$ şi $\gamma' \neq \perp$;
- $F(f)(\gamma') = \gamma'$, dacă $\gamma'(x) = 1$ şi $\gamma' \neq \perp$;
- $F(f)(\gamma') = \perp$, dacă $\gamma' = \perp$.

Fie $f_0 = \perp_{(\Gamma_\perp \to \Gamma_\perp)}$ cel mai mic element al mpo complete $[\Gamma_\perp \to \Gamma_\perp]$, adică funcţia dată prin $f_0(\gamma') = \perp$, pentru orice $\gamma' \in \Gamma_\perp$. Calculul celui mai mic punct fix al funcţiei F se reduce la calculul supremului lanţului $L = \{F^n(f_0) | n \geq 0\}$. Ca urmare, vom determina întâi elementele acestui lanţ. Are loc:

$$F(f_0)(\gamma') = \begin{cases} \perp, & \text{dacă } \gamma'(x) \neq 1 \\ \gamma', & \text{dacă } \gamma'(x) = 1 \end{cases}$$

$$\begin{aligned}
F^2(f_0)(\gamma') &= \begin{cases} \perp, & \text{dacă } \gamma'(x) \neq 1 \text{ şi } \gamma'(x) \neq 2 \\ \gamma'[y/\gamma'(y) * 2][x/1], & \text{dacă } \gamma'(x) = 2 \\ \gamma', & \text{dacă } \gamma'(x) = 1 \end{cases} \\
&= \begin{cases} \perp, & \text{dacă } \gamma'(x) \neq 1 \text{ şi } \gamma'(x) \neq 2 \\ \gamma'[y/\gamma'(y) * 2][x/1], & \text{dacă } \gamma'(x) = 2 \\ \gamma'[y/\gamma'(y) * 1][x/1], & \text{dacă } \gamma'(x) = 1 \end{cases}
\end{aligned}$$

pentru orice γ'. Printr-o simplă inducţie matematică obţinem:

$$F^n(f_0)(\gamma') = \begin{cases} \perp, & \text{dacă } \gamma'(x) < 1 \text{ sau } \gamma'(x) > n \\ \gamma'[y/\gamma'(y) * j * \cdots * 2 * 1][x/1], & \text{dacă } \gamma'(x) = j \text{ şi } 1 \leq j \leq n, \end{cases}$$

pentru orice γ' şi $n \geq 1$.

Atunci, este uşor de văzut că are loc:

$$\mu(F)(\gamma') = \begin{cases} \perp, & \text{dacă } \gamma'(x) < 1 \\ \gamma'[y/\gamma'(y) * n * \cdots * 2 * 1][x/1], & \text{dacă } \gamma'(x) = n \geq 1, \end{cases}$$

pentru orice γ'. Inlocuind γ' cu $\gamma[y/1]$ obţinem

$$\mu(F)(\gamma[y/1]) = \begin{cases} \perp, & \text{dacă } \gamma(x) < 1 \\ \gamma[y/1 * n * \cdots * 2 * 1][x/1], & \text{dacă } \gamma(x) = n \geq 1, \end{cases}$$

pentru orice $\gamma \neq$. Atunci, semantica programului nostru va fi:

$$\mathcal{M}(S)(\gamma) = \begin{cases} \gamma[y/1 * n * \cdots * 2 * 1][x/1], & \text{dacă } \gamma(x) = n \geq 1 \\ \text{nedefinită}, & \text{dacă } \gamma(x) < 1, \end{cases}$$

pentru orice γ.

De exemplu, dacă alegem γ astfel încât $\gamma[x] = 5$, atunci $\mathcal{M}(S)(\gamma) = 1 * 5 * 4 * 3 * 2 * 1$.

Bibliografie

[1] Agrawal, Kayal, Saxena. *PRIMES is in P*, Annals of Mathematics 160(2), 2004, 781-793.

[2] M. Armbrust, J. Schmidt. *Zum Cayleyschen Darstellungssatz*, Mathematische Annalen 154, 1964, 70–72.

[3] R.B. Ash. *Information Theory*, Wiley, 1965.

[4] P. Bachman. *Die analytische Zahlentheorie*, Teubner, Leipzig, 1894.

[5] P. Bernays. *A System of Axiomatic Set Theory II*, Journal of Symbolic Logic 6, 1941, 1–17.

[6] P. Bernays. *A System of Axiomatic Set Theory VII*, Journal of Symbolic Logic 19, 1954, 81–96.

[7] J. Berstel, D. Perrin. *Theory of Codes*, Academic Press, 1985.

[8] G. Birkhoff. *On the Combination of Subalgebras*, Proceedings of the Cambridge Philosophical Society 29, 1933, 441–464.

[9] G. Birkhoff. *On the Structure of Abstract Algebras*, Proceedings of the Cambridge Philosophical Society 31, 1935, 433–454.

[10] G. Birkhoff. *On Groups of Automorphisms*, Rev. Un. Math. Argentina 11, 1946, 155–157 (în spaniolă).

[11] G. Birkhoff, O. Frink. *Representations of Lattices by Sets*, Transactions of the American Mathematical Society 64, 1948, 299–316.

[12] G. Birkhoff. *Lattice Theory*, Colloquium Publications vol. 25 of the American Mathematical Society, 1995 (a 8-a ediţie).

[13] G. Boole. *Mathematical Analysis of Logic, Being an Essay Toward a Calculus of Deductive Reasoning*, Macmillan, Barclay and Macmillan, London, 1847.

[14] G. Boole. *An investigation into the Laws of Thought, on Which are Founded the Mathematical Theories of Logic and Probabilities*, Walton and Maberley, London, 1854.

[15] A. Borgers. *Development of the Notion of Set and of the Axioms of Sets*, Synthese 7, 1949, 374–390.

[16] N. Bourbaki. *Théorie des ensembles*, Actualites Scientifiques et Industrielles 846, Herman et Cie, Paris, 1939.

[17] N. Bourbaki. *Théorie des ensembles*, Ch. 1–2, Paris, 1954 (a 2-a ediţie, 1960); Ch. 3, Paris, 1956 (a 2-a ediţie, 1963).

[18] N. Bourbaki. *General Topology*, Addison-Wesley, Reading, Mass., 1968.

[19] S. Burris, H.P. Sankappanavar. *A Course in Universal Algebra*, Springer-Verlag, 1981.

[20] *Mechanizable Proofs About Parallel Processes*, Proceedings of the 14th Annual IEEE Symposium on Switching and Automata Theory, 1973, 34.

[21] G. Cantor. *Ein Beitrag zur Mannigfaltigkeitslehre*, Journal für Mathematik 84, 1878, 242–258 (de asemenea în [25], 119–138).

[22] G. Cantor. *Über unendliche, lineare Punktmannigfaltigkeiten (V)*, Mathematische Annalen 21, 1883, 545–591.

[23] G. Cantor. *Beiträge zur Begründung der transfiniten Mengenlehre I*, Mathematische Annalen 46, 1895, 418–512.

[24] G. Cantor. *Beiträge zur Begründung der transfiniten Mengenlehre (II)*, Mathematische Annalen 49, 1897, 207–246.

[25] G. Cantor. *Gesammelte Abhandlungen mathematischen und philosophischen Inhalts*, E. Zermelo (ed.), Berlin, 1932.

[26] C.C. Chang. *Some General Theorems on Direct Products and Their Applications in the Theory of Models*, Nederl. Akad. Wetensch. Proc. ser. A 57, 1954, 592–598.

[27] A. Church. Annals of Mathematics 34, 1933, 863–.

[28] A. Church. *An Unsolvable Problem of Elementary Number Theory*, American Journal of Mathematics 58, 1936, 345–363.

[29] A.H. Clifford, G.B. Preston. *The Algebraic Theory of Semigroups*, Mathematical Surveys 7, vol. 1, American Mathematical Society, Providence, 1961.

[30] A.H. Clifford, G.B. Preston. *The Algebraic Theory of Semigroups*, Mathematical Surveys 7, vol. 2, American Mathematical Society, Providence, 1967.

[31] P. Cohen. *The Independence of the Continuum Hyphothesis I, II*, Proceedings of the National Academy of Sciences (USA) 50, 1963, 1143–1148 (de asemenea în P. Cohen: *Set Theory and the Continuum Hypothesis*, W.A. Benjamin, New York, 1966).

[32] P.M. Cohn. *Universal Algebra*, a doua ediţie, Reidel Publishing Company, 1981.

[33] P.M. Cohn. *Classic Algebra*, John Wiley & Sons, 2000.

[34] I. Csiszár, J. Körner. *Information Theory: Coding Theorems for Discrete Memoryless Systems*, Academic Press, 1981.

[35] J. Daemen, V. Rijmen. *The Design of Rijndael*, Springer-Verlag, 2002.

[36] G. Davida. *Chosen Signature Cryptanalysis of the RSA Public Key Cryptosystem*, Technical report TR-CS-82-2, Dept. of Electrical Engineering and Computer Science, University of Wisconsin, 1982.

[37] A.C. Davis. *A Characterization of Complete Lattices*, Pacific Journal of Mathematics 5, 1955, 311–319.

[38] R. Dedekind. *Was sind und was sollen die Zahlen?*, Braunschweig, 1888 (a 6-a ediţie, Braunschweig, 1930).

[39] R. Dedekind. *Über die von drei Moduln erzeugte Dualgruppe*, Mathematische Annalen 53, 1900, 371–403.

[40] R. Dedekind. *Gesammelte mathematische Werke I, II, III*, Volume editate de R. Fricke, E. Noether şi O. Ore, Braunschweig, 1930–1932.

[41] K. Devlin. *The Joy of Sets. Fundamentals of Contemporary Set Theory*, Springer-Verlag, a 2-a ediţie, 1993.

[42] W. Diffie, M.E. Hellman. *Multiuser Cryptographic Techniques*, Proceedings of AFIPS National Computer Conference, 1976, 109–112.

[43] W. Diffie, M.E. Hellman. *New Directions in Cryptography*, IEEE Transactions on Information Theory 6, 1976, 644–654.

[44] P. Dubreil. *Contribution a la theorie de demi-groupes*, Mem. Acad. Sci. France 2(63), 1941.

[45] P. Dubreil-Jacotin. *Sur l'immersion d'un semi-groupe dans un groupe*, C.R. Acad. Sci. Paris 225, 1947, 787–788.

[46] Electronic Frontier Foundation. *Cracking DES. Secrets of Encryption Research, Wiretap Politics & Chip Design*, O'Reilly, 1998.

[47] T. ElGamal. *A Public Key Cryptosuystem and a Digital Sinature Based on Discrete Logarithms*, IEEE Transactions on Information Theory 31, 1985, 469–472.

[48] Federal Register. *Proposed Federal Information Processing Standard for Digital Signature Standard (DSS)*, Federal Register 169, 1991, 42980–42982.

[49] P.A. Fejer, D.A. Simovici, *Foundations of Computer Science. Volume I: Sets, Relations and Induction*, Springer-Verlag, 1991.

[50] Federal Information Processing Standard Publication 186-2. *Digital Signature Standard (DSS)*, National Institute of Standards and Technology (NIST), 2000.

[51] Federal Information Processing Standard Publication 197. *Advanced Encryption Standard*, National Institute of Standards and Technology (NIST), 2001.

[52] A. Fraenkel. , Journal fr die Reine und Angewandte Mathematik (A. L. Crelle), vol. 145, 1914.

[53] A. Fraenkel. *Zu den Grundlagen der Cantor-Zermeloschen Mengenlehre*, Mathematische Annalen 86, 1922, 230–237.

[54] A. Fraenkel. *Abstract Set Theory*, a 2-a ediţie, North-Holland, 1961.

[55] A. Fraenkel, Y. Bar-Hillel. *Foundations of Set Theory*, North-Holland, 1958.

[56] A. Fraenkel, Y. Bar-Hillel, A. Levy. *Foundations of Set Theory*, a 2-a ediţie, North-Holland, 1984.

[57] G. Frege. *Die Grundlagen der Aritmetik. Eine logischmathematische Untersuchung über den Begriff der Zahl*, Breslau, 1884.

[58] G. Frege. *Grundgesetze der Aritmetik*, Vol. I, Jena, 1893; Vol. II, Jena, 1903.

[59] L. Fuchs. *On Subdirect Unions*, Acta Math. Sci. Hungar. 3, 1952, 103–120.

[60] C.F. Gauss. *Disquisitiones Arithmeticae*, revised English translation by W.C. Waterhouse, Springer-Verlag, 1986.

[61] P. Geach, M. Black. *Translations from the Philosophical Writings of Gottlob Frege*, Blackwell, Oxford, 1952.

[62] G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M. Mislove, D.S. Scott. *Continuous Lattices and Domains*, Encyclopedia of Mathematics and Its Applications, vol. 93, 2003.

[63] E.N. Gilbert, E.F. Moore. *Variable length binary encodings*, Bell System Tech. J. 38, 1959, 933–967.

[64] K. Gödel. *The Consistency of the Axiom of Choice and the Generalized Continuum Hypothesis*, Proceedings of the National Academy of Sciences

[65] K. Gödel. *Consistency Proof for the Generalized Continuum Hypothesis*, Proceedings of the National Academy of Sciences (USA) 25, 1938, 220–224.

[66] G. Grätzer, E.T. Schmidt. *Characterizations of Congruence Lattices of Abstract Algebras*, Acta Sci. Math. (Szeged) 24, 1963, 34–59.

[67] G. Grätzer. *Universal Algebra*, Springer Verlag, 1979 (a doua ediţie).

[68] J.A. Green. *On the Structure of Semigroups*, Ann. Math. 54, 1951, 163–172.

[69] F.M. Hall. *An Introduction to Abstract Algebra*, Cambridge University Press, 1969.

[70] P.R. Halmos. *Naive Set Theory*, Springer-Verlag, 1974.

[71] W.R. Hamilton. *On Quaternions or on a New System of Imaginaries in Algebra*, Phil. Mag. 3rd Ser., 1844, 10–13.

[72] R.W. Hamming. *Coding and Information Theory*, Prentice-Hall, 1986.

[73] G.H. Hardy, E.M. Wright. *An Introduction to the Theory of Numbers*, Clarendon Press, Oxford, a 5-a ediţie, 1990.

[74] F. Hartogs. *Über das Problem der Wohlordnung*, Mathematische Annalen 76, 1915, 438–443.

[75] F. Hausdorff. *Grundzüge der Mengenlehre*, Leipzig, 1914.

[76] M.E. Hellman. *The Mathematic sof Public-Key Cryptography*, Scientific American 241, 1979, 146–157.

[77] P.M. Higgins. *Algebras with a Scheme of Operators*, Mathematische Nachrichten 27, 1963, 115–132.

[78] P.M. Higgins. *Techniques of Semigroup Theory*, Oxford University Press, 1992.

[79] R. Hill. *A First Course in Coding Theory*, Clarendon Press, 1993.

[80] D. Hilbert. *Die Theorie der algebraischen Zahlk¨orper*, Jahresbericht der Deutschen Mathematiker Vereiningung, Vol. 4, 1897.

[81] J.M. Howie. *An Introduction to Semigroup Theory*, Academic Press, 1976.

[82] K. Hrbacek, T. Jech. *Introduction to Set Theory*, Marcel Dekker, 1978.

[83] L.K. Hua. *Introduction to Number Theory*, Springer-Verlag, Berlin, 1982.

[84] D.A. Huffman. *A Method for the Construction of Minimum Redundancy Codes*, Proceedings of IRE 40, 1952, 1098–1101.

[85] Th.W. Hungerford. *Algebra*, a 5-a ediţie, Springer-Verlag, 1989.

[86] J.R. Isbell. *Subobjects, Adequacy, Completeness and Categories of Algebras*, Rozprawy Mat. 36, 1964, 33 pag.

[87] E. Jacobsthal. *Über den Aufbau der transfiniten Arithmetik*, Mathematische Annalen 66, 1909, 145–194.

[88] T.J. Jech. *About the axiom of choice*, în J. Bairwise (ed.): *Handbook of Mathematical Logic (Part B)*, Amsterdam, North-Holland, 1977.

[89] T.J. Jech. *Set Theory*, Springer-Verlag, 1978 (a 2-a ediţie, 1997).

[90] D. Kahn. *The Codebreakers: The Story of Secret Writing*, Macmillan Publishing Co., 1967.

[91] B. Knaster. *Un theéorème sur les fonctions d'ensembles*, Ann. Soc. Polon. Math. 6, 1928, 133–134.

[92] L.G. Kraft. *A Device for Quantizing, Grouping, and Coding Amplitude Modulated Pulses*, M.S. Thesis, Electrical Engineering Department, Massachusets Institute of Technology, 1949.

[93] Kranakis. *Primality and Cryptography*, Wiley-Teubner, Series on Applicable Theory in Computer Science, 1986.

[94] K. Kunen. *Set Theory. An Introduction to Independence Proofs*, North Holland, 1980.

[95] K. Kuratowski. *Sur la notion de l'ordre dans le théorie des ensembles*, Fundamenta Mathematicae 2, 1921, 161–171.

[96] K. Kuratowski. *Une methode d'elimination des nombres transfinis des raisonnements mathematiques*, Fundamenta Mathematicae 3, 1922, 76–108.

[97] K. Kuratowski, A. Mostowski. *Set Theory*, North-Holland, 1968.

[98] G. Lallement. *Semigroups and Combinatorial Applications*, John Wiley & Sons, 1979.

[99] E. Landau. *Handbuch der Lehre von der Verteilung der Primzahlen*, Teubner, Leipzig, 1909.

[100] E. Landau. *Vorlesungen über Zahlentheorie*, Hirzel, Leipzig, 1927.

[101] S. Lang. *Linear Algebra*, Springer-Verlag, 1987.

[102] A.K. Lenstra. *Memo on RSA Signature Generation in the Presence of Faults*, personal communication, 1996.

[103] C.H. Lewis, B.K. Rosen. *Recursively Defined Data Types (I)*, Proceedings of the ACM Symposium on Principles of Programming Languages, 1973, 125–138.

[104] S. Leśniewski. *Grundzüge eines neuen Systems der Grundlagen der Mathematik*, Fundamenta Mathematicae 14, 1929, 1–81.

[105] F.W. Levi. *On Semigroups*, Bull. Calcutta Math. Soc. 36, 1944, 141–146.

[106] A. Levy. *The Independence of Certain Consequences of the Axiom of Choice*, Fundamenta Mathematicae 54, 1964, 135–157.

[107] A. Levy. *Basic Set Theory*, Springer-Verlag, 1979.

[108] B. Levi. *Intorno alla teoria degli aggregati*, Royale Istituto Lombardo di Science e Lettere, Rendiconti 2, 1902, 863–868.

[109] R. Lidl, H. Niederreiter. *Introduction to Finite Fields and Their Applications*, Cambridge University Press, 1986.

[110] Van Lindt. *Introduction to Coding Theory*, Springer-Verlag, 1982.

[111] E.S. Ljapin. *Semigroups*, Nauka, Moscow, 1960 (în ruseşte).

[112] J. Loeckx, K. Sieber. *The Foundations of Program Verification*, John Wiley and Sons, 1984 (a 2-a ediţie, 1987).

[113] J. Loeckx, H.-D. Ehrich, M. Wolf. *Specification of Abstract Data Types*, Wiley & Teubner, 1996.

[114] F.J. MacWilliams, N.J.A. Sloane. *The Theory of Error-Correcting Codes*, North-Holland, 1977.

[115] Z. Manna. *Mathematical Theory of Computation*, McGraw-Hill, 1974.

[116] R.C. Merkle. *Secure Communication Over Insecure Channels*, Communications of the ACM 4, 1978, 294–299.

[117] R.C. Merkle, M. Hellman. *Hiding Information and Signatures in Trapdoor Knapsacks*, IEEE Transactions on Information Theory 5, 1978, 525–530.

[118] R.C. Merkle. *Secrecy, Authentication, and Public Key Systems*, Ph.D. dissertation, Stanford University, 1979.

[119] G. Markowski. *Categories of Chain-Complete Posets*, IBM Technical Report RC 5100, T.J. Watson Research Center, Yorktown Heights, 1974.

[120] G. Markowski. *Chain-Complete Posets and Directed Sets with Applications*, Algebra Universalis 6, 1976, 53–68.

[121] G. Markowski. *Bases for Chain-Complete Posets*, IBM Journal of Research Development, 1976, 138–147.

[122] J. McCarthy. *A Basis for a Mathematical Theory of Computation*, in P. Braffort and D. Hirschberg (eds.), Computer Programming and Formal Systems, North-Holland, 1963, 33–70.

[123] R.J. McEliece. *The Theory of Information and Coding*, Cambridge University Press, 2002.

[124] F. McWilliams, J. Sloane. *The Theory of Error Correcting Codes*, North-Holland, 1977.

[125] K. Meinke, J.V. Tucker. *Universal Algebra*, Handbook of Logic in Computer Science (S. Abramsky, D. Gabbay, T.S.E. Maibaum, eds.), vol. 1, Oxford University Press, 1993, 189–411.

[126] R. McEliece. *The Theory of Information and Coding*, Addison-Wesley, 1977.

[127] B. McMillan. *Two Inegalities Implied by Unique Decipherability*, IRE Transactions on Information Theory IT-2, 1956, 114–116.

[128] E.J. McShane. *Partial Orderings and Moore-Smith Limits*, Americam Mathematical Monthly 59, 1952, 1–11.

[129] E.J. McShane. *Order-Preserving Maps and Integration Processes*, Annals of Mathematical Studies 31, Princeton, 1953.

[130] M. Mignotte. , 1983.

[131] D. Mirimanoff. *Les antinomies de Russell et de Burali-Forti et le probleme fondamental de la theorie des ensembles*, L'Enseignement Mathematique 19, 1917, 37–52.

[132] J.C. Mitchell. *Foundations of Programming Languages*, The MIT Press, 1996.

[133] E.H. Moore, H.L. Smith. *A General Theory of Limits*, American Journal of Mathematics 44, 1922, 102–121.

[134] A. Mostowski. *Über die Unabhängigkeit des Wohlordnungssatzes vom Ordnungsprinzip*, Fundamenta Mathematicae 32, 1939, 201–252.

[135] J. von Neumann. *Zur Einführung der transfiniten Zahlen*, Acta Litterarum ac Scientiarum Regiae Universitatis Hungaricae Francisco-Josephinae, Sectio Scientiarum Mathematicarum 1, 1923, 199-208.

[136] J. von Neumann. *Eine Axiomatisierung der Mengenlehre*, Journal für Mathematik 154, 1925, 219–240 (corrections in Journal für Mathematik 155, 1926, 128).

[137] J. von Neumann. *On Regular Rings*, Proceedings of the National Academy of Sciences of the United States of America 22, 1936, 707–713.

[138] J. von Neumann. *Die Axiomatisierung der Mengenlehre*, Mathematische Zeitschrift 27, 1928, 669–752.

[139] H.R. Nielson, F. Nielson, Ch. Hankin. *Principles of Program Analysis*, Springer-Verlag, 1998.

[140] H.R. Nielson, F. Nielson. *Semantics with Applications: A Formal Introduction*, Wiley Professional Computing, 1992 (ediţie revizuită în iulie 1999, disponibilă on-line din pagina autorilor).

[141] NIST 185. *Digital Signature Standard*, National Institute of Standards and Technology, Federal Information Processing Standards Publication 185, U.S. Department of Commerce, 1994.

[142] G. Peano. *Démonstration de l'intégrabilité des équations différentielles ordinaires*, Mathematische Annalen 37, 1890, 182–228.

[143] G. Peano. *Formulaire de Mathématiques*, Torino, 1895 (a 5-a ediţie sub denumirea *Formulario Mathematico*, Torino, 1905-1908).

[144] A. Precupanu. *Bazele analizei matematice*, Editura Polirom, Iaşi, 1998.

[145] M. Petrich. *Introduction to Semigroups*, Merill, Columbus, Ohio, 1973.

[146] E. Post. *A Variant of a Recursively Unsolvable Problem*, Bulletin of the American Mathematical Society 52, 1946, 264–268.

[147] W.V. Quine. *Mathematical Logic*, New York, 1940.

[148] D. Rees. *On Semi-groups*, Proc. Cambridge Phil. Soc. 36, 1940, 387–400.

[149] H. Reichel. *Initial Computability, Algebraic Specifications, and Partial Algebras*, Oxford University Press, 1987.

[150] L. Rieger. *A Contribution to Gödel's Axiomatic Set Theory I*, Czechoslovak Mathematical Journal 7 (82), 1957, 323–357.

[151] R.L. Rivest, A. Shamir, L.M. Adleman. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, Communications of the ACM 2, 1978, 120–126.

[152] S. Roman. *Coding and Information Theory*, Springer-Verlag, 1992.

[153] B.K. Rosen. *Program Equivalence and Context-Free Grammars*, Journal of Computer and System Science, 1973

[154] K.H. Rosen. *Elementary Number Theory and Its Applications*, Addison Wesley Longman (a 4-a ediţie), 2000.

[155] J.B. Rosser. *The n-th Prime is Greater Than $n \ln n$*, Proceedings of London Mathematical Society 2, 1939, 21–44.

[156] J.B. Rosser, L. Schoenfeld. *Approximate Formulas for Some Funcţions of Prime Numbers*, Illinois Journal of Mathematics 6, 1962, 64–89.

[157] B. Russell. *The Priciples of Mathematics*, London, 1903 (a 2-a ediţie, London, 1937).

[158] A. Salomaa. *Jewels of Formal Language Theory*, Computer Science Press, 1981.

[159] D. Salomon. *Data Compression. The Complete Reference*, Springer-Verlag (a 3-a ediţie), 1998.

[160] A.A. Sardinas, P.W. Patterson. *A Necessary and Sufficient Condition for the Unique Decomposition of Coded Messages*, IRE Internat. Conv. Rec. 8, 1953, 104–108.

[161] K. Sayood. *Introduction to Data Compression*, Morgan Kaufmann Publishers (a 2-a ediţie), 2000.

[162] B. Schneier. *Applied Cryptography*, John Wiley & Sons, 1996.

[163] M. Schönfinkel. *Über die Bausteine der mathematischen Logik*, Mathematische Annalen, vol. 92, 1924.

[164] E. Schröder. *Vorlesungen über die Algebra und Logik*, 1890 (1891, 1895, 1905).

[165] M.P. Schützenberger. *Une théorie algébrique du codage*, Séminaire Dubreil-Pisot, Expose no. 15, 1955-1956).

[166] D. Shanks. *Class Number, a Theory of Factorization, and Genera*, Symposium of Pure Mathematics, 1972.

[167] C.E. Shannon. *A Mathematical theory of Communication*, Bell Syst. Tech. J. 27, 1948, 379–423 şi 623–656.

[168] H.N. Shapiro. *Introduction to the Theory of Numbers*, John Wiley & Sons, 1983.

[169] W. Sierpiński. *Elementary Theory of Numbers*, Państwowe Wydawnictwo Naukowe, 1964

[170] T. Skolem. *Einige Bermerkungen zur axiomatischen Begründung der Mengenlehre*, Wiss. Vorträge gehalten auf dem 5 Kongress der scandinav. Mathematiker in Helsingfors, 1922, 217–232.

[171] D. Slepian. *Some Further Theory on Group Codes*, Bell System Tech. Journal 39, 1960, 1219–1252.

[172] N.J.A. Sloane. *Recent Bounds for Codes, Sphere Packing and Related Problems Obtained by Linear Programming and Other Methods*, Contemporary Mathematics 9, 1982, 153–85.

[173] E. Steinitz. *Bedingt konvergente Reihen und konvexe Systeme*, J. Reine Angew. Math., 143, 1913, 128–175.

[174] J.E. Stoy. *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory*, The MIT Press, 1977.

[175] P. Suppes. *Axiomatic Set Theory*, Dover, New York, 1972.

[176] A.K. Suschkewitsch. *Über die endlichen Gruppen ohne das Gesets der eindeutigen Umkehrbarkeit*, Mathematische Annalen 99, 1928, 30–50.

[177] A.K. Suschkewitsch. *Theory of Generalized Groups*, Kharkov, 1937 (în ruseşte).

[178] A. Tarski. *Sur quelques théorèmes qui équivalent à l'axiome du choix*, Fundamenta Mathematicae 5, 1924, 147–154.

[179] A. Tarski. *General Principles of Induction and Recursion. The Notion of Rank in Axiomatic Set Theory and Some of its Applications*, Bulletin of the American Mathematical Society 61, 1955, 442–443.

[180] A. Thue. *Über unendliche Zeichenreihen*, Videnskapsselskapets Skrifter, I. Mat.-naturv. Klasse, Kristiania, 1906, 1–22.

[181] A. Thue. *Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen*, Videnskapsselskapets Skrifter, I. Mat.-naturv. Klasse, Kristiania, 1912, 1–67.

[182] F.L. Ţiplea. *Introducere în teoria mulţimilor*, Editura Universităţii "Al.I.Cuza", 1998.

[183] F.L. Ţiplea, E. Mäkinen, C. Enea. *SE-Systems, Timing Mechanisms, and Time-Varying Codes*, International Journal of Computer Mathematics 79(9), 2002.

[184] F.L. Ţiplea, E. Mäkinen, D. Trincă, C. Enea. *Characterization Results for Time-Varying Code*, Fundamenta Informaticae 52, 2003, 1–13.

[185] F.L. Ţiplea, S. Iftene, C. Hriţcu, I. Goriac, R. Gordan, E. Erbiceanu: *MpNT: A Multi-Precision Number Theory Package. Number Theoretic Algorithms (I)*, Technical Report 03-02, Faculty of Computer Science, "Al.I.Cuza" University of Iasi, Romania, 2003, 98 pages.

[186] F.L. Ţiplea, C. Enea. *Abstractions of Abstract Data Types*, Acta Informatica, 2005 (to appear).

[187] F.L. Ţiplea. *Structuri algebrice de bază în informatică*, 2005 (în curs de apariţie).

[188] F.L. Ţiplea. *Criptografie*, (în pregătire).

[189] J.W. Tukey. *Convergence and Uniformity in Topology*, Annales of Mathematical Studies 2, Priceton, 1940.

[190] R.L. Vaught. *Set Theory. An Introduction*, Birkhäuser (a 2-a ediţie), 1995.

[191] J. Vuillemin. *Correct and Optimal Implementations of Recursion in a Simple Programming Language*, Proceedings of the 5th Annual ACM Symposium on Theory of Computing, 1973, 224–.

[192] H. Weyl. *Raum, Zeit und Materie*, Berlin, 1923.

[193] A.N. Whitehead. *A Treatise on Universal Algebra*, Cambridge University Press, 1898.

[194] A.N. Whitehead, B. Russell. *Principia Mathematica I,II,III*, Cambridge, 1910, 1912, 1913.

[195] N. Wiener. *A Simplification of the Logic of Relations*, Proceedings of the Cambridge Philosophical Society 17, 1914, 387-390.

[196] M. Wiener. *Cryptanalysis of short RSA secret exponent*, IEEE Transactions on Information Theory 36, 1990, 553–558.

[197] H. Wussing. *Genesis of the Abstract Group Concept*, MIT Press, 1984.

[198] F.S. Wolk. *Dedekind Completeness and a Fixed-Point Theorem*, Canadian Journal of Mathematics, vol. IX, no. 3, 1957, 400–405.

[199] E. Zermelo. *Beweis, das jede Menge wohlgeordnet werden kann*, Mathematische Annalen 59, 1904, 514–516.

[200] E. Zermelo. *Untersuchung über die Grundlagen der Mengenlehre (I)*, Mathematische Annalen 65, 1908, 261–281.

[201] E. Zermelo. *Über Grenzzahlen und Mengenbereiche*, Fundamenta Mathematicae 16, 1930, 29–47.

[202] E. Zermelo. *Grundlagen einer allgemeinen Theorie der mathematishe Satzsysteme*, Fundamenta Mathematicae 25, 1935, 136–146.

[203] M. Zorn. *A Remark on Method in Transfinite Algebra*, Bulletin of the American Mathematical Society 41, 1935, 667–670.

[204] N. Yoneda. *On the Homology Theory of Modules*, Journal of the Faculty of Sciences of Tokyo I-7, 1954, 193–227.