

Ferucio Laurențiu Țiplea este profesor universitar doctor la Facultatea de Informatică a Universității „Alexandru Ioan Cuza” din Iași. Domeniile de cercetare în care activează sunt: modelare și verificare formală (bazată pe elemente de algebră, logică și rețele Petri), complexitatea calculului, criptografie și securitatea informației. În fiecare din aceste domenii, a publicat articole de cercetare în reviste de circulație internațională și a comunicat lucrări la conferințe internaționale. A conferențiat ca profesor invitat la universități din Germania, Franța, Ungaria, Statele Unite ale Americii, Japonia și Grecia și a beneficiat de granturi și burse oferite de DAAD, Academia Germană, Monbusho și Fulbright. Din ianuarie 2004 până în aprilie 2006, a activat ca Visiting Professor la University of Central Florida, School of Electrical Engineering and Computer Science, unde a predat cursul COT3100H: Honors Introduction to Discrete Structures, bazat pe prima ediție a prezentei cărți. Pentru mai multe detalii pe linie didactică și de cercetare, se poate consulta <https://profs.info.uaic.ro/~ftiplea>.

Ferucio Laurențiu Țiplea

Fundamentele algebrice ale informaticii

Referenți științifici:

Prof. univ. dr. **Mirela Ștefănescu**, Universitatea Ovidius, Constanța
Prof. univ. dr. **Virgil Emil Căzănescu**, Universitatea din București

Ediția a doua revăzută și adăugită

Coperta: Manuela Oboroceanu
Redactor: dr. Marius-Nicușor Grigore

ISBN: 978-606-714-630-1

© Editura Universității „Alexandru Ioan Cuza” din Iași, 2021
700109 – Iași, str. Pinului, nr. 1A, tel./fax: (0232) 314947
<http://www.editura.uaic.ro> e-mail: editura@uaic.ro

Director: prof. univ. dr. **Constantin Dram**


EDITURA UNIVERSITĂȚII „ALEXANDRU IOAN CUZA” DIN IAȘI
2021

Cuprins

Descrierea CIP a Bibliotecii Naționale a României

ȚIPLEA, FERUCIO LAURENTIU

Fundamentele algebrice ale informaticii / Ferucio Laurențiu Țiplea. - Ed. a 2-a,
rev. și adăug.. - Iași : Editura Universității „Al. I. Cuza”, 2021

Conține bibliografie. - Index

ISBN 978-606-714-630-1

004

Cuprins	5
Prefață	11
Prefață la ediția a II-a	13
1 Concepte fundamentale	15
1.1 Multimi	15
1.1.1 Ce este o mulțime?	15
1.1.2 Operații cu mulțimi	31
1.1.3 Axioma infinitului. Numere naturale	34
1.1.3.1 Axioma infinitului	34
1.1.3.2 Numere naturale	36
1.1.3.3 Ordonare și inducție	38
1.1.3.4 Mulțimi finite și infinite	43
1.1.3.5 Recursie	43
1.2 Relații și funcții	49
1.2.1 Relații	49
1.2.2 Relații de echivalență	58
1.2.3 Funcții și operații	62
1.2.4 Familii indexate de mulțimi	73
1.2.5 Axioma alegerii	77
1.2.6 Relații de ordine	83
1.3 Închideri	84
1.3.1 Închideri. Inducție structurală	84
1.3.2 Închideri ale unei relații binare	87
1.3.3 Definiții inductive/recursive	89
1.4 Sisteme relationale și algebrel universale	93
1.4.1 Sisteme relationale	93
1.4.2 Mulțimi parțial ordonate	96
1.4.2.1 Concepte de bază	96
1.4.2.2 Dualitate	100
1.4.2.3 Proprietăți de bază ale supremumului și infimumului .	100
1.4.2.4 Construcții de mpo	104

1.4.3	Latici	110
1.4.3.1	Laticea ca mulțime parțial ordonată	110
1.4.3.2	Laticea ca structură algebrică	113
1.4.3.3	Latici distributive și modulare	117
1.4.4	Algebre universale dintr-un punct de vedere elementar	124
1.4.4.1	Algebre	125
1.4.4.2	Subalgebrelor. Ordin	127
1.4.4.3	Homomorfisme și congruențe	130
1.4.5	Algebrelor Booleene	133
1.5	Numere ordinale și cardinale	139
1.5.1	Mulțimi bine ordonate	139
1.5.2	Numere ordinale	141
1.5.3	Axioma înlocuirii. Inducție și recursie pe ordinali	145
1.5.4	Principiul bunei ordonări și alte propoziții echivalente Axiomei alegerii	152
1.5.5	Numere cardinale	153
2	Elemente de teoria numerelor	157
2.1	Divizibilitate. Numere prime	157
2.2	Cel mai mare divizor comun	164
2.3	Congruențe	172
2.4	Funcția lui Euler	175
2.5	Ecuații congruențiale	178
2.6	Teorema chineză a resturilor	181
2.7	Reziduozație pătratică	186
2.7.1	Congruențe pătratice	186
2.7.2	Reziduuri pătratice și simbolul Legendre	188
2.7.3	Simbolul Jacobi	194
2.8	Complexitatea operațiilor	196
2.8.1	Ordine de mărime	196
2.8.2	Timpul de execuție al unui algoritm	199
3	Semigrupuri și monoizi	203
3.1	Definiții și exemple	203
3.2	Relațiile lui Green	207
3.3	Clase remarcabile de semigrupuri și monoizi	210
3.3.1	Monoizi de cuvinte	210
3.3.2	Semigrupuri de transformări	215
3.3.3	Semigrupuri și monoizi ciclici	218
3.3.4	Semigrupuri regulate și inverse	221
3.4	Semigrupuri și monoizi liberi	224
3.4.1	Definiții. Exemple. Proprietăți de bază	224
3.4.2	Rezultate de caracterizare	228
3.4.3	Submonoizi liberi	233

3.5	Aplicații: coduri de lungime variabilă	235
3.5.1	Definiții. Exemple. Proprietăți de bază	236
3.5.2	Rezultate de caracterizare	241
3.5.3	Măsura unui cod	247
3.5.4	Coduri Huffman	253
3.5.5	Entropie. Limita compresiei	261
4	Grupuri	269
4.1	Definiții. Exemple. Proprietăți de bază	269
4.2	Subgrupuri. Teorema lui Lagrange	274
4.3	Subgrupuri normale	277
4.4	Grupuri ciclice	281
4.5	Grupul \mathbb{Z}_m^*	284
4.6	Problema logaritmului discret	290
4.7	Aplicații: criptografie cu chei publice	292
4.7.1	Introducere în criptografie	292
4.7.2	Criptosistemul RSA	298
4.7.2.1	Descrierea criptosistemului	299
4.7.2.2	Criptanaliza RSA	302
4.7.3	Semnături digitale	304
4.7.3.1	Introducere	304
4.7.3.2	Semnătura ElGamal	307
4.7.3.3	Semnătura DSS	311
5	Inele și corpuși	315
5.1	Definiții. Exemple. Proprietăți de bază	315
5.2	Homomorfisme, subinle și ideale	322
5.3	Caracteristica unui inel	327
5.4	Inele de polinoame	331
5.4.1	Polinoame. Proprietăți de bază	332
5.4.2	Extensiile, elemente algebrice, descompuneri	336
5.5	Corpuși finite	341
5.6	Aplicații: criptosistemul Rijndael	343
6	Spații vectoriale	351
6.1	Definiții. Exemple. Proprietăți de bază	351
6.2	Bază și dimensiune	356
6.3	Funcții liniare	361
6.4	Sume directe și spații vectoriale către	363
6.5	Produs scalar. Ortogonalitate. Spațiu dual	366
6.6	Aplicații: coduri detectoare și corectoare de erori	368
6.6.1	Introducere	369
6.6.1.1	Transmiterea informației prin canale cu zgomot	369
6.6.1.2	Detectarea și corectarea erorilor	372

6.6.1.3	Determinarea cuvântului cod transmis	374
6.6.1.4	Coduri detectoare și corectoare de erori	377
6.6.1.5	Problema fundamentală a teoriei codurilor bloc	379
6.6.2	Coduri liniare	382
6.6.2.1	Definiții. Exemple. Proprietăți de bază	382
6.6.2.2	Decodificare Slepian și sindrom	387
7	Teoria mulțimilor parțial ordonate	393
7.1	Completitudine	393
7.1.1	Completitudine prin submulțimi. Latici complete	393
7.1.2	Completitudine prin mulțimi dirijate	397
7.1.3	Completitudine prin lanțuri	400
7.2	Teoria de punct fix a mulțimilor parțial ordonate	408
7.2.1	Funcții continue	408
7.2.2	Puncte fixe	416
7.2.3	Inducție de punct fix	424
7.3	Aplicații: semantica și analiza programelor	428
7.3.1	Semantica programelor	428
7.3.1.1	λ -notație	428
7.3.1.2	Programe recursive	434
7.3.1.3	Semantica denotațională a programelor recursive	437
7.3.1.4	Programe structurate	441
7.3.1.5	Semantica denotațională a programelor structurate	443
7.3.2	Analiza și verificarea programelor	449
7.3.2.1	Analiza programelor	450
7.3.2.2	Verificarea programelor	455
8	Algebre universale	461
8.1	Structuri sortate	462
8.2	Signaturi și algebre	466
8.3	Subalgebre. Inducție structurală	474
8.4	Congruențe și algebre cât	477
8.4.1	Definiții. Exemple. Proprietăți de bază	477
8.4.2	Congruențe principale	484
8.5	Homomorfisme de algebre	485
8.5.1	Definiții. Exemple. Proprietăți de bază	485
8.5.2	Structura algebraică a mulțimilor $End(\mathbf{A})$ și $Aut(\mathbf{A})$	490
8.5.3	Homomorfisme și congruențe	492
8.5.4	Teoreme de izomorfism	494
8.6	Produse de algebre	496
8.6.1	Produse directe de algebre	496
8.6.2	Algebre decompozabile	500
8.6.3	Produse subdirecte de algebre	504

8.7	Algebre libere	505
8.7.1	Algebre de termi	505
8.7.2	Algebre libere. Definiții și proprietăți de bază	511
8.7.3	Teorema lui Birkhoff de existență a algebrelor libere	515
8.8	Logică ecuațională	519
8.8.1	Ecuații și modele	519
8.8.2	Deducție ecuațională	522
8.8.3	Axiomatizare	530
8.9	Aplicații: semantica limbajelor de programare și specificare algebraică a tipurilor abstracte de date	532
8.9.1	Semantica limbajelor de programare	532
8.9.1.1	Signatura asociată unei gramatici independente de context	532
8.9.1.2	Semantica programelor structurate	541
8.9.1.3	Traduceri de programe și compilare	544
8.9.2	Specificarea algebraică a tipurilor abstracte de date	552
8.9.2.1	Introducere	552
8.9.2.2	Specificații inițiale ale tipurilor abstracte de date	555
8.9.2.3	Tratarea exceptiilor	564
Bibliografie		573
Lista figurilor		587
Index		591

Prefață

Informatica, prin natura ei, utilizează din plin elemente de algebră într-o multitudine de domenii: criptografie, coduri, semantica limbajelor de programare, tipuri abstrakte de date, teoria recursiei, sisteme de scriere, limbaje formale etc. O bună înțelegere și manipulare a conceptelor din aceste domenii cere o bună înțelegere a multor elemente de bază din algebră. Semantica limbajelor de programare face apel la concepte variate de completitudine în mulțimi parțial ordonate, teoria tipurilor abstracte de date face apel la algebrel universale, tratările moderne în teoria limbajelor formale fac apel la semi-inele și serii de puteri, teoria corpurilor finite este indispensabilă codurilor bloc, iar lista aceasta poate continua destul de mult. Din acest punct de vedere, scrierea unei cărți de elemente de algebră pentru informaticieni nu este un lucru simplu. Există multe aspecte care trebuie luate în calcul, aspecte ce au condus autorul la următoarele trei principii naturale:

- *Principiul selecției materialului.* Ce elemente de algebră constituie o bază indispensabilă informaticienilor? Cum prezentăm aceste elemente de algebră astfel încât ele să fie cât mai la îndemâna lor? Prezentarea acestora trebuie să păstreze rigoarea matematică și, în același timp, să le facă ușor de asimilat;
- *Principiul exemplificării.* Elementele ce urmează a fi prezentate au nevoie de exemplificare consistentă. Trebuie arătat clar la ce sunt necesare acestea și cum se utilizează. Nu este suficient să se spună că teoria spațiilor vectoriale este importantă pentru construcția codurilor detectoare și corectoare de erori; este cu totul altceva să arătăm cum se definesc și construiesc astfel de coduri utilizând aparatul spațiilor vectoriale;
- *Principiul echilibrului.* Păstrarea unui echilibru între importanța teoretică și importanța practică a elementelor selectate constituie un alt aspect important. Un astfel de echilibru nu poate fi universal. De exemplu, teoria avansată a domeniilor semantice necesită elemente de algebră destul de complexe, accesibile unei clase destul de reduse de cititori (cercetătorilor în domeniu) și cu un grad scăzut de aplicabilitate practică. La polul opus, criptanaliza face apel la elemente destul de adânci de teoria numerelor, accesibile unei clase reduse de cititori, dar cu importanță practică foarte mare. Cum păstrăm atunci un echilibru între dificultatea conceptelor abordate și importanța teoretică și practică a acestora?

Acestea sunt cele trei principii care au ghidat și ghidează autorul în predarea cursului de *Fundamentele algebrice ale informaticii* la Facultatea de Informatică a Universității “Alexandru Ioan Cuza” din Iași, începând cu 1994. Acestea sunt cele trei principii care ne-au ghidat în redactarea materialului de față: o selecție corespunzătoare a elementelor de algebră necesare informaticienilor, exemplificări consistente și păstrarea unui echilibru între importanța teoretică și cea practică.

Exceptând primul capitol, ce prezintă concepte fundamentale necesare tuturor celor ce abordează domenii în care au de manipulat concepte matematice de bază (funcții, relații, mulțimi parțial ordonate etc.), toate celelalte capitole conțin aplicații majore în informatică: teoria codurilor de lungime variabilă, compresia datelor, teoria codurilor detectoare și corectoare de erori, criptografie, semantica limbajelor de programare și tipuri abstracte de date. Pe lângă adăugarea de aplicații consistente care să motiveze pe deplin concepțele introduse, s-a acordat atenție deosebită modului de introducere a acestora. Acolo unde a fost posibil s-au încercat abordări unitare și discuții ample (a se vedea, de exemplu, discuția asupra tipurilor de completitudine din Capitolul 7). Sperăm ca, prin conținutul ei, cartea să motiveze și să ajute cititorul informatician.

Dorim să mulțumim profesorei Mirela Ștefănescu de la Universitatea Ovidius din Constanța și profesorului Virgil Emil Căzănescu de la Universitatea din București pentru timpul dedicat lecturii acestui material și pentru remarcile și sugestiile oferite cu multă amabilitate. Acestea au condus la îmbunătățirea prezentării materialului în fața căruia se află cititorul. În particular, capitolul 8 al cărții a fost rescris complet ca urmare a unor sugestii ale profesorului Virgil Emil Căzănescu (detalii sunt prezentate în introducerea premergătoare capitolului).

Colegii Sorin Iftene, Cătălin Bîrjoveanu, Ioana Olga (Leahu) și Codruț Matei, prin predarea de seminarii pe o astfel de tematică, au făcut observații utile de-a lungul anilor, iar Ana-Maria Minea a lecturat întregul material cu atenție deosebită. Tuturor acestora le suntem îndatorați.

Forma finală a acestei cărți a fost pregătită în timp ce autorul a ocupat poziția de Visiting Professor la School of Electrical Engineering and Computer Science, University of Central Florida (Orlando, USA), în perioada ianuarie 2004 - mai 2006. Materialul a fost predat studenților americanii în cadrul cursului *COT3100H: Honors Introduction to Discrete Structures* în primăvara anului 2006. Mulțumim pe această cale colegilor de la University of Central Florida pentru oferirea posibilității de a predă acest material, pentru *feedback*-ul lor și pentru condițiile excelente de lucru oferite în această perioadă.

Suportul familiei a fost de neprețuit pe întreg parcursul scrierii acestei cărți.

Prefață la ediția a II-a

Fundamentele algebrice ale informaticii reprezintă suportul cursului cu același nume predat de autor la Facultatea de Informatică a Universității “Alexandru Ioan Cuza” din Iași, încă din anul 1994. Dacă până în 2006, când a fost publicată prima ediție a acestei cărți de către Editura Polirom, cursul *Fundamentele algebrice ale informaticii* a suferit modificări minore dar continue de la an la an, perioada 2006–2020 a arătat o stabilitate mult mai mare a conținutului acestuia. Explicația acestui fenomen este că se poate de naturală și pleacă de la cerințele generale ale studentului informatician relativ la matematica discretă. Astfel, prezenta carte oferă un suport matematic riguros pentru domenii precum criptografia, teoria codurilor de lungime variabilă, teoria codurilor detectoare și corectoare de erori, limbi formale, semantica limbajelor de programare etc. Specializarea pe oricare din aceste domenii necesită informații suplimentare specifice.

Ca urmare a celor spuse mai sus, prezenta ediție a cărții aduce modificări minore ediției anterioare, concretizate prin corectarea unor mici erori. Suplimentar, ea propune o nouă secțiune capitolului de teoria numerelor, referitoare la reziduozitate pătratică. Sperăm ca prin această nouă ediție să venim și mai mult în întâmpinarea cerințelor studentului informatician.

Dorim să mulțumim tuturor celor ce au făcut comentarii asupra materialului, colegilor care au predat seminarii la *Fundamentele algebrice ale informaticii*, precum și Editurii Universității “Alexandru Ioan Cuza” din Iași pentru promptitudinea cu care a sprijinit re-editarea acestui material.

Iași, 15 martie 2021
Ferucio Laurențiu Țiplea

Capitolul 1

Concepțe fundamentale

1.1. Mulțimi

În această secțiune vom prezenta câteva concepte de bază de teoria mulțimilor, concepte ce vor fi utilizate pe parcursul acestei lucrări. Pentru detalii, cititorul este îndrumat către [93, 102, 207].

1.1.1. Ce este o mulțime?

Conceptul Cantorian de mulțime. Conceptul de mulțime stă la temelia matematicii moderne, fiind un concept larg utilizat în orice domeniu. Teoria mulțimilor (studiul abstract al mulțimilor), aşa cum o utilizăm astăzi, a fost inițiată de Georg Cantor în ultimul sfert al secolului al XIX-lea [26]. Abordarea lui a condus însă la *contradicții* (numite și *paradoxuri*), remediu “aparent” al acestora fiind abordarea axiomatică.

Conform definiției date de Cantor, prin *mulțime* înțelegem

orice colecție de obiecte distincte și bine definite ale intuiției și ale gândirii noastre, considerată ca un tot (întreg, ca o unitate).

Noțiunea de mulțime trebuie considerată ca un concept primitiv, suficient de bine înțeles intuitiv, care nu este precis definit, dar care poate fi utilizat în definirea altor concepte particulare. Așadar, motivați de “definiția” lui Cantor, să considerăm că o mulțime este o colecție de “obiecte” numite *elementele* mulțimii și să presupunem că există măcar o mulțime.

Dacă A este o mulțime și a este un obiect (arbitrар), atunci a poate fi sau nu în mulțimea A . În primul caz vom folosi exprimarea *a este element al mulțimii A* sau

a aparține mulțimii A sau a este conținut în A sau A conține a și vom nota $a \in A$; vom scrie $a \notin A$, dacă a nu este element al mulțimii A ¹.

Menționăm explicit că nu considerăm noțiunea de obiect ca o noțiune primară a teoriei mulțimilor. Așadar, avem libertatea de a gândi ceea ce este un obiect. De exemplu, putem gândi anumite mulțimi ca fiind obiecte componente ale altrei mulțimi. Fie R mulțimea

$$R = \{x \mid x \text{ este mulțime și } x \notin x\}.$$

Conform definiției lui Cantor, R este mulțime. Mulțimea numerelor naturale este element al mulțimii R . Este mai dificil de găsit un exemplu de mulțime ce nu este element al mulțimii R , dar aceasta nu are nici o importanță relativ la statutul de mulțime a lui R . Însă, constatăm că are loc

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

ceea ce constituie un paradox. Acesta este așa-numitul *paradox al lui Russell* [221].

Este natural să ne întrebăm atunci care este cauza ce conduce la acest paradox. Constatăm că definiția mulțimii R este bazată pe următorul principiu numit **Axioma abstracției sau a construcției de mulțimi**, principiu ce a fost introdus de Gottlob Frege în 1893 [62]:

Axioma abstracției. Date o proprietate ce poate fi sau nu îndeplinită de obiecte, există o mulțime ce constă exact din obiectele ce satisfac această proprietate.

În cazul mulțimii R , proprietatea este

$$P(x) : x \text{ este mulțime și } x \notin x.$$

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

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

obținemmulțimea $A = \{1, 3, 5, 7, 9\}$. Diferența dintre construcțiile celor două mulțimi constă în aceea că mulțimea A este obținută prin selectarea obiectelor dintr-o mulțime dată (cea a numerelor naturale) prin intermediul unei proprietăți, ceea ce nu se întâmplă în cazul mulțimii R .

Descoperirea de paradoxuri în teoria cantoriană a mulțimilor a avut efecte dintre cele mai neplăcute pentru mulți matematicieni care și-au bazat studiile și 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,

¹Simbolul " \in " a fost folosit pentru prima dată de matematicianul Giuseppe Peano [164], el fiind de fapt o variație grafică a primei litere a cuvântului grecesc " $\epsilon\sigma\tau\nu$ " ce înseamnă "este".

a fost nevoie să opreasă pentru o perioadă publicarea acestora². Mai tragic a fost însă cu lucrarea în două volume a lui Gottlob Frege, despre bazele aritmeticii, care tocmai fusese terminată [62] și care utilizează Axioma abstracției. În cel de-al doilea volum, când Frege luase deja cunoștință de paradoxul lui Russell, acesta a inserat o anexă din care prezentăm mai jos un fragment (traducerea din original este după [67], pag. 234):

"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 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. Într-o perioadă s-a crezut că apariția de paradoxuri poate distruga teoria cantoriană a mulțimilor 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 așezarea ei pe baze axiomatice, logistice sau intuiționiste. Dintre sistemele axiomatice propuse, *sistemul Zermelo-Fraenkel cu Axioma alegerii*, abreviat ZFC, este astăzi unul dintre cele mai utilizate.

Restul acestei secțiuni va fi dedicat unei prezentări extrem de succinte a sistemului ZFC. Pentru detalii cititorul este îndrumat către [207] (atrăgând atenția asupra faptului că sistemul ZFC, aşa cum este prezentat în [207], pornește de la premisa că universul de discurs al variabilelor poate conține obiecte ce nu sunt mulțimi, pe când abordarea pe care o vom prezenta în cele ce urmează pleacă de la premisa că universul de discurs al variabilelor conține numai mulțimi). O discuție detaliată asupra diferenței dintre aceste abordări poate fi găsită în [207]).

Chiar dacă prezentarea pe care o vom face acestui sistem va fi extrem de succintă, încurajăm cititorul să citească secțiunea până la capăt deoarece credem că ea este suficient de edificatoare relativ la modul și la necesitatea introducerii fiecărei axiome. Ce trebuie să aibă cititorul în minte atunci când dorește să asimileze un sistem axonomic al teoriei mulțimilor? Ca un minim necesar, spunem că orice sistem axonomic al teoriei mulțimilor trebuie să asigure:

²În prefața la a treia ediție a lucrării [42], sau în [44], pag. 449.

³"a paradise created by Cantor which nobody shall ever expel us" (conform cu [58], pag. 240).

- existența a cel puțin unei mulțimi;
- metode de construcție de noi mulțimi prin operații de intersecție, diferență și reuniune, pornind de la mulțimi existente;
- definiția perechii ordonate a două elemente arbitrar (acest concept stă la baza definiției produsului cartezian și a relațiilor, deci a relațiilor de echivalență, de ordine și a funcțiilor);
- existența mulțimilor infinite;
- construcția numerelor naturale (care stau apoi la baza construcției numerelor întregi, raționale și reale).

Sistemul axiomatic trebuie să fie consistent. Evident, există și alte cerințe, dar ceea ce am menționat mai sus constituie un minim necesar atunci când se dorește elaborarea unui astfel de sistem.

Pentru a putea prezenta axiomele și construcțiile aferente într-o manieră precisă, avem nevoie de un *limbaj formal* care, în cadrul sistemului ZFC, este logica cu predicate de ordinul întâi, la care se adaugă două predicate noi, *predicatul de egalitate* și *predicatul binar de apartenență*. Menționăm încă de la început că apartenența este un predicat primitiv, care nu se definește. Obiectele au proprietatea de a apartine sau nu mulțimilor, proprietate ce este primitivă.

Formulele atomice sunt de forma

$$x \in y \text{ și } x = y,$$

pe baza cărora se construiesc noi *formule* prin intermediul operatorilor logici clasici și a cuantificatorilor:

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

(parantezele vor fi eliminate ori de câte ori nu vor putea apărea ambiguități). Vom adopta notația $\varphi(x_1, \dots, x_n)$ pentru a specifica că variabilele libere ale formulei φ sunt printre variabilele x_1, \dots, 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 în care vor lua valori variabilele, numit și *universul de discurs*. Din punct de vedere al teoriei mulțimilor, proprietatea fundamentală prin care se poate face distincție între obiectele universului de discurs este proprietatea de apartenență: un obiect poate *conține* alte obiecte sau nici unul. Dacă un obiect conține obiecte atunci el va fi referit ca *mulțime*; altfel, ca *obiect individual*. Termenul de “element” va însemna pentru noi “obiect al unei mulțimi”. Este natural să se presupună că fiecare obiect al universului de discurs este element al unei mulțimi (măcar al mulțimii formate doar din obiectul în cauză). Întrebarea fundamentală ce se pune acum este următoarea:

“câtă” obiecte individuale și “câtă” mulțimi considerăm în univers? Trebuie să admitem că existența a cel puțin unui obiect este cerută atât din rațiuni filozofice cât și practice; ea este necesară *fondării* universului. Pe de altă parte, anticipând asupra intersecției mulțimilor, constatăm că avem nevoie de un obiect care să reprezinte rezultatul intersecției a două mulțimi fără elemente comune. Este natural ca acest obiect să fie fără elemente, deci obiect individual, și să nu depindă de mulțimile sursă. Din considerente tehnice este important ca și acest obiect să fie referit ca mulțime; ca urmare, prin mulțime vom înțelege acele obiecte ce conțin obiecte sau acest obiect particular bine precizat (uzual, un astfel de obiect individual este numit *mulțimea vidă*). Vrem să accentuăm însă că, în timp ce existența a cel puțin unui obiect individual este cerută din rațiuni filozofice și practice, referirea la un obiect individual ca fiind mulțimea vidă este numai din rațiuni de conveniență și simplitate. Acum avem de analizat următoarele două variante:

- (1) universul de discurs conține numai mulțimi și, în plus, mulțimea vidă ca singur obiect individual [58, 102, 124, 215];
- (2) universul de discurs conține și alte obiecte individuale pe lângă cel desemnat a fi mulțimea vidă [153, 16, 199, 111, 207].

Majoritatea sistemelor ZFC în varianta (1) asigură fondarea universului de discurs și existența mulțimii vide

- fie prin intermediul unei axiome, ușor numită *Axioma existenței* sau a *mulțimii vide*, ce postulează existența unei mulțimi fără nici un obiect (de exemplu, [93]),
- fie presupunând tacit că universul de discurs conține cel puțin o mulțime (universul de discurs al variabilelor logicii de ordinul întâi trebuie să conțină cel puțin un obiect), de la care se deduce, pe baza Axiomei separării, existența unei mulțimi ce nu conține nici un obiect (de exemplu, [124, 60]).

Ceea ce a ieșit în evidență este că, pentru scopuri matematice, postularea existenței doar a unui singur obiect individual (mulțimea vidă) este suficientă⁴. Aceasta este varianta pe care o vom adopta și noi (adică, (1)). Atragem atenția asupra faptului că ne vom referi adesea la [207] unde este adoptată varianta (2) și, ca urmare, trebuie acordată atenție diferenței care există în formularea unora dintre axiome.

Așa cum s-a văzut, nu orice colecție de obiecte este o mulțime (cazul mulțimii R din paradoxul lui Russell). De multe ori însă avem de exprimat proprietăți asupra unor obiecte, proprietăți ce nu depind de faptul că obiectele pot fi “colectate” sau nu într-o mulțime. Ca urmare, vom utiliza termenul de *clasă* cu sensul intuitiv de “colecție” a tuturor obiectelor ce satisfac o anumită formulă și vom fi precauți în folosirea unei notății de tipul $\{x|\varphi(x)\}$, atunci când nu știm dacă prin aceasta se definește o mulțime.

⁴ Discuții asupra acestui aspect pot fi găsite în [57, 156, 19]. Pe de altă parte, dacă sistemul ZFC în varianta (1) este consistent, atunci și sistemul ZFC în varianta (2) este consistent [153, 123, 172].

Mulțimile sunt cazuri particulare de clase, însă există clase ce nu sunt mulțimi; acestea vor fi numite *clase proprii*. Atragem atenția că, din punct de vedere formal, clasele nu există ca entități ale obiectului nostru de studiu. Ele trebuie gândite numai ca abrevieri pentru formule ce nu le implică și pe ele. Ca urmare, nu va exista nici o distincție formală între clase și formule; distincția este numai informală în prezentarea noastră (în metalimbaj) ⁵.

Pentru clase vom utiliza terminologii și notații similare celor de la mulțimi (în discuția de mai jos **A** și **B** vor fi clase asociate formulelor φ și ψ):

- “ $x \in \mathbf{A}$ ” pentru “ $\varphi(x)$ ”; (apartenență)
- “ $\mathbf{A} \subseteq \mathbf{B}$ ” pentru “ $(\forall x)(\varphi(x) \Rightarrow \psi(x))$ ”; (incluziune)
- “ $\mathbf{A} = \mathbf{B}$ ” pentru “ $(\forall x)(\varphi(x) \Leftrightarrow \psi(x))$ ”; (egalitate)
- “ $\mathbf{A} \cup \mathbf{B}$ ” pentru “ $(\forall x)(\varphi(x) \vee \psi(x))$ ”; (reuniune)
- “ $\mathbf{A} - \mathbf{B}$ ” pentru “ $(\forall x)(\varphi(x) \wedge \neg\psi(x))$ ”; (diferență)
- etc.

Din cele de mai sus se vede clar că lucrul cu clase nu este altceva decât lucrul cu formule. Însă este mai comod din punct de vedere tehnic, dar poate și intuitiv, să spunem “ $x \in \mathbf{A} \cup \mathbf{B}$ ” în loc de “ $\varphi(x) \vee \psi(x)$ ”.

Cu aceste elemente pregătitoare putem începe discuția axiomelor de bază ale sistemului ZFC.

Mulțimile sunt complet specificate de elementele pe care le conțin. Acest fapt trebuie introdus axiomatic.

Axioma extensionalității. Două mulțimi A și B sunt egale și notăm $A = B$, dacă pentru orice obiect x are loc $x \in A$ dacă și numai dacă $x \in B$.

Dacă două mulțimi A și B nu sunt egale, atunci vom nota $A \neq B$. Axioma extensionalității nu este o presupunere trivială asupra proprietății de apartenență. Să considerăm exemplul dat de Paul Richard Halmos [80], în care anumite entități sunt perfect caracterizate de mulțimea descendenților (strămoșilor). Identificăm atunci entitatea cu mulțimea strămoșilor ei. Dacă A și B sunt două astfel de entități și $A = B$, atunci pentru orice obiect x avem $x \in A$ dacă și numai dacă $x \in B$. Reciproca nu este în general adevărată, ca urmare a faptului că pot exista entități diferite cu aceeași strămoși (cazul în care entitățile ar fi persoane umane, de exemplu).

Este ușor de văzut că au loc următoarele proprietăți:

- $A = A$; (reflexivitate)

⁵Necesitatea de a distinge între mulțimi și clase a fost observată și de Cantor. Astfel, într-o scrisoare din 1899 către Dedekind, acesta a utilizat expresia de “mulțime inconsistentă” (a se vedea [59], pag. 97, și [28], pag. 443-448).

- dacă $A = B$, atunci $B = A$; (simetrie)
- dacă $A = B$ și $B = C$, atunci $A = C$. (tranzițivitate)

Definiția 1.1.1.1. Fie A și B două mulțimi.

- (1) Spunem că A este *submulțime* a mulțimii B și notăm $A \subseteq B$, dacă orice element al mulțimii A este element al mulțimii B .
- (2) Spunem că A este *submulțime proprie* a mulțimii B și notăm $A \subset B$, dacă $A \subseteq B$ și $A \neq B$.

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

Teorema 1.1.1.1. Fie A , B și C mulțimi. Atunci au loc următoarele proprietăți:

- (1) $A \subseteq A$;
- (2) dacă $A \subseteq B$ și $B \subseteq C$, atunci $A \subseteq C$;
- (3) $A = B$ dacă și numai dacă $A \subseteq B$ și $B \subseteq A$;
- (4) $A \subseteq B$ dacă și numai dacă $A \subset B$ sau $A = B$;
- (5) $A \not\subseteq A$;
- (6) dacă $A \subset B$, atunci $B \not\subseteq A$;
- (7) dacă $A \subset B$ și $B \subseteq C$, sau $A \subseteq B$ și $B \subset C$, atunci $A \subset C$.

Demonstrație. (1) și (2) urmează direct de la Definiția 1.1.1.1(1). (3) este o altă exprimare, utilizând inclusiunea, a Axiomei extensionalității.

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

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

(5) Dacă am presupune prin contradicție că $A \subset A$, atunci conform definiției ar exista $a \in A$ astfel încât $a \notin A$, ceea ce constituie o contradicție.

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

(7) Să presupunem că are loc $A \subset B$ și $B \subseteq C$. De la (4) și (2) obținem $A \subseteq C$. Dacă presupunem că $A = C$, atunci ipoteza se reduce la $A \subset B$ și $B \subseteq A$, ceea ce constituie o contradicție. În mod similar se discută și celălalt caz. \square

Pentru a stabili egalitatea a două mulțimi A și B , conform Teoremei 1.1.1.1(3), avem de arătat că oricare dintre cele două mulțimi este inclusă în celalăță. Această metodă de demonstrație este numită adesea *demonstrația prin dublă inclusiune*.

Existența a cel puțin unei mulțimi în universul de discurs al variabilelor trebuie asigurată axiomatice. Aceasta se poate face prin introducerea unei axiome care să postuleze existența a cel puțin unei mulțimi, sau printr-o axiomă care să postuleze existența unei mulțimi particulare sau de tip particular. Axioma pe care o vom considera va postula existența unei mulțimi fără nici un element⁶.

Axioma de existență a mulțimii vide. Există mulțimi fără nici un element.

În baza Axiomei extensionalității, orice două mulțimi fără nici un element sunt egale și, deci, există o unică mulțime fără nici un element. Aceasta este numită *mulțimea vidă* și este notată prin \emptyset . Este clar că \emptyset este submulțime proprie a oricărei mulțimi diferite de ea însăși și nici o mulțime nu este submulțime proprie a ei.

Reformulăm acum Axioma abstracției utilizată în abordarea cantoriană a mulțimilor, în modul următor:

Axioma separării. Pentru orice formulă $\varphi(x)$ și mulțime U există o mulțime ce conține toate elementele din U ce satisfac P și numai pe acestea.

Axioma separării este de fapt o schemă de axiomă, pentru fiecare caz particular al formulei $P(x)$ obținându-se o axiomă.

Axioma separării exclude paradoxul lui Russell formulat corespunzător acesteia. Adică, dacă considerăm o mulțime arbitrară U și definim $R_U = \{x \in U | x \notin x\}$, atunci contradicția:

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

nu mai poate fi obținută. În adevăr,

- dacă $R_U \in R_U$, atunci urmează că $R_U \in U$ și $R_U \notin R_U$, ceea ce constituie o contradicție;
- dacă $R_U \notin R_U$, atunci $R_U \notin U$ sau $R_U \in R_U$; rezultă deci că $R_U \notin U$.

Am obținut astfel că $R_U \notin U$, adică, indiferent de ce mulțimea U să ar alege, mulțimea R_U nu este element al ei. Mai mult, are loc:

Teorema 1.1.1.2. Nu există nici o mulțime U care să conțină ca element orice mulțime.

Demonstrație. Pentru orice mulțime U , mulțimea R_U , definită ca mai sus, nu este element al mulțimii U . Ca urmare, nu poate exista o mulțime U care să conțină orice mulțime, deoarece atunci ea ar trebui să conțină și R_U . \square

⁶Această axiomă nu este necesară, existența a cel puțin unei mulțimi va fi asigurată de Axioma infinitului. Însă, dacă vom merge pe o astfel de variantă, va trebui să amânăm introducerea multor concepte și probabil că anumite aspecte ale discuției noastre nu vor părea prea naturale.

Teorema 1.1.1.2 afirmă că nu există o mulțime a tuturor mulțimilor sau, altfel spus, clasa tuturor mulțimilor este o clasă proprie.

Concluzia pe care o desprindem din cele de mai sus este că Axioma separării conduce la schimbări semnificative eliminând paradoxuri de tipul paradoxului lui Russell.

Dacă în locul Axiomei de existență a mulțimii vide am fi adoptat o axiomă care să postuleze existența a cel puțin unei mulțimi A , atunci existența mulțimii vide s-ar fi putut obține prin intermediul Axiomei separării, considerând $\{x \in A | x \neq x\}$ (care, prin Axioma extensionalității, este unică).

Axioma separării permite introducerea a două operații fundamentale cu mulțimi, *intersecția* și *diferența*. Considerând o clasă nevidă C de mulțimi (adică C conține cel puțin o mulțime), există o mulțime $\bigcap C$ ce conține exact obiectele comune tuturor mulțimilor din C . Aceasta se poate obține aplicând Axioma separării unei mulțimi arbitrară A din C :

$$\bigcap C = \{a \in A | (\forall B \in C)(a \in B)\}.$$

Axioma extensionalității asigură unicitatea acestei mulțimi, care se numește *intersecția clasei* C . Atragem atenția asupra următoarelor două aspecte:

- aşa cum am spus, clasele sunt abrevieri pentru formule. Ca urmare, expresia $"(\forall B \in C)(a \in B)"$ este de fapt o formulă;
- clasa C trebuie să fie nevidă pentru a considera intersecția ei, deoarece altfel o definiție de tipul:

$$\{a | (\forall B)(B \in C \Rightarrow a \in B)\}$$

conduce la *clasa universală* V ce conține toate obiectele universului de discurs.

Convenim ca prin *familie de mulțimi* să înțelegem orice mulțime ale cărei elemente sunt mulțimi⁷. Vom nota familiile de mulțimi prin litere runde ale alfabetului ($\mathcal{A}, \mathcal{B}, \mathcal{C}$ etc.). Discuția de mai sus conduce la existența *intersecției oricărei familii nevide de mulțimi*. 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 mulțimi A și B sunt numite *disjuncte* dacă $A \cap B = \emptyset$. O familie de mulțimi este numită *familie disjunctă de mulțimi* dacă mulțimile componente sunt disjuncte două căte două.

Diferența a două mulțimi A și B , notată $A - B$, se obține cu ușurință de la Axioma separării:

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

⁷Sistemul ZFC, în varianta adoptată, este dezvoltat peste un sistem de discurs format numai din mulțimi. Ca urmare, orice obiect al acestuia este o familie de mulțimi. Utilizarea termenului de "familie de mulțimi" vrea să atragă și mai mult atenția asupra acestui aspect. Pe de altă parte, în special în aplicații pentru informatică, vom considera diverse obiecte (construcții) care nu vor trebui gândite neapărat ca mulțimi. De exemplu, dacă $\Sigma = \{a, b, c\}$ este un alfabet, atunci $\{\{a, b\}, \{a, c\}\}$ este o familie de mulțimi peste Σ , dar $\{a, \{a, b\}, \{a, c\}\}$ nu va considera că familia de mulțimi peste Σ cu toate că elementele a, b și c , fiind luate din universul de discurs, sunt mulțimi.

Uneori diferența $A - B$ mai poartă și denumirea de *complementara lui B relativ la A* sau *în raport cu A*.

Nici una din axiomele introduse până acum nu asigură existența unor mulțimi de forma $\{a\}$ sau $\{a, b\}$. Pentru aceasta va trebui să considerăm o nouă axiomă.

Axioma împerecherii. Pentru orice două obiecte a și b (nu neapărat distințe) există o mulțime ce conține obiectele a și b și numai pe acestea.

Fie a și b obiecte. Axioma extensionalității asigură unicitatea mulțimii ce conține obiectele a și b și 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 și v au loc următoarele proprietăți:

- (1) $z \in \{x, y\}$ dacă și numai dacă $z = x$ sau $z = y$;
- (2) $\{x, y\} = \{u, v\}$ dacă și numai dacă $x = u$ și $y = v$ sau $x = v$ și $y = u$;
- (3) $\{x\} = \{y\}$ dacă și numai dacă $x = y$;
- (4) $\{x\} = \{u, v\}$ dacă și numai dacă $x = u = v$.

Demonstrație. (1), (2) și (3) decurg imediat de la definiții și 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 aparțină mulțimii $\{u, v\}$. Dacă presupunem că $x = u$ atunci, în baza aceleiași teoreme, deducem că $v \in \{x\}$, adică $x = v$. Deci $x = u = v$.

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

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

- obiectele x și 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 [109].

Definiția 1.1.1.2. Se numește *perechea ordonată* a obiectelor x și y mulțimea notată (x, y) și definită prin $(x, y) = \{\{x\}, \{x, y\}\}$.

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

Teorema 1.1.1.4. $(x, y) = (u, v)$ dacă și numai dacă $x = u$ și $y = v$ ⁸.

Demonstrație. Să presupunem întâi că $(x, y) = (u, v)$. Dacă $x = y$, atunci $\{\{x\}\} = \{\{u\}, \{u, v\}\}$, de unde urmează că $x = u = v$ (Teorema 1.1.3).

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

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

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 și pentru astfel de 3-uple, adică $(x, y, z) = (x', y', z')$ dacă și numai dacă $x = x'$, $y = y'$ și $z = z'$.

Ceea ce trebuie să remarcăm este că prin Axioma împerecherii putem construi mulțimi cu cel mult două elemente. Trecerea la mulțimi cu mai mult de două elemente trebuie făcută axiomatic.

Axioma reuniunii. Pentru orice familie de mulțimi \mathcal{A} există o mulțime ce conține elementele componente ale mulțimilor conținute de \mathcal{A} și numai pe acestea.

Axioma extensionalității asigură că, pentru orice familie de mulțimi \mathcal{A} , există exact o mulțime ca cea postulată de Axioma reuniunii; această mulțime se numește *reuniunea* familiei \mathcal{A} și 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 proprietăți de bază ale reuniunii unei familii de mulțimi.

Teorema 1.1.1.5. Fie A și B mulțimi, iar \mathcal{A} și \mathcal{B} familii de mulțimi. Atunci au loc următoarele proprietăți:

- (1) $\bigcup \emptyset = \emptyset$;
- (2) $\bigcup \{A\} = A$;
- (3) $\bigcup \mathcal{A} = \emptyset$ dacă și numai dacă $\mathcal{A} = \emptyset$ sau $\mathcal{A} = \{\emptyset\}$;

⁸O altă variantă de a defini perechea ordonată a două obiecte x și y este cea propusă de Norbert Wiener în 1914 [222], prin care $(x, y) = \{\{x\}, \{y\}\}$. Se poate arăta că și o astfel de definiție satisfac Teorema 1.1.1.4, dar, spre deosebire de definiția lui Kuratowski, aceasta implică un nou obiect, 0 (care va fi definit mai târziu).

Există variante în care noțiunea de pereche ordonată se consideră ca o noțiune primitivă, și atunci enunțul Teoremei 1.1.1.4 se introduce ca axiomă (a se vedea [19, 156]). Justificarea constă în faptul că majoritatea aplicațiilor matematice ale acestei noțiuni utilizează Teorema 1.1.1.4 și nu definiția.

- (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 $\bigcup \mathcal{A} \subseteq \mathcal{B}$.

Possibilitatea colectării tuturor submulțimilor unei mulțimi într-o mulțime trebuie introdusă axiomatic.

Axioma părților. Pentru orice mulțime A , există o mulțime ce conține ca elemente toate submulțimile mulțimii A și numai pe acestea.

Axioma extensionalității asigură că pentru orice mulțime A există o unică mulțime ca cea postulată de Axioma părților; această mulțime se numește *mulțimea părților* mulțimii A și se notează prin $\mathcal{P}(A)$.

Teorema 1.1.1.6. Fie A și B mulțimi. Atunci au loc următoarele proprietăți:

- (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 părților permite introducerea conceptului de produs cartezian a două mulțimi.

Definiția 1.1.1.3. Fie A și B două mulțimi. Numim *produsul cartezian* sau *direct* al mulțimilor A și B , mulțimea notată $A \times B$ și definită prin:

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

Existența produsului cartezian al mulțimilor A și B urmează de la Axioma separării, aplicată mulțimii $\mathcal{P}(\mathcal{P}(A \cup B))$,

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

iar unicitatea lui de la Axioma extensionalității.

Dacă una dintre mulțimile A sau B este vidă, atunci produsul cartezian al lor este mulțimea vidă și reciproc. Notăm că $A \times B$ nu este același cu $B \times A$, exceptând cazul în care $A = B$ sau cazul în care una dintre aceste două mulțimi este mulțimea vidă.

Produsul cartezian poate fi extins în mod natural la mai mult de două mulțimi. Considerând de exemplu mulțimile A , B și C , putem defini:

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

Construcțiile (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:

- $A^n = \underbrace{A \times \cdots \times A}_{n \text{ ori}}$, dacă $n \geq 2$;
- $A^1 = A$ și $A^0 = \{\emptyset\}$.⁹

În matematică, dar nu numai, suntem interesati în a construi corespondențe (asocieri) între diverse tipuri de obiecte. Cel mai frecvent sunt întâlnite corespondențele între două tipuri de obiecte, nu neapărat distinse. Perechea ordonată (a, b) poate fi o alegere bună pentru a exprima corespondența (asocierea) dintre a și b , mai ales atunci când dorim să surprindem și o anumită "relație de precedență" între a și b .¹⁰ Ca urmare, o mulțime de perechi ordonate va modela o corespondență (asociere) între două tipuri de obiecte. Astfel de mulțimi vor fi numite *relații binare*.

Definiția 1.1.1.4. Se numește *relație binară* orice mulțime ale cărei elemente sunt perechi ordonate.¹¹

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

Fiind dată o relație ρ , vom nota

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

și

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

În baza Axiomelor separării și reuniunii, $\text{Dom}(\rho)$ și $\text{Cod}(\rho)$ sunt mulțimi. În adevăr,

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

În mod similar, putem arăta că $\text{Cod}(\rho)$ este mulțime. Mulțimea $\text{Dom}(\rho)$ se numește *domeniu* relației ρ , iar $\text{Cod}(\rho)$, *codomeniul* relației ρ .

Putem spune că ρ este relație dacă există două mulțimi A și B astfel încât $\rho \subseteq A \times B$. Reciproc, orice submulțime a unui produs cartezian este relație. Aceasta face ca adesea relațiile $\rho \subseteq A \times B$ să mai fie numite și *relații de la A la B*, iar atunci când $B = A$, *relații (binare) pe A*.

Mulțimea vidă este relație (de la A la B), numită *relația vidă*.

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

⁹Anticipând câteva concepte și notări care vor fi prezentate ulterior, dar cu care cititorul este probabil familiarizat, justificăm definiția " $A^0 = \{\emptyset\}$ " astfel. Așa cum vom vedea, numerele naturale vor fi definite ca mulțimi: $0 = \emptyset$, $1 = \{0\}$, $2 = \{0, 1\}$ etc. Atunci, un n -uplu poate fi gândit ca o funcție de la mulțimea n la mulțimea A , iar mulțimea tuturor acestor funcții (n -uple) este notată prin A^n (în general, prin A^B se va nota mulțimea tuturor funcțiilor de la B la A). În cazul $n = 0 = \emptyset$, există o singură funcție de la \emptyset la A și anume funcția vidă. Ca urmare, $A^0 = \{\emptyset\}$.

¹⁰O altă posibilă exprimare a asocierii dintre a și b ar putea fi specificată prin intermediu mulțimii $\{a, b\}$. În acest caz însă se pierde "ordinea" în care sunt considerate obiectele a și b .

¹¹În limbaj logic, ρ este relație binară dacă $(\forall x)(x \in \rho \Rightarrow (\exists y)(\exists z)(x = (y, z)))$.

Definiția 1.1.1.5. O relație binară f este numită *relație funcțională* sau *funcție* dacă are loc:

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

Relația vidă este funcție, numită și *funcția vidă*.

Pentru funcții se utilizează în mod frecvent notația $f(a) = b$ 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.

Funcțiile fiind relații, putem vorbi de *domeniu* și *codomeniu* acestora. Domeniul unei funcții mai poartă denumirea și de *domeniu de definiție* al funcției. Domeniul și codomeniul funcției vide sunt mulțimea vidă.

O funcție f este numită *funcție de la A la B* sau *funcție definită pe A și cu valori în B* și notăm $f : A \rightarrow B$, dacă $\text{Dom}(f) = A$ și $\text{Cod}(f) \subseteq B$. Funcția vidă este funcție de la A la B numai dacă $A = \emptyset$.

Mulțimea tuturor funcțiilor de la A la B se notează prin $(A \rightarrow B)$ sau B^A .

Definiția 1.1.1.6. Fie f o funcție de la A la B .

(1) f este numită *funcție injectivă* sau *injecție* 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 *funcție surjectivă* sau *surjecție* dacă are loc:

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

(3) f este numită *funcție bijectivă* sau *bijecție* dacă este atât funcție injectivă, cât și funcție surjectivă.

Uneori funcțiile injective sunt numite *funcții 1 – 1*, iar cele surjective, *pe*. Funcția vidă de la \emptyset la B este injectivă; ea este surjectivă (deci și bijectivă) doar dacă $B = \emptyset$.

Atunci când există o funcție bijectivă de la o mulțime A la o mulțime B , vom mai nota $A \sim B$ și vom spune că A și B sunt *echipotente*¹², iar dacă există o funcție 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, ce rămâne nerezolvată de axioanele prezентate până acum: există mulțimi ce sunt elemente ale lor însăși? Dacă teoria mulțimilor (sistemuaxiomatic pe care îl prezentăm) ar fi bazată pe Axioma abstracției, atunci răspunsul ar fi pozitiv: mulțimea tuturor mulțimilor, a cărei existență ar fi asigurată de Axioma abstracției,

¹²Terminologia de "mulțimi echipotente", care înseamnă "mulțimi cu același număr de elemente", este justificată prin aceea că o funcție bijectivă pune în corespondență "unu-la-unu" elementele a două mulțimi. Echipotența joacă un rol important în definirea numerelor naturale, a numerelor ordonale și cardinale.

are această proprietate. Axiomele considerate până acum nu conduc nici la existența și nici la non-existența unei astfel de mulțimi [172]. Similar, constatăm că este dificil să ne imaginăm mulțimi A, B, C cu proprietatea $A \in B$ și $B \in A$ sau $A \in B$, $B \in C$ și $C \in A$. Aceste remarci conduc la necesitatea considerării unei noi axiome.

Axioma regularității. Pentru orice mulțime nevidă A există $x \in A$ astfel încât $x \cap A = \emptyset$.

Putem acum demonstra:

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

Demonstrație. Presupunem prin contradicție că există o mulțime A cu proprietatea $A \in A$. Aplicăm Axioma regularității mulțimii $\{A\}$. Atunci există $x \in \{A\}$ astfel încât $x \cap A = \emptyset$. Forma particulară a mulțimii $\{A\}$ conduce la faptul că x trebuie să fie A și, în consecință, $A \cap \{A\} = \emptyset$, ceea ce contrazice faptul că $A \in A$. \square

Teorema 1.1.1.8. Nu există mulțimi A și B astfel încât $A \in B$ și $B \in A$.

Demonstrație. Să presupunem că există două mulțimi A și B astfel încât $A \in B$ și $B \in A$. Aplicăm Axioma regularității mulțimii $\{A, B\}$. Atunci, există $x \in \{A, B\}$ astfel încât $x \cap \{A, B\} = \emptyset$. Elementul x poate fi A sau B . Dacă $x = A$, atunci $A \cap \{A, B\} = \emptyset$, ceea ce contrazice faptul că $B \in A$ și $B \in \{A, B\}$. Se raționează similar pentru cazul $x = B$. \square

Axioma regularității este consistentă cu celelalte axiome ale sistemului ZFC și independentă de acestea [70, 172]. Este posibil să construi sisteme ale teoriei mulțimilor care să contrazică această axiomă. Două exemple în acest sens sunt sistemul lui Lesniewski [121] și cel al lui Quine [169].

Axioma regularității are consecințe naturale, cum este cea din teorema următoare (care nu poate fi demonstrată pe baza celorlalte axiome).

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

Demonstrație. Presupunem prin contradicție că există o mulțime nevidă A astfel încât $A \subseteq A \times A$. Atunci elementele mulțimii $A \times A$ sunt mulțimi nevide. Ipoteza și definiția produsului cartezian conduce la faptul că elementele mulțimilor A și $\bigcup A$ sunt, de asemenea, mulțimi nevide. Fie $B = A \cup \bigcup A$. Axioma regularității asigură existența unei mulțimi $x \in B$ astfel încât $x \cap B = \emptyset$. Avem de analizat următoarele două cazuri posibile:

- $x \in A$. Atunci $x \subseteq \bigcup A$ și, deoarece x este mulțime nevidă, urmează că $x \cap \bigcup A \neq \emptyset$, ceea ce contrazice faptul că $x \cap B = \emptyset$;
- $x \in \bigcup A$. Conform ipotezei și definiției produsului cartezian, x este ori de forma $\{a\}$, ori de forma $\{a, b\}$, unde $a, b \in A$. Ca urmare, $x \cap A \neq \emptyset$, ceea ce contrazice faptul că $x \cap B = \emptyset$.

Cum ambele cazuri au condus la o contradicție concluzionăm că ipoteza de la care am plecat este falsă, ceea ce încheie demonstrația teoremei. \square

Axioma regularității este o presupunere foarte tare. În cele ce urmează vom încerca să evităm pe cât posibil utilizarea ei.

Sistemul ZFC conține încă trei axiome ce vor fi prezentate în secțiuni separate deoarece ele necesită o discuție mai amplă. Vom face totuși o scurtă introducere pentru fiecare dintre ele.

Axiomele deja prezentate nu asigură existența mulțimilor infinite, ceea ce va trebui făcut axiomatic. *Axioma infinitului* va fi cea care va asigura existența acestor mulțimi.

Definiția 1.1.1.7. Fie A o mulțime. Se numește *sistem* peste A orice submulțime $S \subseteq \mathcal{P}(A)$.

Un sistem peste o mulțime A este o familie de mulțimi. Reciproc, orice familie de mulțimi \mathcal{A} poate fi gândită ca un sistem peste $\bigcup \mathcal{A}$ (această mulțime există în baza Axiomei reuniunii). Din acest motiv vom utiliza în egală măsură aceste noțiuni.

Un sistem nevid peste o mulțime A , format din mulțimi nevide, disjuncte două câte două și a cărui reuniune este A , se numește *partiție a mulțimii* A . Elementele unei partiții sunt numite *blocuri*, iar mulțimea tuturor partițiilor mulțimii A se notează prin $\text{Part}(A)$. În cazul $A \neq \emptyset$, $\text{Part}(A)$ este nevidă deoarece este formată cel puțin din partiția ce conține singurul bloc A .

Definiția 1.1.1.8. Fie \mathcal{A} o familie de mulțimi (un sistem). Se numește *mulțime de reprezentanți* pentru \mathcal{A} orice mulțime C cu proprietatea că pentru fiecare $A \in \mathcal{A}$ mulțimea C conține exact un element din A și nu conține alte elemente.

Întrebarea naturală care se pune acum este următoarea: dată o familie de mulțimi \mathcal{A} , există mulțimi de reprezentanți pentru ea? Încercările de a răspunde acestei întrebări pe baza axiomelor prezentate până acum au eşuat. Aceasta a condus la introducerea unei noi axiome, *Axioma alegerii*, care să postuleze existența unei mulțimi de reprezentanți pentru orice familie de mulțimi disjuncte și nevide.

Uneori suntem interesați să construim mulțimi B ce “corespond” unor mulțimi date A în manieră funcțională. Dacă am cunoaște o mulțime C din care să selectăm obiectele ce vor alcătui B , atunci am putea aplica Axioma separării. De multe ori însă C este o clasă proprie. Dar dacă obiectele din B sunt construite pornind de la obiectele mulțimii A , în manieră funcțională, atunci intuiția ne spune că B ar trebui să fie mulțime. Însă nici una dintre axiomele prezentate până acum nu ne asigură aceasta, ceea ce face necesară introducerea unei noi axiome, ce va fi *Axioma înlocuirii*.

Acestea sunt celelalte 3 axiome pe care le vom discuta în acest capitol dar, aşa cum am spus, fiecare dintre ele va fi prezentată la momentul potrivit.

1.1.2. Operații cu mulțimi

În secțiunea anterioară au fost introduse în mod implicit un număr de operații cu mulțimi: intersecție, diferență, reuniune și produs cartezian. Vom adăuga la acestea câteva operații noi și vom prezenta unele proprietăți de bază ale lor.

Teorema 1.1.2.1. Fie A , B și C mulțimi. 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) $A \cup A = A$; (idempotență)
- (4) $A \cup \emptyset = A$;
- (5) $\mathcal{P}(A) \cup \mathcal{P}(B) \subseteq \mathcal{P}(A \cup B)$.

Demonstrație. (1) Vom folosi metoda dublei incluziuni. Fie $a \in A \cup (B \cup C)$. Conform definiției reuniunii, avem $a \in A$ sau $a \in B \cup C$. Dacă $a \in A$, atunci $a \in A \cup B$ și, în consecință, $a \in (A \cup B) \cup C$. Dacă $a \in B \cup C$, atunci $a \in B$ sau $a \in C$. În cazul $a \in B$ obținem $a \in A \cup B$ și, în consecință, $a \in (A \cup B) \cup C$, iar în cazul $a \in C$ obținem $a \in (A \cup B) \cup C$. 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 definiția reuniunii și faptul că $\{A, B\} = \{B, A\}$.
- (3) Observăm că are loc relația $\{A, A\} = \{A\}$. Ca urmare ne rămâne de demonstrat că $\bigcup \{A\} = A$, ceea ce urmează direct de la definiția reuniunii.
- (4) Un element a este în $A \cup \emptyset$ dacă și 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 $X \subseteq A \cup B$; deci $X \in \mathcal{P}(A \cup B)$. Se procedează similar și în cazul $X \in \mathcal{P}(B)$. \square

Demonstrațiile următoarelor două teoreme sunt lăsate în seama cititorului.

Teorema 1.1.2.2. Fie A , B și C mulțimi. 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) $A \cap A = A$; (idempotență)
- (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 proprietății de asociativitate a reuniunii și intersecției, 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 de $(A \cup B) \cup C$ ($(A \cap B) \cap C$). Evident, această scriere poate fi extinsă la o reuniune (intersecție) finită de mulțimi.

Teorema 1.1.2.3. Fie A o mulțime și \mathcal{C} o familie de mulțimi. Atunci au loc următoarele reguli de distributivitate:

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

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

Interpretăm proprietățile din Teorema 1.1.2.3 prin aceea că *intersecția este distributivă față de reuniune atât la stânga, cât și la dreapta*. În mod similar, *reuniunea este distributivă față de intersecție la stânga și la dreapta*.

Următoarea teoremă prezintă câteva proprietăți de bază ale diferenței de mulțimi.

Teorema 1.1.2.4. Fie A și B mulțimi, iar \mathcal{C} o familie de mulțimi. Atunci:

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

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

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

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

$$(5) \text{ dacă } A \cap B = \emptyset, \text{ atunci } A - B = A;$$

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

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

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

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

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

Fie U o mulțime. Complementara unei submulțimi $A \subseteq U$ în raport cu U se mai numește și *complementara absolută a lui A relativ la (în raport cu) U* sau, mai simplu, *complementara lui A* (dar în acest caz U trebuie subînțeleasă din context). Ea se notează prin \bar{A} .

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

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

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

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

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

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

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

$$(7) A \subseteq B \text{ dacă și numai dacă } \bar{B} \subseteq \bar{A}$$

(complementara este în raport cu U).

Corolarul 1.1.2.1. (Legile lui De Morgan)

Fie U , A și B mulțimi astfel încât $A \cup B \subseteq U$. Atunci au loc relațiile:

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

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

(complementara este în raport cu U).

Definiția 1.1.2.1. Fie A și B două mulțimi. Numim *diferență simetrică* a mulțimilor A și B mulțimea $A \Delta B = (A - B) \cup (B - A)$.

Conform Axiomei reuniunii, există o unică mulțime $A \Delta B$. Deci, Definiția 1.1.2.1 este consistentă.

Operațiile \cup , \cap , $-$ și Δ au fost studiate în mod sistematic pentru prima dată de către George Boole [15]. Din acest motiv, ele sunt numite astăzi *operații Booleene*. Ele pot fi reprezentate grafic prin aşa-numitele *diagrame Venn*, ca în Figura 1.1.

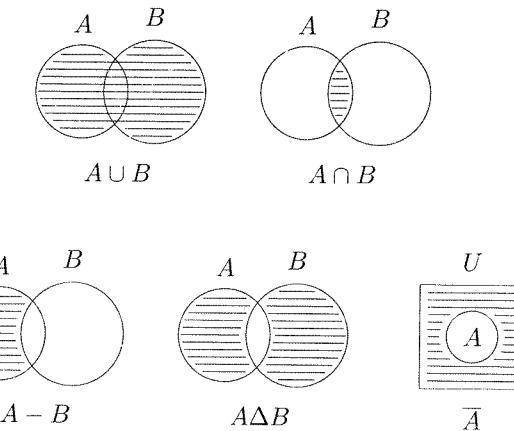


Figura 1.1: Reprezentarea operațiilor Booleene cu mulțimi prin diagrame Venn

Următoarea teoremă ne spune că produsul cartezian este *distributiv la stânga și la dreapta față de reuniune și de intersecție* (demonstrația este lăsată în seama cititorului).

Teorema 1.1.2.6. Fie A, B, C, D mulțimi și \mathcal{A} o familie de mulțimi. Atunci au loc următoarele proprietăți:

- (1) $A \times \bigcup \mathcal{A} = \bigcup \{A \times X | X \in \mathcal{A}\}$ și $\bigcup \mathcal{A} \times A = \bigcup \{X \times A | X \in \mathcal{A}\}$;
- (2) $A \times \bigcap \mathcal{A} = \bigcap \{A \times X | X \in \mathcal{A}\}$ și $\bigcap \mathcal{A} \times A = \bigcap \{X \times A | X \in \mathcal{A}\}$, cu condiția 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. Axioma infinitului. Numere naturale

Mulțimea numerelor naturale este, fără doar și poate, primul exemplu de mulțime infinită la care ne-am gândi dacă ni s-ar cere să dăm un exemplu de o astfel de mulțime. Introducerea ei este însă un proces destul de complex, care a stat în atenția cercetătorilor multe zeci de ani. În cele ce urmează vom prezenta o modalitate de introducere a mulțimii numerelor naturale fără a face apel la numere ordinară [93, 102], după care vom prezenta câteva proprietăți fundamentale ale ei.

1.1.3.1. Axioma infinitului

Așa cum s-a menținat, axiomele prezentate în Secțiunea 1.1.1 nu asigură existența mulțimilor infinite. Observăm însă că încă nu am introdus concepțele de *mulțime finită* și *mulțime infinită*. În mod clar, o mulțime va fi infinită dacă nu este finită. Dar ce este o mulțime finită? Uzual, conceptul acesta se introduce prin intermediul conceptului de număr natural. O mulțime este finită dacă există un număr natural n astfel încât ea are exact n elemente. Concluzia este că ar trebui introduce întâi numerele naturale. Cea mai elegantă variantă de introducere a numerelor naturale este prin intermediul ordinalilor. Această abordare este potrivită pentru o carte orientată numai pe teoria mulțimilor. Altfel, abordarea prin teoria ordinalilor poate părea nenaturală și greoală. Prezentarea noastră va urma [93, 102, 207], unde numerele naturale sunt introduse fără a face apel la teoria ordinalilor, într-o manieră destul de elegantă.

Noțiunea de număr natural este poate una dintre cele mai vechi și importante noțiuni ale matematicii. Intuitiv, știm ce reprezintă 1, 2, 3, ... însă, dacă suntem puși să explicăm aceasta, constatăm că nu este chiar atât de simplu. Ce înseamnă 2? Am putea spune că 2 reprezintă proprietatea comună pe care o au toate mulțimile cu ... două obiecte! În mod clar, definiția este circulară; dar aceasta este ceea ce am dori să surprindem. Putem porni atunci de la observația, intuitiv clară, că \emptyset nu are nici un element, $\{\emptyset\}$ are un element, $\{\emptyset, \{\emptyset\}\}$ are două elemente etc. Am spune că

“2 este ceea ce au în comun toate mulțimile cu același număr de elemente ca și $\{\emptyset, \{\emptyset\}\}$ ”.

Această nouă definiție face apel în continuare la noțiunea de număr deoarece ea utilizează exprimarea “același număr de elemente”. Observația fundamentală, ce aparține lui Cantor, este că proprietatea “mulțimile A și B au același număr de elemente” poate fi exprimată fără a face apel la noțiunea de număr natural. Putem accepta faptul intuitiv că A și B au “același număr de elemente” dacă și numai dacă există o corespondență bijectivă între A și B . Atunci, putem defini numerele naturale prin intermediul unor reprezentanți din clasa mulțimilor ce se află în bijecție. Astfel, mulțimea vidă, care nu are nici un element, ar putea fi aleasă ca reprezentant pentru numărul natural 0; mai mult, acest reprezentant este unic. Pentru numărul natural 1 pornim de la observația că deja am definit un obiect particular, 0. Atunci, am putea lua $1 = \{\emptyset\}$ ($\{\emptyset\}$ va fi reprezentant al tuturor mulțimilor cu un element). Procedeul continuă prin $2 = \{0, 1\}$, $3 = \{0, 1, 2\}$, ... Suntem astfel conduși la a defini numărul natural n ca fiind mulțimea vidă dacă n este 0 și ca fiind mulțimea tuturor numerelor naturale mai mici decât el, altfel.

Definiția 1.1.3.1. Fie x o mulțime. Mulțimea $S(x) = x \cup \{x\}$ este numită *succesoarea* mulțimii x .

Definiția 1.1.3.2. Spunem că o mulțime x este *inductivă* dacă:

- (1) $\emptyset \in x$;
- (2) $S(y) \in x$, pentru orice $y \in x$.

Axiomele menționate până acum nu conduc nicidcum la existența mulțimilor inductive. Pe de altă parte, posibilitatea colectării unui “număr infinit” de obiecte într-o mulțime reprezintă esența teoriei mulțimilor și argumentul de bază al utilității ei în multe ramuri ale matematicii. Ca urmare, va trebui să extindem sistemul axiomatic, adoptând o nouă axiomă.

Axioma infinitului. Există mulțimi inductive.

Vom prezenta în continuare câteva proprietăți utile ale mulțimilor inductive.

Definiția 1.1.3.3. O mulțime A este numită *tranzitivă* dacă $a \subseteq A$, pentru orice $a \in A$.

Direct de la definiții obținem:

Lema 1.1.3.1.

- (1) \emptyset este mulțime tranzitivă.
- (2) Dacă A este mulțime tranzitivă, atunci $S(A)$ este mulțime tranzitivă.

Teorema 1.1.3.1. Dacă A este o mulțime inductivă, atunci și următoarele mulțimi sunt inductive:

1. $A' = \{a \in A \mid a \subseteq A\}$;
2. $A' = \{a \in A \mid a \text{ este tranzitivă}\}$;
3. $A' = \{a \in A \mid a = \emptyset \vee (\exists b \in A)(a = S(b))\}$;
4. $A' = \{a \in A \mid a \text{ este tranzitivă} \wedge (\forall b \subseteq a)(b \neq \emptyset \Rightarrow (\exists c \in b)(\forall x \in b)(x \not\in c))\}$.

Demonstrație. Se utilizează din plin definițiile și/sau Lema 1.1.3.1. Vom exemplifica (2).

$\emptyset \in A'$ deoarece \emptyset este tranzitivă și este element al mulțimi A (A fiind inductivă). Dacă $a \in A'$, atunci a este tranzitivă și, conform Lemei 1.1.3.1, $S(a)$ este tranzitivă. Cum $S(a) \in A$, deoarece $a \in A$ și A este inductivă, urmează că $S(a) \in A'$. \square

1.1.3.2. Numere naturale

Fie \mathcal{I} clasa mulțimilor inductive. În baza Axiomei infinitului, această clasă este nevidă și, prin urmare, $\bigcap \mathcal{I}$ este mulțime. Atunci, definim *mulțimea numerelor naturale* \mathbf{N} ca fiind $\mathbf{N} = \bigcap \mathcal{I}$. Elementele acestei mulțimi vor fi numite *numere naturale*. Uzual, notăm

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

Vom prezenta în cele ce urmează câteva proprietăți de bază ale mulțimii \mathbf{N} și ale elementelor ei. Multe dintre aceste proprietăți se obțin drept consecințe directe ale Teoremei 1.1.3.1. Întâi, observăm că \mathbf{N} este cea mai mică mulțime inductivă și, ca urmare, ea va coincide cu toate mulțimile \mathbf{N}' obținute ca în Teorema 1.1.3.1. Adică are loc:

$$\begin{aligned} \mathbf{N} &= \{n \in \mathbf{N} \mid n \subseteq \mathbf{N}\} \\ &= \{n \in \mathbf{N} \mid n \text{ este tranzitivă}\} \\ &= \{n \in \mathbf{N} \mid n = \emptyset \vee (\exists m \in \mathbf{N})(n = S(m))\} \\ &= \{n \in \mathbf{N} \mid n \text{ este tranzitivă} \wedge (\forall m \subseteq n)(m \neq \emptyset \Rightarrow (\exists k \in m)(\forall x \in m)(x \not\in k))\}. \end{aligned}$$

Direct de la această simplă observație obținem următoarele proprietăți ale numerelor naturale (proprietăți pe care le cunoaștem din clasele primare ca fiind proprietăți implicit satisfăcute de aceste numere).

Corolarul 1.1.3.1.

- (1) \mathbf{N} este tranzitivă.
- (2) Orice număr natural este mulțime tranzitivă.
- (3) Dacă $n \in m$ și $m \in k$, atunci $n \in k$, pentru orice $n, m, k \in \mathbf{N}$.

- (4) $\bigcup S(n) = n$, pentru orice $n \in \mathbf{N}$.
- (5) Dacă $S(n) \subseteq S(m)$, atunci $n \subseteq m$, pentru orice $n, m \in \mathbf{N}$.
- (6) Dacă $S(n) = S(m)$, atunci $n = m$, pentru orice $n, m \in \mathbf{N}$.
- (7) $n \notin n$, pentru orice $n \in \mathbf{N}$.
- (8) $n \neq S(n)$, pentru orice $n \in \mathbf{N}$.
- (9) Nu există $n, m \in \mathbf{N}$ astfel încât $n \in m \in S(n)$.
- (10) Pentru orice $n, m \in \mathbf{N}$, cel mult una dintre relațiile $n = m$, $n \in m$ sau $m \in n$ poate avea loc.
- (11) Orice număr natural este ori 0, ori succesorul unui alt număr natural.
- (12) Orice submulțime nevidă a unui număr natural are cel mai mic element în raport cu “ \in ” (adică, pentru orice număr natural n și $m \subseteq n$, dacă $m \neq \emptyset$, atunci există $k \in m$ cu proprietatea $x \notin k$, pentru orice $x \in m$).

Demonstrație. (1), (2), (11) și (12) urmează direct de la sirul de egalități de mai sus.

Pentru (3), în baza tranzitivității, obținem $n \in m$ și $m \subseteq k$, ceea ce conduce la $n \in k$.

(4) $\bigcup S(n) = (\bigcup n) \cup (\bigcup \{n\}) = (\bigcup n) \cup n = n$ (ultima egalitate urmează de la faptul că n este tranzitivă, și astfel $x \subseteq n$, pentru orice $x \in n$).

(5) Relația $S(n) \subseteq S(m)$ conduce la $n = m$ sau $n \in m$. Cea de a doua relație, în baza tranzitivității, implică $n \subseteq m$.

(6) urmează de la (5).

(7) Mulțimea $\mathbf{N}' = \{n \in \mathbf{N} \mid n \notin n\}$ este tranzitivă (ceea ce este foarte ușor de arătat). Ca urmare, $\mathbf{N} = \mathbf{N}'$, ceea ce ne spune că $n \notin n$, pentru orice $n \in \mathbf{N}$.

(8) Dacă am presupune $n = S(n)$, atunci am avea $n \in n$, ceea ce ar contrazice (7).

(9) Presupunem că există n și m astfel încât $n \in m \in S(n)$. Relația $m \in S(n)$ conduce la $m \in n$ sau $m = n$. Oricare dintre acestea, combinată cu $n \in m$, conduce la contradicție cu (7).

(10) Dacă $n = m$, atunci nu poate avea loc nici una dintre celelalte două relații deoarece s-ar contrazice (7). Dacă are loc $n \in m$ și am presupune că are loc și $m \in n$, atunci (3) ar conduce la $n \in n$, ceea ce contrazice din nou (7). \square

Proprietatea $n \notin n$ (Corolarul 1.1.3.1(7) urmează direct de la Axioma regularității. Însă, aşa cum am spus în Secțiunea 1.1.1, vom evita pe cât posibil ca în demonstrații de tipul “ $A \notin A$ ” să facem apel la această axiomă. Ea este o presupunere foarte puternică și utilitatea ei se justifică doar acolo unde nu se poate demonstra altfel faptul că $A \notin A$.

1.1.3.3. Ordonare și inducție

Definim pe \mathbb{N} relația binară $<$ prin

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

pentru orice $n, m \in \mathbb{N}$. În mod ușual, $n \leq m$ dacă și numai dacă $n < m$ sau $n = m$. Adoptăm următoarele terminologii (care vor fi generalizate în Secțiunea 1.2):

- dacă $n < m$, atunci vom spune că n este mai mic decât m ;
- dacă $n \leq m$, atunci vom spune că n este mai mic sau egal cu m ;
- dacă $n = S(m)$, atunci vom spune că n este succesor imediat al lui m (terminologia este justificată de Corolarul 1.1.3.1(9));
- spunem că n și m sunt *comparabile* (în raport cu $<$ sau, în mod echivalent, cu \in), dacă are loc exact una dintre relațiile: $n = m$, sau $n < m$, sau $m < n$;
- spunem că un număr natural n este cel mai mic element al mulțimii nevide $A \subseteq \mathbb{N}$, dacă $n \in A$ și $x \notin n$, pentru orice $x \in A$;
- spunem că un număr natural n este element maximal al mulțimii nevide $A \subseteq \mathbb{N}$, dacă $n \in A$ și $n \notin x$, pentru orice $x \in A$.

Corolarul 1.1.3.1(3) ne spune că relația $<$ pe numere naturale este *tranzitivă*, adică satisfacă $n < k$ ori de câte ori există m astfel încât $n < m$ și $m < k$, iar punctul (7) al aceluiași corolar ne spune că această relație este și *ireflexivă*, adică $n \notin n$, pentru orice $n \in \mathbb{N}$. Dacă, în plus, orice două numere naturale n și m ar fi comparabile, atunci $<$ ar fi o *ordine totală strictă* pe \mathbb{N} ¹³. Corolarul 1.1.3.1(10) ne spune că pentru orice două numere naturale n și m , cel mult una dintre relațiile $n = m$ sau $n < m$ sau $m < n$ poate avea loc, lăsând liberă posibilitatea ca nici una dintre aceste relații să nu fie satisfăcută. Vom arăta că, în adevăr, $<$ este o ordine totală strictă pe \mathbb{N} , dar pentru aceasta vom avea nevoie mai întâi de o tehnică de demonstrație, numită *principiu inducției matematice*.

Teorema 1.1.3.2. (Principiu inducției matematice)

Fie $P(x)$ o proprietate astfel încât:

- (i) $P(0)$;
- (ii) pentru orice $k \in \mathbb{N}$, $P(k)$ implică $P(S(k))$.

Atunci, P este satisfăcută de toate numerele naturale.

¹³ Aceste concepte, tranzitivitate, ireflexivitate și ordine totală strictă, precum și altele, vor fi discutate în detaliu în Secțiunea 1.2.

Demonstrație. (i) și (ii) arată că mulțimea $A = \{k \in \mathbb{N} | P(k)\}$ este inductivă. Cum \mathbb{N} este cea mai mică mulțime inductivă, obținem $\mathbb{N} \subseteq A$, ceea ce demonstrează teorema. \square

Aplicarea Prinzipiului inducției în situații concrete constă în parcurgerea următoarelor etape:

- se alege (fixeză) proprietatea P despre care se dorește a se arăta că este satisfăcută de toate numerele naturale;
- se verifică faptul că P este satisfăcută de 0 (această etapă se numește *baza inducției*);
- 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 numește *pasul inductiv*).

Dacă atât baza inducției, cât și pasul inductiv au fost parcuse cu succes, atunci, în baza Prinzipiului inducției, deducem că proprietatea P este satisfăcută de toate numerele naturale.

Demonstrațiile ce utilizează exclusiv Prinzipiul inducției sau variante ale acestuia, aşa cum vom prezenta în continuare, sunt numite *demonstrații prin inducție (matematică)*.

Teorema 1.1.3.3.

- (1) Relația $<$ pe \mathbb{N} este ordine totală strictă.
- (2) Orice submulțime nevidă a mulțimii \mathbb{N} are cel mai mic element în raport cu relația $<$ ¹⁴.

Demonstrație. (1) Conform discuției de mai sus, ceea ce ne rămâne de arătat este că oricare două numere naturale n și m sunt comparabile. Demonstrația acestui fapt o vom face prin inducție matematică arătând că proprietatea

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

este satisfăcută de orice număr natural:

- $P(0)$: vom arăta, utilizând iarăși inducția, că 0 este comparabil cu orice $m \in \mathbb{N}$:
 - evident, 0 este comparabil cu 0 ($0 = 0$);
 - presupunem că 0 este comparabil cu m . Ca urmare, ori $0 \in m$, ori $0 = m$. În ambele cazuri avem $0 \in S(m) = m \cup \{m\}$, deci 0 este comparabil cu $S(m)$.

¹⁴ Această proprietate va sta la baza definiției mulțimilor bine ordonate.

Principiul inducției matematice asigură atunci că 0 este comparabil cu orice număr natural m :

- presupunem că n este comparabil cu orice $m \in \mathbb{N}$. Vom arăta prin inducție că $S(n)$ este comparabil cu orice $m \in \mathbb{N}$:
 - 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 $m = n$. În primul caz are loc $S(m) \in S(n)$, iar în cel de-al doilea $S(m) = S(n)$, ceea ce arată că $S(m)$ și $S(n)$ sunt comparabile.

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

Am obținut astfel, în baza Principiului inducției matematice, că orice două numere naturale sunt comparabile; ca urmare, $<$ este ordine totală strictă.

(2) Să arătăm acum că orice submulțime nevidă a mulțimii \mathbb{N} are un cel mai mic element în raport cu ordinea $<$. Fie $M \subseteq \mathbb{N}$ nevidă și $n \in M$. Mulțimea $S(n) \cap M$ este nevidă și, deoarece $S(n)$ este număr natural, urmează că $S(n) \cap M$ admite cel mai mic element. Este ușor de văzut că acest cel mai mic element este de fapt și cel mai mic element al mulțimii M în raport cu ordinea $<$. \square

Corolarul 1.1.3.2. Dacă o submulțime de numere naturale are un element maximal, atunci acesta este unic (el fiind, astfel, cel mai mare element al acesteia).

Demonstrație. Dacă o submulțime de numere naturale ar avea mai mult de un element maximal, atunci mulțimea acestor elemente maximale ar admite un cel mai mic element care ar contrazice statutul de element maximal al acestuia. \square

Faptul că orice submulțime nevidă a mulțimii \mathbb{N} are un cel mai mic element permite stabilirea unor noi variante ale Principiului inducției. Prezentăm întâi câteva generalizări ușoare.

Principiul inducției poate fi aplicat pe submulțimi nevide (bine precizate) ale lui \mathbb{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 $k < n$.
- În adevăr, considerând proprietatea Q dată prin $Q(k) = P(k)$, pentru $k \leq n$, și $Q(k)$ satisfăcută pentru orice $k > n$, (a) și (b) conduc la:
- (c) $Q(0)$;
 - (d) $Q(k)$ implică $Q(S(k))$, pentru orice $k \in \mathbb{N}$,

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

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

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

Evident, se pot imagina și alte tipuri de submulțimi 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 $\{k \in \mathbb{N}|k \geq n_0\} \subseteq \{k \in \mathbb{N}|P(k)\}$ (cititorul este invitat să argumenteze această variantă a Principiului inducției).

Pentru variantele pe care le vom prezenta în continuare vom utiliza din plin Teorema 1.1.3.3(2). Dacă A este o mulțime 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 $k' \in A$. În adevăr, mulțimea $\{a \in A|k < a\}$ este nevidă și are cel mai mic element, care este succesorul imediat al lui k .

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

- (i) $P(0)$;
- (ii) pentru orice $k \in \mathbb{N}$, $((\forall j \leq k)(P(j)) \Rightarrow P(S(k)))$.

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

Demonstrație. Presupunem prin contradicție că există un număr natural n ce nu satisface P . Fie A mulțimea tuturor acestor numere. A este nevidă dar nu conține 0 (deoarece are loc (i)). Proprietatea P este satisfăcută de toate numerele naturale mai mici decât \perp_A și, atunci, (ii) conduce la faptul că P este satisfăcută de \perp_A , ceea ce intră în contradicție cu $\perp_A \notin A$. \square

Propoziția 1.1.3.2. Fie $A \subseteq \mathbb{N}$ și $P(x)$ o proprietate astfel încât:

- (i) $P(\perp_A)$;
- (ii) pentru orice $k \in A$ ce nu este cel mai mare element al mulțimii A ,

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

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

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

Demonstrație. Considerăm proprietatea $Q(x)$ dată prin:

- (1) pentru orice $x \in A$, Q este satisfăcută de x dacă și numai dacă P este satisfăcută de x ;
- (2) Q este satisfăcută de orice $x \in \mathbf{N} - A$.

Utilizând Propoziția 1.1.3.1, arătăm că proprietatea Q satisfac ipotezele Prinzipiului inducției:

- 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 și 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) și ipoteza propoziției conduc la faptul că $S(k)$ satisfac 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 satisfac $m \leq k$, și atunci, pe baza ipotezei inducțive, urmează că m satisfac Q . Ipoteza propoziției și (1) conduc la faptul că $S(k)$ satisfac Q .

Prinzipiul inducției aplicat proprietății 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 propoziția. \square

Pentru Prinzipiul din Propoziția 1.1.3.2 se poate da o variantă asemănătoare celei din Propoziția 1.1.3.1. Demonstrația acesteia o lăsăm în seama cititorului.

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

- (i) $P(\perp_A)$;
- (ii) pentru orice $k \in A$ ce nu este cel mai mare element al mulțimii 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 $n \in A$.

1.1.3.4. Mulțimi finite și infinite

Avem acum posibilitatea de a introduce într-o manieră simplă și elegantă conceptul de *mulțime finită*¹⁵.

Definiția 1.1.3.4. O mulțime A este numită *finită* dacă există un număr natural n astfel încât A și n sunt echipotente. Vom mai spune în acest caz că A are n elemente și vom nota $|A| = n$. Dacă A nu este finită, atunci vom spune că ea este *infinite*.

Secvențele sunt “înșiruirii” finite sau infinite de elemente; ele apar frecvent în considerații matematice. În analiza matematică secvențele infinite sunt uzuale numite *șiruri*.

Definiția 1.1.3.5. Se numește *secvență* de elemente peste A orice funcție f cu domeniul un număr natural sau \mathbf{N} și cu valori în A . Dacă domeniul este un număr natural n , atunci secvența este numită *finită* sau de *lungime* n ; altfel ea este numită *infinite*.

Secvențele sunt funcții și, ca urmare, putem vorbi despre *domeniul* și *codomeniul* unei secvențe. Domeniul va fi întotdeauna un număr natural sau \mathbf{N} . Există o unică secvență de lungime 0, și anume funcția vidă; ea va fi numită *secvență vidă*.

Uzual, secvențele infinite sunt noteate 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:

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

unde $a_i = f(i)$, f fiind secvența în cauză. Uneori croșetele “⟨” și “⟩” sunt înlocuite prin paranteze rotunde sau accolade, iar în cazul secvențelor finite ele sunt eliminate cu precădere¹⁶.

1.1.3.5. Recursie

Definirea operațiilor de bază pe mulțimea numerelor naturale, cum ar fi adunarea și înmulțirea, constituie un alt obstacol pe care trebuie să îl trecem. Menționăm întâi că o *operație binară* pe o mulțime A nu este altceva decât o funcție de la $A \times A$ cu valori în A .

Caracterul inductiv al mulțimii numerelor naturale face loc ideii definirii “inductive” de funcții al căror domeniu este această mulțime, dar nu numai. De exemplu, adunarea poate fi definită prin:

¹⁵Există abordări ale acestui concept ce nu fac apel la numere naturale. Însă aceste abordări sunt în mod necesar artificiale.

¹⁶Atunci când sunt utilizate parantezele rotunde, distincția dintre secvențe și familii indexate de mulțimi (ce vor fi introduse în Secțiunea 1.2.4) urmează a fi dedusă din context. De fapt, trebuie să remarcăm că în cazul în care A este o familie de mulțimi, secvențele peste A sunt cazuri particulare de familii indexate de mulțimi (mulțimea de index este un număr natural sau mulțimea \mathbf{N}).

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

În cazul funcțiilor, astfel de proceduri (metodologii, scheme de definiție) sunt numite *definiții recursive/recurente sau scheme de recursie/recurență*. În general, se procedează astfel:

- se definește funcția pentru 0;
- dacă funcția a fost definită pentru $n \in \mathbb{N}$, atunci se arată cum se definește pentru $S(n)$.

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

Teorema 1.1.3.4. (Teorema recursiei)

Fie A o mulțime, $a \in A$ și $h : \mathbb{N} \times A \rightarrow A$ o funcție. Atunci există o unică funcție $f : \mathbb{N} \rightarrow A$ astfel încât:

- $f(0) = a$;
- $f(S(n)) = h(n, f(n))$, pentru orice $n \in \mathbb{N}$.

Demonstrație. Fie F mulțimea tuturor funcțiilor g al căror domeniu este un număr natural diferit de 0, ce iau valori în A , și care verifică:

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

Este ușor de văzut că F este mulțime nevidă (F conține funcția $g : \{0\} \rightarrow A$ dată prin $g(0) = a$).

Arătăm că oricare două funcții $g, g' \in F$ sunt compatibile. Fie $g, g' \in F$. Există numerele naturale $k, m \in \mathbb{N} - \{0\}$ astfel încât $\text{Dom}(g) = k$ și $\text{Dom}(g') = m$. Presupunem că $k \leq m$. Prin inducție finită 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

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

În baza Prinzipiului inducției finitare obținem că g și g' sunt compatibile. Ca urmare, F este mulțime de funcții compatibile, ceea ce conduce la faptul că există funcția $f = \bigcup F$ cu domeniul $\bigcup_{g \in F} \text{Dom}(g)$.

Arătăm că $\text{Dom}(f) = \mathbb{N}$. Domeniul funcției f este submulțime a mulțimii \mathbb{N} . Dacă presupunem că $\mathbb{N} - \text{Dom}(f)$ este nevidă, atunci ea va avea un cel mai mic element, fie acesta x . Este clar că $x > 0$ și astfel există y astfel încât $x = S(y)$. Numărul y este în domeniul funcției f și, prin urmare, există $g \in F$ astfel încât $y \in \text{Dom}(g)$. Mai mult, nu există $z \geq x$ astfel încât $z \in \text{Dom}(g)$. Adică, $\text{Dom}(g) = x$. Vom arăta că există o funcție $g' \in F$ al cărei domeniu conține x .

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

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

Arătăm că g' satisfacă (*):

- $g'(0) = g(0) = a$;
- fie z astfel încât $S(z) \in \text{Dom}(g')$. Dacă $S(z) \in \text{Dom}(g)$, 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 definiția funcției g' urmează că

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

Ca urmare g' satisfacă (*) și, deci, $g' \in F$. Aceasta contrazice presupunerea conform căreia $x \notin \text{Dom}(f)$ și, deci, $\text{Dom}(f) = \mathbb{N}$.

Arătăm că funcția f satisfacă (i) și (ii) ale teoremei. Conform definiției ei, $f(0) = g(0) = a$, pentru orice $g \in F$ și, deci, f satisfacă (i). Fie $x \in \text{Dom}(f)$. Atunci există $g \in F$ astfel încât $S(x) \in \text{Dom}(g)$. Deoarece F este mulțime de funcții compatibile urmează că

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

ceea ce ne arată că f satisfacă (ii).

Unicitatea funcției f se obține astfel. Dacă ar exista o altă funcție g ce satisfacă (i) și (ii), atunci prin inducție după $n \in \mathbb{N}$ arătăm că $f(n) = g(n)$, ceea ce va conduce la $f = g$. În adevăr, $f(0) = a = g(0)$ și, 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 $n \in \mathbb{N}$, ceea ce arată că $f = g$. □

Funcțiile cu domeniul \mathbb{N} sunt secvențe infinite și reciproc. Ca urmare, Teorema recursiei poate fi reformulată în termeni de secvențe, astfel¹⁷:

- dată o mulțime A , $a \in A$ și o funcție $h : \mathbb{N} \times A \rightarrow A$, există o unică secvență infinită $\langle a_i | i \geq 0 \rangle$ astfel încât:

¹⁷Și celelalte variante de recursie, ce vor fi prezentate pe parcursul acestui capitol, pot fi reformulate în termeni de secvențe.

- $a_0 = a$;
- $a_{n+1} = h(n, a_n)$, pentru orice $n \in \mathbb{N}$.

Așadar, a defini recursiv o funcție cu domeniul \mathbb{N} revine la a defini o secvență infinită în care orice element al ei, exceptând primul, este "construit" pe baza elementului anterior:

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

Uneori, este bine de gândit această definiție și în modul următor: inițial (la pasul 0) funcția f este definită prin a , la pasul 1 funcția f este definită prin $h(0, f(0))$, la pasul 2 funcția f este definită prin $h(1, f(1))$ etc.

Operațiile binare, cum ar fi de exemplu adunarea, înmulțirea etc., sunt funcții de două variabile (definite pe produsul cartezian a două mulțimi). Teorema recursiei poate fi utilizată și pentru a defini astfel de funcții, pornind de la următoarea remarcă. Fie $f : A \times B \rightarrow C$ o funcție. Dacă fixăm unul dintre argumente, iar celălalt îl păstrăm variabil, de exemplu al doilea fix și primul variabil, atunci pentru fiecare valoare $b \in B$ dată celui de-al doilea argument obținem o funcție cu un singur argument, $f_b : A \rightarrow C$, cu proprietatea $f_b(a) = f(a, b)$, pentru orice $a \in A$. Atunci, a defini funcția f revine la a defini funcțiile f_b , pentru orice $b \in B$. Dacă B este mulțimea numerelor naturale, atunci putem utiliza Teorema recursiei pentru a defini o funcție $F : \mathbb{N} \rightarrow C^A$ astfel încât $F(b) = f_b$, pentru orice $b \in B = \mathbb{N}$; adică, F va defini funcțiile f_b pentru orice $b \in B$. Aceasta va fi de fapt ideea de demonstrație a următoarei teoreme.

Teorema 1.1.3.5. (Varianța parametrică a Teoremei recursiei)

Fie A și P mulțimi, iar $g : P \rightarrow A$ și $h : P \times \mathbb{N} \times A \rightarrow A$ funcții. Atunci, există o unică funcție $f : P \times \mathbb{N} \rightarrow A$ 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 $p \in P$ și $n \in \mathbb{N}$.

Demonstrație. Fie $f_0 : P \rightarrow A$ dată prin $f_0(p) = g(p)$ și $H : \mathbb{N} \times A^P \rightarrow A^P$ data prin $H(n, \varphi)(p) = h(p, n, \varphi(p))$, pentru orice $p \in P$, $n \in \mathbb{N}$ și $\varphi \in A^P$ (este ușor de văzut că aceste funcții există). Teorema recursiei va conduce la existența unei unice funcții $F : \mathbb{N} \rightarrow A^P$ astfel încât:

- $F(0) = f_0$;
- $F(S(n)) = H(n, F(n))$, pentru orice $n \in \mathbb{N}$.

Definim atunci $f : P \times \mathbb{N} \rightarrow A$ prin $f(p, n) = F(n)(p)$, pentru orice $p \in P$ și $n \in \mathbb{N}$. f este funcție și arătăm că ea satisfac 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 $p \in P$ și $n \in \mathbb{N}$.

Unicitatea funcției f se stabilește ca în Teorema 1.1.3.4. \square

Demonstrația Teoremei 1.1.3.5 ne arată clar că a defini recursiv o funcție f de la $P \times \mathbb{N}$ la A înseamnă a defini o secvență infinită de funcții de la P la A ,

$$f_0, f_1, f_2, \dots$$

Funcția f va fi atunci dată prin $f(p, n) = f_n(p)$, pentru orice $p \in P$ și $n \in \mathbb{N}$. Altfel spus, funcția f "condensează" secvența infinită de mai sus. Diferența dintre Teorema 1.1.3.4 și Teorema 1.1.3.5 este dată de "natura" elementelor secvenței infinite definite.

Prezentăm o nouă demonstrație a Teoremei 1.1.3.5, bazată pe fixarea primului argument al funcției f .

Pentru orice element $p \in P$, Teorema recursiei asigură existența unei unice funcții $f_p : \mathbb{N} \rightarrow A$ astfel încât:

- (i) $f_p(0) = g(p)$;
- (ii) $f_p(S(n)) = h_p(n, f_p(n))$, pentru orice $n \in \mathbb{N}$,

unde h_p este funcția $h_p(n, x) = h(p, n, x)$, pentru orice $n, x \in \mathbb{N}$.

Funcția $f = \bigcup_{p \in P} f_p$ verifică teorema.

Vom arăta acum modul în care se pot introduce riguroș operațiile de bază cu numere naturale.

Teorema 1.1.3.6.

- (1) Există o unică operație $+ : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ astfel încât:
 - (a) $+(m, 0) = m$, pentru orice $m \in \mathbb{N}$;
 - (b) $+(m, S(n)) = S(+ (m, n))$, pentru orice $m, n \in \mathbb{N}$.
- (2) Există o unică operație $\cdot : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ astfel încât:
 - (a) $\cdot(m, 0) = 0$, pentru orice $m \in \mathbb{N}$;
 - (b) $\cdot(m, S(n)) = +(\cdot(m, n), m)$, pentru orice $m, n \in \mathbb{N}$.
- (3) Există o unică operație $\hat{\cdot} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ astfel încât:
 - (a) $\hat{\cdot}(m, 0) = 1$, pentru orice $m \in \mathbb{N}$;
 - (b) $\hat{\cdot}(m, S(n)) = \cdot(\hat{\cdot}(m, n), m)$, pentru orice $m, n \in \mathbb{N}$.

(4) Există o unică operație $S' : \mathbb{N} \rightarrow \mathbb{N}$ astfel încât:

- (a) $S'(0) = 0$;
- (b) $S'(S(n)) = n$, pentru orice $n \in \mathbb{N}$.

(5) Există o unică operație $\bullet : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ astfel încât:

- (a) $\bullet(m, 0) = m$, pentru orice $m \in \mathbb{N}$;
- (b) $\bullet(m, S(n)) = S'(\bullet(m, n))$, pentru orice $m, n \in \mathbb{N}$.

Demonstrație. În Teorema 1.1.3.5 considerăm $A = P = \mathbb{N}$ și:

- $g(p) = p$ și $h(p, n, x) = S(x)$, pentru orice $p, n, x \in \mathbb{N}$ (pentru operația $+$);
- $g(p) = 0$ și $h(p, n, x) = +(x, p)$, pentru orice $p, n, x \in \mathbb{N}$ (pentru operația \cdot);
- $g(p) = 1$ și $h(p, n, x) = \cdot(x, p)$, pentru orice $p, n, x \in \mathbb{N}$ (pentru operația \wedge);
- $g(p) = 0$ și $h(p, n, x) = n$, pentru orice $p, n, x \in \mathbb{N}$ (pentru operația S');
- $g(p) = p$ și $h(p, n, x) = S'(x)$, pentru orice $p, n, x \in \mathbb{N}$ (pentru operația \bullet).

Unica funcție a cărei existență este asigurată de această teoremă este întocmai $+$, respectiv, \cdot , \wedge , S' , \bullet . \square

Operația $+(\cdot, \wedge, \bullet)$ este numită *operația de adunare (înmulțire, ridicare la putere, diferență, scădere aritmetică)* pe \mathbb{N} ; ușual vom folosi notația infix pentru ele, adică vom scrie $m + n$ ($m \cdot n$, $m \wedge n$, $m \bullet n$) în loc de $+(m, n)$ ($\cdot(m, n)$, $\wedge(m, n)$, $\bullet(m, n)$). Semnele operațiilor de înmulțire și de ridicare la putere se omit cu precădere, utilizându-se mn și m^n pentru $m \cdot n$ și, respectiv, $m \wedge n$. De la Teorema 1.1.3.6 rezultă că are loc:

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

ceea ce permite utilizarea notației $m + 1$ pentru $S(m)$, care este mult mai intuitivă și ușor de manipulat. (a1), (a2), (i1), (i2), (p1), (p2), (d1) și (d2) din Teorema 1.1.3.6 pot fi reformulate astfel:

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

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

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

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

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

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

$$(d1') m \bullet 0 = m;$$

$$(d2') m \bullet (n + 1) = S'(m \bullet n),$$

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

Introducerea mulțimii numerelor naturale, împreună cu operațiile de bază cu acestea, constituie un obiectiv major pe care considerăm că l-am dus la bun sfârșit. Din acest punct mai departe vom presupune că cititorul este familiarizat cu proprietățile de bază ale numerelor naturale și operațiile cu acestea. De asemenea, presupunem că este cunoscut modul de introducere a celorlalte sisteme de numere: *întregi* (\mathbf{Z}), *raționale* (\mathbf{Q}), *reale* (\mathbf{R}) și *complexe* (\mathbf{C}), precum și a operațiilor de bază pe acestea (pentru detalii, indicăm [207]). Prin \mathbf{Z}^* vom nota $\mathbf{Z} - \{0\}$, prin \mathbf{Z}_+ vom nota $\{x \in \mathbf{Z} | x \geq 0\}$, iar prin \mathbf{Z}_+^* vom nota $\{x \in \mathbf{Z} | x > 0\}$. Aceste notații vor fi extinse și la \mathbf{Q} și \mathbf{R} , iar notația “ $*$ ” și la \mathbf{C} .

1.2. Relații și funcții

În Secțiunea 1.1 s-a introdus în manieră axiomatică conceptul de mulțime și, pe baza acestuia, concepțele de pereche ordonată, de relație, de funcție și de număr natural. Toate acestea sunt fundamentale în matematică, ele constituind baza tuturor celorlalte concepții matematice.

În această secțiune vom aprofunda studiul acestor concepții de bază.

1.2.1. Relații

Relațiile binare (Secțiunea 1.1.1) sunt mulțimi de perechi ordonate. Mulțimea vidă este relație, numită *relația vidă*. Notația $a \rho b$ este utilizată frecvent pentru a specifica faptul că (a, b) este element al relației ρ .

Deoarece relații sunt mulțimi, putem construi reuniunea, intersecția, diferența și complementara lor, care sunt relații; egalitatea relațiilor este egalitate de mulțimi. $Dom(\rho)$ și $Cod(\rho)$ desemnează domeniul, respectiv, codomeniul relației ρ .

Exemplul 1.2.1.1. Fie A și B mulțimi.

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

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

este numită *relația de egalitate* pe A sau *identitatea* pe A sau *diagonala* lui $A \times A$ (frecvent notată și prin ι_A ¹⁸).

¹⁸Notația ι_A este de preferat notației $=_A$ care poate reduce lizibilitatea textului, cum ar fi de exemplu în scrierile de forma “ $\rho ==_A$ ”.

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

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

este numită *relația de apartenență* pe A .

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

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

este numită *relația de incluziune* pe A . Înlocuind \subseteq prin \subset , obținem *relația de incluziune strictă* pe A , notată prin \subset_A .

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

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

este numită *relația completă* de la A la B . În cazul $A = B$, notația $\omega_{A,B}$ va fi simplificată la ω_A , care este numită *relația completă* pe A .

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

Definiția 1.2.1.1. Fie ρ o relație binară și A o mulțime. *Restricția relației ρ la A* este relația binară notată $\rho|_A$ și dată prin

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

Relația $\rho|_A$ este intersecția a două relații, $\rho|_A = \rho \cap \omega_A$. Acest fapt permite dezvoltarea unor proprietăți ale relației $\rho|_A$ uzând de diverse proprietăți ale intersecției de relații.

Evident, restricția unei relații binare se poate face restrângând doar domeniul sau doar codomeniul acesteia, sau restrângându-le pe ambele, dar în mod diferit. În cazul Definiției 1.2.1.1, atât domeniul, cât și codomeniul sunt restricționate prin intermediul aceleiași mulțimi A .

Este adesea utilă să reprezintă grafic relațiile binare. Reprezentarea grafică a unei relații ρ se face printr-un graf orientat ale cărui noduri sunt etichetate cu elementele mulțimii $Dom(\rho) \cup Cod(\rho)$. Pentru fiecare pereche $(a, b) \in \rho$ se trasează un arc de la nodul cu eticheta a la nodul cu eticheta b . În mod frecvent nodurile sunt identificate prin etichetele lor (distincția nod-etagă fiind esențială atunci când noduri diferite sunt etichetate cu aceeași etagă). În Figura 1.2 este reprezentată grafic relația

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

punând în evidență atât reprezentarea cu noduri etichetate, cât și cea în care nodurile sunt identificate cu etichetele lor.

Următoarea propoziție prezintă câteva proprietăți elementare ale domeniului și codomeniului unei relații.

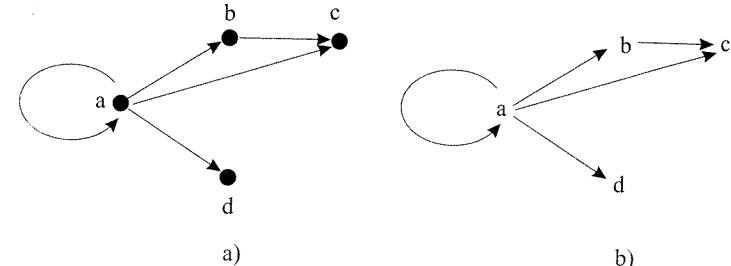


Figura 1.2: Reprezentări grafice ale aceleiași relații binare

Propoziția 1.2.1.1. Fie ρ și σ două relații binare. Atunci au loc următoarele proprietăți:

- (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) $Dom(\rho - \sigma) \subseteq Dom(\rho - \sigma);$
- (6) $Cod(\rho - \sigma) \subseteq Cod(\rho - \sigma);$
- (7) dacă $\rho \subseteq \sigma$, atunci $Dom(\rho) \subseteq Dom(\sigma)$ și $Cod(\rho) \subseteq Cod(\sigma)$.

Atragem atenția asupra inclusiunilor din Propoziția 1.2.1.1. Ele pot fi stricte. De exemplu, dacă $\rho = \{(a, b)\}$ și $\sigma = \{(a, c)\}$, unde $b \neq c$, atunci $Dom(\rho \cap \sigma) = \emptyset$ și $Dom(\rho) \cap Dom(\sigma) = \{a\}$. Ca urmare, $Dom(\rho \cap \sigma) \subset Dom(\rho) \cap Dom(\sigma)$.

Produsul și inversa relațiilor binare sunt “operații” specifice de mare importanță în studiul acestora.

Definiția 1.2.1.2. Fie ρ și σ două relații binare. Relația binară notată $\rho \circ \sigma$ și dată prin

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

este numită *produsul* relațiilor ρ și σ .

Este clar că pentru orice două relații ρ și σ , produsul lor este relație binară (deci, Definiția 1.2.1.2 este consistentă). Dacă ρ este relație de la A la B , iar σ de la C la D , atunci $\rho \circ \sigma$ este relație de la A la D . În plus, dacă $Cod(\rho) \cap Dom(\sigma) = \emptyset$, atunci $\rho \circ \sigma = \emptyset$. În particular $\rho \circ \sigma = \emptyset$, dacă $\rho = \emptyset$ sau $\sigma = \emptyset$.

Următoarea propoziție prezintă câteva proprietăți de bază ale produsului de relații.

Propoziția 1.2.1.2. Fie ρ , σ și θ relații binare, iar A și B mulțimi. Atunci au loc următoarele proprietăți:

- (1) $\text{Dom}(\rho \circ \sigma) \subseteq \text{Dom}(\rho)$;
- (2) $\text{Cod}(\rho \circ \sigma) \subseteq \text{Cod}(\sigma)$;
- (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 $\rho \circ \sigma \subseteq \rho \circ \theta$ și $\sigma \circ \rho \subseteq \theta \circ \rho$;
- (10) $\iota_A \circ \rho \subseteq \rho$ și $\rho \circ \iota_B \subseteq \rho$. În plus, $\iota_A \circ \rho = \rho$ dacă și numai dacă $\text{Dom}(\rho) \subseteq A$ și, $\rho \circ \iota_B = \rho$ dacă și numai dacă $\text{Cod}(\rho) \subseteq B$.

Demonstrație. Vom demonstra doar (10). Fie $(a, b) \in \iota_A \circ \rho$. Atunci, există c astfel încât $(a, c) \in \iota_A$ și $(c, b) \in \rho$. Conform definiției relației ι_A , urmează că $a = c$ și astfel, $(a, b) \in \rho$. Am obținut astfel inclusiunea $\iota_A \circ \rho \subseteq \rho$; inclusiunea $\rho \circ \iota_B \subseteq \rho$ se demonstrează similar.

Să presupunem acum că $\iota_A \circ \rho = \rho$ și să arătăm că $\text{Dom}(\rho) \subseteq A$. Fie $a \in \text{Dom}(\rho)$. Atunci există b astfel încât $(a, b) \in \rho$. Deoarece $\rho = \iota_A \circ \rho$, obținem $(a, b) \in \iota_A \circ \rho$ și astfel, va exista c astfel încât $(a, c) \in \iota_A$ și $(c, b) \in \rho$. Conform definiției relației ι_A avem $c = a$ și, prin urmare, $a \in A$. Am obținut astfel $\text{Dom}(\rho) \subseteq A$.

Reciproc, să presupunem că $\text{Dom}(\rho) \subseteq A$. 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 $\text{Dom}(\rho) \subseteq A$ urmează că $a \in A$ și, atunci, putem scrie $(a, b) \in \iota_A \circ \rho$. Am obținut astfel $\rho = \iota_A \circ \rho$.

Echivalența “ $\rho \circ \iota_B = \rho$ dacă și numai dacă $\text{Cod}(\rho) \subseteq B$ ” se demonstrează similar celei precedente. \square

Atragem atenția asupra inclusiunilor din Propoziția 1.2.1.2. Dacă, de exemplu, există $a \in \text{Dom}(\rho)$ astfel încât

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

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

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

atunci $\text{Cod}(\rho \circ \sigma) \subset \text{Cod}(\sigma)$.

Asociativitatea produsului de relații 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 și c astfel încât $(a, b) \in \rho$, $(b, c) \in \sigma$ și $(c, d) \in \theta$.

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

Definiția 1.2.1.3. Fie ρ o relație binară. *Inversa* relației ρ este relația notată ρ^{-1} și dată prin

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

Inversa unei relații ρ există întotdeauna, iar dacă ρ este relație de la A la B , atunci ρ^{-1} este relație de la B la A . Pentru anumite relații inverse are o notație consacrată. Următorul tabel prezintă câteva dintre aceste notări (A este o mulțime arbitrară):

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

Propoziția 1.2.1.3. Fie ρ și σ relații binare. Atunci au loc următoarele proprietăți:

- (1) $\text{Dom}(\rho^{-1}) = \text{Cod}(\rho)$;
- (2) $\text{Cod}(\rho^{-1}) = \text{Dom}(\rho)$;
- (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) $(\rho \circ \sigma)^{-1} = \sigma^{-1} \circ \rho^{-1}$.

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

Definiția 1.2.1.4. Fie ρ o relație binară, iar A și B mulțimi.

- (1) *Imaginea mulțimii* A prin ρ , notată $\rho(A)$, este mulțimea

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

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

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

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

Propoziția 1.2.1.4. Fie ρ și σ relații binare, iar A și B mulțimi. Atunci au loc următoarele proprietăți:

- (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 și numai dacă $\text{Dom}(\rho) \cap A = \emptyset$;
- (6) $\text{Dom}(\rho) \cap A \subseteq \rho^{-1}(\rho(A))$ și $\text{Cod}(\rho) \cap B \subseteq \rho(\rho^{-1}(B))$;
- (7) $(\rho \circ \sigma)(A) = \sigma(\rho(A))$.

Demonstrație. (1) Afirmația se obține pe baza echivalențelor:

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

pentru orice b .

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

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

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

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

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

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 $a \rho c$ și, pentru orice $b \in B$, $(b, c) \notin \rho$. Aceasta ne arată că $a \in A - B$ și astfel, $c \in \rho(A - B)$. Ca urmare, $\rho(A) - \rho(B) \subseteq \rho(A - B)$.

(5) urmează direct de la faptul că $b \in \rho(A)$ dacă și numai dacă există un element $a \in \text{Dom}(\rho) \cap A$ astfel încât $(a, b) \in \rho$.

(6) Pentru orice $a \in \text{Dom}(\rho) \cap A$, $\{b | a \rho b\} \subseteq \rho(A)$ și, în consecință,

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

În mod similar se obține și inclusiunea $\text{Cod}(\rho) \cap B \subseteq \rho(\rho^{-1}(B))$.

(7) Afirmația se obține pe baza echivalențelor:

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

pentru orice c . \square

Vom prezenta acum câteva tipuri importante de relații binare, precum și simple caracterizări ale acestora.

Definiția 1.2.1.5. Fie ρ o relație binară și A o mulțime.

(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 ¹⁹ dacă are loc

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

¹⁹ Conceptul de relație dirijată apare pentru prima dată în lucrarea lui Moore și Smith asupra unei teorii generale a conceptului de limită [150]. Acest concept de relație dirijată s-a dovedit ulterior de importanță foarte mare în informatică, în studiul semanticii limbajelor de programare și 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 mulțimea $Dom(\rho) \cup Cod(\rho)$.

Teorema 1.2.1.1. Fie ρ o relație binară și $A = Dom(\rho) \cup Cod(\rho)$. Atunci, au loc următoarele proprietăți:

- (1) ρ este reflexivă dacă și numai dacă $\iota_A \subseteq \rho$;
- (2) ρ este ireflexivă dacă și numai dacă $\iota_A \cap \rho = \emptyset$;
- (3) ρ este simetrică dacă și numai dacă $\rho = \rho^{-1}$;
- (4) ρ este antisimetrică dacă și numai dacă $\rho \cap \rho^{-1} \subseteq \iota_A$;
- (5) ρ este asimetrică dacă și numai dacă $\rho \cap \rho^{-1} = \emptyset$;
- (6) ρ este tranzitivă dacă și numai dacă $\rho \circ \rho \subseteq \rho$;
- (7) ρ este conexă dacă și numai dacă $\rho \cup \rho^{-1} \cup \iota_A = A \times A$.

Demonstrație. 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 $(a, b) \in \rho$ și $(b, a) \in \rho$. Relația ρ fiind antisimetrică, deducem $a = b$ și astfel, $(a, b) \in \iota_A$. Am obținut astfel $\rho \cap \rho^{-1} \subseteq \iota_A$. □

Fie ρ o relație pe A . Definim următoarele relații:

- $\rho^0 = \iota_A$;
- $\rho^{n+1} = \rho^n \circ \rho$, 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 relație pe A . Atunci:

- (1) ρ^+ este cea mai mică relație tranzitivă pe A ce include ρ ;
- (2) ρ^* este cea mai mică relație reflexivă și tranzitivă pe A ce include ρ .

Demonstrație. (1) Conform definiției, ρ^+ include ρ . În 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 relație tranzitivă ce include ρ , atunci ea trebuie să includă și ρ^2 . Acum, incluzând ρ și ρ^2 , va trebui să includă și 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) $\iota_A \subseteq \rho^*$ și astfel, ρ^* este reflexivă. Restul se arată ca la (1). □

Relația ρ^+ este numită *închiderea tranzitivă a relației ρ* , iar ρ^* *închiderea reflexivă și tranzitivă a relației ρ* (asupra acestor relații vom reveni în Secțiunea 1.3.2).

Reprezentarea grafică a relațiilor reflexive se simplifică, în mod frecvent, prin eliminarea arcelor de la nod la el însuși. O simplificare mult mai consistentă se face pentru relații tranzitive. Dacă ρ este o relație tranzitivă, atunci reprezentarea grafică a ei se substitue prin reprezentarea grafică doar a perechilor $(a, b) \in \rho$ pentru care nu există c diferit de a și b cu $(a, c) \in \rho$ și $(c, b) \in \rho$. De exemplu, relația

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

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

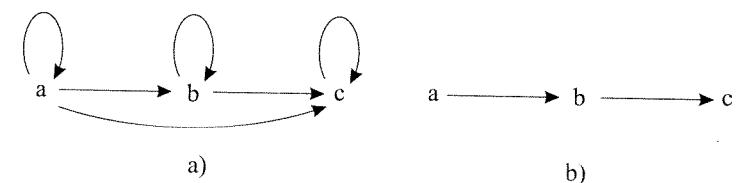


Figura 1.3: Reprezentare simplificată a unei relații reflexive și tranzitive

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

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

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

1.2.2. Relații de echivalență

Clasa relațiilor de echivalență este una dintre cele mai importante clase de relații binare.

Definiția 1.2.2.1. Fie ρ o relație binară pe o mulțime A . Spunem că ρ este *relație de echivalență* pe A dacă ρ este reflexivă, simetrică și tranzitivă pe A .

Dacă ρ este relație de echivalență pe o mulțime A , atunci domeniul și codomeniul ei coincid cu A . Ca urmare a acestui fapt, vom spune adesea că “ ρ este relație de echivalență”, înțelegând că ea este relație de echivalență pe $A = \text{Dom}(\rho)$.

Este ușor de văzut că dacă ρ este relație de echivalență, atunci, pentru orice mulțime B , $\rho|_B$ este relație de echivalență pe B .

Relația vidă este relație de echivalență numai pe mulțimea vidă (pe mulțimi nevide ea este simetrică și tranzitivă, dar nu este reflexivă).

Exemplu 1.2.2.1. Fie A o mulțime nevidă. Relația binară $=_A$ (definită în Exemplul 1.2.1.1(1)) este relație de echivalență pe A .

Observația 1.2.2.1. Echipotența, introdusă în Secțiunea 1.1.1, verifică următoarele proprietăți:

- $A \sim A$, pentru orice mulțime A ;
- dacă $A \sim B$, atunci $B \sim A$, pentru orice mulțimi A și B ;
- dacă $A \sim B$ și $B \sim C$, atunci $A \sim C$, pentru orice mulțimi A , B și C .

Ca urmare, echipotența ar avea atributele unei relații de echivalență, dar nu este relație de echivalență deoarece clasa tuturor mulțimilor, peste care s-ar considera echipotență ca relație binară, nu este mulțime. Dacă însă considerăm echipotența peste o familie de mulțimi \mathcal{A} , să o notăm prin $\sim_{\mathcal{A}}$, atunci ea devine relație de echivalență pe \mathcal{A} .

Definiția 1.2.2.2. Fie ρ o relație de echivalență și a un element. Se numește *clasa de echivalență a lui a modulo/relativ la ρ* mulțimea $[a]_{\rho} = \{b | a \rho b\}$.

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

Dacă ρ este o relație de echivalență pe o mulțime A , atunci vom nota prin A/ρ mulțimea

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

și o vom numi *mulțimea cât sau factor indușă de A și ρ* (existența acestei mulțimi este asigurată de Axiomele părților și ale separării). În cazul $A = \emptyset$, $A/\rho = \emptyset$.

Lema 1.2.2.1. Fie ρ o relație de echivalență, iar a și b două elemente. Atunci au loc următoarele proprietăți:

- (1) $a \rho b$ dacă și numai dacă $[a]_{\rho} = [b]_{\rho}$;
- (2) $\neg(a \rho b)$ dacă și numai dacă $[a]_{\rho} \cap [b]_{\rho} = \emptyset$.

Demonstrație. (1) Să presupunem că $a \rho b$. Fie $x \in [a]_{\rho}$. Urmează că $x \rho a$ (în baza simetriei), $x \rho b$ (în baza tranzitivității) și $b \rho x$ (în baza simetriei); deci $x \in [b]_{\rho}$. Am obținut astfel $[a]_{\rho} \subseteq [b]_{\rho}$; similar se arată și cealaltă inclusiune.

Reciproc, dacă presupunem că $[a]_{\rho} = [b]_{\rho}$, atunci $b \in [a]_{\rho}$ (deoarece $b \in [b]_{\rho}$) și astfel, $a \rho b$.

(2) Dacă $\neg(a \rho b)$ atunci $[a]_{\rho} \neq [b]_{\rho}$ (de la (1)). Dacă mulțimile $[a]_{\rho}$ și $[b]_{\rho}$ ar conține elemente comune, fie c un astfel de element, atunci $a \rho c$ și $c \rho b$, ceea ce conduce la $a \rho b$; contradicție.

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

Există o foarte strânsă legătură între mulțimea partițiilor unei mulțimi nevide A , $\text{Part}(A)$, și mulțimea $E(A)$ a tuturor relațiilor de echivalență pe A . Înainte de a prezenta această legătură, introducem conceptul de rafinare între partiții, concept ce este “echivalent” inclusiunii între relații binare.

Definiția 1.2.2.3. Fie A o mulțime nevidă iar S_1 și S_2 două partiții ale lui A . Spunem că S_1 rafinează pe S_2 , și 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 mulțime nevidă.

- (1) Fie S o partiție a mulțimii A și ρ_S relația 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 relație de echivalență pe A .

- (2) Fie ρ o relație de echivalență pe A și S_{ρ} mulțimea tuturor claselor de echivalență induse de ρ . Atunci S_{ρ} este partiție a mulțimii A .
- (3) (a) Dacă S_1 și S_2 sunt partiții ale mulțimii A astfel încât $S_1 \leq S_2$, atunci $\rho_{S_1} \subseteq \rho_{S_2}$.
(b) Dacă ρ_1 și ρ_2 sunt relații de echivalență pe A astfel încât $\rho_1 \subseteq \rho_2$, atunci $S_{\rho_1} \leq S_{\rho_2}$.
- (4) (a) Dacă S este partiție a mulțimii A , atunci $S = S_{\rho_S}$.
(b) Dacă ρ este relație de echivalență pe A , atunci $\rho = \rho_{S_{\rho}}$.

Demonstrație. (1) și (2) necesită doar simple verificări și, ca urmare, vom trece la a demonstra celelalte proprietăți.

(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 $\rho_{S_1} \subseteq \rho_{S_2}$.

Afirmăția de la (3)(b) se obține similar celei precedente.

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

Fie $X \in S$. Considerăm un element arbitrar x din X (există un astfel de element, căci X este nevidă) și arătăm că $X = [x]_{\rho_S}$. Dacă $y \in X$, atunci $x \rho_S y$ și astfel, $y \in [x]_{\rho_S}$; dacă $y \in [x]_{\rho_S}$, atunci $x \rho_S y$ și, deci, x și y sunt în același bloc al partiției 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 echivalență, atunci există un unic bloc X ce conține x . Prinț-un raționament asemănător celui de mai sus se arată că $X = [x]_{\rho_S}$. De la acestea urmează $S = S_{\rho_S}$.

Afirmăția de la (4)(b) se obține similar celei precedente. \square

Putem spune deci că relațiile de echivalență pe o mulțime și partițiile aceleiai mulțimi sunt “descrieri diferite ale aceleiași entități matematice”. Atunci când lucrăm cu astfel de entități, este convenabil de a avea câte un “reprezentant” al fiecărei clase de echivalență. Suntem astfel conduși la a ne întreba asupra existenței unei mulțimi de reprezentanți pentru o partiție. Această chestiune a fost de altfel abordată în Secțiunea 1.1.1 și, aşa cum am menționat, o vom trata complet în secțiunea dedicată Axiomei alegerii.

Funcțiile injective “păstrează” relațiile de echivalență. Fie A o mulțime, ρ o relație pe A și $f : A \rightarrow B$ o funcție. Notăm prin $f(\rho)$ relația

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

Propoziția 1.2.2.1. Fie $f : A \rightarrow B$ o funcție și ρ o relație de echivalență pe A . Dacă f este funcție injectivă, atunci $f(\rho)$ este relație de echivalență pe $f(A)$.

Demonstrație. Reflexivitatea și simetria relației $f(\rho)$ se obțin imediat. Să discutăm tranzitivitatea.

Fie $(x, y), (y, z) \in f(\rho)$. Atunci, există $(a, b), (c, d) \in \rho$ astfel încât $f(a) = x$, $f(b) = y$, $f(c) = y$ și $f(d) = z$. Injectivitatea funcției f conduce la $b = c$, iar tranzitivitatea relației ρ conduce la $(a, d) \in \rho$ și, ca urmare, $(x, z) \in f(\rho)$. Deci $f(\rho)$ este tranzitivă. \square

Atragem atenția asupra necesității proprietății de injectivitate în a obține tranzitivitatea relației $f(\rho)$ (a se vedea demonstrația propoziției). De asemenea, atragem atenția asupra faptului că $f(\rho)$ este relație de echivalență pe $f(A)$ și nu în mod necesar pe B . Aceasta pentru că este posibil să se piardă proprietatea de reflexivitate.

Corolarul 1.2.2.1. Fie $f : A \rightarrow B$ o funcție și ρ o relație de echivalență pe A . Dacă f este funcție bijectivă, atunci $f(\rho)$ este relație de echivalență pe B .

Fie A o mulțime și $\rho, \theta \in E(A)$ astfel încât $\rho \subseteq \theta$. Simpla incluziune a relației ρ în θ ne spune că orice clasă de echivalență în raport cu ρ este inclusă în exact o clasă

de echivalență în raport cu θ . Ca urmare, o clasă de echivalență în raport cu θ este formată din una sau mai multe clase de echivalență în raport cu ρ . Grafic, această situație arată ca în Figura 1.4. Este justificat atunci să spunem că ρ este mai fină

$$\rho = \dots \text{ și } \dots$$

$$\theta = \dots$$

Figura 1.4: ρ este mai fină decât θ

decât θ .

Considerând acum mulțimea A/ρ , putem defini relația binară θ/ρ dată prin

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

pentru orice $a, b \in A$.

Diferența dintre θ și θ/ρ constă în aceea că θ acționează pe mulțimea A , în timp ce θ/ρ acționează pe A/ρ .

Propoziția 1.2.2.2. Fie A o mulțime și ρ, θ, θ_1 și θ_2 relații de echivalență pe A astfel încât $\rho \subseteq \theta \cap \theta_1 \cap \theta_2$. Atunci au loc următoarele proprietăți:

- (1) $\theta/\rho \in E(A/\rho)$;
- (2) orice relație de echivalență pe A/ρ este de forma θ'/ρ , unde $\theta' \in E(A)$ și $\rho \subseteq \theta'$;
- (3) $\rho/\rho = \iota_{A/\rho}$;
- (4) $A^2/\rho = (A/\rho)^2$ (A^2 este relația binară $A \times A$ care, evident, include ρ);
- (5) $\theta_1 \subset \theta_2$ dacă și numai dacă $\theta_1/\rho \subset \theta_2/\rho$;
- (6) $\theta_1 \neq \theta_2$ dacă și numai dacă $\theta_1/\rho \neq \theta_2/\rho$.

Demonstrație. Vom demonstra (2), (3) și (4), restul rămânând în grija cititorului.

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

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

pentru orice $a, b \in A$. Este ușor de văzut că θ' este relație de echivalență pe A .

Fie $a \rho b$. Atunci $[a]_\rho \psi [b]_\rho$, deoarece ψ este reflexivă. Conform definiției relației θ' , urmează $a \theta' b$. Ca urmare, $\rho \subseteq \theta'$. Ne rămâne de arătat că $\psi = \theta'/\rho$. Aceasta urmează însă imediat de la definițiile relațiilor θ' și θ'/ρ .

(3) Au loc relațiile:

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

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

(4) Au loc relațiile:

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

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

Atragem atenția asupra faptului că, în Propoziția 1.2.2.2(3), ρ/ρ este $\iota_{A/\rho}$ și nu $\iota_{A/\rho}$.

1.2.3. Funcții și operații

Funcțiile, o clasă foarte importantă de relații, au fost introduse în Secțiunea 1.1.1. Astfel, s-a spus că o relație f este *funcție* (sau *relație funcțională*) dacă satisfacă

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

Atunci când $\text{Dom}(f) = A$ și $\text{Cod}(f) \subseteq B$ se mai spune că f este *funcție de la A la B* sau că f este *funcție definită pe A și cu valori în B* și se notează $f : A \rightarrow B$.

În informatică în special, este important de considerat și *funcții parțiale de la A la B*, adică funcții f ce au proprietatea $\text{Dom}(f) \subseteq A$ și $\text{Cod}(f) \subseteq B$. Astfel de funcții mai sunt numite *funcții parțiale definite pe A și cu valori în B*. Dacă $\text{Dom}(f) \subset A$, atunci se mai spune că f este *strict parțială pe A*. În contrast, dacă $\text{Dom}(f) = A$, atunci se mai spune că f este *totală pe A*. Relația vidă este funcție parțială de la A la B. Așa cum s-a spus în Secțiunea 1.1.1, ea este funcție totală de la A la B doar dacă $A = \emptyset$.

Prin $(A \rightsquigarrow B)$ vom nota mulțimea tuturor funcțiilor parțiale de la A la B. Evident, $(A \rightarrow B) \subseteq (A \rightsquigarrow B)$.

O funcție parțială $f : A \rightsquigarrow B$ 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 și că f este definită în a*, și 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*, și se notează $f(a) \uparrow$.

Terminologia de funcție “parțială” este justificată de (2).

Noțiunea de imagine a unui element printr-o funcție parțială f poate fi extinsă la submulțimi în mod natural. De exemplu, dacă $C \subseteq A$, atunci vom numi *imaginarea mulțimii C prin f* mulțimea notată $f(C)$ și definită prin

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

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

A specifică o funcție parțială de la A la B înseamnă a preciza pentru fiecare element $a \in A$ dacă funcția este definită sau nu în a ; dacă ea este definită în a , atunci este necesară specificarea imaginii lui a prin respectiva funcție.

Vom prezenta în continuare câteva exemple de (clase de) funcții pe care le vom întâlni frecvent de-a lungul acestei cărți.

Exemplul 1.2.3.1.

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

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

este funcție strict parțială, iar relația g dată prin

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

este funcție (totală) de la A la B. Relația $h = \{(1, a), (1, b)\}$ nu este funcție parțială.

- (2) Funcția parțială $f : A \rightsquigarrow B$ dată prin $f(a) \uparrow$, pentru orice $a \in A$, este numită *funcție total nedefinită* de la A la B. Observăm că ea este de fapt relația vidă.

- (3) Fie $A \subseteq B$. Funcția $f : A \rightarrow B$ dată prin $f(a) = a$, pentru orice $a \in A$, este numită *funcție incluziune*. Uneori ea se mai notează prin $f : A \hookrightarrow B$. În cazul $B = A$ funcția incluziune se mai numește *funcție identică* pe A și se notează prin 1_A sau id_A sau chiar id . Observăm că ea coincide cu relația ι_A .

- (4) Relația completă de la A la B este funcție doar în cazul în care $A = \emptyset$ sau, dacă $A \neq \emptyset$, atunci B conține exact un element.

- (5) Fie A o mulțime și $B \subseteq A$. Funcția $f_B : A \rightarrow \{0, 1\}$ 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 *funcție caracteristică* a mulțimii B relativ la mulțimea A. Pentru $B = A$, f_B este *funcția constantă 1*.

- (6) Funcțiile de tipul $f : \{0, 1\}^n \rightarrow \{0, 1\}$, unde $n \geq 1$, sunt numite *funcții Booleene*.

(7) Fie $n \geq 1$, A_1, \dots, A_n mulțimi nevide și $1 \leq i \leq n$. Funcția

$$pr_i : A_1 \times \dots \times A_n \rightarrow A_i$$

dată prin $pr_i(a_1, \dots, a_n) = a_i$, pentru orice $(a_1, \dots, a_n) \in A_1 \times \dots \times A_n$, se numește *funcția de i -proiecție* asociată produsului $A_1 \times \dots \times A_n$.

Dacă am considera cazul în care una dintre mulțimile A_1, \dots, A_n este mulțimea vidă, atunci funcția de i -proiecție ar fi funcția vidă, chiar și atunci când A_i ar fi nevidă.

(8) Funcțiile de tipul $P : A \rightarrow \{0, 1\}$ sunt numite și *predicate pe A*. Uzual, 0 este interpretat ca fiind valoarea de adevăr “fals”, iar 1 ca fiind valoarea de adevăr “adevărat”. Evident, în locul mulțimii $\{0, 1\}$ se poate alege orice altă mulțime cu două elemente.

(9) În informatică, funcțiile de tipul $f : A \rightarrow B$ date prin

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

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

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

O funcție parțială de la A la B poate fi gândită ca un dispozitiv (Figura 1.5) care, primind la intrare un element $a \in A$, funcționează și la ieșire emite elementul $b = f(a)$, dacă f este definită în a . În ipoteza în care f nu este definită în a , putem gândi că

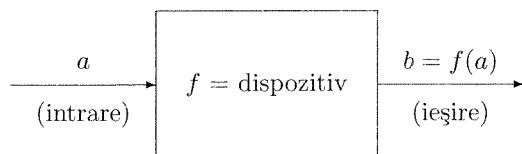


Figura 1.5: Reprezentare schematică a unei funcții

acest dispozitiv ori se oprește fără a emite nimic la ieșire (de exemplu, dispozitivul se blochează), ori lucrează la nesfârșit (din punct de vedere teoretic). Un model practic, adesea considerat în literatura informatică, este cel de automat ce poate oferi o ceașcă de ceai sau o cafea la introducerea unei monede de un anumit tip, să spunem m_1 și respectiv m_2 (presupunem, de exemplu, că ele au prețuri diferite). Considerând că automatul funcționează perfect, pentru m_1 el va oferi un ceai, pentru m_2 , o cafea, iar pentru alte tipuri de monede se va bloca.

Funcțiile fiind relații, putem construi reuniunea, intersecția și diferența lor; egalitatea de funcții este egalitate de relații. Evident, reuniunea a două funcții nu este, în mod necesar, funcție. Intersecția sau diferența a două funcții este funcție.

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

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

Unii autori consideră notația $(a)f = b$ pentru $(a, b) \in f$, și atunci produsul de relații nu se mai schimbă notațional:

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

Cum funcțiile sunt relații, notația f^n se deduce imediat de la acestea. Mai exact, dacă $f : A \rightsquigarrow A$ este o funcție parțială, atunci:

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

Propoziția 1.2.3.1. Dacă $f : A \rightsquigarrow B$ și $g : B \rightsquigarrow C$ sunt funcții parțiale, atunci $g \circ f$ este funcție parțială de la A la C .

Demonstrație. Fie $(a_1, b_1), (a_2, b_2) \in g \circ f$ astfel încât $a_1 = a_2$. Atunci, există c_1 și c_2 cu proprietatea $(a_1, c_1), (a_2, c_2) \in f$ și $(c_1, b_1), (c_2, b_2) \in g$. Egalitatea $a_1 = a_2$, combinată cu faptul că f este funcție, conduce la $c_1 = c_2$ care, la rândul ei, ne arată că $b_1 = b_2$ deoarece g este funcție. Deci $g \circ f$ este funcție parțială. \square

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 $g \circ f = \emptyset$. Grafic, compunerea poate fi reprezentată ca în Figura 1.6.

Relativ la inversa unei funcții parțiale $f : A \rightsquigarrow B$ putem spune că aceasta nu este neapărat funcție parțială. De exemplu, dacă $f(a_1) = f(a_2) = b$ este o funcție de la $A = \{a_1, a_2\}$ la $B = \{b\}$, atunci f^{-1} este relația $\{(b, a_1), (b, a_2)\}$ ce nu este funcție parțială (presupunând $a_1 \neq a_2$).

Concepțele de injectivitate și surjectivitate se extind și la funcții parțiale, exact ca în Definiția 1.1.1.6. Astfel, o funcție parțială f de la A la B este *injectivă* dacă are loc

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

și este *surjectivă* dacă are loc

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

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

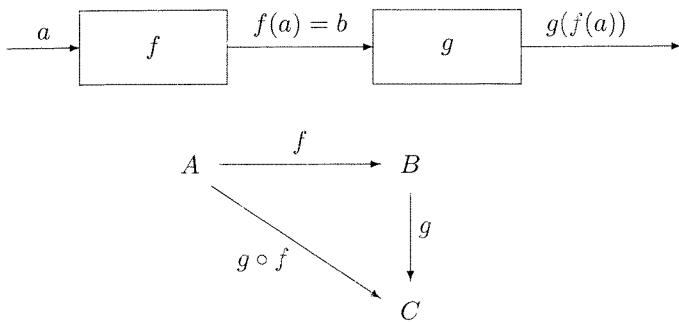


Figura 1.6: Reprezentare grafică a compunerii de funcții

O funcție parțială este *bijectivă* dacă este injectivă și surjectivă. Spre deosebire de cazul funcțiilor totale bijective, inversa unei funcții parțiale bijective este totală și injectivă, dar nu în mod necesar surjectivă.

Funcțiile bijective de la o mulțime A la ea însăși se mai numesc și *permute* ale mulțimii A . Când A este finită, $A = \{a_1, \dots, a_n\}$, permutările $f : A \rightarrow A$ se mai notează prin

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

Propoziția 1.2.3.2. Fie $f : A \rightsquigarrow B$ o funcție parțială.

- (1) Dacă f este injectivă, atunci f^{-1} este funcție parțială injectivă.
- (2) Dacă f este bijectivă, atunci f^{-1} este funcție totală injectivă.
- (3) Dacă f este totală și bijectivă, atunci f^{-1} este totală și bijectivă.

Demonstrație. (1) Să presupunem că f este injectivă. Fie $(b_1, a_1) \in f^{-1}$ și $(b_2, a_2) \in f^{-1}$ 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 funcție parțială.

Pentru a stabili injectivitatea funcției f^{-1} este suficient să observăm, în raționamentul de mai sus, că dacă vom considera $a_1 = a_2$, atunci faptul că f este funcție conduce la $b_1 = b_2$; adică f^{-1} este injecție.

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

(3) Dacă f este totală și bijectivă, atunci de la (2) obținem că f^{-1} este totală și injectivă. Totalitatea funcției f asigură surjectivitatea funcției f^{-1} . Deci, f^{-1} este totală și bijectivă. \square

Următoarea propoziție, ce este ușor de verificat, prezintă câteva proprietăți elementare ale compunerii funcțiilor parțiale.

Propoziția 1.2.3.3. Fie $f : A \rightsquigarrow B$ și $g : B \rightsquigarrow C$ funcții parțiale.

- (1) Dacă f și g sunt totale, atunci $g \circ f$ este totală.
- (2) Dacă f și g sunt injective, atunci $g \circ f$ este injectivă.
- (3) Dacă f și g sunt surjective, atunci $g \circ f$ este surjectivă.
- (4) Dacă f și g sunt bijective, atunci $g \circ f$ este bijectivă și $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$.

Menționăm că Teorema 1.2.2.1, ce face legătura dintre $Part(A)$ și $E(A)$, ne spune printre altele că există o bijecție ϕ între cele două mulțimi

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

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

Definiția 1.2.3.1. Fie $f : A \rightarrow B$ o funcție.

- (1) Se numește *invers la stânga* al funcției f orice funcție $g : B \rightarrow A$ cu proprietatea $g \circ f = 1_A$.
- (2) Se numește *invers la dreapta* al funcției f orice funcție $g : B \rightarrow A$ cu proprietatea $f \circ g = 1_B$.

Propoziția 1.2.3.4. Fie $f : A \rightarrow B$ o funcție.

- (1) Dacă f admite un invers la dreapta, atunci ea este surjecție.
- (2) Dacă $A \neq \emptyset$ și f admite un invers la stânga, atunci f este injecție.

Demonstrație. (1) Fie g un invers la dreapta al funcției f . Considerând $b \in B$, avem $g(b) \in A$ și $f(g(b)) = b$. Deci, f este surjecție.

(2) Fie g un invers la stânga al funcției f și $a, b \in A$. Dacă presupunem că $f(a) = f(b)$, atunci $a = g(f(a)) = g(f(b)) = b$, ceea ce arată că f este injecție. \square

În cadrul Propoziției 1.2.3.4(2) se consideră $A \neq \emptyset$, deoarece pentru $A = \emptyset$, f este funcția vidă, care este injectivă. Asupra reciprocelor afirmațiilor din această propoziție vom reveni în secțiunea dedicată Axiomei alegerii.

Vom prezenta acum câteva rezultate de “descompunere” a funcțiilor.

Teorema 1.2.3.1. Fie $h : A \rightarrow B$ o funcție. Atunci există o mulțime C , o funcție surjectivă $f : A \rightarrow C$ și o funcție injectivă $g : C \rightarrow B$ astfel încât $h = g \circ f$. În plus, pentru orice mulțime C' , funcție surjectivă $f' : A \rightarrow C'$ și funcție injectivă $g' : C' \rightarrow B$ astfel încât $h = g' \circ f'$, există o unică funcție $d : C \rightarrow C'$ astfel încât $f' = d \circ f$ și $g = g' \circ d$ (a se vedea diagrama din Figura 1.7).

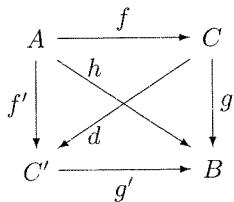


Figura 1.7: Descompunerea din Teorema 1.2.3.1

Demonstrație. Fie $C = h(A)$, $f : A \rightarrow C$ dată prin $f(a) = h(a)$, pentru orice $a \in A$, și $g : C \rightarrow B$ dată prin $g(c) = c$, pentru orice $c \in C$. Este clar că f este surjectie, g este injectie și $h = g \circ f$.

Fie C' , f' și g' ca în enunțul teoremei. Considerăm $d : C \rightarrow C'$ dată prin

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

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

Deoarece f este surjectie, 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 relația $h = g' \circ f'$ conduce la $g'(f'(a_1)) = g'(f'(a_2))$, de unde, pe baza faptului că g' este injectie, deducem $f'(a_1) = f'(a_2)$. Ca urmare, d este bine definită pe C .

Conform definiției funcției d are loc $d \circ f = f'$. Fie acum $a \in A$ și $c \in C$ 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 funcției d decurge cu ușurință după cum urmează. Dacă presupunem că există o altă funcție $d' : C \rightarrow C'$ astfel încât $f' = d' \circ f$ și $g = g' \circ d'$, atunci relația

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

combinată cu faptul că g' este injectie, conduce la $d = d'$. \square

Teorema 1.2.3.1 nu impune nici o restricție asupra funcției h . Ca urmare, h poate fi și funcția vidă. În acest caz $A = \emptyset$, $C = \emptyset$, f este funcția vidă de la \emptyset la \emptyset (deci este surjectivă), g este funcția vidă de la \emptyset la B (deci este injectivă), C' nu poate fi decât \emptyset , ca urmare a surjectivității funcției f' , iar d nu este alta decât funcția vidă de la \emptyset la \emptyset .

Definiția 1.2.3.2. Fie $f : A \rightarrow B$ o funcție. Relația $\ker(f) \subseteq A \times A$ dată prin

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

se numește *nucleul funcției* f .

Este ușor de verificat că pentru orice funcție f de la A la B , $\ker(f)$ este relație de echivalență pe A .

Dată o mulțime A și o relație de echivalență ρ pe A , funcția $f_\rho : A \rightarrow A/\rho$ dată prin $f_\rho(a) = [a]_\rho$, pentru orice $a \in A$, este bine definită și surjectivă. Ea se numește *surjecția naturală indușă de ρ* . Are loc următorul rezultat important.

Teorema 1.2.3.2. (Proprietatea de universalitate a mulțimii cât)

Pentru orice funcție $f : A \rightarrow B$ și orice relație de echivalență ρ pe A ce satisface $\rho \subseteq \ker(f)$, există o unică funcție $g : A/\rho \rightarrow B$ ce satisface $f = g \circ f_\rho$. În plus, dacă $\rho = \ker(f)$, atunci g este injectie, iar dacă f este surjecție, atunci g este surjecție.

Demonstrație. Definim g prin $g([a]_\rho) = f(a)$, pentru orice $a \in A$. Proprietatea $\rho \subseteq \ker(f)$ asigură că g este bine definită. Conform definiției, $f = g \circ f_\rho$. Unicitatea funcției g decurge de la faptul că f_ρ este surjecție, și deci are un invers la dreapta.

Să presupunem că $\rho = \ker(f)$ și fie a și b astfel încât $g([a]_\rho) = g([b]_\rho)$. Atunci $f(a) = f(b)$ care, în baza relației $\rho = \ker(f)$, conduce la $a \rho b$, stabilind astfel injectivitatea funcției g .

Dacă f este surjecție, atunci direct de la definiția funcției g urmează că aceasta este surjecție. \square

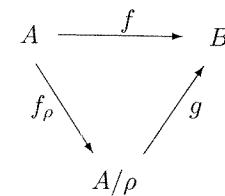


Figura 1.8: Descompunerea din Teorema 1.2.3.2

Corolarul 1.2.3.1. Pentru orice funcție $f : A \rightarrow B$ există o bijecție de la $A/\ker(f)$ la $f(A)$.

Demonstrație. Decurge direct de la Teorema 1.2.3.2 aplicată funcției $f' : A \rightarrow f(A)$ dată prin $f'(a) = f(a)$, pentru orice $a \in A$, și relației $\rho = \ker(f)$ (se observă că are loc $\ker(f') = \ker(f)$). \square

Corolarul 1.2.3.2. Orice funcție $f : A \rightarrow B$ poate fi scrisă ca produs de 3 funcții, $f = k \circ h \circ g$, unde $g : A \rightarrow A/\ker(f)$ este surjecție, $h : A/\ker(f) \rightarrow f(A)$ este bijecție și $k : f(A) \rightarrow B$ este injectie.

Demonstrație. Descompunem întâi f în forma $f = k \circ f'$, unde $f' : A \rightarrow f(A)$ este dată prin $f'(a) = f(a)$ iar $k : f(A) \rightarrow B$ este dată prin $k(c) = c$ (a se vede demonstrația Teoremei 1.2.3.1). Apoi descompunem f' ca în Teorema 1.2.3.2 în forma $f' = h \circ g$, ținând cont de faptul că $\ker(f') = \ker(f)$. Obținem $f = k \circ h \circ g$, unde k ,

h și g satisfac cerințele corolarului.

□

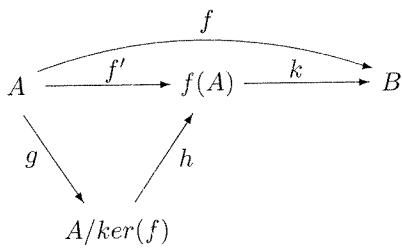


Figura 1.9: Descompunerea din Corolarul 1.2.3.2

Vom prezenta în cele ce urmează câteva proprietăți de bază referitoare la *imaginăea inversă a unei mulțimi* printr-o funcție (ce este caz particular a imaginii inverse a unei mulțimi printr-o relație – a se vedea Definiția 1.2.1.4(2)).

Propoziția 1.2.3.5. Fie $f : A \rightarrow B$ o funcție și $X, Y \subseteq B$. Atunci au loc următoarele proprietăți:

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

Demonstrație. (1) urmează de la echivalențele:

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

pentru orice a .

(2) și (3) se obțin în manieră similară.

□

Propoziția 1.2.3.6. Dacă $f : A \rightarrow B$ este o funcție injectivă, atunci pentru orice $X, Y \subseteq A$ au loc următoarele proprietăți:

- (1) dacă $X \subset Y$, atunci $f(X) \subset f(Y)$;
- (2) $f(X \cap Y) = f(X) \cap f(Y)$;
- (3) $f(X - Y) = f(X) - f(Y)$.

Demonstrație. Se utilizează Propoziția 1.2.1.4(2)(3)(4) și proprietatea de injectivitate a funcției f .

Propoziția 1.2.3.7. Dacă $f : A \rightarrow B$ este o funcție, $X \subseteq A$ și $Y \subseteq f(A)$, atunci au loc următoarele proprietăți:

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

Demonstrație. (1) se obține pe baza implicațiilor:

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

pentru orice b , și a Propoziției 1.2.1.4(6).

(2) urmează direct de la Propoziția 1.2.1.4(6).

□

Operațiile (partiale) sunt funcții (partiale). De exemplu, o funcție (parțială) f de la $A \times B$ la C poate fi gândită ca o operație (parțială) ce acționează pe elementele a două mulțimi, A și B , și produce rezultate în C . Dacă operația este parțială, atunci pot exista perechi (a, b) asupra căror operația să nu acționeze. De exemplu, inversul multiplicativ într-un corp este definit pentru elementele diferite de 0 (unitatea aditivă).

Operațiile (partiale) de la A^n la A , unde n este un număr natural, se numesc *operații (partiale) n-are pe A*. Numărul natural n este numit *aritatea* sau *tipul* operației, iar atunci când operația este notată prin f , aritatea ei se mai notează prin $\text{ar}(f)$. În cazul în care $n = 0$, reamintim că A^0 este $\{\emptyset\}$ și, ca urmare, o operație 0-ară (numită și *operație nulară* sau *constantă*²⁰) va asocia un element din A mulțimii vide. Această asociere este constantă în sensul că ea nu depinde de elementele mulțimii A . Atunci, a specifica o operație nulară revine la a fixa un element din A . Justificații de aceasta, vom folosi adesea terminologia “fie $a \in A$ o operație nulară pe A ” sau “fie a o constantă din A ”.

Operațiile parțiale fiind definite prin intermediul funcțiilor parțiale, iar acestea fiind relații, înseamnă că operațiile sunt relații. Mai exact, o operație parțială n -ară peste A este o relație $(n+1)$ -ară pe A .

Operațiile (partiale) de la o mulțime A sau $A_1 \times A_2$ la o mulțime B pot fi specificate prin așa numitele *tabele Cayley*. De exemplu, operația unară f pe $A = \{a, b\}$ dată prin $f(a) = b$ și $f(b) = a$ poate fi specificată prin tabelul

f	a	b
	b	a

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

²⁰Termenul folosit este cel de *constantă* și nu de *operație constantă*.

g	a	b
a	a	b
b	a	a

Pentru operații (parțiale) binare se folosește adesea *notația infix*, semnul operației fiind între elemente (această notație am folosit-o și la relații binare). De exemplu, dacă $+$ sau \cdot sunt operații, notația infix asupra elementelor a și b va fi $a + b$, respectiv, $a \cdot b$. Pentru operații unare se folosește adesea și notația *exponent*. De exemplu, B' poate reprezenta complementara lui B (semnul operației fiind $'$).

Până acum au fost considerate un număr de operații pe mulțimi, relații și funcții (reuniune, intersecție, diferență, produs cartezian, produs de relații sau funcții, inversă etc.), precum și unele proprietăți ale acestora adnotate în dreapta prin cuvinte de tipul "asociativitate", "comutativitate" etc.

Definiția 1.2.3.3. Fie f o operație binară pe o mulțime A .

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

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

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 $a \in A$.

În cazul operațiilor binare comutative, tabelul Cayley poate fi redus la jumătate, considerând doar valorile de deasupra (sau de dedesubtul) diagonalei, incluzând și diagonala. Dacă operația este și idempotentă, atunci poate fi eliminată și diagonala.

Definiția 1.2.3.4. Fie f și g operații binare pe o mulțime A .

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

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

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

(3) Spunem că f este *distributivă față de g* dacă este distributivă la stânga și la dreapta față de g .

1.2.4. Familii indexate de mulțimi

În această secțiune vom considera noțiunea de familie indexată de mulțimi și vom generaliza reuniunea, intersecția și produsul cartezian la astfel de familii.

Definiția 1.2.4.1. Fie A și I două mulțimi. O *familie de elemente peste A indexată prin I* , sau *familie I -indexată peste A* , este o funcție $f : I \rightarrow A$.

Uzual, dacă f este o familie I -indexată de elemente peste A , atunci vom utiliza pentru ea și notația $(a_i | i \in I)$, unde $f(i) = a_i$, pentru orice $i \in I$. Atunci când I se subînțelege din context vom simplifica această notație la (a_i) , iar dacă $I \neq \emptyset$, atunci vom spune că *familia este nevidă*. Mulțimea I este numită *mulțime de indecsi*, iar elementele ei, *indecsi*. În cazul în care elementele familiei sunt mulțimi ($f(i)$ este mulțime, pentru orice $i \in I$), vom vorbi de *familie I -indexată de mulțimi*, iar dacă mulțimile sunt disjuncte două câte două, vom spune că avem de-a face cu o *familie indexată disjunctă*. Atunci când I și A sunt subînțelese din context sau nu este necesar să le specifică, vom simplifica terminologia renunțând la ele.

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

- referirea la o mulțime se face printr-un index;
- pentru indecsi diferiți este posibil ca mulțimile referite să fie egale (din acest punct de vedere putem gândi familiile indexate de mulțimi ca fiind colecții de mulțimi în care anumite mulțimi pot apărea de mai multe ori).

Definiția 1.2.4.2. Fie $(A_i | i \in I)$ o familie indexată de mulțimi 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 mulțimea:

$$\bigcup(A_i | i \in I) = \{a \in A | (\exists i \in I)(a \in A_i)\}.$$

(2) Dacă $I \neq \emptyset$, atunci *intersecția familiei* $(A_i | i \in I)$, notată $\bigcap(A_i | i \in I)$ sau $\bigcap_{i \in I} A_i$, este mulțimea:

$$\bigcap(A_i | i \in I) = \{a \in A | (\forall i \in I)(a \in A_i)\}.$$

(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 mulțimea tuturor funcțiilor $f : I \rightarrow \bigcup(A_i | i \in I)$ cu proprietatea $f(i) \in A_i$, pentru orice $i \in I$.

În cazul $I = \emptyset$ vom considera $\bigcap(A_i | i \in I) = A$. Observăm că dacă $I = \emptyset$, atunci $\bigcup(A_i | i \in I) = \emptyset$ și $\prod(A_i | i \in I) = \{\emptyset\}$, iar dacă $I \neq \emptyset$ și există $i \in I$ astfel încât

$A_i = \emptyset$, atunci $\prod(A_i | i \in I) = \emptyset$ ²¹. De asemenea, dacă $A_i = A$ pentru orice $i \in I$, atunci $\prod(A_i | i \in I) = A^I$, adică produsul familiei este mulțimea tuturor funcțiilor de la I la A . Notațional, atunci când I se subînțelege din context, vom scrie $\bigcup \mathcal{A}$ ($\bigcap \mathcal{A}$, $\prod \mathcal{A}$) în loc de $\bigcup_{i \in I} \mathcal{A}$ ($\bigcap_{i \in I} \mathcal{A}$, $\prod_{i \in I} \mathcal{A}$).

Existența reuniunii și a intersecției rezultă cu ușurință pe baza Axiomei separării. Pentru a demonstra existența produsului se aplică Axioma separării mulțimii $\mathcal{P}(I \times \bigcup(A_i | i \in I))$.

Prezentăm în continuare câteva proprietăți de bază ale reuniunii, ale intersecției și ale produsului de familii indexate de mulțimi.

Propoziția 1.2.4.1. Fie $(A_i | i \in I)$ și $(B_i | i \in I)$ două familii indexate de mulțimi și A o mulțime. Atunci au loc următoarele proprietăți:

- (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 $\bigcup_{i \in I} A_i \subseteq A$.

Demonstrație. Vom demonstra ca exemplu doar (6), care se obține pe baza echivalențelor:

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

pentru orice a . \square

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

²¹Dacă $I \neq \emptyset$ și $A_i \neq \emptyset$, pentru orice $i \in I$, nu putem deduce, pe baza axiomelor prezentate până acum, că $\prod(A_i | i \in I) \neq \emptyset$ (a se vedea Secțiunea 1.2.5 dedicată Axiomei alegerii). \square

Lema 1.2.4.1. Fie $(A_i | i \in I)$ o familie indexată de mulțimi. Atunci, $\bigcup_{i \in I} A_i$ este unică mulțime A ce satisface proprietățile:

- (i) $A_i \subseteq A$, pentru orice $i \in I$;
 - (ii) dacă B este o mulțime ce satisface $A_i \subseteq B$, pentru orice $i \in I$, atunci $A \subseteq B$.
- Similar, $\bigcap_{i \in I} A_i$ este unică mulțime A ce satisface proprietățile:
- (i') $A \subseteq A_i$, pentru orice $i \in I$;
 - (ii') dacă B este o mulțime ce satisface $B \subseteq A_i$, pentru orice $i \in I$, atunci $B \subseteq A$.

Demonstrație. De la definiții și Propoziția 1.2.4.1(1)(10)(11). \square

Propoziția 1.2.4.2. Fie $(B_j | j \in J)$ o familie indexată de mulțimi și I reuniunea acestei familii. Atunci, pentru orice familie indexată de mulțimi $(A_i | i \in I)$, au loc următoarele proprietăți:

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

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

S este cea mai mică mulțime ce include mulțimile 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 mulțimile A_i , $i \in I$. Minimalitatea mulțimii 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 mulțime ce include mulțimile S_j , pentru orice $j \in J$. Însă pentru orice $j \in J$, $S_j \subseteq S$. Minimalitatea mulțimii $\bigcup_{j \in J} S_j$ (Lema 1.2.4.1) conduce la $\bigcup_{j \in J} S_j \subseteq S$, care, combinată cu incluziunea anterioară, furnizează (1). \square

Propoziția 1.2.4.3. Fie $(A_i | i \in I)$ o familie indexată de mulțimi și f o permutare a mulțimii I . Atunci au loc următoarele proprietăți:

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

Demonstrație. Ca și în cazul Propoziției 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 mulțime ce include mulțimile A_i , $i \in I$. Pentru orice $i \in I$, S include A_j , unde $j = f^{-1}(i)$, și deci va include și $A_{f(j)}$ (care este de fapt A_i). Dacă există o altă mulțime B care include toate mulțimile A_i , atunci ea include și S conform definiției acesteia și a noțiunii de permutare. Lema 1.2.4.1 conduce atunci la (1). \square

Propoziția 1.2.4.4. Fie $(B_j | j \in J)$ o familie indexată de mulțimi, I reuniunea ei și $K = \{C \in \mathcal{P}(I) | (\forall j \in J)(C \cap B_j \neq \emptyset)\}$.

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

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

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

Demonstrație. Demonstrăm doar (1), (2) obținându-se în manieră similară. Fie $C \in K$ și $j \in J$. Conform definiției mulțimii K , avem $C \cap B_j \neq \emptyset$. Pentru orice $i \in C \cap B_j$, Propoziția 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 aceste incluziuni sunt satisfăcute pentru orice $j \in J$, de la Propoziția 1.2.4.1(10) obținem

$$\bigcap_{i \in C} A_i \subseteq \bigcap_{j \in J} (\bigcup_{i \in B_j} A_i),$$

și apoi, de la punctul (11) al aceleiași propoziții, 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 a din mulțimea $\bigcap_{j \in J} (\bigcup_{i \in B_j} A_i)$ și fie $C = \{i \in I | a \in A_i\}$. Pentru orice $j \in J$, avem $a \in \bigcup_{i \in B_j} A_i$, și astfel, există $i \in B_j$ astfel încât $a \in A_i$. Deci, $i \in C$, ceea ce arată că $C \cap B_j \neq \emptyset$ și, 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 obținut astfel că $a \in \bigcup_{C \in K} (\bigcap_{i \in C} A_i)$, și (1) este demonstrată. \square

Ne îndreptăm acum atenția asupra unor proprietăți ale imaginilor și ale imaginilor inverse ale reuniunilor și ale intersecțiilor de familii indexate de mulțimi.

Propoziția 1.2.4.5. Fie $(A_i | i \in I)$ o familie indexată de mulțimi și f o funcție. Atunci au loc următoarele proprietăți:

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

(2) $f(\bigcap_{i \in I} A_i) \subseteq \bigcap_{i \in I} f(A_i)$. Dacă f este injectivă, atunci relația are loc prin egalitate.

Demonstrație. Au loc echivalențele:

$$\begin{aligned} b \in f(\bigcup_{i \in I} A_i) &\Leftrightarrow \exists a \in \bigcup_{i \in I} A_i : b = f(a) \\ &\Leftrightarrow \exists i \in I, \exists a \in A_i : b = f(a) \\ &\Leftrightarrow \exists i \in I : b \in f(A_i) \\ &\Leftrightarrow b \in \bigcup_{i \in I} f(A_i), \end{aligned}$$

pentru orice b , ceea ce demonstrează (1). Similar se arată și (2). \square

Demonstrația următoarei propoziții este lăsată în grija cititorului.

Propoziția 1.2.4.6. Fie $(A_i | i \in I)$ o familie indexată de mulțimi și f o funcție. Atunci au loc următoarele proprietăți:

$$(1) \quad f^{-1}(\bigcup_{i \in I} A_i) = \bigcup_{i \in I} f^{-1}(A_i);$$

$$(2) \quad f^{-1}(\bigcap_{i \in I} A_i) = \bigcap_{i \in I} f^{-1}(A_i).$$

Definiția 1.2.4.3. Fie $(A_i | i \in I)$ o familie indexată nevidă de mulțimi și $i \in I$. Funcția de i -proiecție asociată familiei $(A_i | i \in I)$ este funcția

$$pr_i : \prod(A_i | i \in I) \rightarrow A_i$$

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 $\prod(A_i | i \in I) = \emptyset$ și, deci, pr_i este funcția vidă, pentru orice $i \in I$. Altfel, pr_i este funcție surjectivă.

Teorema 1.2.4.1. (Proprietatea de universalitate a produsului)

Fie $(A_i | i \in I)$ o familie indexată nevidă de mulțimi, A o mulțime nevidă și g_i funcții de la A la A_i , pentru orice $i \in I$. Atunci există o unică funcție $f : A \rightarrow \prod(A_i | i \in I)$ astfel încât $g_i = pr_i \circ f$.

Demonstrație. 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 funcția f prin $f(a) = x_a$, pentru orice $a \in A$, și constatăm cu ușurință că ea satisfac proprietatea $g_i = pr_i \circ f$, pentru orice $i \in I$.

Unicitatea funcției f se obține astfel. Dacă ar exista o altă funcție 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 $a \in A$ și $i \in I$. Dar aceasta nu înseamnă altceva decât că are loc $f = f'$. \square

1.2.5. Axioma alegerii

Axioma alegerii este, fără doar și poate, una dintre cele mai discutate axiome din matematică, după axioma paralelelor introdusă de Euclid cu mai mult de 2000 de ani în urmă. Această axiomă spune că, pentru orice familie \mathcal{A} de mulțimi disjuncte și nevide, există mulțimi de reprezentanți.

Fără îndoială că orice cititor se întreabă de ce existența unei astfel de mulțimi nu poate fi dedusă pe baza celorlalte axiome, în special a Axiomei separării, ca urmare a faptului că orice mulțime de reprezentanți pentru \mathcal{A} este submulțime a mulțimii $\bigcup \mathcal{A}$. Probabil că răspunsul va fi intuit chiar de acesta în încercarea de a defini proprietăți prin care să fie selectate astfel de submulțimi. Însă trebuie remarcat că numai proprietatea de a fi mulțime de reprezentanți pentru \mathcal{A} nu definește unic o astfel de mulțime. În adevăr, dacă \mathcal{A} conține mulțimi cu cel puțin două elemente²² și C este o mulțime de reprezentanți pentru \mathcal{A} , atunci mulțimea $C' = (C - A \cap C) \cup \{y\}$, unde $A \in \mathcal{A}$ are cel puțin două elemente și $y \in A - (A \cap C)$, este de asemenea mulțime de reprezentanți pentru \mathcal{A} . În plus, $C \neq C'$.

Evident că, pentru anumite familii particulare \mathcal{A} , se poate aplica Axioma separării pentru a deduce existența unei mulțimi C , ca mai sus. De exemplu, dacă \mathcal{A} este o familie de mulțimi disjuncte și nevide de numere naturale, atunci mulțimea ce conține cel mai mic număr natural al fiecărei mulțimi componente a familiei \mathcal{A} (un astfel de număr natural există întotdeauna), și numai pe acestea, satisface proprietatea de mai sus. Mai mult, această mulțime se poate obține cu ajutorul Axiomei separării aplicate mulțimii $\bigcup \mathcal{A}$ și formulei

$$P(x) : (\exists A)(A \in \mathcal{A} \wedge x = \min(A)).$$

Deci ceea ce face ca în acest caz să obținem, prin Axioma separării, o astfel de mulțime este proprietatea de existență a celui mai mic număr natural al oricărei submulțimi nevide de numere naturale. Situația se schimbă complet dacă în loc de submulțimi de numere naturale considerăm submulțimi arbitrarne nevide de numere reale. În general, într-un astfel de caz nu cunoaștem o regulă prin care să “selectăm” câte un element din fiecare mulțime componentă a familiei \mathcal{A} (exceptând cazul în care fiecare mulțime componentă are o anumită proprietate specifică ce permite localizarea a exact unui element).

Ca urmare, caracterul existențial al unei mulțimi de reprezentanți pentru \mathcal{A} trebuie înțeles în sensul că o astfel de mulțime poate fi specificată printr-o proprietate “bine-definită” a obiectelor ei, iar axiomele teoriei mulțimilor pot asigura demonstrarea faptului că ea este unica mulțime definită prin respectiva proprietate și că este mulțime de reprezentanți pentru \mathcal{A} .

Axioma alegerii. Pentru orice familie de mulțimi disjuncte și nevide există mulțimi de reprezentanți.

Vom abrevia Axioma alegerii prin (AC) (de la “Axiom of Choice”). Se pare că prima mențiune explicită a necesității unei astfel de axiome a fost făcută de Giuseppe Peano în 1890 [163] în legătură cu o demonstrație de existență asupra unor sisteme de ecuații diferențiale ordinare. Necesitatea unei astfel de axiome a fost remarcată și de Beppo Levi în 1902 [125] în studii de cardinalitate. Formularea explicită a ei a fost

²²Dacă orice mulțime $A \in \mathcal{A}$ are exact un element, atunci $\bigcup \mathcal{A}$ este mulțime de reprezentanți pentru \mathcal{A} .

făcută de Ernst Zermelo în 1904 [227], care a utilizat-o apoi în demonstrarea unor rezultate asupra ordinilor bune. Consistența acestei axiome (faptul că ea nu conduce la contradicții atunci când este adăugată celorlalte axiome ale teoriei mulțimilor) a fost demonstrată de Kurt Gödel în 1938 [70], în timp ce independența ei față de celelalte axiome ale teoriei mulțimilor (inclusiv Axioma infinitului și Axioma înlăturării) ce va fi prezentată ulterior) a fost stabilită în 1963 de Paul Cohen [35].

Menționăm că Axioma alegerii nu este necesară în cazul familiilor formate doar dintr-o singură mulțime nevidă, $\mathcal{A} = \{A\}$. În adevăr, deoarece A este nevidă, urmează că există un element $x_0 \in A$, iar Axioma împerecherii conduce la existența mulțimii $\{x_0\}$ care este mulțime de reprezentanți pentru \mathcal{A} . Prin inducție matematică putem generaliza această observație la familiile finite de mulțimi disjuncte și nevide.

Prezentăm în continuare câteva din formele echivalente ale Axiomei alegerii.

Teorema 1.2.5.1. Axioma alegerii este echivalentă cu oricare din următoarele două afirmații:

- (AC1) pentru orice familie de mulțimi nevide există mulțimi de reprezentanți;
- (AC2) pentru orice mulțime A , familia $\mathcal{P}(A) - \{\emptyset\}$ admite mulțimi de reprezentanți.

Demonstrație. Este clar că $(AC1)$ implică atât (AC) , cât și $(AC2)$.

Arătăm că $(AC2)$ implică $(AC1)$. Fie \mathcal{A} o familie de mulțimi nevide. Considerăm $A = \bigcup \mathcal{A}$ și, de la $(AC2)$, urmează că există mulțimi de reprezentanți pentru mulțimea $\mathcal{P}(A) - \{\emptyset\}$; fie C o astfel de mulțime. Este evident că pentru orice $X \in \mathcal{A}$ are loc $X \in \mathcal{P}(A) - \{\emptyset\}$. Atunci mulțimea

$$D = \{c \in C \mid (\exists X \in \mathcal{A})(c \in X)\}$$

este mulțime de reprezentanți pentru familia \mathcal{A} .

Arătăm că (AC) implică $(AC1)$. Fie \mathcal{A} o familie de mulțimi nevide. Definim $\mathcal{B} = \{\{X\} \times X \mid X \in \mathcal{A}\}$, care este familie de mulțimi disjuncte și nevide. (AC) asigură atunci existența unei mulțimi C de reprezentanți pentru \mathcal{B} . Orice element al mulțimii C este de forma (X, x) , unde $X \in \mathcal{A}$ și $x \in X$. Atunci, mulțimea

$$D = \{x \mid (\exists X \in \mathcal{A})((X, x) \in C)\}$$

este mulțime de reprezentanți pentru familia \mathcal{A} . \square

Acceptând Axioma alegerii, putem răspunde pozitiv întrebării puse în Secțiunea 1.2.2, adică, pentru orice partitură există mulțimi de reprezentanți.

Definiția 1.2.5.1.

- (1) Se numește *funcție de alegere* pentru o familie de mulțimi \mathcal{A} orice funcție f de la \mathcal{A} la $\bigcup \mathcal{A}$ cu proprietatea $f(A) \in A$, pentru orice $A \in \mathcal{A}$ cu $A \neq \emptyset$.

- (2) Se numește *funcție de alegere* pentru o familie indexată de mulțimi $(A_i | i \in I)$ orice funcție f de la I la $\bigcup(A_i | i \in I)$ cu proprietatea $f(i) \in A_i$, pentru orice $i \in I$ cu $A_i \neq \emptyset$.

Teorema 1.2.5.2. Axioma alegerii este echivalentă cu oricare din următoarele 5 afirmații:

(AC') pentru orice familie de mulțimi disjuncte și nevide există funcții de alegere;

$(AC1')$ pentru orice familie de mulțimi nevide există funcții de alegere;

$(AC2')$ pentru orice mulțime A există funcții de alegere pentru $\mathcal{P}(A) - \{\emptyset\}$;

$(AC3')$ pentru orice familie indexată de mulțimi nevide există funcții de alegere;

$(AC4')$ produsul oricărei familii indexate de mulțimi nevide este nevid.

Demonstrație. Direct de la definiție urmează că $(AC1')$ implică atât (AC') , cât și $(AC2')$.

Să arătăm că (AC') implică (AC) . Fie \mathcal{A} o familie de mulțimi disjuncte și nevide și $f : \mathcal{A} \rightarrow \bigcup \mathcal{A}$ o funcție de alegere pentru ea. Este ușor de văzut că mulțimea $C = f(\mathcal{A})$ este mulțime de reprezentanți pentru \mathcal{A} . Același raționament poate fi utilizat pentru a arăta că $(AC1')$ implică $(AC1)$ și $(AC2')$ implică $(AC2)$.

Arătăm că $(AC1)$ implică $(AC1')$. Fie deci \mathcal{A} o familie de mulțimi nevide și C o mulțime de reprezentanți ai ei. Funcția

$$f = \{(X, x) | X \in \mathcal{A} \wedge C \cap X = \{x\}\}$$

este funcție de alegere pentru \mathcal{A} .

Este clar că $(AC3')$ și $(AC4')$ sunt echivalente. Arătăm că $(AC2')$ implică $(AC4')$. Fie $(A_i | i \in I)$ o familie indexată de mulțimi nevide și $A = \bigcup(A_i | i \in I)$. $(AC2')$ conduce la existența unei funcții de alegere f pentru $\mathcal{P}(A) - \{\emptyset\}$. Ca urmare, pentru orice $i \in I$ avem $f(A_i) \in A_i$ (aceste mulțimi sunt nevide), și astfel, $\prod(A_i | i \in I) \neq \emptyset$ deoarece el conține măcar elementul g dat prin $g(i) = f(A_i)$, pentru orice $i \in I$. Are loc deci $(AC4')$.

Reciproc, fie A o mulțime. Conform cu $(AC4')$ avem

$$\prod(X | X \in \mathcal{P}(A) - \{\emptyset\}) \neq \emptyset.$$

Atunci orice funcție $g \in \prod(X | X \in \mathcal{P}(A) - \{\emptyset\})$ este funcție de alegere pentru mulțimea $\mathcal{P}(A) - \{\emptyset\}$. Deci are loc $(AC2')$. \square

Teorema 1.2.5.3. Axioma alegerii este echivalentă cu oricare din următoarele trei afirmații:

$(AC3)$ pentru orice funcție surjectivă există un invers la dreapta;

$(AC4)$ pentru orice relație binară R există o funcție f astfel încât $f \subseteq R$ și $\text{Dom}(f) = \text{Dom}(R)$;

$(AC5)$ pentru orice funcție f al cărei codomeniu este o familie de mulțimi există o funcție g cu proprietatea $g(x) \in f(x)$, pentru orice $x \in \text{Dom}(f)$ cu $f(x) \neq \emptyset$.

Demonstrație. Arătăm că $(AC2')$ implică $(AC3)$. Fie $f : A \rightarrow B$ o funcție surjectivă. Considerăm submulțimile

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

pentru orice $b \in B$. Funcția f fiind surjectivă, urmează că $A_b \neq \emptyset$, pentru orice $b \in B$, iar $(AC2')$ conduce la existența unei funcții de alegere h pentru $\mathcal{P}(A) - \{\emptyset\}$.

Definim funcția g de la B la A prin $g(b) = h(A_b)$, pentru orice $b \in B$. Este ușor de verificat că $f \circ g = 1_B$.

Arătăm că $(AC3)$ implică $(AC4)$. Fie R o relație binară. Considerăm funcția $h : R \rightarrow \text{Dom}(R)$ dată prin $h(x, y) = x$, pentru orice $(x, y) \in R$. Este clar că această funcție este surjectivă, și atunci $(AC3)$ asigură existența unei funcții g de la $\text{Dom}(R)$ la R astfel încât $h \circ g = 1_{\text{Dom}(R)}$.

Definim funcția $f : \text{Dom}(R) \rightarrow \text{Cod}(R)$ prin $f(x) = y$, pentru orice $x \in \text{Dom}(R)$, unde $g(x) = (x, y)$. Are loc $\text{Dom}(f) = \text{Dom}(R)$ și $f \subseteq R$.

Arătăm că $(AC4)$ implică $(AC5)$. Fie f o funcție al cărei codomeniu este o familie de mulțimi. Definim relația R_f prin

$$R_f = \{(x, y) | x \in \text{Dom}(f) \wedge f(x) \neq \emptyset \wedge y \in f(x)\}.$$

Atunci, $(AC4)$ conduce la existența unei funcții g ce satisfac proprietățile $g \subseteq R_f$ și $\text{Dom}(g) = \text{Dom}(R_f)$. g este funcția căutată.

Încheiem demonstrația teoremei prin a arăta că $(AC5)$ implică $(AC2')$. Fie A o mulțime. Considerând funcția identitate f pe $\mathcal{P}(A)$, $(AC5)$ conduce la existența unei funcții g de la $\mathcal{P}(A) - \{\emptyset\}$ la A cu proprietatea $g(B) \in f(B) = B$, pentru orice $B \in \mathcal{P}(A) - \{\emptyset\}$. Deci, g este funcție de alegere pentru A . \square

Diagrama din Figura 1.10 reprezintă schematic implicațiile dintre formele echivalente ale Axiomei alegerii ce au fost prezentate până acum.

Revenim acum la întrebarea formulată după Propoziția 1.2.3.4 . Are loc:

Corolarul 1.2.5.1. Fie $f : A \rightarrow B$ o funcție.

(1) f este surjecție dacă și numai dacă admite un invers la dreapta.

(2) Dacă $A \neq \emptyset$, atunci f este injectie dacă și numai dacă admite un invers la stânga.

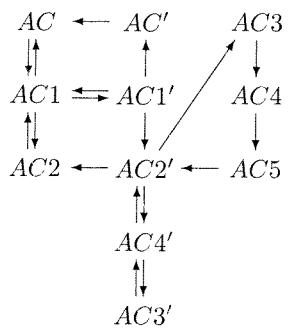


Figura 1.10: Afirmații echivalente Axiomei alegerii

Demonstrație. (1) urmează de la Propoziția 1.2.3.4(1) și Teorema 1.2.5.3.

(2) Fie $a_0 \in A$ (A este nevidă). Definim funcția $g : B \rightarrow A$ prin

$$g(b) = \begin{cases} a, & \text{dacă } b \in f(A) \text{ și } f(a) = b \\ a_0, & \text{dacă } b \in B - f(A), \end{cases}$$

pentru orice $b \in B$.

O simplă verificare ne arată că g este invers la stânga al funcției f . Acum, (2) urmează de la aceasta și Propoziția 1.2.3.4(2). \square

Dorim să subliniem necesitatea utilizării Axiomei alegerii în demonstrarea punctului (1) al Corolarului 1.2.5.1, dar nu a punctului (2). De asemenea, urmărind cu atenție demonstrația, constatăm că o funcție surjectivă (injectivă) poate avea mai mulți inversi la dreapta (stânga).

Corolarul 1.2.5.2. Fie f o funcție de la A la B și g o funcție de la B la A . Atunci, au loc următoarele proprietăți:

- (1) f este bijecție dacă și numai dacă are un același invers la stânga și la dreapta și, în acest caz, inversul este unic;
- (2) dacă $g \circ f = 1_A$ și $f \circ g = 1_B$, atunci f și g sunt bijecții.

Demonstrație. (1) Este clar că dacă f are un același invers la stânga și la dreapta, atunci ea este bijecție. Reciproc, dacă f este bijecție, atunci f^{-1} este atât invers la stânga cât și la dreapta. Să arătăm că acest invers este unic. Dacă ar mai exista un invers la stânga, g , atunci,

$$f^{-1} = 1_A \circ f^{-1} = (g \circ f) \circ f^{-1} = g \circ (f \circ f^{-1}) = g \circ 1_B = g,$$

și similar dacă g ar fi invers la dreapta.

(2) urmează direct de la (1). \square

1.2.6. Relații de ordine

O altă clasă importantă de relații binare, pe lângă cea a relațiilor de echivalență și a funcțiilor, este cea a relațiilor de ordine.

Definiția 1.2.6.1. Fie ρ o relație binară pe o mulțime A .

- (1) ρ este numită *relație de pre-ordine* sau *quasi-ordine* pe A dacă ρ este reflexivă și tranzitivă pe A și, în acest caz, cuplul (A, ρ) se numește *mulțime pre-ordonată* sau *quasi-ordonată*.
- (2) ρ este numită *relație de ordine parțială* pe A dacă ρ este reflexivă, antisimetrică și tranzitivă pe A și, în acest caz, cuplul (A, ρ) se numește *mulțime parțial ordonată* (abreviat, mpo).
- (3) ρ este numită *relație de ordine parțială strictă* pe A dacă ρ este ireflexivă și tranzitivă pe A și, în acest caz, cuplul (A, ρ) se numește *mulțime parțial ordonată strict* (abreviat, mpos).
- (4) ρ este numită *relație de ordine totală* pe A dacă ρ este ordine parțială și conexă pe A și, în acest caz, cuplul (A, ρ) se numește *mulțime total ordonată* (abreviat, mto) sau *mulțime liniar ordonată* sau *lanț*.
- (5) ρ este numită *relație de ordine totală strictă* pe A dacă ρ este ordine parțială strictă și conexă pe A și, în acest caz, cuplul (A, ρ) se numește *mulțime total ordonată strict* (abreviat, mtos).

Nu este dificil de văzut că ireflexivitatea și tranzitivitatea implică antisimetria, iar antisimetria este echivalentă cu antisimetria plus ireflexivitatea. Ca urmare, putem spune că relațiile de ordine parțială strictă sunt relații antisimetrice și tranzitive. Relațiile definite mai sus vor fi referite în general ca fiind *relații de ordine*.

Relația vidă satisfac oricare dintre proprietățile din Definiția 1.2.6.1. Ca urmare, perechea (\emptyset, \emptyset) este mpo, mto etc.

Definiția 1.2.6.2.

- (1) Se numește *mulțime dirijată* orice cuplu (A, ρ) format dintr-o mulțime nevidă A și o relație ρ dirijată pe A .
- (2) Se numește *mulțime filtrată* orice cuplu (A, ρ) format dintr-o mulțime nevidă A și o relație ρ filtrată pe A .

Este ușor de văzut că orice mulțime total ordonată nevidă este atât dirijată, cât și filtrată.

Frecvent, conceptul de mulțime dirijată (filtrată) este cuplat cu unul dintre conceptele din Definiția 1.2.6.1(1)(2)(3). Astfel, o *mulțime parțial ordonată dirijată* este o mulțime parțial ordonată ce este și dirijată.

Reprezentarea grafică a perechilor (A, ρ) , unde ρ poate fi o relație binară arbitrară pe A , se face ca în Secțiunea 1.2.1, cu deosebirea că nodurile grafului nu sunt date de $\text{Dom}(\rho) \cup \text{Cod}(\rho)$, ci de A . Ca urmare, graful asociat relației poate conține *noduri izolate* (noduri ce nu sunt extremități ale nici unui arc). Evident, în cazul în care ρ este reflexivă și/sau tranzitivă se poate recurge la simplificarea reprezentării grafice, așa cum a fost menționat în Secțiunea 1.2.1.

Exemplul 1.2.6.1. Fie A o mulțime nevidă. Relația de inclusiune pe A , \subseteq_A , este reflexivă, antisimetrică și tranzitivă. Ca urmare, (A, \subseteq_A) este mpo.

Structurile (A, ρ) din Definiția 1.2.6.1 se generalizează la *structuri relationale* ce vor fi prezentate în Secțiunea 1.4.1.

1.3. Închideri

Închiderea unei mulțimi la o familie de constructori este una dintre operațiile de bază în matematică și în informatică. Dacă din punct de vedere pur matematic suntem adesea interesați doar de existența închiderii (ca fiind intersecția tuturor mulțimilor ce includ mulțimea în cauză și sunt închise la familia respectivă de constructori), din punct de vedere informatic lucrurile stau puțin altfel. Ne interesează nu numai existența închiderii, dar și o manieră constructivă de obținere a obiectelor închiderii. Mai suntem interesați și într-o ordine de aplicare a constructorilor, atunci când este posibil de stabilit o astfel de ordine, și ne mai interesează și metode de demonstrație de proprietăți ale obiectelor închiderii. Mai sunt și alte probleme de care suntem interesați, cum ar fi unicitatea construcției obiectelor închiderii. Toate acestea vor fi discutate în această secțiune, urmând în principal [207].

1.3.1. Închideri. Inducție structurală

Începem secțiunea prin a stabili câteva convenții ce vor fi utilizate ori de câte ori va fi vorba despre închideri.

Vom fi interesați în a realiza închideri ale unei mulțimi la o mulțime \mathcal{R} de relații. Din rațiuni tehnice vom considera că fiecare relație $r \in \mathcal{R}$ este de la V^{nr} la V , unde V este o mulțime fixată pentru toate relațiile din \mathcal{R} , iar n_r este un număr natural ce depinde de r ²³. Reamintim că pentru $n = 0$ avem, prin convenție matematică,

²³ Specificarea unei același mulțimi V pentru toate relațiile din \mathcal{R} are doar scop tehnic. Toată teoria din această secțiune s-ar fi putut dezvolta considerând o mulțime de relații \mathcal{R} ce are proprietatea că, pentru fiecare $r \in \mathcal{R}$, r conține numai perechi de forma $((a_1, \dots, a_n), a)$. Într-un astfel

$V^0 = \{\emptyset\}$. Ca urmare, a specifica o relație de la V^0 la V revine la a specifica o submulțime a mulțimii V , ceea ce ne permite să considerăm astfel de relații ca fiind submulțimi ale lui V . În cazul $n = 1$, perechile $((a_1), a) \in r$ vor fi noteate simplificat prin (a_1, a) .

Pentru $A \subseteq V$ și $r \in \mathcal{R}$, vom nota prin $r(A^{nr})$ mulțimea

$$r(A^{nr}) = \begin{cases} r, & \text{dacă } n_r = 0 \\ \{a \in V | (\exists a_1, \dots, a_{n_r} \in A)((a_1, \dots, a_{n_r}), a) \in r\}, & \text{altfel,} \end{cases}$$

iar prin $\mathcal{R}(A)$, mulțimea

$$\mathcal{R}(A) = \bigcup_{r \in \mathcal{R}} r(A^{nr}).$$

Definiția 1.3.1.1. Fie A o mulțime și \mathcal{R} o mulțime de relații.

- (1) Spunem că A este *închisă la $r \in \mathcal{R}$* dacă $r(A^{nr}) \subseteq A$.
- (2) Spunem că o mulțime B este *închiderea mulțimii A la mulțimea de relații \mathcal{R}* , sau că este *\mathcal{R} -închiderea mulțimii A* , și notăm $B = \mathcal{R}[\![A]\!]$, dacă B este cea mai mică mulțime, în sensul inclusiunii, ce include A și este închisă la fiecare relație $r \in \mathcal{R}$.

În 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 mulțimi la o mulțime de relații există întotdeauna.

Teorema 1.3.1.1. Fie A o mulțime și \mathcal{R} o mulțime de relații pe V . Atunci există o unică mulțime B ce este închiderea mulțimii A la \mathcal{R} .

Demonstrație. Fie sirul de mulțimi:

- $B_0 = A$;
- $B_{m+1} = B_m \cup \mathcal{R}(B_m)$, pentru orice $m \in \mathbb{N}$.

Este clar că au loc inclusiunile

$$A = B_0 \subseteq B_1 \subseteq \dots \subseteq B_m \subseteq \dots$$

Fie $B = \bigcup_{m \geq 0} B_m$. Arătăm că B satisfac teorema. B include A și

$$\begin{aligned} \mathcal{R}(B) &= \mathcal{R}(\bigcup_{m \geq 0} B_m) \\ &= \bigcup_{m \geq 0} \mathcal{R}(B_m) \\ &\subseteq \bigcup_{m \geq 1} B_m \\ &= B, \end{aligned}$$

de căz, mulțimea V nu ar fi fost altă decât mulțimea tuturor elementelor ce apar în toate perechile relațiilor din \mathcal{R} .

ceea ce arată că B este închisă la \mathcal{R} (ultima egalitate se bazează pe faptul că $B_0 \subseteq B_1$).

Fie B' o mulțime ce include A și este închisă la \mathcal{R} . Prin inducție matematică se arată că, pentru orice $m \geq 0$, are loc $B_m \subseteq B'$ și, deci

$$B = \bigcup_{m \geq 0} B_m \subseteq B'.$$

Ca urmare, B este cea mai mică mulțime ce include A și este închisă la \mathcal{R} , fiind astfel unică mulțime cu aceste proprietăți. \square

Observația 1.3.1.1.

(1) Analizând demonstrația Teoremei 1.3.1.1, constatăm că închiderea mulțimii A la mulțimea \mathcal{R} de relații este "limita" ("supremumul") sirului de mulțimi:

- $B_0 = A$;
- $B_{m+1} = B_m \cup \mathcal{R}(B_m)$, pentru orice $m \geq 0$,

adică $B = \bigcup_{m \geq 0} B_m$.

(2) Mulțimea $\mathcal{A} = \{X \subseteq V \mid A \subseteq X \wedge X \text{ este închisă la } \mathcal{R}\}$ este nevidă ($V \in \mathcal{A}$) și $\bigcap \mathcal{A}$ este închiderea mulțimii A la \mathcal{R} . Ca urmare, putem spune că închiderea mulțimii A la \mathcal{R} este intersecția tuturor submulțimilor X ale mulțimii V ce includ A și sunt închise la \mathcal{R} .

Următoarea teoremă, cunoscută ca fiind *Principiul inducției structurale*, poate fi gândită ca echivalentă Principiului inducției matematice pentru mulțimi definite prin închidere. Ea este un instrument matematic foarte important prin care se pot demonstra anumite proprietăți ale elementelor închiderii unei mulțimi.

Teorema 1.3.1.2. (principiul inducției structurale)

Fie $B = \mathcal{R}[\![A]\!]$ și P o proprietate astfel încât:

- (i) $P(a)$, pentru orice $a \in A$;
- (ii) $(P(a_1) \wedge \dots \wedge P(a_{n_r}) \Rightarrow P(a))$, pentru orice $r \in \mathcal{R}$ și $a_1, \dots, a_{n_r}, a \in B$ cu $((a_1, \dots, a_{n_r}), a) \in r$ (în cazul $n_r = 0$, proprietatea P se înțelege a fi satisfăcută de orice $a \in r$).

Atunci, P este satisfăcută de toate elementele $b \in B$.

Demonstrație. Fie $B' = \{b \in B \mid P(b)\}$. Atunci $A \subseteq B'$ (de la (i)) și B' este închisă la \mathcal{R} (de la (ii)). Ca urmare, $B \subseteq B'$. \square

Observația 1.3.1.2.

- (1) Se poate da o demonstrație a Teoremei 1.3.1.2 pornind de la observația că B este supremumul sirului de mulțimi $\langle B_m \mid m \geq 0 \rangle$ descris în Observația 1.3.1.1. Astfel,

prinț-o simplă inducție 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 observații ne arată suplimentar că ceea ce se poate demonstra prin inducție structurală se poate demonstra și prin inducție matematică. În concluzie, inducția structurală nu este "mai puternică" decât inducția matematică. Însă utilizarea inducției structurale acolo unde este cazul conduce în general la simplificarea tehnică a demonstrațiilor oferind un plus de naturalețe și de eleganță.

Un aspect important ce trebuie discutat este cel legat de "ordinea" în care se face închiderea unei mulțimi atunci când sunt considerate mai mult decât o relație. În cadrul sirului de mulțimi din demonstrația Teoremei 1.3.1.1 nu este impusă o anume ordine, toate relațiile fiind aplicate la fiecare pas. Am putea să ne întrebăm: dacă realizăm întâi închiderea mulțimii A la r_0 , apoi a mulțimii $r_0[\![A]\!]$ la r_1 etc., obținem $\{r_i \mid i \geq 0\}[\![A]\!]$? Răspunsul este negativ în general, iar în secțiunea următoare vom prezenta câteva exemple în acest sens.

1.3.2. Închideri ale unei relații binare

Vom nota prin $r(\rho)$ ($s(\rho), t(\rho)$) *închiderea reflexivă* (simetrică, tranzitivă) a relației ρ ; *închiderea la echivalență* a relației ρ va fi notată prin $equiv(\rho)$ sau \equiv_ρ . Adică, închiderea reflexivă (simetrică, tranzitivă, la echivalență) a relației ρ este cea mai mică relație ce include ρ și este reflexivă (simetrică, tranzitivă, relație de echivalență). Oricare dintre aceste închideri pot fi obținute recurgând la construcția din demonstrația Teoremei 1.3.1.1 (a se vedea și Observația 1.3.1.1). De exemplu, dacă considerăm o relație θ ce conține toate 2-uplurile de forma $((a, b), (b, c)), (a, c)$, unde $a, b, c \in A$, atunci obținem cu ușurință că $t(\rho) = \theta[\![\rho]\!]$ (presupunând că $\rho \subseteq A^2$).

În cele ce urmează vom arăta cum pot fi obținute închiderile de mai sus într-un mod mai ușor. Menționăm întâi că vom realiza și închideri multiple ale unei relații binare. De exemplu, vom folosi scrierea $trs(\rho)$ pentru a specifica faptul că relația $trs(\rho)$ se obține prin închiderea relației ρ la simetrie, apoi a relației astfel obținute la reflexivitate și, în final, la tranzitivitate. Formal, $trs(\rho) = t(r(s(\rho)))$.

Teorema 1.3.2.1. Fie ρ o relație binară pe o mulțime A . Atunci au loc următoarele proprietăți:

- (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) $rt(\rho) = \rho^* = \bigcup_{n \geq 0} \rho^n$.

Demonstrație. (1) Orice relație reflexivă pe A include ι_A și, ca urmare, orice relație reflexivă ce include ρ va include și $\rho \cup \iota_A$. Deci cea mai mică relație reflexivă ce include ρ este $\rho \cup \iota_A$, adică, $r(\rho) = \rho \cup \iota_A$.

(2) Orice relație simetrică ce include ρ trebuie să includă și ρ^{-1} (conform definiției). Ca urmare, cea mai mică relație simetrică ce include ρ este $\rho \cup \rho^{-1}$, adică $s(\rho) = \rho \cup \rho^{-1}$.

(3) urmează de la Corolarul 1.2.1.1(1).

$$(4) \quad rt(\rho) = r(t(\rho)) = r(\rho^+) = \rho^+ \cup \iota_A = \rho^+ \cup \rho^0 = \bigcup_{n \geq 0} \rho^n = \rho^*. \quad \square$$

Teorema 1.3.2.1(3) justifică terminologia de închidere tranzitivă atribuită relației ρ^+ în Secțiunea 1.2.1, iar Teorema 1.3.2.1(4) ne arată că *închiderea reflexivă și tranzitivă* a relației ρ (a se vedea și Secțiunea 1.2.1) se poate obține prin efectuarea închiderii relației ρ la tranzitivitate și apoi a relației obținute la reflexivitate.

Propoziția 1.3.2.1. Fie ρ și σ relații binare pe o mulțime A . Atunci au loc următoarele proprietăți:

- (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 1.3.2.2. Fie ρ o relație binară. Atunci au loc următoarele proprietăți:

- (1) $sr(\rho) = rs(\rho);$
- (2) $tr(\rho) = rt(\rho);$
- (3) dacă ρ este simetrică, atunci $t(\rho)$ este simetrică.

Demonstrație. Vom folosi din plin Propoziția 1.3.2.1.

$$(1) \quad sr(\rho) = s(\rho \cup \iota_A) = s(\rho) \cup \iota_A = r(s(\rho)) = rs(\rho).$$

$$(2) \quad tr(\rho) = t(\rho \cup \iota_A) = t(\rho) \cup \iota_A = r(t(\rho)) = rt(\rho).$$

(3) Vom arăta că $t(\rho) = t(\rho)^{-1}$ utilizând faptul că $\rho = \rho^{-1}$ (ρ este simetrică).

Are loc

$$(t(\rho))^{-1} = \left(\bigcup_{n \geq 1} \rho^n \right)^{-1} = \bigcup_{n \geq 1} (\rho^{-1})^n = \bigcup_{n \geq 1} (\rho)^n = t(\rho),$$

ceea ce stabilește simetria relației $t(\rho)$. \square

Punctele (1) și (2) ale Teoremei 1.3.2.2 ne arată că ordinea închiderii la simetrie/tranzitivitate și reflexivitate poate fi permuatată fără a afecta rezultatul final, în timp ce punctul (3) ne spune că închiderea tranzitivă nu “distrugă” simetria.

Închiderea simetrică poate distruga însă tranzitivitatea, așa 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 relațiile:

$$st(\rho) = \{(1, 1), (2, 2), (3, 3), (1, 2), (2, 1), (1, 3), (3, 1)\},$$

$$ts(\rho) = \{(1, 1), (2, 2), (3, 3), (1, 2), (2, 1), (1, 3), (3, 1), (2, 3), (3, 2)\},$$

care ne arată că $st(\rho) \neq ts(\rho)$.

Teorema 1.3.2.3. Fie ρ o relație binară pe o mulțime A . Atunci

$$\equiv_\rho = tsr(\rho) = trs(\rho) = rts(\rho).$$

Demonstrație. Utilizând Teorema 1.3.2.2, obținem sirul de egalități

$$tsr(\rho) = trs(\rho) = rts(\rho)$$

și faptul că $tsr(\rho)$ este relație de echivalență. În plus, $\rho \subseteq tsr(\rho)$.

Dacă σ este o relație de echivalență pe A ce include ρ , atunci

$$r(\rho) \subseteq r(\sigma) = \sigma, \quad sr(\rho) \subseteq s(\sigma) = \sigma \quad și \quad tsr(\rho) \subseteq t(\sigma) = \sigma;$$

ca urmare, $tsr(\rho)$ este cea mai mică echivalență pe A ce include ρ . \square

Teorema anterioară ne spune că pentru a realiza închiderea la echivalență a unei relații ρ este important de a realiza întâi închiderea simetrică și apoi închiderea tranzitivă; închiderea reflexivă poate fi realizată oricând.

1.3.3. Definiții inductive/recursive

O construcție inductivă de mulțimi are ca scop elaborarea unei mulțimi pornind de la o mulțime dată (de elemente de bază) și adăugând, pas cu pas, noi elemente.

Considerăm următorul exemplu din logica matematică. Fie A o mulțime nevidă astfel încât $A \cap \{\neg, \vee, \wedge, (,)\} = \emptyset$. Formulele propoziționale peste A sunt definite prin:

- (a) orice element al mulțimii A este formulă propozițională;
- (b) dacă w_1 și w_2 sunt formule propoziționale, atunci

$$(\neg w_1), (w_1 \vee w_2) și (w_1 \wedge w_2)$$

sunt formule propoziționale;

(c) formulele propoziționale sunt definite numai ca la (a) sau (b).

Un cititor pretențios din punct de vedere al rigorii matematice se poate întreba: este aceasta o “definiție” matematică a noțiunii de formulă propozițională (peste A)? Există o mulțime a tuturor formulelor propoziționale (peste A)? Cum pot fi demonstreate anumite proprietăți ale formulelor propoziționale?

Pentru a răspunde acestor întrebări, vom recrise “definiția” de mai sus în termeni de închidere a unei mulțimi. Considerăm relațiile r_0 și r_1 ce conțin toate perechile de forma $(x, (\neg x))$ și, respectiv, $((x, y), (x \vee y))$ și $((x, y), (x \wedge y))$, unde $x, y \in (A \cup \{\neg, \vee, \wedge, (\cdot)\})^+$. Atunci, presupunând că există mulțimea tuturor formulelor propoziționale peste A , fie aceasta $FP(A)$, punctele (a), (b) și (c) ne spun că:

(a') $FP(A)$ include A (de la (a));

(b') $FP(A)$ este închisă la r_0 și r_1 (de la (b));

(c') $FP(A)$ este cea mai mică mulțime, în sensul incluziunii, cu proprietățile (a') și (b') (de la (c)).

Adică, $FP(A)$ este închiderea mulțimii A la mulțimea $\{r_0, r_1\}$ de relații, mulțime ce există în baza Teoremei 1.3.1.1. Ca urmare, $FP(A)$ există și este reuniunea șirului de mulțimi $\langle B_m | m \geq 0 \rangle$ 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 inducției structurale poate fi aplicat mulțimii $FP(A)$ pentru a demonstra anumite proprietăți ale elementelor acesteia. Astfel, pentru a arăta că în orice formulă propozițională $w \in FP(A)$ 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 $w = (w_1 \wedge w_2)$ și presupunem proprietatea adevărată pentru w_1 și w_2 , atunci ea va fi adevărată și pentru w .

Discuția purtată până acum conduce la următoarea definiție. Spunem că o mulțime B este *definită inductiv* dacă există o mulțime A și o mulțime \mathcal{R} de relații astfel încât $B = \mathcal{R} \llbracket A \rrbracket$ ²⁴. Aceasta nu este o nouă definiție; nu am făcut altceva decât să atribuim o nouă terminologie noțiunii de închidere a unei mulțimi, să scoatem în evidență faptul că închiderea unei mulțimi poate acționa ca metodă de definiție de mulțimi și să justificăm “formularea” utilizată frecvent în descrierea inductivă a (obiectelor) unei mulțimi.

²⁴Unii autori [129, 53] atribuie terminologia de *constructor* relațiilor ce intervin într-o astfel de definiție. Analiza exemplului de mai sus credem că justifică cititorului această terminologie.

Fie $B = \mathcal{R} \llbracket A \rrbracket$. Observația 1.3.1.1(1) ne spune că pentru orice $b \in B$ există o secvență de elemente

$$a_1, \dots, a_i, \dots, a_n = b$$

astfel încât, pentru orice $1 \leq i \leq n$, are loc

– $a_i \in A$, sau

– există $r \in \mathcal{R}$ și $j_1, \dots, j_{n_r} < i$ astfel încât $((a_{j_1}, \dots, a_{j_{n_r}}), a_i) \in r$ (în cazul $n_r = 0$ înțelegem că $a_i \in r$).

O astfel de secvență poartă denumirea de *construcție/definiție inductivă* a lui b . Ca urmare, B este mulțimea tuturor elementelor ce au cel puțin o construcție inductivă de la A și \mathcal{R} .

Un alt aspect important pe care trebuie să-l discutăm este cel legat de definiția prin recursie/recurență a unor funcții al căror domeniu este o mulțime definită inductiv. Observația de la care plecăm constă în aceea că o funcție este o relație, deci o mulțime, și atunci, a defini recursiv o funcție revine la a defini inductiv o mulțime. Dar să fixăm întâi cu exactitate problematica pe care o urmărim.

Fie B o mulțime definită inductiv de A și \mathcal{R} , C o mulțime, g o funcție de la A la C^{n_r} la C (în cazul $n_r = 0$, $h(r)$ poate fi identificată cu un element din C). Ne punem problema existenței unei funcții $f : B \rightarrow C$ cu proprietățile:

(i) $f(a) = g(a)$, pentru orice $a \in A$;

(ii) $f(a) = h(r)(f(a_1), \dots, f(a_{n_r}))$, pentru orice a, a_1, \dots, a_{n_r} ce satisfac $((a_1, \dots, a_{n_r}), a) \in r$ și $h(r)(f(a_1), \dots, f(a_{n_r})) \downarrow$

(egalitatea de la (ii) trebuie înțeleasă ca egalitate de funcții parțial definite. Astfel, dacă $h(r)$ nu este definită pe $(f(a_1), \dots, f(a_{n_r}))$, atunci f nu va fi definită pe a).

Evident, pentru ca o astfel de funcție să existe este necesar ca pentru orice $((a_1, \dots, a_{n_r}), a) \in r$ și $((a'_1, \dots, a'_{n_r}), a) \in r'$ să avem

(iii) $h(r)(f(a_1), \dots, f(a_{n_r})) = h(r')(f(a'_1), \dots, f(a'_{n_r}))$.

O astfel de condiție, pe care am dorit-o satisfăcută a priori de definiția funcției f , implică însăși funcția f . Ea va fi însă satisfăcută dacă presupunem că pentru orice $a \in B$, ori $a \in A$, ori există o unică relație r și un unic n_r -uplu (a_1, \dots, a_{n_r}) astfel încât $((a_1, \dots, a_{n_r}), a) \in r$.

Așa cum am spus, o funcție este o relație, și deci o mulțime. Atunci a defini f cu proprietățile (i) și (ii) revine la a defini o mulțime $f \subseteq B \times C$ astfel încât:

(a) $(a, g(a)) \in f$, pentru orice $a \in A$;

(b) dacă $(a_1, b_1), \dots, (a_{n_r}, b_{n_r}) \in f$, $((a_1, \dots, a_{n_r}), a) \in r$ și $h(r)(b_1, \dots, b_{n_r}) \downarrow$, atunci $(a, h(r)(b_1, \dots, b_{n_r})) \in f$;

(c) f este cea mai mică mulțime, în sensul inclusiunii, cu proprietățile (a) și (b).

Următoarea lemă va stabili existența unei astfel de mulțimi.

Lema 1.3.3.1. Fie B mulțimea definită inductiv de A și \mathcal{R} , C o mulțime, g o funcție de la A la C și h o funcție ce asociază fiecărei relații $r \in \mathcal{R}$ o funcție $h(r)$ (parțială sau totală) de la C^{nr} la C . Atunci există o unică mulțime $f \subseteq B \times C$ cu proprietățile (a), (b) și (c).

Demonstrație. Fie mulțimea $A' = \{(a, g(a)) | a \in A\}$. Pentru fiecare relație r considerăm o nouă relație r' astfel încât

$$(((a_1, b_1), \dots, (a_{n_r}, b_{n_r})), (a, h(r)(b_1, \dots, b_{n_r}))) \in r'$$

dacă și numai dacă

$$(a_1, \dots, a_{n_r}), a \in r, \quad b_1, \dots, b_{n_r} \in C \text{ și } h(r)(b_1, \dots, b_{n_r}) \downarrow.$$

Fie $\mathcal{R}' = (r' | r \in \mathcal{R})$. Există atunci o unică mulțime B' ce este închiderea mulțimii A' la \mathcal{R}' . Pe baza definiției închiderii și a Prinzipiului inducției structurale se obține cu ușurință că $f = B'$ este unică mulțime ce satisface lema. \square

Mulțimea f din Lema 1.3.3.1 nu este în mod necesar funcție.

Definiția 1.3.3.1. O mulțime B spunem că este *liber inductiv definită* de A și \mathcal{R} dacă B este definită inductiv de A și \mathcal{R} și, pentru orice $a \in B$,

- ori $a \in A$,
- ori există o unică relație $r \in \mathcal{R}$ și un unic n_r -uplu (a_1, \dots, a_{n_r}) astfel încât $((a_1, \dots, a_{n_r}), a) \in r$ (în cazul $n_r = 0$ înțelegem că $a \in r$).

Teorema 1.3.3.1. (Teorema recursiei)

Fie B o mulțime definită inductiv de A și \mathcal{R} , C o mulțime, g o funcție de la A la C și h o funcție ce asociază fiecărei c-relații $r \in \mathcal{R}$ o funcție $h(r)$ (parțială sau totală) de la C^{nr} la C . Dacă B este liber inductiv definită de A și \mathcal{R} , atunci există o unică funcție (parțială) $f : B \rightarrow C$ astfel încât:

- (i) $f(a) = g(a)$, pentru orice $a \in A$;
- (ii) $f(a) = h(r)(f(a_1), \dots, f(a_{n_r}))$, pentru orice a, a_1, \dots, a_{n_r} ce satisface $((a_1, \dots, a_{n_r}), a) \in r$ și $h(r)(f(a_1), \dots, f(a_{n_r})) \downarrow$

(egalitatea de la (ii) este înțeleasă ca egalitate de funcții parțial definite).

Demonstrație. Lema 1.3.3.1 asigură existența unei mulțimi $f \subseteq B \times C$ cu proprietățile (a), (b) și (c). Prin inducție structurală și utilizând ipoteza (B este liber inductiv definită) se arată că f satisface teorema. \square

O ușoară modificare a demonstrației Lemei 1.3.3.1 permite stabilirea următoarei variante a Teoremei recursiei.

Teorema 1.3.3.2. Fie B o mulțime definită inductiv de A și \mathcal{R} , C o mulțime, g o funcție de la A la C și h o funcție ce asociază fiecărei relații $r \in \mathcal{R}$ o funcție $h(r)$ (parțială sau totală) de la $B^{nr} \times C^{nr}$ la C . Dacă B este liber inductiv definită de A și \mathcal{R} , atunci există o unică funcție (parțială) $f : B \rightarrow C$ astfel încât:

- (i) $f(a) = g(a)$, pentru orice $a \in A$;
- (ii) $f(a) = h(r)(a_1, \dots, a_{n_r}, f(a_1), \dots, f(a_{n_r}))$, pentru orice a, a_1, \dots, a_{n_r} ce satisface $((a_1, \dots, a_{n_r}), a) \in r$ și $h(r)(a_1, \dots, a_{n_r}, f(a_1), \dots, f(a_{n_r})) \downarrow$.

1.4. Sisteme relaționale și algebrelle universale

Structurile relaționale și algebrele universale constituie un cadru general prin care pot fi introduse mulțimile parțial ordonate, semigrupurile, grupurile etc. și prin care pot fi studiate proprietăți comune ale acestora.

1.4.1. Sisteme relaționale

De multe ori apare necesitatea de a considera cupluri formate dintr-o mulțime de bază și anumite relații pe acea mulțime. Aceste cupluri se numesc *sisteme relaționale*.

Definiția 1.4.1.1. Se numește *sistem relațional* orice cuplu $\mathcal{R} = (A; R)$, unde A este o mulțime arbitrară, iar R este o familie de relații pe A .

Dacă $\mathcal{R} = (A; R)$ este un sistem relațional și $R = \{\rho_1, \dots, \rho_n\}$, $n \geq 1$, atunci vom mai nota \mathcal{R} prin $(A, \rho_1, \dots, \rho_n)$. Mulțimea A va fi numită *mulțimea suport* a sistemului relațional \mathcal{R} . În cazul particular în care R este formată doar dintr-o singură relație binară ρ vom spune că $(A; R)$ este o *structură (relațională)*.

Structurile introduse în Definiția 1.2.6.1 sunt cazuri particulare de structuri relaționale.

Reprezentarea grafică a unei structuri relaționale se face așa cum este menționat în Secțiunea 1.2.6.

Definiția 1.4.1.2. Fie (A, ρ) o structură relațională. (B, σ) este numită *substructură* a structurii (A, ρ) dacă $B \subseteq A$ și $\sigma = \rho|_B$.

Observăm că o substructură (B, σ) a unei structuri (A, ρ) este complet determinată de submulțimea B . Ca urmare, ne vom referi uneori la substructuri ale unei structuri ca fiind submulțimi ale mulțimii suport a acestora.

Atunci când pentru o anumită structură relațională este adoptată o terminologie specifică, aşa cum ar fi de exemplu cea de "mulțime parțial ordonată", terminologia de substructură relațională va fi modificată corespunzător, cum ar fi de exemplu "submulțime parțial ordonată", abreviată *sub-mpo*.

Definiția 1.4.1.3. Fie (A, ρ) o structură, $a, b \in A$ și $B \subseteq A$.

- (1) Se numește *lanț al structurii* (A, ρ) orice submulțime $B \subseteq A$ cu proprietatea că $(B, \rho|_B)$ este lanț.
- (2) Se numește *submulțime dirijată a structurii* (A, \leq) orice submulțime nevidă $B \subseteq A$ cu proprietatea că $(B, \rho|_B)$ este mulțime dirijată.
- (3) Se numește *submulțime filtrată a structurii* (A, \leq) orice submulțime nevidă $B \subseteq A$ cu proprietatea că $(B, \rho|_B)$ este mulțime filtrată.
- (4) Mulțimea $[a, b] = \{x \in A | a \leq x \leq b\}$ este numită *segmentul sau intervalul induș de a și b în M*.
- (5) Mulțimea $B^{\uparrow} = \{x \in A | (\exists a \in B)(a \leq x)\}$ este numită *mulțimea succesorilor mulțimii B*.
- (6) Mulțimea $B^{\downarrow} = \{x \in A | (\exists a \in B)(x \leq a)\}$ este numită *mulțimea predecesorilor mulțimii B*.

Atunci când $B = \{a\}$ vom scrie mai simplu a^{\uparrow} (a^{\downarrow}) în loc de $\{a\}^{\uparrow}$ ($\{a\}^{\downarrow}$). Aceste notații nu trebuie confundate cu notațiile $f(a)^{\uparrow}$ sau $f(a)^{\downarrow}$.

Este ușor de văzut că are loc $[a, b] = a^{\uparrow} \cap b^{\downarrow}$, pentru orice a și b .

Conceptul de morfism de structuri este unul dintre cele mai importante concepte în teoria structurilor relaționale.

Definiția 1.4.1.4. Fie $M = (A, \rho)$ și $N = (B, \sigma)$ două structuri. Numim *homomorfism* sau *morfism* de la M la N orice funcție $f : A \rightarrow B$ 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 însăși se mai numesc și *endomorfisme*.

În Figura 1.11 sunt reprezentate grafic 3 structuri. Funcția f dată prin

$$f(a) = x = f(b), \quad f(c) = y \quad \text{și} \quad f(d) = z$$

este un morfism de la structura din Figura 1.11(a) la structura din Figura 1.11(b), dar este ușor de văzut că nu există nici un morfism de la structura din Figura 1.11(b) la cea din Figura 1.11(c).

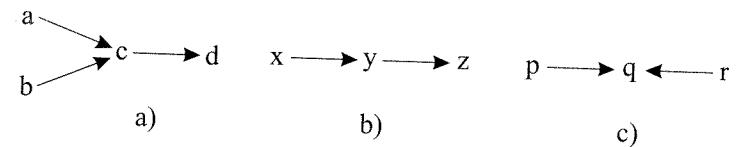


Figura 1.11: Structuri relaționale

Morfismele se mai numesc și *funcții monotone*²⁵. Ele au proprietatea că păstrează relația de ordine pe domeniul de definiție, ceea ce în Definiția 1.4.1.4 este specificat prin " $x \rho y \Rightarrow f(x) \sigma f(y)$ ". Ca o consecință importantă, funcțiile monotone păstrează *lanțurile (submulțimile dirijate, submulțimile filtrate)* în sensul că, dacă $f : M \rightarrow N$ este funcție monotonă de la M la N , atunci pentru orice lanț (submulțime dirijată, submulțime filtrată) B în M , $f(B)$ este lanț (submulțime dirijată, submulțime filtrată) în N .

În 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 (funcție monotonă strictă)*. Mulțimea tuturor funcțiilor monotone de la M la N va fi notată prin $(M \rightarrow_m N)$ sau $(A \rightarrow_m B)$ atunci când relațiile ρ și σ sunt subînțelese din context.

Definiția 1.4.1.5. Fie $M = (A, \rho)$ și $N = (B, \sigma)$ două structuri. Numim *izomorfism* de la M la N orice funcție bijectivă $f : A \rightarrow B$ pentru care f este morfism de la M la N și f^{-1} este morfism de la N la M .

Izomorfismele de la o structură la ea însăși se mai numesc și *automorfisme*.

Propoziția 1.4.1.1. Fie $M = (A, \rho)$ și $N = (B, \sigma)$ două structuri. O funcție f de la M la N este izomorfism de structuri dacă și numai dacă f este funcție bijectivă și are loc

$$(\forall x, y \in A)(x \rho y \Leftrightarrow f(x) \sigma f(y)).$$

Demonstrație. Dacă f este izomorfism de la M la N , atunci f este funcție bijectivă și au loc proprietățile

$$(\forall x, y \in A)(x \rho y \Rightarrow f(x) \sigma f(y))$$

și

$$(\forall x', y' \in B)(x' \sigma y' \Rightarrow f^{-1}(x') \rho f^{-1}(y')).$$

Alegând $x' = f(x)$ și $y' = f(y)$ în cea de a doua relație și combinând cu prima obținem

$$(\forall x, y \in A)(x \rho y \Leftrightarrow f(x) \sigma f(y)).$$

²⁵Unii autori folosesc termenul de "funcție monotonă" doar pentru morfisme între mpo. Considerăm însă că este potrivit să utilizăm această terminologie și pentru morfisme între structuri arbitrară, aşa cum s-a introdus în Definiția 1.4.1.4.

Demonstrația în sens invers decurge similar celei de mai sus. \square

Este clar că dacă f este izomorfism, atunci atât f , cât și f^{-1} sunt morfisme stricte. Dacă există cel puțin un izomorfism de la M la N , atunci vom spune că M și N sunt izomorfe și vom nota $M \cong N$. În plus, dacă f este un izomorfism de la M la N și dorim să specificăm aceasta atunci vom folosi notația $M \cong_f N$.

Propoziția 1.4.1.2. Componere de morfisme (izomorfisme) de structuri este morfism (izomorfism) de structuri.

Demonstrație. Fie $M = (A, \rho)$, $N = (B, \sigma)$ și $P = (C, \theta)$ trei structuri, iar $f : A \rightarrow B$ și $g : B \rightarrow C$ 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.

Demonstrația decurge similar în cazul izomorfismelor. \square

Fie $M = (A, \rho)$ și $N = (B, \sigma)$ două structuri și f un morfism de la M la N . Multimea $A /_{Ker(f)}$ poate fi organizată ca o structură considerând relația 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 $M /_{Ker(f)}$.

Propoziția 1.4.1.3. Fie f un morfism de la $M = (A, \rho)$ la $N = (B, \sigma)$. Funcția $g : A \rightarrow A /_{Ker(f)}$ dată prin $g(x) = [x]_{Ker(f)}$, pentru orice $x \in A$, este morfism de la M la $M /_{Ker(f)}$.

Demonstrație. 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. \square

1.4.2. Multimi parțial ordonate

Multimile parțial ordonate sunt structuri de importanță majoră. În această secțiune vom prezenta câteva din concepțele și proprietățile de bază ale acestora.

1.4.2.1. Concepțe de bază

Fie $M = (A, \rho)$ o mpo. Cel mai adesea relația ρ se notează prin \leq . Prin $<$ vom nota relația $\leq - i_A$. Inversa relației \leq ($<$) va fi notată prin \geq ($>$).

Dacă $a, b \in A$ și $a \leq b$ sau $b \leq a$, atunci vom spune că a și 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 . În ipoteza suplimentară în care $a \neq b$ și 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 *precedesor imediat* al lui b , iar b este numit *succesor imediat* al lui a .

Este ușor de văzut că o ordine parțială ρ este totală (pe A) dacă și numai dacă $\rho \cup \rho^{-1} = A \times A$. Într-un lanț, orice două elemente sunt comparabile. Se mai folosește adesea și terminologia de *antilanț* pentru mpo în care orice două elemente sunt incomparabile.

În cazul multimilor parțial ordonate, pe lângă reprezentarea grafică introdusă în Secțiunea 1.2.6, o altă reprezentare frecvent întâlnită este cea prin *diagramă Hasse*²⁶. Aceste diagrame se construiesc similar reprezentărilor specifice mpo utilizate de noi dar cu deosebirea că arcele ce unesc nodurile grafului sunt neorientate. În acest caz, orientarea este suplinită desenând nodul ce succede un alt nod mai sus decât acesta, pe verticală. De exemplu, în Figura 1.12 sunt reprezentate prin diagrame Hasse 4 mpo. Ultimile două reprezentări sunt ale aceleiași mpo a cărei relație binară este

$$\{(a, a), (b, b), (c, c), (d, d), (e, e), (f, f), (a, d), (a, e), (b, d), (b, f), (c, e), (c, f)\}.$$

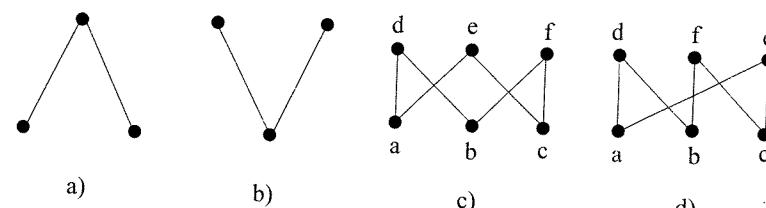


Figura 1.12: Diagrame Hasse

Definiția 1.4.2.1. Fie $M = (A, \leq)$ o mpo și $B \subseteq A$.

- (1) Un element $a \in A$ este numit *majorant* al mulțimii B dacă $b \leq a$, pentru orice $b \in B$.
- (2) Un element $a \in A$ este numit *cel mai mic majorant* al mulțimii B dacă el este majorant al lui B și pentru orice alt majorant a' al mulțimii B are loc $a \leq a'$.
- (3) B este numită *majorată* sau *mărginită superior* dacă există cel puțin 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 $b \leq b'$ sunt incomparabile.

²⁶După numele matematicianului german Helmut Hasse (1898–1979), care le-a utilizat pentru prima dată cu scopul reprezentării grafice a mpo.

- (5) Un element $b \in B$ este numit *cel mai mare element* (al mulțimii B) dacă $b' \leq b$, pentru orice $b' \in B$.
- (6) Un element $a \in A$ este numit *minorant* al mulțimii B dacă $a \leq b$, pentru orice $b \in B$.
- (7) Un element $a \in A$ este numit *cel mai mare minorant* al mulțimii B dacă el este minorant al lui B și pentru orice alt minorant a' al mulțimii B are loc $a' \leq a$.
- (8) B este numită *minorată* sau *mărginită inferior* dacă există cel puțin un minorant al ei.
- (9) Un element $b \in B$ este numit *minimal* (în B) dacă pentru orice $b' \in B$, $b \leq b'$ sau b și b' sunt incomparabile.
- (10) Un element $b \in B$ este numit *cel mai mic element* (al mulțimii B) dacă $b \leq b'$, pentru orice $b' \in B$.
- (11) B este numită *mărginită* dacă este mărginită inferior și superior.

Fie $M = (A, \leq)$ o mpo și $B \subseteq A$. Vom nota prin B^+ mulțimea tuturor majoranților mulțimii B în M , și prin B^- mulțimea tuturor minoranților mulțimii B în M . Atragem atenția asupra faptului că $B^+ \subseteq B^\uparrow$, dar inclusivitatea poate fi strictă (similar pentru $B^- \subseteq B^\downarrow$).

Este clar că dacă există cel mai mic majorant (cel mai mare minorant) al mulțimii B , atunci acesta este unic, el fiind cel mai mic element al mulțimii B^+ (cel mai mare element al mulțimii B^-). Acest element, atunci când există, va fi notat prin $\text{lub}_M(B)$ ($\text{glb}_M(B)$) sau $\text{sup}_M(B)$ ($\text{inf}_M(B)$) sau $\vee_M(b|b \in B)$ ($\wedge_M(b|b \in B)$) și va mai fi numit și *supremumul* (*infimumul*) mulțimii B . În cazul în care B are două elemente, $B = \{b_1, b_2\}$, ultima notație 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 mulțimii B , dacă există, este unic; el se notează (atunci când există) prin $\perp_{M,B}$ ($\top_{M,B}$) sau $\min_M(B)$ ($\max_M(B)$). Indicele M va fi întotdeauna eliminat atunci când se subînțelege din context.

Dacă elementul $a \in A$ are proprietatea $b < a$ pentru orice $b \in B$, atunci el este numit și *majorant strict* al (*margine superioară strictă* a) mulțimii B , iar B spune că este *majorată strict* (*mărginită superior strictă*). Dacă există cel mai mic majorant strict al mulțimii B , atunci acesta este numit și *succesor imediat* al lui B . Orice majorant strict este majorant, dar o mulțime poate avea majoranți fără a avea majoranți stricti. Notiunile de *minorant strict* și *mulțime minorată strict* (*mărginită inferior strictă*) se introduc în manieră similară. Cel mai mare minorant strict al unei mulțimi, atunci când există, este numit și *precedesor imediat* al acelei mulțimi.

Exemplul 1.4.2.1. Fie A o mulțime. 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 $S = \mathcal{P}(A)$, $\bigcap S = \emptyset$ și $\bigcup S = A$).

Utilizând concepțele introduse în Definiția 1.4.2.1, mpo dirijate (filtrate) pot fi definite ca fiind mpo ce au proprietatea că orice submulțime cu unul sau două elemente admite cel puțin un majorant (minorant). Evident, această proprietate este echivalentă cu a spune că orice submulțime finită și nevidă admite cel puțin un majorant (minorant).

Orice mpo care are cel mai mare element (cel mai mic element) este dirijată (filtrată).

Am spus în Secțiunea 1.4.1 că funcțiile monotone păstrează lanțurile și submulțimile dirijate și filtrate. Acest concept de păstrare a unei proprietăți poate fi extins. De exemplu, putem spune că funcția $f : M \rightarrow M'$ păstrează supremul submulțimilor nevide dacă pentru orice submulțime nevidă $B \subseteq A$ pentru care există $\sup_M(B)$, există și $\sup_{M'}(f(B))$ și are loc $\sup_{M'}(f(B)) = f(\sup_M(B))$. În manieră similară putem discuta despre funcții ce păstrează infimul etc.

Fie A o mulțime și $PO(A)$ mulțimea tuturor ordinilor parțiale pe A . Această mulțime este nevidă, deoarece $\iota_A \in PO(A)$, și poate fi structurată ca o mpo prin relația de incluziune pe relații de ordine parțială:

$$\rho \leq \theta \Leftrightarrow \rho \subseteq \theta,$$

pentru orice $\rho, \theta \in PO(A)$.

Propoziția 1.4.2.1. O ordine parțială ρ pe o mulțime A este totală dacă și numai dacă este element maximal al mulțimii parțial ordonate $(PO(A), \leq)$.

Demonstrație. Fie $\rho \in PO(A)$ o ordine totală. Presupunem prin contradicție că ρ nu este element maximal în $(PO(A), \leq)$. Atunci există $\theta \in PO(A)$ astfel încât $\rho < \theta$. Adică $\rho \subset \theta$ și, deci, există $a, b \in A$ astfel încât $a \neq b$ și $(a, b) \in \theta - \rho$. În plus, $(b, a) \notin \rho$, deoarece altfel am avea $(a, b) \in \theta$ și $(b, a) \in \theta$, de unde ar urma $a = b$. Deci ρ nu este ordine totală pe A , ceea ce constituie o contradicție.

Reciproc, presupunem că ρ este element maximal al mulțimii parțial ordonate $(PO(A), \leq)$ dar nu este ordine totală. Ca urmare, există $a, b \in A$ astfel încât $a \neq b$, $(a, b) \notin \rho$ și $(b, a) \notin \rho$. Este ușor de văzut că există o ordine parțială θ pe A ce include $\rho \cup \{(a, b)\}$. Dar atunci θ extinde strict relația ρ , ceea ce intră în contradicție cu maximalitatea relației ρ . \square

Teorema 1.4.2.1. (Teorema de reprezentare a mpilor)

Orice mpo (A, \leq) este izomorfă cu o mpo de forma (S, \subseteq_S) , unde S este un sistem peste A .

Demonstrație. Fie (A, \leq) o mpo. Pentru orice element $a \in A$ considerăm mulțimea $A_a = \{b \in A | b \leq a\}$ și fie $S = \{A_a | a \in A\}$. Cuplul (S, \subseteq_S) este mulțime parțial ordonată. Definim $f : A \rightarrow S$ prin $f(a) = A_a$. Această funcție este bijecție și 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) și (S, \subseteq_S) . \square

1.4.2.2. Dualitate

Este ușor de văzut că dacă ρ este relație de ordine parțială pe o mulțime A , atunci ρ^{-1} este de asemenea relație de ordine parțială pe A .

Definiția 1.4.2.2. Fie $M = (A, \leq)$ o mpo. Mulțimea parțial ordonată $M^{-1} = (A, \geq)$ este numită *duala* mulțimii parțial ordonate M .

Este util de remarcat că anumite concepțe valide pentru mpo au o contraparte pentru duală. De exemplu, dacă a este majorant pentru submulțimea B a mpo (A, \leq) , atunci a este minorant pentru B în duală (A, \geq) . Următorul tabel prezintă concepțele duale întâlnite până acum (alte concepțe duale vor fi întâlnite pe parcursul lucrării).

Concept	Concept dual
majorant	minorant
minorant	majorant
mulțime mărginită superior	mulțime mărginită inferior
mulțime mărginită inferior	mulțime 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 concepțele ce apar într-o afirmație asupra unei mpo M sunt înlăciute prin conceptul dual corespunzător, atunci se obține ceea ce se numește *afirmația duală* afirmației date. Dacă afirmația inițială este validă în M , atunci afirmația duală va fi validă în duala lui M . Aceasta este aşa numitul *Principiu al dualității pentru mpo* în baza căruia multe dintre demonstrații pot fi reduse la jumătate.

1.4.2.3. Proprietăți de bază ale supremumului și infimumului

Vom prezenta în continuare câteva simple proprietăți referitoare la cel mai mic majorant al unei submulțimi într-o mpo, proprietăți ce vor avea multiple aplicații în secțiunile următoare. Evident, toate acestea pot fi dualizate conținând astfel la proprietăți similare pentru cel mai mare minorant.

Propoziția 1.4.2.2. Fie $M = (A, \leq)$ o mpo. Atunci au loc următoarele proprietăți:

- (1) există $\sup(\emptyset)$ dacă și numai dacă M are cel mai mic element. În plus, dacă există cel mai mic element, atunci $\sup(\emptyset) = \perp_M = \inf(A)$;

- (2) există $\inf(\emptyset)$ dacă și numai dacă M are cel mai mare element. În plus, dacă există cel mai mare element, atunci $\inf(\emptyset) = \top_M = \sup(A)$.

Demonstrație. (1) Dacă există $\sup(\emptyset)$, atunci mulțimea de majoranți ai mulțimii vide este nevidă și $\sup(\emptyset)$ este cel mai mic element al acesteia. Însă observăm că mulțimea de majoranți ai mulțimii vide este A (în limbaj logic, faptul că $a \in A$ este majorant al mulțimii 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, existența $\sup(\emptyset)$ conduce la existența celui mai mic element al mulțimii A , deci a mpo M .

Reciproc, dacă există \perp_M , atunci mulțimea de majoranți ai mulțimii vide este nevidă, deoarece conține \perp_M care, evident, este și $\sup(\emptyset)$.

Este clar că dacă există \perp_M atunci $\sup(\emptyset) = \perp_M = \inf(A)$.
(2) se demonstrează similar proprietății de la (1). \square

Atragem explicit atenția asupra faptului că supremul mulțimii vide poate exista doar dacă mulțimea vidă este considerată submulțime a unei mpo nevide. Altfel spus, $\sup(\emptyset)$ în (\emptyset, \emptyset) nu există.

Următoarea propoziție se obține cu ușurință de la definiții și Propoziția 1.4.2.2.

Propoziția 1.4.2.3. Fie $M = (A, \leq)$ o mpo. Atunci următoarele afirmații sunt echivalente:

- (1) pentru orice submulțime $B \subseteq A$ există $\sup(B)$;
- (2) există \perp_M și pentru orice submulțime nevidă $B \subseteq A$ există $\sup(B)$;
- (3) pentru orice submulțime $B \subseteq A$ există $\inf(B)$;
- (4) există \top_M și pentru orice submulțime nevidă $B \subseteq A$ există $\inf(B)$;
- (5) Pentru orice submulțime $B \subseteq A$ există $\sup(B)$ și $\inf(B)$;
- (6) există \perp_M și \top_M , și pentru orice submulțime nevidă $B \subseteq A$ există $\sup(B)$ și $\inf(B)$.

Propoziția 1.4.2.4. Fie $M = (A, \leq)$ o mpo și $B \subseteq A$. Atunci au loc următoarele proprietăți:

- (1) dacă există $\sup(B)$, atunci există $\inf(B^+)$ și reciproc. În 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 $\{\inf(B)\}^-$ și $\sup(B)^-$ în loc de $\{\sup(B)\}^-$).

Demonstrație. Vom demonstra doar (1). Să presupunem că există $\text{sup}(B)$. Aceasta este cel mai mic majorant al mulțimii B^+ și, ca urmare, el coincide cu $\text{inf}(B^+)$. \square

Evident, Propoziția 1.4.2.4 poate fi dualizată.

Următoarea lemă va avea aplicații majore în Secțiunea 7.1, dar este importantă și ca rezultat de sine stătător.

Lema 1.4.2.1. (Lema *minsup–majinf*)

Fie $M = (A, \leq)$ o mpo. Pentru orice submulțime nevidă $B \subseteq A$ cu proprietatea $B^- \neq \emptyset$ are loc:

$$(\forall C \subseteq B^-)((\exists \text{sup}(C)) \Rightarrow \text{sup}(C) \in B^-).$$

Demonstrație. Fie $C \subseteq B^-$ astfel încât există $\text{sup}(C)$.

Dacă $C = \emptyset$, atunci $\text{sup}(C) = \perp_M$, ceea ce ne spune că $\text{sup}(C) \in B^-$. Să presupunem că C este nevidă. Orice element din B este majorant pentru C . Cum $\text{sup}(C)$ este cel mai mic majorant al mulțimii C , el va fi mai mic decât orice element din B . Ca urmare, $\text{sup}(C) \in B^-$. \square

Interpretăm această foarte importantă lemă prin aceea că mulțimea minoranților unei submulțimi nevide este închisă la supremum (de aici provine prima parte, *minsup*, din denumirea lemei). Prin dualizare, obținem că mulțimea majoranților unei submulțimi nevide este închisă la infimum (de aici provine a doua parte, *majinf*, din denumirea lemei).

Propoziția 1.4.2.5. Fie $M = (A, \leq)$ o mpo și $B = \{a_{ij} \in A \mid i \in I, j \in J\}$, unde I și J sunt mulțimi nevide. Dacă au loc:

(1) există $b_i = \text{sup}(\{a_{ij} \mid j \in J\})$, pentru orice $i \in I$;

(2) există $u = \text{sup}(\{b_i \mid i \in I\})$,

atunci există $\text{sup}(\{a_{ij} \mid i \in I, j \in J\})$ și $\text{sup}(\{a_{ij} \mid i \in I, j \in J\}) = u$.

Demonstrație. Presupunem că au loc (1) și (2). Atunci, este clar că u este majorant al mulțimii $\{a_{ij} \mid i \in I, j \in J\}$.

Dacă v este un alt majorant al acestei mulțimi, atunci pentru orice i , $b_i \leq v$ deoarece b_i este cel mai mic majorant al mulțimii $\{a_{ij} \mid j \in J\}$ (v fiind majorant al acestei mulțimi). Cum u este cel mai mic majorant al mulțimii $\{b_i \mid i \in I\}$, urmează $u \leq v$. Deci, $u = \text{sup}(\{a_{ij} \mid i \in I, j \in J\})$. \square

Corolarul 1.4.2.1. Fie $M = (A, \leq)$ o mpo și $B = \{a_{ij} \in A \mid i \in I, j \in J\}$, unde I și J sunt mulțimi nevide. Dacă au loc:

(1) există $b_i = \text{sup}(\{a_{ij} \mid j \in J\})$, pentru orice $i \in I$;

(2) există $u = \text{sup}(\{b_i \mid i \in I\})$;

(3) există $c_j = \text{sup}(\{a_{ij} \mid i \in I\})$, pentru orice $j \in J$;

(4) există $v = \text{sup}(\{c_j \mid j \in J\})$,

atunci există $\text{sup}(\{a_{ij} \mid i \in I, j \in J\})$ și $\text{sup}(\{a_{ij} \mid i \in I, j \in J\}) = u = v$.

Demonstrație. Direct de la Propoziția 1.4.2.5. \square

Deci calculul supremumului unei mulțimi dublu indexate se poate face calculând supremumul după unul din indecesi și apoi după celălalt (atunci când aceștia există).

Corolarul 1.4.2.2. Fie $M = (A, \leq)$ o mpo și $X, Y \subseteq A$ două submulțimi ale lui A . Dacă există $\text{sup}(X)$, $\text{sup}(Y)$ și $\text{sup}(\{\text{sup}(X), \text{sup}(Y)\})$, atunci există și $\text{sup}(X \cup Y)$ și acesta este $\text{sup}(\{\text{sup}(X), \text{sup}(Y)\})$.

Demonstrație. Evident, putem presupune că atât X , cât și Y sunt nevide. Corolarul poate fi demonstrat direct, similar Propoziției 1.4.2.5, dar poate fi obținut și drept caz particular al acesteia considerând $X = \{a_{1j} \mid j \in J\}$ și $Y = \{a_{2k} \mid k \in K\}$ cu $J \subseteq K$ sau $K \subseteq J$. Mai mult, putem presupune că $J = K$. În adevăr, dacă am presupune că $K \subset J$, atunci repetând un element din Y și indexându-l cu indecesi $2k$ cu $k \in J - K$, obținem o nouă mulțime $Y' = \{a_{2j} \mid j \in J\}$ pentru care $\text{sup}(Y') = \text{sup}(Y)$ și $X \cup Y = X \cup Y'$. \square

Definiția 1.4.2.3. Fie $M = (A, \leq)$ o mpo și $B, C \subseteq A$. Spunem că C este cofinală în B dacă pentru orice $b \in B$ există $c \in C$ astfel încât $b \leq c$.

Orice submulțime este cofinală în \emptyset , dar \emptyset nu este cofinală în nici o submulțime nevidă.

Propoziția 1.4.2.6. Fie $M = (A, \leq)$ o mpo și B, C submulțimi ale lui A astfel încât C este cofinală în B . Dacă există $\text{sup}(B)$ și $\text{sup}(C)$, atunci $\text{sup}(B) \leq \text{sup}(C)$.

Demonstrație. Deoarece C este cofinală în B , $\text{sup}(C)$ este majorant pentru B . Deci, $\text{sup}(B) \leq \text{sup}(C)$. \square

Propoziția 1.4.2.7. Fie $M = (A, \leq)$ o mpo și B, C submulțimi ale lui A astfel încât C este cofinală în B și $C \subseteq B$. Dacă există unul din $\text{sup}(B)$ sau $\text{sup}(C)$, atunci există și celălalt și ele sunt egale.

Demonstrație. Se observă că $B^+ = C^+$, de la care urmează propoziția. \square

Propoziția 1.4.2.8. Fie $f : M \rightarrow M'$ un morfism de mpo și $B \subseteq A$.

(1) Dacă există $\text{sup}(B)$, atunci $f(\text{sup}(B))$ este majorant pentru $f(B)$. Dacă în plus există și $\text{sup}(f(B))$, atunci $\text{sup}(f(B)) \leq' f(\text{sup}(B))$.

(2) Dacă există $\text{sup}(B)$ și f este izomorfism, atunci există și $\text{sup}(f(B))$ și are loc $\text{sup}(f(B)) = f(\text{sup}(B))$.

Demonstrație. (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 $\sup(B)$ și 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 obține că $f^{-1}(a')$ este majorant pentru $f^{-1}(f(B)) = B$ și $f^{-1}(a') < f^{-1}(f(\sup(B))) = \sup(B)$, ceea ce ar contrazice faptul că $\sup(B)$ este cel mai mic majorant pentru B (s-a folosit faptul că f este bijecție și f^{-1} este morfism strict).

Deci $f(\sup(B))$ este cel mai mic majorant pentru $f(B)$, ceea ce înseamnă că are loc $\sup(f(B)) = f(\sup(B))$. \square

Corolarul 1.4.2.3. Fie $f : M \rightarrow M'$ un morfism de mpo.

(1) Dacă există \perp_M și $\perp_{M'}$, atunci $\perp_{M'} \leq' f(\perp_M)$.

(2) Dacă există \perp_M și f este izomorfism, atunci există și $\perp_{M'}$ și $\perp_{M'} = f(\perp_M)$.

Demonstrație. Atât (1), cât și (2) pot fi demonstrate direct sau utilizând Propoziția 1.4.2.8 cu $B = \{\perp_M\}$. \square

Putem spune deci că izomorfismele păstrează supremumul submulțimilor, atunci când acesta există. În particular, izomorfismele păstrează cel mai mic element, atunci când acesta există.

1.4.2.4. Construcții de mpo

Arătăm în continuare cum putem construi noi mulțimi parțial ordonate pornind de la mulțimi parțial ordonate date. Vom spune că o familie de mpo (indexată sau nu) este *familie disjunctă de mpo* dacă mulțimile suport sunt disjuncte două câte două.

Mpo plate. Fie A o mulțime. Considerăm un nou element, notat \perp_A sau \perp , atunci când A este subînțeleasă din context ($\perp_A \notin A$), și fie $A_\perp = A \cup \{\perp_A\}$.

Definiția 1.4.2.4. Fie A o mulțime. *Mulțimea parțial ordonată plată indușă de A* este mulțimea parțial 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 ușor că, în adevăr, (A_\perp, \leq) este mpo. Elementul \perp_A este cel mai mic element al mulțimii parțial ordonate (A_\perp, \leq) și orice element $a \in A$ este element maximal. (A_\perp, \leq) are cel mai mare element dacă și numai dacă $|A| \leq 1$.

Această construcție este importantă în special atunci când A are cel puțin 2 elemente. Prin intermediul ei se introduce un cel mai mic element, elementele mulțimii A fiind tratate egal.

Dacă \perp este un element arbitrar, atunci vom nota prin \perp mulțimea parțial ordonată $\perp = (\{\perp\}, \{(\perp, \perp)\})$.

Pe mpo plate proprietatea de monotonie admite caracterizări foarte simple.

Propoziția 1.4.2.9. Fie $f : A_\perp \rightarrow B_\perp$. Atunci f este monotonă dacă și numai dacă $f(\perp_A) = \perp_B$ sau există $c \in B_\perp$ astfel încât $f(a) = c$, pentru orice $a \in A_\perp$.

Demonstrație. Urmează direct de la definiția ordinii parțiale pe mpo plate. \square

Intersecție de mpo. *Intersecția unei familii $((A_i, \leq_i)|i \in I)$ de mpo este definită prin*

$$\bigcap_{i \in I} (A_i, \leq_i) = (\bigcap_{i \in I} A_i, \bigcap_{i \in I} \leq_i).$$

Este cât se poate de clar că intersecție de mpo este mpo. De asemenea, este ușor de verificat că intersecția unei familii de sub-mpo ale unei mpo M este sub-mpo a mpo M .

Intersecția mpo din Figura 1.13(a)(b) este reprezentată grafic în Figura 1.13(c).

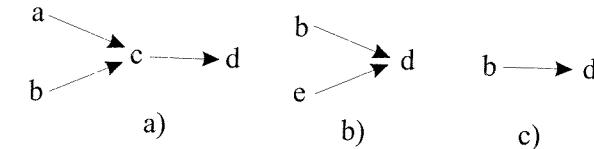


Figura 1.13: Intersecție de mpo

Reuniune de mpo. *Reuniunea unei familii $((A_i, \leq_i)|i \in I)$ de mpo este definită prin*

$$\bigcup_{i \in I} (A_i, \leq_i) = (\bigcup_{i \in I} A_i, \bigcup_{i \in I} \leq_i).$$

Spre deosebire de intersecție, reuniunea unei familii de mpo poate să nu mai fie mpo. Dacă însă familia este disjunctă, atunci și reuniunea este mpo.

Reuniunea mpo din Figura 1.13(a)(b) este reprezentată grafic în Figura 1.14.

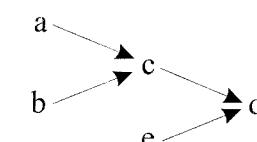


Figura 1.14: Reuniune de mpo

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.

Propoziția 1.4.2.10. Fie $\mathcal{I} = (I, \leq)$ o mto și $((A_i, \leq_i) | i \in I)$ o familie disjunctă de mpo. Atunci relația binară \leq' pe $\bigcup_{i \in I} A_i$ dată prin

$$x \leq' y \Leftrightarrow (\exists i, j \in I)(x \in A_i \wedge y \in A_j \wedge (\text{ori } i < j \text{ ori } (i = j \wedge x \leq_i y))),$$

pentru orice $x, y \in \bigcup_{i \in I} A_i$, este relație de ordine parțială.

Demonstrație. Vom verifica reflexivitatea, antisimetria și tranzitivitatea relației \leq' . Fie $x \in \bigcup_{i \in I} A_i$. Deoarece mulțimile acestei familii sunt disjuncte, urmează că există un unic $i \in I$ astfel încât $x \in A_i$. Reflexivitatea relației \leq_i conduce la $x \leq' x$, adică \leq' este reflexivă.

Considerăm $x, y \in \bigcup_{i \in I} A_i$ astfel încât $x \leq' y$ și $y \leq' x$. Utilizând iarăși faptul că mulțimile acestei familii sunt disjuncte obținem că există două unice elemente $i, j \in I$ astfel încât $x \in A_i$ și $y \in A_j$. Cuplul (I, \leq) este mulțime total ordonată și, ca urmare, are loc doar unul dintre următoarele cazuri: $i < j$, $i = j$ sau $i > j$. Cazul $i < j$ este imposibil deoarece $y \leq'_i x$ și, similar, cazul $i > j$. Urmează atunci că $i = j$ și, pe baza antisimetriei relației \leq_i obținem $x = y$. Deci, \leq' este antisimetrică.

Tranzitivitatea relației \leq' se stabilește în mod similar. \square

Această propoziție asigură consistența conceptului de *sumă ordonată* de mpo.

Definiția 1.4.2.5. Fie $\mathcal{I} = (I, \leq)$ o mto și $((A_i, \leq_i) | i \in I)$ o familie disjunctă de mpo. *Suma ordonată* a acestei familii, notată $\sum_{i \in I}^o (A_i, \leq_i)$ sau $\oplus_{i \in I} (A_i, \leq_i)$, este definită ca fiind mulțimea parțial ordonată

$$\sum_{i \in I}^o (A_i, \leq_i) = (\bigcup_{i \in I} A_i, \leq'),$$

unde \leq' este relația de ordine parțială din Propoziția 1.4.2.10.

Suma ordonată a două mpo disjuncte (A_1, \leq_1) și (A_2, \leq_2) , considerate în această ordine, va mai fi notată prin

$$(A_1, \leq_1) \oplus (A_2, \leq_2),$$

notație care se extinde în mod natural la un număr finit arbitrar de mpo disjuncte.

Dacă A este o mulțime, atunci putem scrie

$$(A_\perp, \leq) = \perp \oplus (A, \iota_A) = \perp \oplus \left(\bigcup_{a \in A} (\{a\}, \iota_{\{a\}}) \right).$$

Constatăm că operatorul de sumă ordonată este asociativ.

Suma ordonată a mpo din Figura 1.15(a)(b) este reprezentată grafic în Figura 1.15(c).

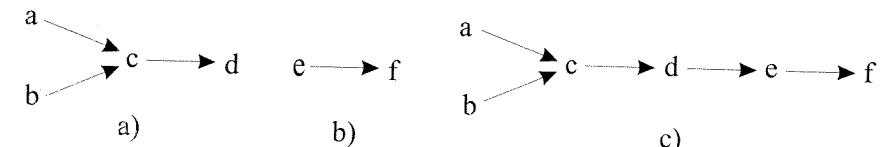


Figura 1.15: Sumă de mpo

Produse de mpo. Fie $n \geq 1$ un număr natural și (A_i, \leq_i) mpo, $1 \leq i \leq n$. Produsul cartezian $A_1 \times \dots \times A_n$ poate fi organizat ca mpo considerând relația binară \leq dată prin

$$(a_1, \dots, a_n) \leq (b_1, \dots, b_n) \Leftrightarrow (\forall 1 \leq i \leq n)(a_i \leq_i b_i),$$

pentru orice $(a_1, \dots, a_n), (b_1, \dots, b_n) \in A_1 \times \dots \times A_n$.

Este trivial de arătat că această relație binară este ordine parțială pe mulțimea $A_1 \times \dots \times A_n$. Ca urmare, $(A_1 \times \dots \times A_n, \leq)$ este mpo. Vom nota această mpo prin $\times_{i=1}^n (A_i, \leq_i)$ sau $(A_1, \leq_1) \times \dots \times (A_n, \leq_n)$ și o vom numi *produsul cartezian* al mpilor $(A_1, \leq_1), \dots, (A_n, \leq_n)$.

În Figura 1.16(c) este reprezentat grafic produsul cartezian al mpo din Figura 1.16(a)(b).

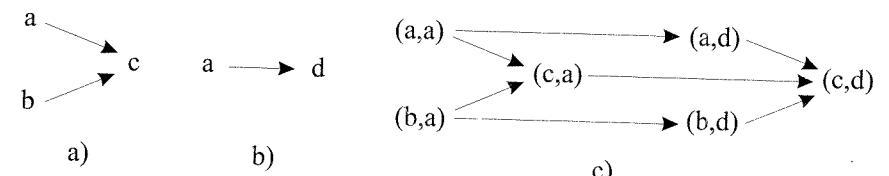


Figura 1.16: Produs cartezian de mpo

Propoziția 1.4.2.11. Fie $f : A_\perp^n \rightarrow B_\perp$, unde $n \geq 1$. Dacă f este monotonă, atunci $f(\perp_A, \dots, \perp_A) = \perp_B$ sau există $c \in B_\perp$ astfel încât $f(a) = c$, pentru orice $a \in A_\perp^n$.

Demonstrație. Dacă $f(\perp_A, \dots, \perp_A) = c$ și $c \neq \perp_B$, atunci se arată cu ușurință (utilizând ordinea parțială pe mpo plate) că $f(a) = c$ pentru orice $a \in A_\perp^n$. \square

Reciproca Propoziției 1.4.2.11 nu este în general adevărată dacă $n \geq 2$. De exemplu, funcția $f : \mathbf{R}_\perp^2 \rightarrow \mathbf{R}_\perp$ dată prin

$$f(x, y) = \begin{cases} x/y, & \text{dacă } x, y \in \mathbf{R} \text{ și } y \neq 0 \\ 0, & \text{dacă } x \in \mathbf{R} \text{ și } y = \perp \\ \perp, & \text{altfel,} \end{cases}$$

unde x/y este împărțirea uzuală pe \mathbf{R} , satisfacă $f(\perp, \perp) = \perp$, dar f nu este monotonă deoarece $f(1, \perp) = 0 \not\leq 1 = f(1, 1)$.

Propoziția 1.4.2.11 ne spune că o funcție ne-constantă $f : A_{\perp}^n \rightarrow B_{\perp}$ trebuie să satisfacă în mod necesar $f(\perp_A, \dots, \perp_A) = \perp_B$ pentru a putea fi monotonă. Evident, aceasta nu este suficient.

Definiția 1.4.2.6. O funcție $f : A_{\perp}^n \rightarrow B_{\perp}$, unde A și B sunt două mulțimi arbitrară și $n \geq 1$, este numită *extinsă naturală* dacă $f(a_1, \dots, a_n) = \perp$, ori de câte ori există $1 \leq i \leq n$ astfel încât $a_i = \perp_A$, pentru orice $(a_1, \dots, a_n) \in A^n$.

Propoziția 1.4.2.12. Orice funcție extinsă naturală este monotonă.

Demonstrație. Urmează direct de la definiția ordinii parțiale pe mpo plate. \square

Oricarei funcții $f : A^n \rightarrow B$, unde A și B sunt două mulțimi arbitrară și $n \geq 1$, i se poate asocia în mod unic o *extensie naturală* $f' : A_{\perp}^n \rightarrow B_{\perp}$ dată prin

$$f'(a_1, \dots, a_n) = \begin{cases} f(a_1, \dots, a_n), & \text{dacă } (\forall 1 \leq i \leq n)(a_i \neq \perp_A) \\ \perp_B, & \text{altfel,} \end{cases}$$

pentru orice $(a_1, \dots, a_n) \in A^n$.

Exemplul 1.4.2.2.

(1) Funcția $f : \mathbf{R}_{\perp}^2 \rightarrow \mathbf{R}_{\perp}$ 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 naturală și, prin urmare, este monotonă.

(2) Predicatul de egalitate $= : A^2 \rightarrow \{0, 1\}$ poate fi extins la A_{\perp} astfel:

1. natural, ceea ce va conduce la faptul că această extensie este funcție monotonă. Uzual, extensia naturală a acestui predicat se notează tot prin $=$ sau prin $=_w$ și se mai numește *predicatul de egalitate slabă*;

2. $=_s : A_{\perp}^2 \rightarrow \{0, 1\}_{\perp}$ prin

$$=_s(a, b) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{dacă } ((a, b \in A \wedge a = b) \vee (a = b = \perp_A)) \\ 0, & \text{altfel,} \end{cases}$$

pentru orice $a, b \in A_{\perp}$. $=_s$ se mai numește și *predicatul de egalitate tare*.

Este ușor de văzut că $=_s$ nu este o funcție monotonă deoarece

$$=_s(\perp, a) \not\leq =_s(a, a),$$

pentru orice $a \in A$ (presupunând că A este nevidă).

(3) Fie funcția *if_then_else* : $\{0, 1\} \times A \times A \rightarrow A$ dată prin

$$\text{if_then_else}(b, x, y) = \begin{cases} x, & \text{dacă } b = 1 \\ y, & \text{dacă } b = 0, \end{cases}$$

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 funcții la $\{0, 1\}_{\perp} \times A_{\perp} \times A_{\perp}$, notată tot prin *if_then_else* și dată prin

$$\text{if_then_else}(b, x, y) = \begin{cases} x, & \text{dacă } b = 1 \\ y, & \text{dacă } b = 0 \\ \perp_A, & \text{dacă } b = \perp_{\{0, 1\}} \end{cases}$$

este funcție monotonă chiar dacă nu este extensie naturală. Aceasta se poate arăta cu ușurință luând în discuție cele 3 cazuri posibile pentru b .

Produsul cartezian al unei familii finite de mulțimi este generalizat prin intermediul produsului direct la familii arbitrară de mulțimi. Aceeași generalizare se poate aplica și pentru mpo. Înainte 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 mulțimea tuturor funcțiilor f de la I la $\bigcup_{i \in I} A_i$ cu proprietatea $f(i) \in A_i$, pentru orice $i \in I$. Următoarea propoziție, a cărei demonstrație este imediată, va asigura consistența noțiunii de produs direct de mpo.

Propoziția 1.4.2.13. Fie $((A_i, \leq_i) | i \in I)$ o familie de mpo. Atunci relația 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 relație de ordine parțială.

Definiția 1.4.2.7. Fie $((A_i, \leq_i) | i \in I)$ o familie de mulțimi parțial ordonate. *Produsul direct* al familiei $((A_i, \leq_i) | i \in I)$, notat $\prod_{i \in I}^o (A_i, \leq_i)$ sau $\otimes_{i \in I} (A_i, \leq_i)$, este definit ca fiind mulțimea parțial ordonată

$$\prod_{i \in I}^o (A_i, \leq_i) = (\prod_{i \in I} A_i, \leq'),$$

unde \leq' este relația de ordine parțială din Propoziția 1.4.2.13.

Cazul familiilor nedisjuncte. Așa cum observăm, proprietatea de disjunctivitate a familiei $((A_i, \leq_i) | i \in I)$ este esențială în definiția reuniunii și a sumei ordonate. În cazul în care această proprietate nu este asigurată se poate recurge la diverse variante prin care să se poată introduce conceptul de reuniune și de sumă ordonată de mpo. Una dintre variantele des întâlnite este de a considera $A_i \times \{i\}$, pentru orice $i \in I$, și de a defini reuniunea sau produsul având în vedere aceste mulțimi. Astfel, definim *reuniunica 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 relația \leq' este def

$$x \leq' y \Leftrightarrow (\exists i \in I)(x = (a, i)$$

pentru orice $x, y \in A$ (și în acest caz se a
pe $\bigcup_{i \in I} (A_i \times \{i\})$).

În manieră similară se poate defini \leq
 $((A_i, \leq_i) | i \in I)$ de mpo, notată prin $\sum_{i \in I}^{\text{ord}}$

mpo
"multimi parțial ordonate"
 ordine parțială
 a unei familii

1.4.3. Latici

Structurile laticeale, ce își au originea în studiile lui George Boole asupra logicii [14], sunt cazuri particulare de mulțimi parțial ordonate ce apar în cele mai variate domenii: matematică, fizică, informatică, biologie, geologie etc. Cunoașterea proprietăților de bază ale acestora este nu numai benefică, dar și necesară.

1.4.3.1. Laticea ca mulțime parțial ordonată

Laticile pot fi introduse în două moduri (echivalente): ca mulțimi parțial ordonate sau ca algebre. În această secțiune vom discuta prima variantă.

Definiția 1.4.3.1. 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 și sup-semilatice.

Observăm că perechea (\emptyset, \emptyset) este atât inf-semilatice cât și sup-semilatice și 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 1.4.3.1.

- (1) Orice lanț este latice.
- (2) Dacă A este o mulțime arbitrară, atunci $(\mathcal{P}(A), \subseteq)$ este latice.
- (3) Orice familie de mulțimi ce este închisă la reuniune și la intersecție este latice (în raport cu inclusiunea).

Pentru latici se utilizează în mod ușual reprezentarea grafică prin diagrame Hasse.

Propoziția 1.4.3.1. Fie $M = (A, \leq)$ o mpo. M este inf-semilatice dacă și numai dacă pentru orice submulțime finită și nevidă $B \subseteq A$ există $\inf(B)$.

Demonstrație. Este clar că dacă pentru orice submulțime finită și nevidă $B \subseteq A$ există $\inf(B)$ atunci M este inf-semilatice.

Reciproca se obține prin aplicarea repetată a Corolarului 1.4.2.2. □

Propoziția 1.4.3.1 poate fi dualizată pentru cazul sup-semilaticilor și, împreună conduc la faptul că o mpo M este latice dacă și numai dacă pentru orice submulțime finită și nevidă $B \subseteq A$ există $\inf(B)$ și $\sup(B)$.

O generalizare naturală a conceptului de latice, prin prisma Propoziției 1.4.3.1, este următoarea.

Definiția 1.4.3.2. Fie $M = (A, \leq)$ o mpo.

- (1) M este numită *inf-semilatice completă* dacă pentru orice submulțime nevidă $B \subseteq A$ există $\inf(B)$.
- (2) M este numită *sup-semilatice completă* dacă pentru orice submulțime nevidă $B \subseteq A$ există $\sup(B)$.
- (3) M este numită *latice completă* dacă este atât inf-semilatice completă, cât și sup-semilatice completă.

Și în acest caz observăm că structura (\emptyset, \emptyset) este atât inf-semilatice completă, cât și sup-semilatice completă și latice completă.

Observația 1.4.3.1. 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 și cel mai mare element. În mod ușual, în 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 propoziție urmează direct de la definiții și Propoziția 1.4.3.1.

Propoziția 1.4.3.2. Orice inf-semilatice (sup-semilatice, latice) finită este inf-semilatice (sup-semilatice, latice) completă.

Utilizând echivalențele din Propoziția 1.4.2.3, demonstrația următoarei teoreme este imediată.

Teorema 1.4.3.1. Fie $M = (A, \leq)$ o mpo nevidă. Atunci următoarele afirmații sunt echivalente:

- (1) M este latice completă;

- (2) Pentru orice submulțime $B \subseteq A$, există $\inf(B)$;
 (3) M este inf-semilatice completă ce are cel mai mare element.

Evident, Teorema 1.4.3.1 poate fi dualizată înlocuind “inf” cu “sup” și “cel mai mare element” cu “cel mai mic element”.

Exemplul 1.4.3.2. (Laticea completă a submulțimilor unei mulțimi)

Fie A o mulțime și $S \subseteq \mathcal{P}(A)$ un sistem peste A astfel încât $A \in S$ și S este închis la intersecții de familii nevide cu elemente din S . În baza Teoremei 1.4.3.1, (S, \subseteq) este latice completă. În particular, $(\mathcal{P}(A), \subseteq)$ este latice completă, numită *laticea (completă a) submulțimilor mulțimii* A .

Exemplul 1.4.3.3. (Laticea completă a relațiilor de echivalență)

Mulțimea relațiilor de echivalență $E(A)$ peste o mulțime A , cu incluziunea, formează latice completă, numită *laticea completă a relațiilor de echivalență pe* A . În adevăr, fie $(\rho_i | i \in I)$ o familie nevidă de relații de echivalență pe A . Este trivial de verificat că $\bigcap_{i \in I} \rho_i$ este relație de echivalență pe A , ea fiind $\inf(\rho_i | i \in I)$.

Cum A^2 este cea mai mare relație de echivalență pe A , în baza Teoremei 1.4.3.1, $(E(A), \subseteq)$ este latice completă.

Evident, este interesant de știut care este $\sup(\rho_i | i \in I)$ în această latice. Putem răspunde foarte simplu prin: $\sup(\rho_i | i \in I)$ este închiderea la echivalență a relației $\bigcup_{i \in I} \rho_i$. Dacă apelăm la modul de construcție a închiderii, obținem că $\sup(\rho_i | i \in I)$ este relația θ dată prin

$$\begin{aligned} x \theta y &\Leftrightarrow (\exists x_0, \dots, x_n)(x_0 = x \wedge y = x_n \wedge \\ &(\forall 1 \leq j \leq n)(\exists i_j \in I)(x_{j-1} \rho_{i_j} x_j)), \end{aligned}$$

pentru orice $x, y \in A$.

Diagrama din Figura 1.17 prezintă schematic relația dintre inf-semilatici complete, sup-semilatici complete și latici complete.

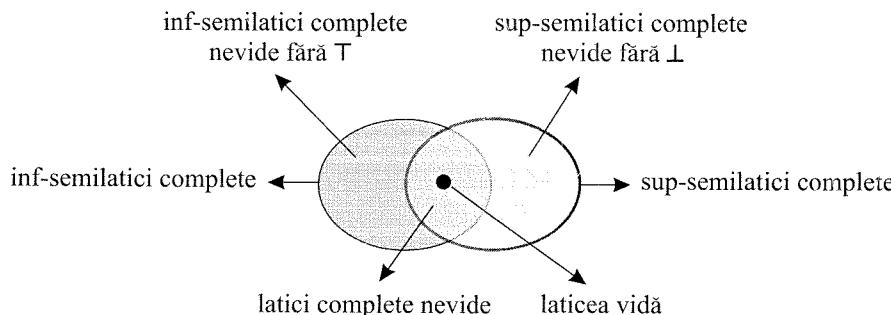


Figura 1.17: Inf-semilatici complete, sup-semilatici complete și latici complete

Asupra laticilor complete vom reveni în Secțiunea 7.1.1.

1.4.3.2. Laticea ca structură algebrică

Fie $M = (A, \leq)$ o latice. Definiția laticei cât și faptul că supremumul este unic asigură consistența definiției unei operații binare pe A , notată \vee ²⁷ și dată prin

$$\vee(a, b) = \sup(\{a, b\}),$$

pentru orice $a, b \in A$. În mod similar putem defini operația binară \wedge dată prin

$$\wedge(a, b) = \inf(\{a, b\}),$$

pentru orice $a, b \in A$. Este ușor de văzut că aceste două operații satisfac următoarele proprietăți:

- $a \vee b = b \vee a$ și $a \wedge b = b \wedge a$; (comutativitate)
- $a \vee (b \vee c) = (a \vee b) \vee c$ și $a \wedge (b \wedge c) = (a \wedge b) \wedge c$; (asociativitate)
- $a \wedge (a \vee b) = a$ și $a \vee (a \wedge b) = a$; (absorbție)

pentru orice $a, b \in A$. În plus,

- $a \leq b \Leftrightarrow a \vee b = b \Leftrightarrow a \wedge b = a$,

pentru orice $a, b \in A$.

Proprietatea de absorbție conduce la idempotența operațiilor \vee și \wedge . În adevăr,

$$a \vee a = a \vee (a \wedge (a \vee a)) = a,$$

pentru orice a (prima egalitate se obține înlocuind al doilea a din membrul stâng al egalității prin $a \wedge (a \vee a)$, iar a doua egalitate se obține aplicând absorbția în forma $a \vee (a \wedge b) = a$). În 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 mulțime și \vee și \wedge sunt două operații binare pe A ce satisfac proprietățile de comutativitate, asociativitate și absorbție (ca mai sus), relația binară \leq dată prin

$$a \leq b \Leftrightarrow a \vee b = b,$$

pentru orice $a, b \in A$, structurează A ca o latice. Teorema de mai jos ne arată aceasta.

Teorema 1.4.3.2.

- (1) Fie $M = (A, \leq)$ o latice. Atunci structura $M^a = (A, \vee, \wedge)$, unde \vee și \wedge sunt operațiile binare pe A date prin

$$\vee(a, b) = \sup(\{a, b\}) \text{ și } \wedge(a, b) = \inf(\{a, b\}),$$

²⁷Această notație nu trebuie confundată cu notația pentru supremum introdusă în Secțiunea 1.4.2. De fapt, am evitat să utilizăm acea notație până acum întocmai pentru a nu crea confuzii. Această remarcă va fi valabilă și pentru notația \wedge ce urmează a fi introdusă ca notăție de operație binară.

pentru orice $a, b \in A$, verifică proprietățile de comutativitate, asociativitate și absorbție. În plus,

$$a \leq b \Leftrightarrow a \vee b = b \Leftrightarrow a \wedge b = a,$$

pentru orice $a, b \in A$.

- (2) Fie $M = (A, \vee, \wedge)$ o structură formată dintr-o mulțime A și două operații binare pe A ce verifică proprietățile de comutativitate, asociativitate și absorbție. Atunci cuplul $M^o = (A, \leq)$, unde \leq este relația binară dată prin

$$a \leq b \Leftrightarrow a \vee b = b,$$

pentru orice $a, b \in A$, este latice. În plus, $\sup(\{a, b\}) = a \vee b$ și $\inf(\{a, b\}) = a \wedge b$, pentru orice $a, b \in A$.

- (3) Fie $M = (A, \leq)$ o latice. Atunci $(M^o)^o = M$.
(4) Fie $M = (A, \vee, \wedge)$ o structură ca la (2). Atunci $(M^o)^o = M$.

Demonstrație. Lăsăm (1), (3) și (4) în seama cititorului și ne vom ocupa de (2). Reflexivitatea relației binare \leq decurge imediat de la proprietatea de idempotență a operației \vee (proprietate indusă de absorbție). În adevăr, relația $a \vee a = a$ ne spune că $a \leq a$, pentru orice a .

Fie $a, b \in A$ astfel încât $a \leq b$ și $b \leq a$. Atunci $a \vee b = b$ și $b \vee a = a$, iar comutativitatea conduce la

$$a = a \vee b = b \vee a = b,$$

care stabilește antisimetria relației \leq .

Fie $a, b, c \in A$ astfel încât $a \leq b$ și $b \leq c$. Atunci $a \vee b = b$ și $b \vee c = c$, iar asociativitatea conduce la

$$a \vee c = a \vee (b \vee c) = (a \vee b) \vee c = b \vee c = c,$$

care ne arată că $a \leq c$, și astfel \leq este tranzitivă.

Ca urmare, \leq este ordine parțială pe A .

Fie $a, b \in A$. Vom arăta că există $\sup(\{a, b\})$ și $\inf(\{a, b\})$. Fie $c = a \vee b$ (acest element există deoarece \vee este operație binară pe A , deci definită pentru orice două elemente din A). Arătăm că c este majorant pentru $\{a, b\}$. În 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 $a \leq c$. În mod similar obținem $b \leq c$, ceea ce conduce la faptul că c este majorant pentru $\{a, b\}$.

Dacă d este un alt majorant pentru $\{a, b\}$, atunci $a \leq d$ și $b \leq d$, ceea ce conduce la $a \vee d = d$ și $b \vee d = d$. Atunci,

$$d = a \vee d = a \vee (b \vee d) = (a \vee b) \vee d = c \vee d,$$

care ne arată că $c \leq d$. Deci $c = \sup(\{a, b\})$. În mod similar se arată că are loc $\inf(\{a, b\}) = a \wedge b$. \square

Ca urmare, o latice poate fi văzută atât ca mulțime parțial ordonată, cât și ca structură algebrică. În plus, atunci când lucrăm cu latice putem folosi \vee (\wedge) atât pentru a specifica supremumul (infimumul), cât și ca operație binară.

Privind laticea ca structură algebrică, putem recurge la specificări ale acesteia prin intermediul diagramelor operațiilor \vee și \wedge . Cum aceste operații sunt comutative și idempotente, tabelele lor pot fi reduse la jumătate, renunțând și la diagonală. Ca urmare, ambele tabele pot fi cumulate în unul singur (fără diagonală). De exemplu, tabelul

$\vee \wedge$	0	a	b	1
0	0	0	0	0
a	a	0	a	a
b	b	1	b	b
1	1	1	1	1

specifică laticea ale cărei operații 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 proprietăți simple ce au loc în latice.

Propoziția 1.4.3.3. Fie $M = (A, \leq)$ o latice și $a, b_i \in A$, unde $1 \leq i \leq n$ și $n \geq 1$. 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 $\bigvee_{i=1}^n b_i \leq a$.

Demonstrație. 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 mulțimii $\{b_i | 1 \leq i \leq n\}$. Procedăm similar pentru cealaltă proprietate (sau prin dualizare). \square

Propoziția 1.4.3.4. (Proprietăți de idempotență)

Fie $M = (A, \leq)$ o latice. Atunci, pentru orice $a \in A$ și $n \geq 1$, au loc proprietățile de idempotență $\bigvee_{i=1}^n a = a$ și $\bigwedge_{i=1}^n a = a$.

Demonstrație. Pentru $n = 2$ proprietatea a fost deja demonstrată. Cazul general se poate obține cu ușurință prin inducție matematică. \square

Propoziția 1.4.3.5. Fie $M = (A, \leq)$ o latice și $a_i, b_i \in A$, unde $1 \leq i \leq n$ și $n \geq 1$. Dacă $a_i \leq b_i$, pentru orice i , atunci $\bigvee_{i=1}^n a_i \leq \bigvee_{i=1}^n b_i$ și $\bigwedge_{i=1}^n a_i \leq \bigwedge_{i=1}^n b_i$.

Demonstrație. Vom demonstra propoziția 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 enunțul propoziției se obține în mod similar utilizând $a_1 \wedge b_1 = a_1$ și $a_2 \wedge b_2 = a_2$. \square

Propoziția 1.4.3.6. (Proprietatea min-max)

Fie $M = (A, \leq)$ o latice și $a_{ij} \in A$, unde $1 \leq i \leq m$, $1 \leq j \leq n$ și $m, n \geq 1$. Atunci, are loc:

$$\bigvee_{j=1}^n \left(\bigwedge_{i=1}^m a_{ij} \right) \leq \bigwedge_{i=1}^m \left(\bigvee_{j=1}^n a_{ij} \right).$$

Demonstrație. Vom demonstra proprietatea, drept exemplu, pentru $m = 2$ și $n = 3$ (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

$$(a_{11} \wedge a_{21}) \vee (a_{12} \wedge a_{22}) \vee (a_{13} \wedge a_{23}) \leq a_{11} \vee a_{12} \vee a_{13}$$

și

$$(a_{11} \wedge a_{21}) \vee (a_{12} \wedge a_{22}) \vee (a_{13} \wedge a_{23}) \leq a_{21} \vee a_{22} \vee a_{23}$$

(conform Propoziției 1.4.3.3). Prima inegalitate este indușă de $a_{11} \wedge a_{21} \leq a_{11}$, $a_{12} \wedge a_{22} \leq a_{12}$ și $a_{13} \wedge a_{23} \leq a_{13}$ prin aplicarea Propoziției 1.4.3.5.

În mod similar se obține și cea de a doua inegalitate. \square

Dacă gândim elementele a_{ij} din Propoziția 1.4.3.6 ca fiind distribuite într-o matrice (cu notația uzuală), atunci proprietatea min-max ne spune că supremumul infimumurilor calculate pe coloane este mai mic sau cel mult egal cu infimumul supremelor calculate pe linii.

Corolarul 1.4.3.1. (Inegalități de distributivitate)

În orice latice $M = (A, \leq)$ au loc proprietățile

$$a \vee (b \wedge c) \leq (a \vee b) \wedge (a \vee c)$$

și

$$a \wedge (b \vee c) \geq (a \wedge b) \vee (a \wedge c),$$

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

Demonstrație. Este suficient de demonstrat doar una dintre aceste proprietăți deoarece cealaltă se obține prin dualizare. Ca urmare, vom demonstra prima proprietate, care de fapt decurge imediat din Propoziția 1.4.3.6 considerând $m = n = 2$, $a_{11} = a_{21} = a$, $a_{12} = b$ și $a_{22} = c$. \square

Într-o latice elementul $(a \wedge b) \vee (b \wedge c) \vee (c \wedge a)$ se numește *mediana* elementelor a , b și c . El satisfac următoarea proprietate:

Corolarul 1.4.3.2. (Proprietatea mediană)

În 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 $a, b, c \in A$.

Demonstrație. Aceasta urmează direct de la Propoziția 1.4.3.6 alegând $m = n = 3$, $a_{11} = a_{13} = a_{31} = a$, $a_{12} = a_{21} = a_{22} = b$ și $a_{23} = a_{32} = a_{33} = c$. \square

Corolarul 1.4.3.3. (Inegalități modulare)

În orice latice $M = (A, \leq)$ au loc proprietățile

$$a \leq c \Rightarrow a \vee (b \wedge c) \leq (a \vee b) \wedge c$$

și

$$a \geq c \Rightarrow a \wedge (b \vee c) \geq (a \wedge b) \vee c,$$

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

Demonstrație. Vom demonstra doar prima proprietate. Presupunând $a \leq c$ obținem $a \vee c = c$ și

$$\begin{aligned} a \vee (b \wedge c) &\leq (a \vee b) \wedge (a \vee c) \\ &= (a \vee b) \wedge c \end{aligned}$$

(inegalitatea urmează de la Corolarul 1.4.3.1). \square

1.4.3.3. Latici distributive și modulare

Corolarul 1.4.3.1 ne arată că orice latice satisfac anumite legi de distributivitate. Ele sunt satisfăcute prin “inegalitate”. Există latici în care aceste legi sunt satisfăcute prin egalitate, cum ar fi laticea submulțimilor unei mulțimi, și există latici în care aceste proprietăți nu pot fi satisfăcute prin egalitate. Un exemplu în acest sens îl constituie laticile M_3 și N_5 din Figura 1.18²⁸. Argumentăm aceasta doar pentru laticea M_3 . Au loc relațiile:

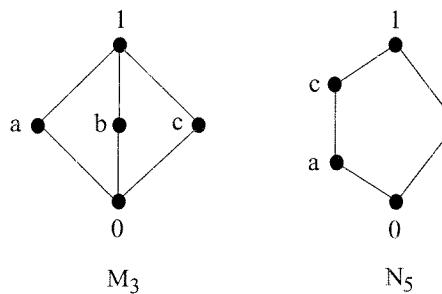
$$a \wedge (b \vee c) = a \wedge 1 = a > 0 = 0 \vee 0 = (a \wedge b) \vee (a \wedge c).$$

Laticile pentru care operațiile \vee și \wedge sunt complet distributive sunt numite *latici distributive*. Ele au fost introduse de Ernst Schröder [186].

Definiția 1.4.3.3. O latice $M = (A, \leq)$ este numită *distributivă* dacă satisfac proprietățile

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

²⁸ Aceste latici mai pot fi întâlnite și sub denumirea de laticile D_5 și M_5 sau laticile *diamant* și, respectiv, *pentagon*.

Figura 1.18: Laticile M_3 și N_5

și

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c),$$

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

Este ușor de văzut că doar una dintre proprietățile din Definiția 1.4.3.3 este nece-
sară, cealaltă putându-se obține cu ușurință. De exemplu, dacă o considerăm pe
prima, atunci cea de a doua se obține astfel:

$$\begin{aligned} (a \wedge b) \vee (a \wedge c) &= ((a \wedge b) \vee a) \wedge ((a \wedge b) \vee c) \\ &= a \wedge ((a \vee c) \wedge (b \vee c)) \\ &= (a \wedge (a \vee c)) \wedge (b \vee c) \\ &= a \wedge (b \vee c) \end{aligned}$$

(s-a aplicat prima relație, absorbția și distributivitatea).

Exemplul 1.4.3.4.

- (1) Laticea submulților unei mulțimi este distributivă.
- (2) Orice lanț este latice distributivă.

Teorema 1.4.3.3. O latice $M = (A, \leq)$ este distributivă dacă și numai dacă are loc

$$(a \wedge b) \vee (b \wedge c) \vee (c \wedge a) = (a \vee b) \wedge (b \vee c) \wedge (c \vee a),$$

pentru orice $a, b, c \in A$.**Demonstrație.** Să presupunem că M este latice distributivă. Atunci:

$$\begin{aligned} (a \wedge b) \vee (b \wedge c) \vee (c \wedge a) &= [(a \wedge b) \vee (b \wedge c)] \vee (c \wedge a) \\ &= [(a \vee c) \wedge b] \vee (a \wedge c) \\ &= [(a \vee c) \vee (a \wedge c)] \wedge [b \vee (a \wedge c)] \\ &= (a \vee c) \wedge [(b \vee a) \wedge (b \vee c)] \\ &= (a \vee b) \wedge (b \vee c) \wedge (c \vee a) \end{aligned}$$

(s-au folosit egalitățile din Definiția 1.4.3.3)

Reciproc, presupunem că are loc relația din teoremă. Vom arăta că are loc

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c),$$

pentru orice $a, b, c \in A$.Fie $d = (a \vee b) \wedge (a \vee c)$. Aplicăm egalitatea din teoremă elementelor $a, b \vee c$ și d . Obținem

$$(*) \quad (a \vee (b \vee c)) \wedge ((b \vee c) \vee d) \wedge (d \vee a) = [a \wedge (b \vee c)] \vee [(b \vee c) \wedge d] \vee (d \wedge a).$$

Arătăm că membrul stâng al egalității $(*)$ este d . Are loc

$$a \wedge d = a \wedge (a \vee b) \wedge (a \vee c) = a \wedge (a \vee c) = a$$

(în baza absorbției), ceea ce arată că $a \leq d$, și deci $d \vee a = d$. Acum, $d \leq a \vee b \vee c$ deoarece $a \vee b \leq a \vee b \vee c$ și $a \vee c \leq a \vee b \vee c$. În plus, relația $d \leq d \vee b \vee c$ este trivial satisfăcută. Deci membrul stâng al egalității $(*)$ este d .Arătăm că membrul drept al egalității $(*)$ este $a \vee (b \wedge c)$. Are loc

$$(b \vee c) \wedge d = (b \vee c) \wedge (a \vee b) \wedge (a \vee c) = (b \wedge c) \vee (a \wedge b) \vee (a \wedge c)$$

(conform ipotezei). Cum $d \wedge a = a$, utilizând repetat absorbția, membrul drept al egalității $(*)$ devine $a \vee (b \wedge c)$, ceea ce încheie demonstrația. \square

Următoarea propoziție urmează cu ușurință de la definiții.

Propoziția 1.4.3.7.

- (1) Duala unei latici distributive este latice distributivă.
- (2) Orice sublatice a unei latici distributive este latice distributivă.
- (3) Imaginea homomorfă a unei latici distributive este o latice distributivă.
- (4) Produs cartezian de latici nevide este latice distributivă dacă și numai dacă fiecare latice a produsului este distributivă.

Laticile M_3 și N_5 joacă un rol fundamental în caracterizarea distributivității.**Teorema 1.4.3.4.** O latice $M = (A, \leq)$ este distributivă dacă și numai dacă nu conține nici o sublatice izomorfă cu una din laticile M_3 sau N_5 .**Demonstrație.** Dacă M este distributivă, atunci ea nu poate conține nici o sublatice izomorfă cu M_3 sau N_5 , deoarece aceste latici nu sunt distributive, și altfel s-ar contrazice Propoziția 1.4.3.7.

Reciproc, presupunem că M nu conține nici o sublatice izomorfă cu una din laticile M_3 sau N_5 , dar cu toate acestea M nu este distributivă. Atunci, conform Corolarului 1.4.3.2 și Teoremei 1.4.3.3, există $a, b, c \in A$ astfel încât

$$(a \wedge b) \vee (b \wedge c) \vee (c \wedge a) < (a \vee b) \wedge (b \vee c) \wedge (c \vee a).$$

Fie s membrul stâng și r membrul drept al acestei inegalități. Deci, $s < r$.

În orice latice are loc proprietatea

$$(*) \quad (\forall u, v, w)(u \leq w \Rightarrow u \vee (v \wedge w) \leq (u \vee v) \wedge w)$$

(a se vedea Corolarul 1.4.3.3). Vom arăta că, deoarece M nu conține nici o sublatice izomorfă cu N_5 , cea de a doua inegalitate din $(*)$ trebuie să fie satisfăcută prin egalitate ori de câte ori $u < w$.

Dacă $u = v$ sau $v = w$ sau atât u cât și v sunt mai mici strict decât w (u și v pot fi comparabile între ele sau nu), atunci se verifică imediat că cea de a doua inegalitate din $(*)$ este satisfăcută prin egalitate.

Presupunem că $u < w$ și v este incomparabil atât cu u , cât și cu v , dar cea de a doua inegalitate din $(*)$ este satisfăcută prin inegalitate strictă. Fie $\alpha = u \vee (v \wedge w)$ și $\beta = (u \vee v) \wedge w$. Deci, $\alpha < \beta$. Vom arăta că $M' = \{v \wedge w, v, u \vee v, \alpha, \beta\}$ este sublatice a laticei M ce este izomorfă cu N_5 , ceea ce va constitui o contradicție.

În primul rând, $v \wedge w < v$, deoarece, dacă ar avea loc $v \wedge w = v$, am obține $v \leq w$, și apoi $u \vee v \leq w$, de unde ar urma cu ușurință $\alpha = u \vee v = \beta$, ceea ce ar fi o contradicție. În mod similar se obține $v < u \vee v$. Elementele α și $v \wedge w$ trebuie să fie distincte deoarece, altfel, am obține $u \leq v \wedge w \leq v$, ceea ce ar conduce la $\alpha = \beta$. În mod similar obținem $\beta < u \vee v$, $\alpha \vee v = u \vee v$ și $\beta \wedge v = v \wedge w$ (a se vedea diagrama din Figura 1.19(a)). Ca urmare, M' este sublatice a laticei M , izomorfă cu N_5 , ceea

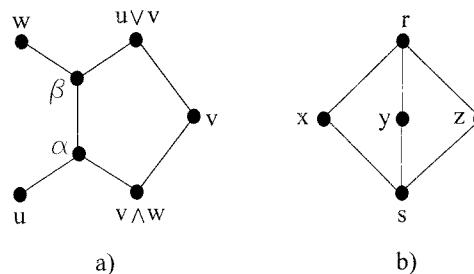


Figura 1.19: a) Sublaticea $\{v \wedge w, v, u \vee v, \alpha, \beta\}$; b) Sublaticea $\{s, r, x, y, z\}$

ce constituie o contradicție.

Finalizăm acum demonstrația teoremei considerând următoarele cazuri:

Cazul 1. Cel puțin două dintre elementele a, b și c coincid. Fie, de exemplu, $a = b$. Un simplu calcul, utilizând absorbția, ne arată că $s = a = r$, ceea ce conduce la contradicție. Deci acest caz nu este posibil;

Cazul 2. a, b și c sunt distințe două câte două, două dintre ele sunt comparabile prin relația \leq , fie de exemplu $a < c$, iar al treilea este incomparabil cu celelalte două. Este ușor atunci de văzut că mulțimea $\{s, r, b, b \wedge c, a \vee b\}$ formează o sublatice ce este izomorfă cu N_5 (a se vedea discuția de mai sus), ceea ce conduce la contradicție cu ipoteza;

Cazul 3. a, b și c sunt distințe și comparabile două câte două. Fie, de exemplu, $a < b < c$. Un simplu calcul ne arată că $s = b = r$, ceea ce conduce la contradicție. Deci, acest caz nu este posibil;

Cazul 4. a, b și c sunt distințe și incomparabile două câte două. În acest caz considerăm $x = s \vee (a \wedge r)$, $y = s \vee (b \wedge r)$ și $z = s \vee (c \wedge r)$ și vom arăta că $\{x, y, z, r, s\}$ formează sublatice izomorfă cu M_3 , conducând astfel la contradicție cu ipoteza.

Deoarece:

$$\begin{aligned} a \wedge r &= a \wedge (a \vee b) \wedge (c \vee a) \wedge (b \vee c) \\ &= a \wedge (c \vee a) \wedge (b \vee c) \\ &= a \wedge (b \vee c). \end{aligned}$$

deducem că $x = s \vee [(a \wedge (b \vee c))]$. În mod similar obținem $y = s \vee [(b \wedge (c \vee a))]$ și $z = s \vee [(c \wedge (a \vee b))]$. Acum avem toate elementele necesare pentru a verifica dacă mulțimea $\{x, y, z, r, s\}$ formează sublatice. În adevăr, în baza relațiilor de mai sus că și a relației $(*)$ obținem:

$$\begin{aligned} x \vee y &= s \vee (a \wedge (b \vee c)) \vee (b \wedge (c \vee a)) \\ &= s \vee (((a \wedge (b \vee c)) \vee b) \wedge (c \vee a)) \\ &= s \vee ((b \wedge a) \wedge (b \vee c) \wedge (c \vee a)) \\ &= s \vee r \\ &= r \end{aligned}$$

(a două egalitate se bazează pe $a \wedge (b \vee c) \leq c \vee a$ și $(*)$, iar a doua pe $b \leq b \vee c$ și $(*)$). În mod similar deducem $x \vee z = y \vee z = r$ și $x \wedge y = y \wedge z = z \wedge x = s$.

Ca urmare, $\{x, y, z, r, s\}$ formează sublatice a cărei reprezentare grafică este cea din Figura 1.19(b). Această latice este izomorfă cu M_3 , conducând astfel la contradicție cu ipoteza.

În concluzie, presupunerea făcută este falsă, ceea ce ne arată că M este latice distributivă. \square

Exemplul 1.4.3.5. Laticile M și M' din Figura 1.20 nu sunt distributive deoarece conțin sublatici izomorfe cu N_5 . Laticea M'' din aceeași figură nu este distributivă deoarece conține sublatici izomorfe cu M_3 .

Dorim să punctăm faptul că structurile M_3 și N_5 trebuie să fie sublatici (via un izomorfism) în laticea despre care vrem să arătăm că nu este distributivă. De exemplu, submulțimea $\{0, a, b, d, 1\}$ a laticei M are forma laticei N_5 , dar ea nu este sublatice a laticei M . Concluzia conform căreia M nu este distributivă o obținem în baza faptului că submulțimea $\{0, a, d, e, 1\}$ este sublatice a laticei M , ce este izomorfă cu N_5 .

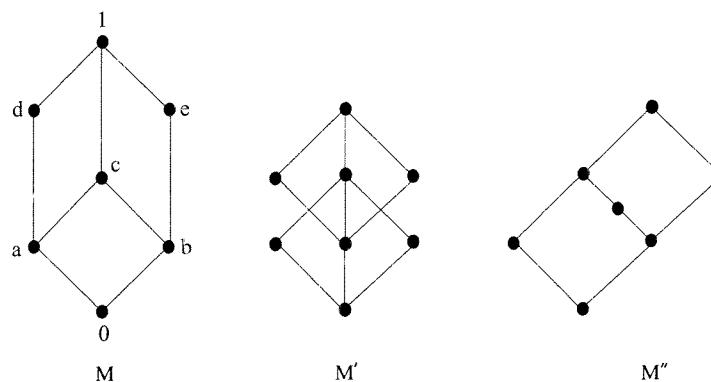


Figura 1.20: Exemplu de aplicare a Teoremei 1.4.3.4

Corolarul 1.4.3.4. O latice $M = (A, \leq)$ este distributivă dacă și numai dacă relațiile $a \vee x = a \vee y$ și $a \wedge x = a \wedge y$ conduc la $x = y$, pentru orice $a, x, y \in A$.

Demonstrație. Dacă presupunem că M este distributivă, atunci:

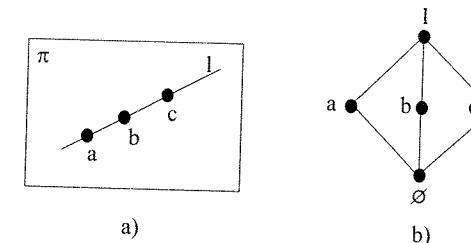
$$\begin{aligned} x &= x \vee (a \wedge x) \\ &= x \vee (a \wedge y) \\ &= (x \vee a) \wedge (x \vee y) \\ &= (a \vee y) \wedge (x \vee y) \\ &= (a \wedge x) \vee y \\ &= (a \wedge y) \vee y \\ &= y. \end{aligned}$$

Reciproc, dacă presupunem că au loc relațiile din corolar, dar laticea nu este distributivă, atunci ea va conține cel puțin o sublatice izomorfă cu M_3 sau N_5 . Însă în nici una din aceste latice nu au loc relațiile din corolar. De exemplu, în laticea M_3 are loc $b \vee a = b \vee c$ și $b \wedge a = b \wedge c$, dar $a \neq c$. Ca urmare, am ajuns la o contradicție ce ne arată că presupunerea făcută este falsă. \square

O alta clasă importantă de latice este cea a laticilor modulare. Această clasă de latice, introdusă de Dedekind în 1897 [43], se obține prin transformarea uneia din inegalitățile din Corolarul 1.4.3.3 în egalitate (cealaltă inegalitate va fi satisfăcută prin egalitate, ceea ce se obține prin dualizare). Necesitatea introducerii acestor tipuri de latice apare datorită faptului că multe latice întâlnite în practică nu sunt distributive. Cel mai simplu exemplu, și poate unul din cele mai importante, este cel al geometriei proiective. Fie π un plan. Considerăm mulțimea A a tuturor punctelor planului π și a dreptelor acestuia, la care adăugăm mulțimea vidă (ca cel mai mic element) și planul π (ca cel mai mare element). Relația \leq dată prin

- $a \leq a$, pentru orice $a \in A$;
 - $\emptyset \leq a \leq \pi$, pentru orice $a \in A$;
 - $a \leq l$, pentru orice punct a și dreaptă l astfel încât a este punct pe dreapta l ,
- este o ordine parțială pe A . Mai mult, (A, \leq) formează o sublatice. În raport cu această ordine parțială observăm că:
- supremumul a două puncte distincte este unică dreaptă ce le conține;
 - infimumul a două drepte concurente este punctul lor de intersecție.

Să considerăm acum 3 puncte distincte a, b și c pe o aceeași dreaptă l , împreună cu mulțimea vidă (a se vedea Figura 1.21(a)). Această submulțime formează o sublatice a laticei (A, \leq) , ce are reprezentarea grafică din Figura 1.21(b). Ca urmare, laticea

Figura 1.21: Sublaticea $\{\emptyset, a, b, c, l\}$

(A, \leq) nu este distributivă.

Pe de altă parte, dacă a este un punct pe o dreaptă l , ceea ce este echivalent cu $a \leq l$, iar b nu este pe dreapta l , atunci are loc

$$a \vee (b \wedge l) = (a \vee b) \wedge (a \vee l) = (a \vee b) \wedge l$$

(a se vedea Figura 1.22). Mai mult, pentru orice trei elemente $x, y, z \in A$, dacă $x \leq z$,

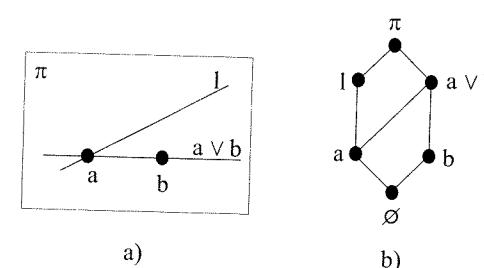


Figura 1.22: Exemplificare a proprietății de modularitate a laticilor

atunci

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) = (x \vee y) \wedge z.$$

Această proprietate specială a fost descoperită de Dedekind care a formalizat-o prin introducerea laticilor modulare.

Definiția 1.4.3.4. O latice $M = (A, \leq)$ este numită *modulară* dacă are loc

$$a \leq c \Rightarrow a \vee (b \wedge c) = (a \vee b) \wedge c,$$

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

Este clar că orice latice distributivă este și modulară. Laticea M_3 este modulară, dar laticea N_5 nu este. Ca și în cazul laticilor distributivi, putem demonstra următoarea propoziție.

Propoziția 1.4.3.8.

- (1) Duala unei latici modulare este latice modulară.
- (2) Orice sublatice a unei latici modulare este latice modulară.
- (3) Imaginea homomorfă a unei latici modulare este o latice modulară.
- (4) Produs cartezian de latici nevide este latice modulară dacă și numai dacă fiecare latice a produsului este modulară.

Teorema 1.4.3.4 poate fi formulată pentru latici modulare, astfel:

Teorema 1.4.3.5. O latice $M = (A, \leq)$ este modulară dacă și numai dacă nu conține nici o sublatice izomorfă cu laticea N_5 .

Demonstrație. A se vedea demonstrația Teoremei 1.4.3.4 care include, ca rezultat intermediar, și demonstrația acestei teoreme. \square

Corolarul 1.4.3.4 poate fi și el reformulat pentru cazul laticilor modulare.

Corolarul 1.4.3.5. O latice $M = (A, \leq)$ este modulară dacă și numai dacă relațiile $a \vee x = a \vee y$ și $a \wedge x = a \wedge y$ conduc la $x = y$, pentru orice $a, x, y \in A$ cu $x \leq y$.

Demonstrație. Similară demonstrației Corolarului 1.4.3.4. \square

1.4.4. Algebre universale dintr-un punct de vedere elementar

Scopul acestei secțiuni 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 mulțime și un număr arbitrar de operații definite pe acea mulțime, după care vom discuta concepțele de subalgebră, homomorfism, congruență etc. Toate acestea vor putea fi apoi translate cu mare ușurință la semigrupuri, monoizi, grupuri etc. În Capitolul 8 vom relua conceptul de algebră universală într-un cadru mult mai larg.

1.4.4.1. Algebre

Definiția 1.4.4.1. Se numește *algebră universală* orice cuplu (A, F) format dintr-o mulțime A și o mulțime nevidă F de operații pe A , fiecare operație având asociată o anumită aritate (ce poate fi și 0).

Terminologia de “algebră universală” va fi simplificată frecvent la cea de “algebră”.

Fie (A, F) o algebră. Mulțimea A este numită *mulțimea suport* sau *suportul* algebrei. Dacă suportul este finit, atunci vom spune că algebra este *finită*, iar dacă are cel mult un element, atunci vom spune că algebra este *trivială*. Atunci când F este finită, de exemplu $F = \{f_1, \dots, f_k\}$, vom mai nota algebra prin (A, f_1, \dots, f_k) . Reamintim că aritatea unei operații $f \in F$ se mai notează prin $ar(f)$ (Secțiunea 1.2.3). Operațiile 0-are vor fi numite *constantele algebrei*. Este clar că dacă algebra are cel puțin o constantă, atunci suportul este nevid.

Exemplul 1.4.4.1. 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 concepții; ele vor fi studiate în detaliu în capitolele următoare.

- (1) Un *semigrup* este o algebră (A, \cdot) cu o singură operație binară asociativă ..
- (2) Un *monoid* este o algebră (A, \cdot, e) , unde \cdot este o operație binară asociativă pe A , iar e este o operație nulară pe A ce satisfac

$$e \cdot a = a \cdot e = a,$$

pentru orice $a \in A$. Constanta e se mai notează și prin 1_M și se numește *unitatea monoidului*. Este ușor de văzut că ea este unicul element ce satisfac proprietatea de mai sus. În adevăr, dacă am presupune că mai există un element e' ce satisfac $e' \cdot a = a \cdot e' = a$, 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 întâlniți și sub denumirea de *semigrupuri cu unitate*.

- (3) Un *grup* este o algebră $(A, \cdot, ', e)$, unde \cdot este o operație binară asociativă pe A , e este o operație nulară pe A ce satisfac proprietatea de la (2), iar $'$ este o

operație unară pe A pentru care are loc:

$$a \cdot a' = a' \cdot a = e,$$

pentru orice $a \in A$. e se numește *unitatea grupului*, este unică, și se mai notează prin 1_G . Elementul a' se mai numește *inversul lui a*; el este unic, ceea ce se poate vedea cu ușurință. 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 operația binară este comutativă se numește *semigrup (monoid, grup) comutativ sau abelian*²⁹.
- (5) Un *inel* este o algebră $(A, +, -, 0, \cdot)$, unde $+$ și \cdot sunt operații binare, $-$ este o operație unară, iar 0 este operație nulară pe A ce satisfac proprietățile:

- (i) $(A, +, -, 0)$ este grup comutativ;
- (ii) (A, \cdot) este semigrup;
- (iii) $a \cdot (b + c) = a \cdot b + a \cdot c$ și $(b + c) \cdot a = b \cdot a + c \cdot a$, pentru orice $a, b, c \in A$.

Uzual, operațiile $+$ și \cdot sunt numite *adunarea* și, respectiv, *înmulțirea* (dar nu sunt, în mod necesar, operațiile de adunare și înmulțire pe mulțimi uzuale cum e cea a numerelor întregi). Prima parte a proprietății de la (iii) este numită proprietatea de *distributivitate la stânga a înmulțirii față de adunare*, iar cea de a doua parte, proprietatea de *distributivitate la dreapta a înmulțirii față de adunare*.

Atunci când semigrupul (A, \cdot) este comutativ, inelul este numit *inel comutativ*.

- (6) Un *inel cu unitate* este o algebră $(A, +, -, 0, \cdot, 1)$ definită ca la (5), dar cu diferența că $(A, \cdot, 1)$ este monoid. Dacă acest monoid este comutativ, atunci inelul este numit *inel comutativ cu unitate*.

Atunci când monoizii, grupurile și inelele sunt formate doar din unitate, ele sunt triviale (în cazul inelelor cu unitate, proprietatea de a fi trivial forcează egalitatea între cele două unități).

Ca o remarcă generală, atunci când operația binară va fi notată multiplicativ (prin \cdot , \circ sau $*$), operația unară corespunzătoare va fi notată prin $'$, iar cea nulară prin e sau uneori 1 (eventual indexate). Atunci când operația binară va fi notată aditiv (prin $+$), operația unară corespunzătoare va fi notată prin $-$, iar cea nulară prin 0 . În acest caz, $a - b$ va reprezenta $a + (-b)$.

Definiția 1.4.4.2. Spunem că două algebrelor (A, F) și (A', F') au *același tip* dacă există o bijecție h între F și F' ce păstrează aritatea operațiilor, adică aritatea operației $f \in F$ este aceeași cu aritatea operației $h(f) \in F'$, pentru orice $f \in F$.

²⁹Denumirea de grup “abelian” provine de la numele matematicianului norvegian Niels Abel.

Atunci când vom spune că (A, F) și (A', F') au același tip vom presupune implicit că $f' \in F'$ este corespondentă operației $f \in F$ printr-o bijecție h ce păstrează tipul operațiilor (adică, f' denotă $h(f)$).

În cazul algebrelor cu un număr finit de operații, pe lângă convenția de mai sus, vom adopta și următoarea convenție. Dacă (A, f_1, \dots, f_k) și (A', g_1, \dots, g_k) sunt de același tip, atunci vom presupune că f_i și g_i au aceeași aritate, pentru orice $1 \leq i \leq k$.

Semigrupurile (monoizii, grupurile, inelele) sunt algebrelor de același tip. O mică discuție este necesară în cazul inelelor care au două operații binare. Asocierea operațiilor trebuie înțeleasă în conformitate cu ordinea lor în 5-uplul ce definește inelul (așa cum, de altfel, s-a specificat în convenția adoptată mai sus).

1.4.4.2. Subalgebrelor. Ordin

Conceptul de subalgebră generalizează concepțile clasice de subsemigrup, submonoid, subgrup etc.

Definiția 1.4.4.3. Fie (A, F) și (A', F') două algebrelor de același tip. Spunem că (A', F') este *subalgebră* a algebrei (A, F) dacă $A' \subseteq A$ și $f' = f|_{A'}$, pentru orice $f \in F$.

Definiția 1.4.4.4. Fie (A, F) o algebră și $X \subseteq A$. Spunem că X este *închisă* în (A, F) dacă are loc:

$$(\forall f \in F)(\forall a_1, \dots, a_{ar(f)} \in A)(a_1, \dots, a_{ar(f)} \in X \Rightarrow f(a_1, \dots, a_{ar(f)}) \in X)$$

(în cazul $ar(f) = 0$, cerința se reduce la $f \in X$).

Observația 1.4.4.1.

- (1) Orice subalgebră sau submulțime închisă conține toate constantele algebrei gazdă.
- (2) Dacă (A', F') este subalgebră a algebrei (A, F) , atunci A' este închisă în (A, F) . Reciproc, dacă $A' \subseteq A$ este o submulțime închisă în (A, F) , atunci ea poate fi structurată ca o subalgebră a algebrei (A, F) , considerând pentru orice $f \in F$ operația $f' : (A')^{ar(f)} \rightarrow A'$ dată prin $f' = f|_{A'}$. Mulțimea tuturor acestor operațiilor, notată F' , împreună cu A' formează o subalgebră a algebrei (A, F) .

Este important de menționat că există o diferență între operația f' definită ca mai sus și $f|_{A'}$. Prima are codomeniu A' , pe când a doua are codomeniu A . Evident, această diferență nu este semnificativă, motiv pentru care vom identifica adesea subalgebrele unei algebrelor cu submulțimile închise în acea algebră împreună cu restricțiile operațiilor algebrei gazdă la acea submulțime.

Următoarea propoziție urmează imediat de la definiții.

Propoziția 1.4.4.1. Intersecția oricărei familii nevide de submulțimi închise (subalgebre) ale unei algebrelor (A, F) este submulțime închisă (subalgebră) a algebrelor (A, F) .

Fie (A, F) o algebră și $X \subseteq A$. Conform teoriei închiderii, închiderea acestei mulțimi 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 $m \geq 0$ (reamintim că prin $f(B_m)$ se înțelege mulțimea $\{f(a_1, \dots, a_{ar(f)}) | a_1, \dots, a_{ar(f)} \in B_m\}$).

Este clar că $\langle X \rangle_{(A, F)}$ este închisă în (A, F) și, ca urmare, ea definește o subalgebră a algebrelor (A, F) . Aceasta se numește *subalgebra generată de X* și 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 intersecția tuturor subalgebrelor ce includ X .

Apelând iarăși la teoria închiderii, un element a este în $\langle X \rangle_{(A, F)}$ dacă și numai dacă există o secvență

$$x_1, \dots, x_n = a$$

astfel încât, pentru orice i , are loc:

- $x_i \in X$, sau
- $(\exists f \in F)(\exists i_1, \dots, i_{ar(f)} < i)(x_i = f(x_{i_1}, \dots, x_{i_{ar(f)}}))$.

Definiția 1.4.4.5.

- (1) Spunem că o algebră (A, F) este *generată de $X \subseteq A$* dacă $A = \langle X \rangle_{(A, F)}$.
- (2) Spunem că (A, F) este *finit generată* dacă există o submulțime finită $X \subseteq A$ astfel încât (A, F) este generată de X .

Dacă X generează algebra (A, F) , atunci X se numește *mulțime de generatori* ai algebrelor (A, F) , iar elementele ei, *generatori* ai algebrelor.

O algebră generată doar de un singur element al ei se numește *algebră ciclică*. Particularizând, obținem concepțele de *semigrup ciclic*, *monoid ciclic* și *grup ciclic*.

Exemplul 1.4.4.2.

- (1) (M_2, \circ, e_2) este submonoid al monoidului (M_1, \cdot, e_1) dacă au loc proprietățile $M_2 \subseteq M_1$, $\circ = \cdot|_{M_2}$ și $e_2 = e_1$.

- (2) (G_2, \circ'', e_2) este subgrup al grupului (G_1, \cdot', e_1) dacă au loc proprietățile $G_2 \subseteq G_1$, $\circ = \cdot|_{G_2}$, $'' = '|_{G_2}$ și $e_2 = e_1$.

Este ușor de văzut că putem renunța la cerința " $e_2 = e_1$ " deoarece aceasta se obține combinând primele 3 cerințe:

$$e_2 = a \circ a'' = a \cdot a' = e_1,$$

pentru orice $a \in G_2$.

Definiția 1.4.4.6.

Fie (A, F) o algebră.

- (1) Spunem că (A, F) este de ordin ∞ sau că are ordinul ∞ dacă A este mulțime infinită. Altfel, spunem că algebră 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 .

Vom încheia subsecțiunea printr-un rezultat important ce poate fi utilizat în demonstrarea de proprietăți în algebrelle.

Teorema 1.4.4.1.

(Principiul inducției structurale pentru algebrelle)

Fie (A, F) o algebră generată de o parte X a sa. Dacă P este o proprietate referitoare la elementele algebrelor (A, F) astfel încât

- (1) $P(x)$, pentru orice $x \in X$;
- (2) $(P(a_1) \wedge \dots \wedge P(a_{ar(f)})) \Rightarrow P(f(a_1, \dots, a_{ar(f)}))$, pentru orice $f \in F$ și $a_1, \dots, a_{ar(f)} \in A$,

atunci $P(a)$, pentru orice $a \in A$.

Demonstrație. Se obține direct de la principiul inducției structurale pentru mulțimi inductive definite. \square

Exemplul 1.4.4.3.

- (1) În cazul semigrupurilor, principiul inducției structurale capătă următoarea formă. Fie (S, \cdot) un semigrup generat de $X \subseteq S$ și 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) În cazul grupurilor, principiul inducției structurale poate fi pus în următoarea formă. Fie (G, \cdot', e) un grup generat de $X \subseteq G$ și P o proprietate referitoare la elementele lui. Dacă

- (1) $P(x)$, pentru orice $x \in X$, și $P(e)$;
- (2) $(\forall a, b \in S)(P(a) \wedge P(b) \Rightarrow P(a \cdot b) \wedge P(a'))$,

atunci $P(a)$, pentru orice $a \in G$.

Diferența față de forma din teorema constă în aceea că se verifică $P(e)$ la pasul (1) și nu la pasul (2).

1.4.4.3. Homomorfisme și congruențe

Concepțele de homomorfism și congruență joacă un rol important în studiul structurilor algebrice.

Definiția 1.4.4.7. Fie (A, F) și (A', F') două algebrelă de același tip. Un *homomorfism* de la (A, F) la (A', F') este o aplicație $h : A \rightarrow A'$ ce satisface

$$h(f(a_1, \dots, a_{ar(f)})) = f'(h(a_1), \dots, h(a_{ar(f)})),$$

pentru orice $f \in F$ și $a_1, \dots, a_{ar(f)} \in A$ (atunci când f este o constantă, proprietatea de homomorfism se reduce $h(f) = f'$).

Exemplul 1.4.4.4.

- (1) Un homomorfism 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 homomorfism 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 homomorfism 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 $a, b \in G_1$.

Proprietățile grupului fac ca a doua cerință să nu fie necesară. În adevăr, relația

$$h(a \cdot a') = h(1_{G_1}) = h(a' \cdot a)$$

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 cerință 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 obținem $h(1_{G_1}) = 1_{G_2}$ dacă aplicăm $(h(1_{G_1}))''$.

Uzual, homomorfismele injective sunt numite *monomorfisme*, homomorfismele surjective sunt numite *epimorfisme*, iar homomorfismele bijective sunt numite *izomorfisme*. Un homomorfism de la o algebră (A, F) la ea însăși este numit *endomorfism*. Multimea tuturor endomorfismelor algebrei (A, F) se notează prin $\text{End}(A, F)$. Endomorfismele care sunt și izomorfisme se mai numesc *automorfisme*. Multimea tuturor automorfismelor algebrei (A, F) se notează prin $\text{Aut}(A, F)$.

Următoarele propoziții urmează imediat de la definiții.

Propoziția 1.4.4.2.

- (1) Componere 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) $\text{End}(A, F)$, cu compunerea funcțiilor, formează monoid.
- (4) $\text{Aut}(A, F)$, cu compunerea funcțiilor, formează grup.

Grupul $\text{Aut}(A, F)$ se numește *grupul automorfismelor* algebrei (A, F) .

Propoziția 1.4.4.3. Fie (A, F) și (A', F') algebrelă de același tip și $h : A \rightarrow A'$ 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) .

În baza acestei propoziții este clar că imaginea unei subalgebre printr-un homomorfism este subalgebră. Similar, imaginea inversă a unei subalgebre printr-un homomorfism este subalgebră.

Propoziția 1.4.4.4. Fie (A, F) și (A', F') algebrelă de același tip și $h_1, h_2 : A \rightarrow A'$ două homomorfisme. Dacă (A, F) este generată de X și $h_1(x) = h_2(x)$, pentru orice $x \in X$, atunci $h_1 = h_2$.

Demonstrație. Vom demonstra propoziția prin inducție structurală. Conform ipotezei, ceea ce ne rămâne de arătat este că dacă $f \in F$ și $h_1(a_i) = h_2(a_i)$, pentru orice $a_i \in A$, $1 \leq i \leq ar(f)$, atunci

$$h_1(f(a_1, \dots, a_{ar(f)})) = h_2(f(a_1, \dots, a_{ar(f)})).$$

Aceasta urmează imediat de la ipoteza inductivă și proprietatea de homomorfism a funcțiilor h_1 și h_2 . \square

Deci, două homomorfisme ce coincid pe o mulțime de generatori ai unei algebrelor vor coincide pe întreaga algebră.

Definiția 1.4.4.8. Fie (A, F) o algebră. O congruență în (A, F) este o relație de echivalență ρ pe A ce satisface

$$f(a_1, \dots, a_{ar(f)}) \rho f(a'_1, \dots, a'_{ar(f)}),$$

pentru orice $f \in F$ de aritate $ar(f) > 0$ și orice $a_1, a'_1, \dots, a_{ar(f)}, a'_{ar(f)} \in A$ pentru care are loc $a_i \rho a'_i$, pentru orice $1 \leq i \leq ar(f)$.

Proprietatea din Definiția 1.4.4.8 se mai numește *proprietatea de compatibilitate a relației de echivalență cu operațiile algebrei*.

Vom nota prin $Con(A, F)$ mulțimea tuturor congruențelor algebrei (A, F) .

Exemplul 1.4.4.5. În cazul semigrupurilor, o congruență ρ pe un semigrup (S, \cdot) satisface

$$(\forall a, b, a', b' \in S)(a \rho a' \wedge b \rho b' \Rightarrow (a \cdot b) \rho (a' \cdot b')).$$

Se mai spune că ρ este o relație de echivalență *compatibilă la stânga și la dreapta* cu operatorul “.”. Aceasta pentru că relația de mai sus este echivalentă cu proprietatea

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

Fie (A, F) o algebră și $\rho \in Con(A, F)$. Mulțimea cât

$$A/\rho = \{[a]_\rho \mid a \in A\}$$

poate fi structurată ca o algebră de același tip cu (A, F) într-un mod natural. Pentru fiecare $f \in F$ definim o nouă operăție f_ρ prin

$$f_\rho([a_1]_\rho, \dots, [a_{ar(f)}]_\rho) = [f(a_1, \dots, a_{ar(f)})]_\rho,$$

pentru orice $a_1, \dots, a_{ar(f)} \in A$. Dacă $ar(f) = 0$, atunci $f_\rho = [f]_\rho$.

Acstea operații nu depind de reprezentanții de clasă datorită proprietății de compatibilitate cu operațiile pe care o are congruența ρ . Mai exact, pentru orice $b_i \in [a_i]_\rho$, $1 \leq i \leq ar(f)$, are loc

$$\begin{aligned} f_\rho([b_1]_\rho, \dots, [b_{ar(f)}]_\rho) &= [f(b_1, \dots, b_{ar(f)})]_\rho \\ &= [f(a_1, \dots, a_{ar(f)})]_\rho \\ &= f_\rho([a_1]_\rho, \dots, [a_{ar(f)}]_\rho). \end{aligned}$$

Algebra astfel obținută, notată prin $(A/\rho, F/\rho)$, se numește *algebra cât sau factor indusă de (A, F) și ρ* .

Următoarele propoziții urmează cu ușurință de la definiții.

Propoziția 1.4.4.5. Fie (A, F) o algebră și $\rho \in Con(A, F)$. Atunci, $f : A \rightarrow A/\rho$ dată prin $f(a) = [a]_\rho$, pentru orice $a \in A$, este epimorfism.

Propoziția 1.4.4.6. Dacă f este un homomorfism de la algebra (A, F) la algebra (A', F') , atunci $ker(f)$ este congruență în (A, F) .

Următoarea teoremă are aplicații multiple.

Teorema 1.4.4.2. (Teorema de homomorfism/Prima teoremă de izomorfism) Fie f un epimorfism de la algebra (A, F) la algebra (A', F') . Atunci algebra cât $(A/ker(f), F/ker(f))$ este izomorfă cu algebra (A', F') .

Demonstrație. Fie funcția 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 $a \in ker(f)$, atunci $f(a) = f(b)$, ceea ce arată că definiția funcție h nu depinde de reprezentanții de clasă aleși;
- *f este homomorfism.* Pentru orice $g \in F$ și $a_1, \dots, a_{ar(g)} \in A$ au loc relațiile:

$$\begin{aligned} h(g_{ker(f)}([a_1]_{ker(f)}, \dots, [a_{ar(g)}]_{ker(f)})) &= \\ &= h([g(a_1, \dots, a_{ar(g)})]_{ker(f)}) \\ &= f(g(a_1, \dots, a_{ar(g)})) \\ &= g'(f(a_1, \dots, f(a_{ar(g)}))) \\ &= g'(h([a_1]_{ker(f)}), \dots, h([a_{ar(g)}]_{ker(f)})) \end{aligned}$$

Care ne arată că h este homomorfism;

- *h este funcție injectivă.* Dacă $f(a) = f(b)$, atunci $a \in ker(f)$, ceea ce arată că h este injecție;
- *h este funcție 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 surjectie.

Deci h este izomorfism. \square

1.4.5. Algebrelle Booleene

Un exemplu foarte important de algebră este cel de *algebră Booleană* [15]. Aceste tipuri de algebrelle își au rădăcinile în studiile lui George Boole asupra operațiilor de

reuniune, intersecție și complementară din teoria mulțimilor și asupra operațiilor de conjuncție, disjuncție și negație din logică. Algebrele Booleene vin să extragă esența acestor operații și să ofere un cadru general de studiu al proprietăților lor.

Definiția 1.4.5.1. O algebră Booleană este o algebră $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$, unde \vee și \wedge sunt operații binare pe A , $'$ este o operație unară pe A , iar 0 și 1 sunt operații nulare pe A ce satisfac:

- (1) \vee și \wedge sunt asociative și comutative;
- (2) $x \vee 0 = x$ și $x \wedge 1 = x$, pentru orice $x \in A$;
- (3) \vee și \wedge sunt distributive una față de alta;
- (4) $x \vee x' = 1$ și $x \wedge x' = 0$, pentru orice $x \in A$. (complementariere)

Elementul x' din Definiția 1.4.5.1 este numit *complementul* lui x .

Algebrele Booleene fiind cazuri particulare de algebrelor, orice concept introdus pentru algebrelor se poate transla la algebrelor Booleene. Este ușor de observat că $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$ este o algebră Booleană dacă $(A, \vee, 0)$ și $(A, \wedge, 1)$ sunt monoizi comutativi, operațiile \vee și \wedge sunt distributive una față de celalaltă și are loc proprietatea de complementariere.

Exemplul 1.4.5.1.

- (1) Pentru orice mulțime A , $(\mathcal{P}(A), \cup, \cap, ', \emptyset, A)$ este algebră Booleană (' fiind operația de complementariere a mulțimilor în raport cu A), numită *algebra Booleană a mulțimii părților lui A*;
- (2) Fie $B = \{0, 1\}$, \vee și \wedge adunarea, respectiv, înmulțirea modulo 2, iar $'$ dată prin:

$$0' = 1 \text{ și } 1' = 0.$$

Atunci, $\mathbf{B} = (B, \vee, \wedge, ', 0, 1)$ este algebră Booleană.

O poate fi interpretat ca valoarea de adevăr "fals", 1 ca valoarea de adevăr "adevărat", \vee este disjuncția, \wedge este conjuncția, iar $'$ este negația. Am obținut astfel *algebra Booleană a mulțimii valorilor de adevăr B*;

- (3) Pe mulțimea B^n , unde B este ca la (2) iar $n \geq 1$, definim operațiile \vee , \wedge și $'$ pe componente, utilizând operațiile de la (2). De exemplu,

$$(x_1, \dots, x_n) \vee (y_1, \dots, y_n) = (x_1 \vee y_1, \dots, x_n \vee y_n).$$

Atunci $\mathbf{B}^n = (B^n, \vee, \wedge, ', \mathbf{0}, \mathbf{1})$, unde $\mathbf{0}$ și $\mathbf{1}$ sunt n -upluri formate numai din 0, respectiv, 1, este o algebră Booleană.

Teorema 1.4.5.1. Fie $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$ o algebră Booleană. Atunci, pentru orice $x, y \in A$, au loc următoarele proprietăți:

- (1) $x \vee x = x$ și $x \wedge x = x$; (idempotență)
- (2) $x \vee 1 = 1$ și $x \wedge 0 = 0$;
- (3) $(x \wedge y) \vee x = x$ și $(x \vee y) \wedge x = x$; (absorbție)
- (4) $(x \vee y)' = x' \wedge y'$ și $(x \wedge y)' = x' \vee y'$; (legile lui De Morgan)
- (5) $x \vee y = y$ dacă și numai dacă $x \wedge y = x$.

Demonstrație. Pentru (1), (2), (3) și (4) vom demonstra doar prima relație, cea de a doua obținându-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 $w \vee z = 1$ și $w \wedge z = 0$, atunci $z = w'$. În 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 relațiile $(x \vee y) \vee (x' \wedge y') = 1$ și $(x \vee y) \wedge (x' \wedge y') = 0$.

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 și cea de a doua relație, și astfel prima parte de la (4) este demonstrată.

- (5) Dacă $x \vee y = y$ atunci, utilizând absorbția, obținem

$$x = x \wedge (x \vee y) = x \wedge y.$$

Implicația în sens invers se obține prin dualizare.

□

Fie $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$ o algebră Booleană. Definim relația $\leq \subseteq A \times A$ prin

$$x \leq y \Leftrightarrow x \vee y = y,$$

pentru orice $x, y \in A$. Relațiile $<$, \geq și $>$ se definesc în mod ușor. Aceste relații vor fi numite *relațiile induse* de algebra \mathbf{A} .

Teorema 1.4.5.2. Fie $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$ o algebră Booleană și \leq relația indușă de \mathbf{A} . Atunci au loc următoarele proprietăți:

- (1) \leq este relație de ordine parțială 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 $x \in A$.

Demonstrație. (1) și (3) necesită verificări triviale, iar (2) urmează de la Teorema 1.4.5.1(2)(5). \square

Corolarul 1.4.5.1. Fie $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$ o algebră Booleană. Atunci (A, \leq) , unde \leq este ordinea parțială indușă de \mathbf{A} , este latice pentru care 0 este cel mai mic element și 1 este cel mai mare element.

Definiția 1.4.5.2. 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 și z ambele diferite de x și 0.

Propoziția 1.4.5.1. Fie $\mathbf{A} = (A, \vee, \wedge, ', 0, 1)$ o algebră Booleană. Un element x diferit de 0 este atom dacă și numai dacă nu există $y \in A$ astfel încât $0 < y < x$.

Demonstrație. Presupunem, prin contradicție, că ar exista un atom x și 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 contradicție.

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 și v ambele diferite de 0. Deoarece $u \leq u \vee v = x$, urmează $u < x$, ceea ce contrazice ipoteza. \square

Exemplul 1.4.5.2.

- (1) Pentru algebră Booleană $(\mathcal{P}(A), \cup, \cap, ', \emptyset, A)$ din Exemplul 1.4.5.1(1), în cazul în care A este finită, atomii sunt exact multimile de forma $\{a\}$ cu $a \in A$.

- (2) Algebră \mathbf{B} din Exemplul 1.4.5.1(2) are ca atom doar pe 1.
- (3) Atomii algebrei \mathbf{B}^n din Exemplul 1.4.5.1(3) sunt toate n -uplele ce conțin doar un 1 și în rest numai 0.

În exemplul anterior, pentru oricare dintre cele trei algebrelle considerate, observăm că orice element poate fi scris ca o \vee -combinăție de atomi. În plus, scrierea este unică, exceptând ordinea în care sunt combinați atomii prin \vee . Mai exact, considerând algebră $(\mathcal{P}(A), \cup, \cap, ', \emptyset, A)$, unde A este finită, orice submulțime nevidă $B \subseteq A$ poate fi scrisă ca \cup -combinăție de exact toți atomii $\{a\}$ cu $a \in B$. De fapt, această observație va fi ideea de demonstrație a următoarei teoreme importante.

Teorema 1.4.5.3. 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

$$x = a_1 \vee \dots \vee a_k,$$

unde a_1, \dots, a_k sunt atomi.

Demonstrație. Vom arăta întâi că orice element $x \in A$ diferit de 0 poate fi scris ca o \vee -combinăție de atomi.

Presupunem, prin contradicție, că există elemente $x \in A$ diferite de 0 ce nu pot fi scrise în această formă, și fie S mulțimea acestora. Este clar că S nu conține atomi, și atunci orice element x din S poate fi scris în forma $x = y \vee z$ cu $0 < y < x$ și $0 < z < x$. Considerând un astfel de x cu o astfel de scriere, constatăm că cel puțin unul din elementele y sau z este în S ; fie acesta y . Are loc $y < x$ și, repetând acest procedeu cu y , deducem că există o secvență

$$x = x_0 > y = x_1 > \dots$$

de elemente din S . Deoarece A este finită, rezultă că nu toate elementele din acest sir sunt diferite două câte două și, ca urmare, vor exista două numere naturale k și m cu $k < m$ astfel încât $x_k = x_m$; contradicție cu $x_k > x_m$. Ca urmare, orice element diferit de 0 din A poate fi scris ca o \vee -combinăție de atomi.

Să ne ocupăm acum de unicitatea scrierii. Pentru aceasta trebuie să remarcăm că este suficient să arătăm că orice element $x \in A$ poate fi scris ca o \vee -combinăție a tuturor atomilor a cu $a \leq x$. În adevăr, dacă

$$x = \vee \{a \in A \mid a \text{ este atom și } a \leq x\}$$

și x ar avea și o altă scriere, $x = b_1 \vee \dots \vee b_k$ cu b_1, \dots, b_k atomi, atunci $b_i \leq x$ și, ca urmare,

$$b_i \in \{a \in A \mid a \text{ este atom și } a \leq x\},$$

pentru orice i . Pe de altă parte, dacă a este atom și $a \leq x$, atunci

$$0 \neq a = a \wedge x = a \wedge (b_1 \vee \dots \vee b_k) = (a \wedge b_1) \vee \dots \vee (a \wedge b_k).$$

Cel puțin unul din $a \wedge b_i$ trebuie să fie diferit de 0 și, în consecință, $a \wedge b_i = a = b_i$, pentru cel puțin un i . Aceasta ne arată că a este unul din atomii b_i , și astfel

$$\{b_1, \dots, b_k\} = \{a \in A \mid a \text{ este atom și } a \leq x\}.$$

Deci ceea ce rămâne de arătat este că orice $x \in A$ diferit de 0 poate fi scris în forma

$$x = \vee \{a \in A \mid a \text{ este atom și } a \leq x\}.$$

Evident, elementul 1 poate fi scris în această formă, și fie $1 = a_1 \vee \dots \vee a_n$ o astfel de \vee -combinare a lui 1. Pentru orice $x \in A$ diferit de 0 are loc

$$x = x \wedge 1 = x \wedge (a_1 \vee \dots \vee a_n) = (x \wedge a_1) \vee \dots \vee (x \wedge a_n).$$

Deoarece $0 \leq x \wedge a_i \leq a_i$ și a_i este atom, Propoziția 1.4.5.1 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

$$x = \vee \{a \in A \mid a \text{ este atom și } a \leq x\},$$

încheind demonstrația teoremei. \square

Izomorfismul de algebrelor Booleene este izomorfism de algebrelor. Următoarea teoremă ne spune că orice algebră Booleană finită este complet determinată, până la un izomorfism, de numărul de atomi ai ei.

Teorema 1.4.5.4. Orice două algebrelor Booleene finite cu același număr de atomi sunt izomorfe.

Demonstrație. Fie $\mathbf{A}_1 = (A_1, \vee_1, \wedge_1, ', 0_1, 1_1)$ și $\mathbf{A}_2 = (A_2, \vee_2, \wedge_2, '', 0_2, 1_2)$ două algebrelor Booleene cu atomii a_1, \dots, a_n și, respectiv, b_1, \dots, b_n .

Considerăm funcția $f : A_1 \rightarrow A_2$ dată prin $f(0_1) = 0_2$, $f(1_1) = 1_2$ și $f(a_i) = b_i$, pentru orice i . Extindem f la un unic homomorfism de la \mathbf{A}_1 la \mathbf{A}_2 . Teorema 1.4.5.3 conduce atunci cu ușurință la faptul că acest homomorfism este funcție bijectivă. \square

Următoarele două rezultate urmează direct de la această teoremă.

Corolarul 1.4.5.2. Orice algebră Booleană finită cu n atomi este izomorfă cu algebra Booleană a mulțimii tuturor părților unei mulțimi cu n elemente.

Corolarul 1.4.5.3. Algebra Booleană \mathbf{B}^n din Exemplul 1.4.5.1(3) este izomorfă cu algebra părților mulțimii $\{1, \dots, n\}$.

Există o strânsă legătură între algebrelor Booleene și latici.

Definiția 1.4.5.3. O latice $M = (A, \vee, \wedge)$ este numită *complementată* dacă are un cel mai mic element 0 și se poate defini o operație unară $'$ ce satisfacă:

$$(1) \quad (a')' = a;$$

$$(2) \quad (a \vee b)' = a' \wedge b';$$

$$(3) \quad a \wedge a' = 0,$$

pentru orice $a, b \in A$.

Acum este ușor de văzut că orice algebră Booleană este latice distributivă complementată și reciproc.

1.5. Numere ordinale și cardinale

În această secțiune vom prezenta câteva elemente de bază de teoria numerelor ordinale și cardinale. Vom urma cu precădere [207], unde cititorul interesat poate găsi detalii.

1.5.1. Mulțimi bine ordonate

Mulțimile bine ordonate joacă un rol foarte important în teoria mulțimilor prin aceea că, pe de o parte, ele sunt intens utilizate în introducerea noțiunilor de număr ordinal, număr cardinal și număr natural și, pe de altă parte, pentru astfel de ordini avem la îndemână un instrument foarte puternic de demonstrație și anume metoda inducției.

Definiția 1.5.1.1. Spunem că o multime este *mulțime bine ordonată*, abreviat mbo, dacă orice submulțime nevidă a ei are cel mai mic element.

Structura (\emptyset, \emptyset) este o mbo.

Exemplul 1.5.1.1.

$$(1) \quad (\mathbb{N}, <) \text{ este mbo.}$$

$$(2) \quad (\mathbb{Z}, <) \text{ nu este mbo deoarece, de exemplu, } B = \{x \in \mathbb{Z} \mid x < 0\} \text{ este submulțime nevidă a lui } \mathbb{Z} \text{ ce nu are cel mai mic element. În mod similar, } (\mathbb{Q}, <) \text{ și } (\mathbb{R}, <) \text{ nu sunt mbo.}$$

Următoarea teoremă prezintă un prim rezultat fundamental asupra mulțimilor bine ordonate.

Teorema 1.5.1.1. Fie $M = (A, \leq)$ o mbo și $f : A \rightarrow A$ o funcție strict monotonă. Atunci, pentru orice $x \in A$, are loc $x \leq f(x)$.

Demonstrație. Presupunem, prin contradicție, că există $x \in A$ cu $x > f(x)$. Fie $B = \{x \in A | x > f(x)\}$. Această mulțime este nevidă și, ca urmare a faptului că A este o mbo, are cel mai mic element; fie acesta a . Elementul $b = f(a)$ este în A și, deoarece $b < a$ și f este strict monotonă, are loc $f(b) < f(a) = b$. Deci $b \in B$. Aceasta însă contrazice alegerea lui $a \in B$. \square

Corolarul 1.5.1.1. Există un singur izomorfism de la o mbo la ea însăși, iar acest izomorfism este identitatea.

Demonstrație. Fie $M = (A, \leq)$ o mbo și f un izomorfism de la ea la ea însăși. De la Teorema 1.5.1.1 obținem $a \leq f(a)$ și $a \leq f^{-1}(a)$, pentru orice a , deoarece atât f , cât și f^{-1} sunt funcții strict monotone. f fiind funcție monotonă, urmează că $f(a) \leq f(f^{-1}(a))$, adică $f(a) \leq a$. Combinând această inegalitate cu $a \leq f(a)$ obținem $a = f(a)$, pentru orice $a \in A$, adică f este funcția identitate pe A . \square

O structură $M = (A, \rho)$ este numită *rigidă* dacă singurul ei automorfism este aplicația identică 1_A . Ca urmare, corolarul anterior ne spune că orice mbo este rigidă.

Corolarul 1.5.1.2. Dacă două mbo sunt izomorfe, atunci izomorfismul este unic.

Demonstrație. Fie $M = (A, \leq)$ și $M' = (A', \leq')$ două mbo izomorfe, iar f și g două izomorfisme de la M la M' . Considerăm mulțimea $B = \{x \in A | f(x) \neq g(x)\}$ și arătăm că $B = \emptyset$, ceea ce va conduce la $f = g$. Presupunem, prin contradicție, că B este nevidă. Atunci există cel mai mic element al ei, fie acesta a . Fără a restrâng generalitatea putem considera că $f(a) < g(a)$. Pentru orice $b \in A - \{a\}$ are loc:

- dacă $b < a$, atunci $g(b) = f(b) < f(a)$;
- dacă $b > a$, atunci $g(b) > g(a) > f(a)$.

Ca urmare, nu există $b \in A$ astfel încât $g(b) = f(a)$, ceea ce arată că g nu poate fi izomorfism de la M la M' , ceea ce constituie o contradicție. \square

Definiția 1.5.1.2. Fie $M = (A, \leq)$ o mbo și $a \in A$. Mulțimea $M_a = \{b \in A | b < a\}$ este numită *segmentul initial induș de a în M* .

Corolarul 1.5.1.3. Nici o mbo nu este izomorfă cu un segment inițial al ei însăși.

Demonstrație. Fie $M = (A, \leq)$ o mbo și $a \in A$. Dacă ar exista un izomorfism f de la A la M_a , atunci funcția $g : A \rightarrow A$ dată prin $g(x) = f(x)$, pentru orice $x \in A$, ar fi funcție strict monotonă de la A la A ce ar satisface $g(a) < a$, contrazicând astfel Teorema 1.5.1.1. \square

Următoarea teoremă prezintă un alt doilea rezultat fundamental asupra mulțimilor bine ordonate.

Teorema 1.5.1.2. (Teorema comparației pentru ordini bune)

Pentru orice două mbo $M = (A, \leq)$ și $M' = (A', \leq')$, exact una din următoarele proprietăți este satisfăcută:

1. M este izomorfă cu M' ;
2. M este izomorfă cu un segment inițial al lui M' ;
3. M' este izomorfă cu un segment inițial al lui M .

Demonstrație. Dacă, de exemplu, $A = \emptyset$ și $A' \neq \emptyset$, atunci funcția vidă stabilăse un izomorfism de la M la M'_a , unde b este cel mai mic element al lui M' ; similar se discută cazurile $A' = \emptyset$ și $A \neq \emptyset$, și $A = A' = \emptyset$.

Presupunem acum că A și A' sunt nevide. Considerăm relația binară f de la A la A' dată prin

$$f = \{(a, a') \in A \times A' | M_a \text{ și } M'_{a'} \text{ sunt izomorfe}\}.$$

În baza Corolarelor 1.5.1.2 și 1.5.1.3, este ușor de văzut că f este funcție și au loc următoarele:

- dacă $a, b \in Dom(f)$ și $a \neq b$, atunci $f(a) \neq f(b)$;
- dacă $b' \in Cod(f)$ și $a' < b'$, atunci $a' \in Dom(f)$;
- dacă $f(a) = a'$, $f(b) = b'$ și $a < b$, atunci $a' < b'$.

Asupra domeniului și codomeniului funcției f avem de considerat următoarele cazuri:

- $Dom(f) = A$ și $Cod(f) = A'$. Atunci, f este izomorfism de la M la M' ;
- $Cod(f) \subset A'$. Atunci, $Dom(f) = A$. În adevăr, dacă $Dom(f) \subset A$, atunci $(x_0, y_0) \in f$, unde x_0 este cel mai mic element al mulțimii $A - Dom(f)$, iar y_0 este cel mai mic element al mulțimii $A' - Cod(f)$. Aceasta însă contrazice faptul că $Cod(f)$ nu conține y_0 . Ca urmare, în acest caz f este izomorfism de la M la un segment inițial al lui M' ;
- $Dom(f) \subset A$. Atunci, obținem că mai sus $Cod(f) = A'$, și astfel f este izomorfism de la un segment inițial al lui M la M' .

Cu acestea, demonstrația teoremei este încheiată. \square

1.5.2. Numere ordinale

Echipotența și numerele naturale stau la baza analizei “dimensionale” a mulțimilor finite. Astfel, am spus că o mulțime A este *finită* dacă este echipotentă cu un

număr natural; în caz contrar, A este numită *infinite*. Notiunea de “finit” este strâns legată de ceea ce intuitiv ar însemna “parcugerea mulțimilor element cu element”, iar numerele naturale, prin modul inductiv de definire al lor, creează suficient de clar această imagine.

Cum orice două numere naturale diferite nu pot fi echivalente, rezultă că mulțimile finite pot fi clasificate gradat: mulțimi fără nici un element (singura astfel de mulțime este \emptyset), mulțimi cu un element (echivalente cu 1), mulțimi cu două elemente (echivalente cu 2) etc. În cazul mulțimilor infinite avem o clasificare doar în două mari subclase: *mulțimi numărabile* (echivalente cu \mathbb{N}) și *mulțimi nenumărabile* (care nu sunt echivalente cu \mathbb{N}). Însă, privind cu atenție la modul de formare a numerelor naturale și a mulțimii \mathbb{N} ,

$$\begin{aligned} 0 &= \emptyset \\ 1 &= \{0\} \\ 2 &= \{0, 1\} \\ \dots \\ \mathbb{N} &= \{0, 1, \dots\}, \end{aligned}$$

constatăm că orice număr natural este format din exact toate numerele definite anterior și această proprietate o are și \mathbb{N} . Mai mult, \mathbb{N} este tranzitivă, strict ordonată prin $\in_{\mathbb{N}}$ și orice submulțime nevidă a ei are cel mai mic element. Ceea ce face ca \mathbb{N} să nu fie număr natural este faptul că nu orice submulțime nevidă a ei are cel mai mare element. Dar, faptul că \mathbb{N} este formată din exact toate numerele naturale definite anterior conduce la ideea conceperii lui \mathbb{N} tot ca un *număr*. În acest context, notăm \mathbb{N} prin ω . Atunci “secvența” de mai sus poate fi “continuată” prin:

$$\begin{aligned} 0 &= \emptyset \\ 1 &= \{0\} \\ 2 &= \{0, 1\} \\ \dots \\ \omega &= \{0, 1, \dots\} \\ S(\omega) &= \omega \cup \{\omega\} \\ S(S(\omega)) &= S(\omega) \cup \{S(\omega)\} \\ \dots \end{aligned}$$

Puteam nota sugestiv $\omega + 1$ în loc de $S(\omega)$, $\omega + 2$ în loc de $S(S(\omega))$ etc. Aceste numere: ω , $\omega + 1$, $\omega + 2$ etc. vor fi cazuri particulare a ceea ce vom numi *numere ordonale*, și ele pot fi utilizate acum pentru a “grada dimensional” mulțimile infinite.

Definiția 1.5.2.1. O mulțime x este numită *număr ordinal* sau *ordinal* dacă este tranzitivă și bine ordonată prin apartenență³⁰.

³⁰ Această definiție a numerelor ordonale apare pentru prima dată la Zermelo în perioada 1915-1916,

Exemplul 1.5.2.1.

- (1) Orice număr natural este ordinal. Aceasta se obține cu ușurință de la Corolarul 1.1.3.1(2)(3)(7)(12).
- (2) \mathbb{N} este ordinal (aceasta urmează de la Teorema 1.1.3.3). În acest context el va fi notat prin ω .

Vom utiliza cu precădere litere mici ale alfabetului grecesc pentru a denota ordinali. *Clasa tuturor ordinalilor* va fi notată prin ON .

Definiția 1.5.2.2. Fie α și β ordinali. Spunem că α este mai mic decât β și notăm $\alpha < \beta$, dacă $\alpha \in \beta$.

În mod ușual definim $\alpha \leq \beta$, $\alpha > \beta$ și $\alpha \geq \beta$. Următoarea lemă prezintă câteva proprietăți de bază ale ordinalilor.

Lema 1.5.2.1. Au loc următoarele proprietăți:

- (1) pentru orice ordinal α , $\alpha \not< \alpha$;
- (2) dacă α este ordinal și $\beta < \alpha$, atunci β este ordinal;
- (3) dacă α și β sunt ordinali și $\beta \subset \alpha$, atunci $\beta < \alpha$;
- (4) dacă α și β sunt ordinali, atunci $\alpha \leq \beta$ sau $\beta \leq \alpha$.

Demonstrație. (1) și (2) urmează de la definiție.

(3) Presupunem că $\beta \subset \alpha$ și fie γ cel mai mic element al mulțimii $\alpha - \beta$. Deoarece β este tranzitivă, obținem $\beta = \{x \in \alpha | x < \gamma\}$. Ca urmare, $\beta < \alpha$.

(4) Este clar că $\alpha \cap \beta$ este ordinal. Dacă $\alpha \cap \beta = \alpha$, atunci $\alpha \leq \beta$, iar dacă $\alpha \cap \beta = \beta$, atunci $\beta \leq \alpha$. Să arătăm că nu există alt caz asupra ordinalului $\alpha \cap \beta$. Dacă am presupune că $\alpha \cap \beta \neq \alpha$ și $\alpha \cap \beta \neq \beta$, atunci am obține că $\alpha \cap \beta \subset \alpha$ și $\alpha \cap \beta \subset \beta$, iar (3) ar conduce la $\alpha \cap \beta < \alpha \cap \beta$, ceea ce ar contrazice (1). \square

Corolarul 1.5.2.1. Au loc următoarele proprietăți:

- (1) relația $<$ este ordine totală strictă pe clasa tuturor ordinalilor;
- (2) pentru orice ordinal α , $\alpha = \{\beta | \beta < \alpha\}$;

într-un material nepublicat (conform cu [5], pag. 6). Un studiu detaliat al lor, fără a se referi la ele ca fiind numere ordonale, este făcut de Mirimanoff în 1917 [148]. În 1923, von Neumann recurge la aceeași definiție a noțiunii de ordinal [155]. Astăzi, multe tratate asupra teoriei mulțimilor atribuie noțiunea de ordinal lui von Neumann (a se vedea, de exemplu, [215]).

O altă definiție a noțiunii de ordinal a fost propusă de Mirimanoff [148], definiție conform căreia o mulțime tranzitivă este un ordinal dacă orice două elemente ale ei sunt comparabile prin relația de apartenență (a se vedea și [5]). Echivalenta celor două definiții face apel la Axioma regularității (a se vedea [6], pag. 87).

- (3) pentru orice clasă nevidă \mathbf{C} de ordinali, $\bigcap \mathbf{C}$ este ordinal, fiind cel mai mic ordinal al clasei \mathcal{C} ;
- (4) α este ordinal dacă și numai dacă $S(\alpha)$ este ordinal. În plus, dacă α este ordinal, atunci $S(\alpha)$ este cel mai mic ordinal mai mare decât α .

Definiția 1.5.2.3. Fie α un ordinal.

- (1) α este numit *ordinal successor* dacă există un ordinal β astfel încât $\alpha = S(\beta)$. Altfel, α este numit *ordinal limită*.
- (2) α este numit *ordinal finit* dacă $\alpha = 0$ sau α este ordinal successor și orice $\beta < \alpha$ este sau 0, sau ordinal successor. Altfel, α este numit *ordinal infinit*.

Exemplul 1.5.2.2.

- (1) Pentru orice ordinal α , $S(\alpha)$ este ordinal successor al lui α .
- (2) Se arată prin inducție că orice număr natural diferit de 0 este ordinal successor finit (0 este ordinal finit). Reciproc, anticipând Prinzipiul inducției pe ordinali ce va fi prezentat în Secțiunea 1.5.3, se arată cu ușurință că orice ordinal finit este număr natural.
- (3) Mulțimea tuturor ordinalilor finiți este ordinalul ω . Mai mult, ω este cel mai mic ordinal infinit și cel mai mic ordinal limită diferit de 0.
- (4) $S(\omega), S(S(\omega)), \dots$ sunt ordinali succesiuni.

Propoziția 1.5.2.1. Dacă X este o mulțime nevidă de ordinali, atunci $\bigcup X$ este ordinal și:

- (1) dacă X are cel mai mare element, fie acesta α , atunci $\alpha = \bigcup X$;
- (2) dacă X nu are cel mai mare element, atunci $\bigcup X$ este ordinal limită și are loc $\alpha < \bigcup X$, pentru orice $\alpha \in X$.

În particular, $S(\bigcup X) \notin X$.

Demonstrație. Se arată că $\bigcup X$ este mulțime tranzitivă de ordinali, bine ordonată prin apartenență. Rămâne în seama cititorului să completeze demonstrația acestei propoziții. \square

Ordinalul $\bigcup X$ definit în Propoziția 1.5.2.1 este numit *supremumul* mulțimii de ordinali X ; el este notat în mod ușual prin $\sup(X)$.

Corolarul 1.5.2.2. ON este clasă proprie.

Demonstrație. Dacă ON ar fi mulțime, atunci ON și $S(ON)$ ar fi ordinali ce satisfac $S(ON) \notin ON$, ceea ce contrazice definiția clasei ON. \square

Propoziția 1.5.2.2. Fie α un ordinal.

- (1) α este ordinal limită dacă și numai dacă $(\forall \beta)(\beta < \alpha \Rightarrow S(\beta) < \alpha)$.
- (2) Dacă α este ordinal limită, atunci $\bigcup \alpha = \alpha$. Deci orice ordinal limită este supremumul mulțimii tuturor ordinalilor mai mici decât el.

Demonstrație. (1) Dacă $\alpha = 0$, atunci proprietatea este trivial satisfăcută. Dacă α este ordinal limită diferit de 0, atunci pentru orice $\beta < \alpha$, $S(\beta)$ va fi mai mic decât α deoarece α și $S(\beta)$ sunt comparabili, iar relația $\alpha \leq S(\beta)$ nu poate avea loc (altfel, α ar fi succesor al lui β și deci nu ar fi ordinal limită). Reciproca se obține în mod similar.

(2) Să presupunem că α este ordinal limită. Evident, $\bigcup \alpha \subseteq \alpha$. Dacă această incluziune ar fi strictă, atunci ar exista $\beta \in \alpha - \bigcup \alpha$. β este ordinal (el este element al unui ordinal) și $\beta < \alpha$. Deoarece β nu este element al ordinalului $\bigcup \alpha$, urmează $\bigcup \alpha \leq \beta$.

α este ordinal limită, iar de la (1) deducem $S(\beta) < \alpha$. Obținem astfel $\bigcup \alpha < S(\beta) < \alpha$, ceea ce contrazice faptul că $\bigcup \alpha$ este cel mai mic ordinal mai mare sau egal decât orice ordinal al lui α . \square

1.5.3. Axioma înlocuirii. Inducție și recursie pe ordinali

Așa cum am afirmat la sfârșitul Secțiunii 1.1.1, uneori suntem interesați în a construi mulțimi B ce “corespond” unor mulțimi date A în manieră funcțională. Dacă am cunoaște o mulțime C din care ar trebui să selectăm obiectele ce vor alcătui B , atunci am putea aplica Axioma separării. De multe ori însă C este o clasă proprie (clasa tuturor ordinalilor, de exemplu). Dar, dacă obiectele clasei B sunt construite pornind de la obiectele mulțimii A în manieră funcțională, atunci intuiția ne spune că B ar trebui să fie mulțime. Drept exemplu, să alegem $A = \mathbb{N}$ și B colecția mulțimilor $\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \dots$. Dacă numărului natural n îi asociem mulțimea

$$\underbrace{\{\dots \}}_n \underbrace{\{\emptyset\} \dots \}_{n},$$

atunci am putea spune că avem o corespondență funcțională între A și elementele unei “anumite mulțimi”, iar B ar fi imaginea mulțimii A prin această corespondență; deci B ar fi mulțime. Acest raționament are însă o fisură mare: codomeniul acestei corespondențe funcționale trebuie să includă B și nu este clar absolut deloc cum să demonstreze existența unei mulțimi ce include B (din orice mulțime ce include B , prin Axioma separării am putea obține B). Ca urmare, pentru a demonstra existența mulțimii B avem de demonstrat în prealabil ... existența mulțimii B .

O altă tentativă de a demonstra existența mulțimii B ar fi aceea de a defini și următoarea:

$$\vdash a_0 = \emptyset;$$

$- a_{n+1} = \{a_n\}$, pentru orice $n \geq 0$

și de a reuni elementele lui, ceea ce ar conduce la B . Dificultatea care apare este similară celei precedente: avem nevoie de o funcție $h : C \times \mathbf{N} \rightarrow C$ dată prin $h(x, n) = \{x\}$, unde C este o mulțime ce include B .

Toate acestea conduc la necesitatea considerării unei noi axiome.

Definiția 1.5.3.1. O formulă (proprietate) $P(x, y)$ spunem că este *funcțională pe o mulțime A* dacă, pentru orice $x \in A$, există cel mult un y astfel încât $P(x, y)$.

Axioma înlocuirii. Pentru orice mulțime A și proprietatea funcțională $P(x, y)$ pe A , există o mulțime B astfel încât

$$(\forall y)(y \in B \Leftrightarrow (\exists x)(x \in A \wedge P(x, y))).$$

Această axiomă a fost introdusă independent de Fraenkel [57] și Skolem [193] în 1922. Necessitatea unei astfel de axiome a fost oarecum "simțită" anterior de Cantor ([28], pag. 444) și de Mirimanoff ([148], pag. 49).

Cerința ca proprietatea P în cadrul Axiomei înlocuirii să fie funcțională pe A este esențială. De exemplu, dacă am aplica această axiomă proprietății " $x \subseteq y$ " (care nu este funcțională pe nici o mulțime A) și mulțimii $A = \{\emptyset\}$, atunci mulțimea B a cărei existență ar fi asigurată de ea ar fi mulțimea tuturor mulțimilor!

Așa cum credem că este clar, Axioma înlocuirii poate fi exprimată în termeni de clase. Menționăm întâi că noțiunile de relație și funcție pot fi extinse și la clase; în acest caz ele vor fi numite *c-relație* și, respectiv, *c-funcție*. Astfel, o c-relație (binară) este o clasă de perechi ordonate. Putem vorbi de c-relație peste o clasă \mathbf{C} ca fiind o clasă de perechi ordonate ale căror elemente sunt membre ale clasei \mathbf{C} . c-funcțiile vor fi notate ca și funcțiile cu deosebirea că pentru ele vom folosi litere mari îngroșate (de exemplu, $\mathbf{F} : \mathbf{V} \rightarrow \mathbf{V}$). Domeniul și codomeniul unei c-funcții sunt în general clase. Dacă \mathbf{C} este o *clasă de c-funcții compatibile*, în sens similar funcțiilor, atunci $\bigcup \mathbf{C}$ este c-funcție.

Vom spune că o c-relație ρ pe clasa \mathbf{C} este *limitată (restricționată) la stânga* dacă, pentru orice x , colecția tuturor acelor y ce satisfac $y \rho x$ este mulțime. Similar se definesc relațiile *limitate (restricționate) la dreapta*. Dacă ρ este limitată la stânga (dreapta) pentru orice x din clasa \mathbf{A} , atunci spunem că ρ este limitată la stânga (dreapta) pe \mathbf{A} .

O *clasă bine ordonată* este un cuplu $(\mathbf{A}, <)$ format dintr-o clasă \mathbf{A} și o c-relație $<$ pe \mathbf{A} ce ordonează total strict \mathbf{A} , este limitată la stânga pe \mathbf{A} și orice mulțime nevidă $B \subseteq \mathbf{A}$ are cel mai mic element (în raport cu $<$). Orice relație este c-relație limitată la stânga și la dreapta. Orice mulțime bine ordonată este clasă bine ordonată. Clasele bine ordonate pentru care clasa în cauză este proprie vor fi numite *clase proprii bine ordonate*.

Formularea Axiomei înlocuirii în termeni de clase se poate face în una din următoarele două variante:

- pentru orice c-funcție \mathbf{F} pentru care $\text{Dom}(\mathbf{F})$ este mulțime, $\text{Cod}(\mathbf{F})$ este mulțime;
- pentru orice c-funcție \mathbf{F} și mulțime X există o funcție f astfel încât $\mathbf{F}|_X = f$.

Axioma înlocuirii implică (direct) Axioma separării și, prin intermediul Axiomei părților, pe cea a împerecherii. În adevăr, în cazul Axiomei separării, considerând o mulțime A și o proprietate $P(x)$, existența mulțimii

$$B = \{x \in A | P(x)\}$$

este asigurată și de Axioma înlocuirii aplicată mulțimii A cu proprietatea funcțională " $x = y \wedge P(y)$ ".

Pentru a arăta cum Axioma împerecherii este implicată de cea a înlocuirii, procedăm astfel³¹. Pornind de la mulțimea vidă, Axioma părților conduce la existența mulțimii $\{\emptyset, \{\emptyset\}\}$. Considerând atunci 2 obiecte a și b , existența mulțimii $\{a, b\}$ va fi asigurată de Axioma înlocuirii aplicate mulțimii $\{\emptyset, \{\emptyset\}\}$ cu proprietatea funcțională $P(x, y)$ dată prin

$$“(x = \emptyset \wedge y = a) \vee (x = \{\emptyset\} \wedge y = b)”.$$

Chiar dacă Axioma înlocuirii implică Axiomele separării și împerecherii, vom prefera să le păstrăm și pe acestea în cadrul sistemului axiomatic datorită simplității lor, a faptului că ele au fost introduse în manieră naturală pornind de la probleme relativ simple. În contrast cu acestea, Axioma înlocuirii este o axiomă mult mai complexă care își are locul într-o problematică avansată a teoriei mulțimilor.

Utilizând Axioma înlocuirii obținem:

Teorema 1.5.3.1. Orice mulțime bine ordonată este izomorfă cu un unic ordinal.

Demonstrație. Fie $M = (A, \leq)$ o mulțime bine ordonată. Considerăm c-funcția \mathbf{F} dată prin $\mathbf{F}(a) = \alpha$, unde α este ordinal izomorf cu M_a , pentru orice $a \in A$. Consistența acestei c-funcții urmează astfel. În primul rând, dacă pentru $a \in A$ există α cu $\mathbf{F}(a) = \alpha$, atunci α este unic. Apoi, dacă ar exista $a \in A$ pentru care nu există α cu $\mathbf{F}(a) = \alpha$, atunci fie b cel mai mic element din A cu această proprietate. b nu poate fi cel mai mic element c din A deoarece $\mathbf{F}(c) = 0$. Ca urmare, $M_b \neq \emptyset$. Axioma înlocuirii ne spune că $\mathbf{F}(M_b)$ este mulțime. Atunci este ușor de văzut că $\mathbf{F}(b) = \bigcup \mathbf{F}(M_b)$.

Încheiem demonstrația repetând raționamentul de mai sus. Adică, $\mathbf{F}(A)$ este mulțime conform Axiomei înlocuirii și cel mai mic ordinal care nu este în $\mathbf{F}(A)$ va fi unicul ordinal izomorf cu M . \square

Dată mulțimea bine ordonată $M = (A, <)$, unicul ordinal a cărui existență este asigurată de Teorema 1.5.3.1 este numit *tipul de ordine* al mulțimii M . Uzual, el

³¹Acest rezultat apare pentru prima dată la Zermelo în 1930 [229].

se notează prin $\|M\|$. Acest ordinal poate fi considerat ca reprezentant al "clasei" tuturor mulțimilor bine ordonate izomorfe cu el (menționăm că nu există o mulțime a tuturor mulțimilor bine ordonate izomorfe cu o mulțime bine ordonată dată). Ca urmare, ordinalii furnizează informații "structurale" asupra universului de mulțimi.

Corolarul 1.5.3.1. Fie M_1 și M_2 două mulțimi bine ordonate. Atunci $M_1 \cong M_2$ dacă și numai dacă $\|M_1\| = \|M_2\|$.

Corolarul 1.5.3.2. O mulțime poate fi bine ordonată dacă și numai dacă este echipotentă cu un ordinal.

Demonstrație. Fie A o mulțime ce poate fi bine ordonată printr-o relație binară $<$ pe A . Atunci $(A, <) \cong \|(A, <)\|$, ceea ce arată că A și ordinalul $\|(A, <)\|$ sunt echipotente.

Reciproc, dacă A este echipotentă cu un ordinal α , atunci definim pe A relația binară $<$ prin

$$a < b \Leftrightarrow f(a) \in f(b),$$

pentru orice $a, b \in A$, unde f este o bijectie de la A la α . Se verifică ușor că $(A, <)$ este mulțime bine ordonată. \square

Inducția și recursia pot fi generalizate la ordinali.

Teorema 1.5.3.2. (Principiul inducției transfinite – varianta I)
Fie $P(x)$ o proprietate astfel încât, pentru orice ordinal α , are loc

$$(\forall \beta)(\beta < \alpha \Rightarrow P(\beta)) \Rightarrow P(\alpha).$$

Atunci, P este satisfăcută de orice ordinal.

Demonstrație. Presupunem, prin contradicție, că există ordinali ce nu satisfac P . Corolarul 1.5.2.1(3) asigură existența celui mai mic ordinal α cu această proprietate. Ca urmare, P va fi satisfăcută de toți ordinalii $\beta < \alpha$, ceea ce conduce la faptul că P trebuie să fie satisfăcută și de α , ceea ce constituie o contradicție. \square

Conform Teoremei 1.5.3.2, pentru a arăta că o proprietate este satisfăcută de toți ordinalii este suficient să demonstrăm că proprietatea este satisfăcută de un ordinal (arbitraru), presupunând că toți ordinalii mai mici decât el satisfac proprietatea.

Este utilă uneori o variantă a Principiului inducției transfinite similară inducției clasice.

Teorema 1.5.3.3. (Principiul inducției transfinite – varianta II)
Fie $P(x)$ o proprietate astfel încât:

(i) $P(0)$;

(ii) pentru orice ordinal α , $P(\alpha)$ implică $P(S(\alpha))$;

(iii) pentru orice ordinal limită $\alpha \neq 0$, dacă $P(\beta)$ pentru orice $\beta < \alpha$, atunci $P(\alpha)$. Atunci P este satisfăcută de toți ordinalii.

Demonstrație. Se arată cu ușurință că ipoteza acestei teoreme implică ipoteza Teoremei 1.5.3.2. \square

În Secțiunea 1.1.3.4 s-au introdus concepțele de secvență finită și infinită. Urmărind aceeași abordare introducem conceptul de *secvență transfinิตă*.

Definiția 1.5.3.2. Se numește *secvență transfinิตă* de elemente peste A orice funcție f cu domeniul un număr ordinal și cu valori în A .

Uzual, secvențele transfinite sunt notate prin

$$\langle a_\beta | \beta < \alpha \rangle,$$

unde α este un ordinal.

Variantele de recursie din Secțiunea 1.1.3.5 permit definirea recursivă/inductivă de secvențe infinite. Pentru cazul secvențelor transfinite, teoremele de recursie trebuie adaptate corespunzător. Vom prezenta în continuare două variante ale recursiei transfinite, fără demonstrații. Pentru detalii cititorul este îndrumat către [207]. Menționăm însă că demonstrațiile acestora fac apel la Axioma Înlocuirii.

Teorema 1.5.3.4. (Teorema recursiei transfinite)

Fie $\mathbf{H} : \mathbf{V} \rightarrow \mathbf{V}$ o c-funcție. Atunci există o unică c-funcție $\mathbf{F} : \mathbf{ON} \rightarrow \mathbf{V}$ astfel încât

$$\mathbf{F}(\alpha) = \mathbf{H}(\mathbf{F}|_\alpha),$$

pentru orice ordinal α .

Putem formula o variantă parametrică a Teoremei recursiei, considerând c-funcția \mathbf{H} ca fiind de două variabile, una din ele fiind un parametru. Vom face întâi următoarea notație. Dacă \mathbf{F} este o c-funcție descrisă de formula $F(p, x, y)$, atunci prin \mathbf{F}_p vom nota c-funcția de o variabilă descrisă de formula $F_p(x, y)$ dată prin

$$(\forall x, y)(F_p(x, y) \Leftrightarrow F(p, x, y)).$$

Teorema 1.5.3.5. (Varianta parametrică a Teoremei recursiei transfinite)

Fie $\mathbf{H} : \mathbf{P} \times \mathbf{V} \rightarrow \mathbf{V}$ o c-funcție, unde \mathbf{P} este o clasă. Atunci, există o unică c-funcție $\mathbf{F} : \mathbf{P} \times \mathbf{ON} \rightarrow \mathbf{V}$ astfel încât

$$\mathbf{F}(p, \alpha) = \mathbf{H}(p, \mathbf{F}_p|_\alpha),$$

pentru orice $p \in \mathbf{P}$ și ordinal α .

Teorema recursiei transfinite este un instrument fundamental în definirea aritmeticii numerelor ordinale.

Teorema 1.5.3.6. (Adunare, înmulțire și exponentiere de ordinali³²⁾

- (1) Există o unică c-funcție de două variabile, notată $+$, astfel încât, pentru orice ordinali α și β , au loc relațiile:

- $\alpha + 0 = \alpha;$
- $\alpha + S(\beta) = S(\alpha + \beta);$
- $\alpha + \beta = \sup(\{\alpha + \gamma | \gamma < \beta\}),$ dacă β este ordinal limită, $\beta \neq 0.$

- (2) Există o unică c-funcție de două variabile, notată \cdot , astfel încât, pentru orice ordinali α și β , au loc relațiile:

- $\alpha \cdot 0 = 0;$
- $\alpha \cdot S(\beta) = (\alpha \cdot \beta) + \alpha;$
- $\alpha \cdot \beta = \sup(\{\alpha \cdot \gamma | \gamma < \beta\}),$ dacă β este ordinal limită, $\beta \neq 0.$

- (3) Există o unică c-funcție de două variabile, notată $\hat{\cdot}$, astfel încât, pentru orice ordinali α și β , au loc relațiile:

- $\alpha \hat{\cdot} 0 = 1;$
- $\alpha \hat{\cdot} S(\beta) = (\alpha \hat{\cdot} \beta) \cdot \alpha;$
- $\alpha \hat{\cdot} \beta = \sup(\{\alpha \hat{\cdot} \gamma | \gamma < \beta\}),$ dacă β este ordinal limită, $\beta \neq 0.$

Pentru ridicarea la putere (exponentiere) se utilizează în mod frecvent notația α^β în loc de $\hat{\alpha}^\beta.$ Este clar că restricția adunării (înmulțirii, exponentierii) numerelor ordinarale la numere ordinarale finite (numere naturale) satisfac relațiile recursive ale adunării (înmulțirii, exponentierii) numerelor naturale, și astfel coincide cu aceasta. Iată motivul pentru care am adoptat aceeași notație pentru adunarea (înmulțirea, exponentierea) numerelor ordinarale, ca și pentru numere naturale.

Ca și în cazul adunării numerelor naturale, observăm că $S(\alpha) = \alpha + 1,$ ceea ce ne permite a (re-)nota succesorul imediat al lui α prin $\alpha + 1.$

Înainte de a exemplifica operațiile cu ordinali și de a stabili unele proprietăți ale acestora, vom introduce următoarea noțiune.

Definiția 1.5.3.3. O c-funcție $F : ON \rightarrow ON$ este numită *normală* dacă:

- (i) $\alpha < \beta$ implică $F(\alpha) < F(\beta),$ pentru orice $\alpha, \beta \in ON;$ (strictă monotonie)
- (ii) $F(\alpha) = \sup(\{F(\gamma) | \gamma < \alpha\}),$ pentru orice ordinal limită $\alpha, \alpha \neq 0.$ (continuitate)

³²Definițiile recursive ale operațiilor de adunare și înmulțire pe ordinali apar pentru prima dată în 1909 la Jacobsthal [99], în timp ce exponentierea ordinalilor în variantă recursivă provine de la Cantor din 1897 [27].

În cazul c-funcțiilor de două variabile (așa cum sunt operațiile introduse prin intermediul Teoremei 1.5.3.6), prin fixarea uneia din variabile obținem o funcție de o variabilă și, astfel, vom vorbi de *normalitate în raport cu primul* sau *al doilea argument* (sau, *normalitate la stânga* și, respectiv, *la dreapta*). De exemplu, normalitatea adunării în raport cu al doilea argument înseamnă:

- pentru orice ordinali α, β și $\gamma,$ dacă $\beta < \gamma,$ atunci $\alpha + \beta < \alpha + \gamma;$
- pentru orice ordinali α și $\beta,$ dacă β este ordinal limită diferit de ordinalul 0, atunci

$$\alpha + \sup(\{\gamma | \gamma < \beta\}) = \sup(\{\alpha + \gamma | \gamma < \beta\})$$

(menționăm că $\beta = \sup(\{\gamma | \gamma < \beta\}).$)

Exemplul 1.5.3.1.

- (1) $(\alpha + 1) + 1 = \alpha + (1 + 1) = \alpha + 2, (\alpha + 2) + 1 = \alpha + (2 + 1) = \alpha + 3$ etc.
- (2) $\omega + \omega = \sup(\{\omega + n | n < \omega\})$ și $(\omega + \omega) + \omega = \sup(\{(\omega + \omega) + n | n < \omega\}).$
- (3) $m + \omega = \sup(\{m + n | n < \omega\}) = \omega$ și $\omega + m > \omega,$ pentru orice $m < \omega.$ Deci adunarea ordinalilor nu este comutativă.
- (4) $0 + \omega = 1 + \omega = \omega,$ ceea ce arată că adunarea ordinalilor nu este strict monotonă în primul argument și, ca urmare, nu este cu simplificare la dreapta. Mai mult, ea nu poate fi normală în primul argument.
- (5) $\sup(\{n | n < \omega\}) + 1 = \omega + 1 > \omega = \sup(\{n + 1 | n < \omega\}),$ ceea ce arată că adunarea ordinalilor nu este continuă în primul argument.
- (6) Arătăm prin inducție transfinิตă că, pentru orice ordinal $\alpha,$ are loc $1 + \alpha \leq \alpha + 1.$ În adevăr:

- dacă $\alpha = 0,$ atunci $1 + 0 = 0 + 1;$
- dacă $\alpha = \beta + 1$ și $1 + \beta \leq \beta + 1,$ atunci

$$\begin{aligned} 1 + \alpha &= 1 + (\beta + 1) \\ &= (1 + \beta) + 1 \\ &\leq (\beta + 1) + 1 \\ &= \alpha + 1; \end{aligned}$$

- dacă α este ordinal limită diferit de 0 și $1 + \beta \leq \beta + 1,$ pentru orice $\beta < \alpha,$ atunci

$$\begin{aligned} 1 + \alpha &= \sup(\{1 + \beta | \beta < \alpha\}) \\ &\leq \sup(\{\beta + 1 | \beta < \alpha\}) \\ &= \alpha < \alpha + 1 \end{aligned}$$

(s-a folosit faptul că $\beta + 1 < \alpha,$ pentru orice $\beta < \alpha).$

Pentru detalii asupra aritmeticii ordinalilor cititorul este îndrumat către [207].

1.5.4. Principiul bunei ordonări și alte propoziții echivalente Axiomei alegerii

În Secțiunea 1.5.3 am arătat că orice mulțime bine ordonată este izomorfă cu un ordinal. Rămâne însă o problemă la care ar trebui răspuns: poate fi bine ordonată orice mulțime? Această problemă a fost analizată pentru prima dată de Cantor în 1883 [25], care a considerat-o ca având un răspuns pozitiv “evident”. Intuitiv, o ordonare bună a unei mulțimi arbitrară A s-ar obține astfel: se alege, în mod arbitrar, un element a_0 din A , apoi un element a_1 din $A - \{a_0\}$, apoi un element a_2 din $A - \{a_0, a_1\}$ etc. Acest raționament are un gol major: nu este clar absolut deloc cum se alege un element arbitrar dintr-o mulțime $B \subseteq A$. Dacă submulțimile nevide ale lui A ar avea cel mai mic element în raport cu o ordine totală bine precizată, atunci alegerea unui element dintr-o astfel de mulțime s-ar putea baza pe acest fapt (a_0 ar fi cel mai mic element al mulțimii A , a_1 ar fi cel mai mic element al mulțimii $A - \{a_0\}$ etc). Lucrând însă sub incidența Axiomei alegerii, Zermelo [227] a arătat că raționamentul intuitiv prezentat mai sus poate fi completat la unul corect din punct de vedere formal, ce răspunde pozitiv întrebării lui Cantor. Vom prezenta acest rezultat, cunoscut ca *Principiul bunei ordonări*, fără demonstrație (a se vedea pentru detalii [207]).

Teorema 1.5.4.1. (Principiul bunei ordonări (PBO) ³³⁾

Orice mulțime poate fi bine ordonată.

Combinând Teorema 1.5.4.1 cu Teorema 1.5.3.1 obținem:

Corolarul 1.5.4.1. Orice mulțime este echivalentă cu un ordinal.

Corolarul 1.5.4.1 justifică terminologia frecvent utilizată: “fie $\{a_\beta | \beta < \alpha\}$ o ordonare bună a mulțimii A ”. Conform acestui corolar, orice mulțime A este izomorfă cu un ordinal. Dacă $f : \alpha \rightarrow A$ este un astfel de izomorfism, atunci considerând $a_\beta = f(\beta)$, pentru orice $\beta < \alpha$, putem scrie $A = \{a_\beta | \beta < \alpha\}$.

Vom încheia secțiunea prin prezentarea a două “principii de maximalitate” frecvent întâlnite. În general, un principiu de maximalitate postulează existența unei substructuri maximale (element maximal, lanț maximal, submulțime maximală) a unei structuri date (mpo, familie de mulțimi) ce satisfac anumite proprietăți. Se pare că primul astfel de principiu apare în 1914 la F. Hausdorff [86].

Teorema 1.5.4.2. (Principiul de maximalitate al lui Hausdorff (PMH))

Orice mulțime parțial ordonată are cel puțin un lanț maximal.

Demonstrație. Fie $M = (A, \leq)$ o mpo nevidă (dacă $A = \emptyset$, atunci A este lanț

³³Această teoremă este cunoscută și ca *Teorema bunei ordonări* sau *Teorema lui Zermelo*. Cea de a două denumire provine de la faptul că Zermelo este cel care, în 1904, a introdus Axioma alegerii și a utilizat-o în demonstrarea aceastei teoreme [227].

maximal). Corolarul 1.5.4.1 conduce la existența unui ordinal α și a unei bijectii $f : \alpha \rightarrow A$. Construcția unui lanț maximal al mulțimii M se bazează pe următoarea idee. Elementul $f(0)$ va fi în lanț. Dacă $\beta \in \alpha$, atunci $f(\beta)$ va fi în lanț dacă și numai dacă el este comparabil cu elementele anterioare.

Formal, definim funcția $g : \alpha \rightarrow A$ prin

$$g(\beta) = \begin{cases} f(\beta), & \text{dacă } (\forall \gamma)(\gamma \in \beta \Rightarrow f(\beta) < g(\gamma) \vee g(\gamma) < f(\beta)) \\ f(0), & \text{altfel,} \end{cases}$$

pentru orice $\beta \in \alpha$.

Este ușor de văzut că mulțimea $Cod(g)$ este lanț maximal în M . \square

Unul dintre principiile de maximalitate frecvent întâlnite este *Lema lui Zorn*.

Teorema 1.5.4.3. (Lema lui Zorn (LZ) ³⁴⁾

Fie M o mpo. Dacă orice lanț al lui M are un majorant în M , atunci M are cel puțin un element maximal ³⁵.

Demonstrație. Fie $M = (A, \leq)$ o mpo ce satisfac ipoteza teoremei. De la Teorema 1.5.4.2 urmează că M are un lanț maximal, fie acesta L . Ipoteza teoremei conduce la existența unui majorant x al lui L , iar maximalitatea lanțului L ne arată că x este element maximal al lui M . \square

Lema lui Zorn și Principiul de maximalitate al lui Hausdorff pot fi întâlnite și sub următoarele forme echivalente:

(LZ') Fie M o mpo. Dacă orice lanț al lui M are un majorant în M , atunci, pentru orice $a \in A$, există un element maximal $b \in A$ astfel încât $a \leq b$;

(PMH') Orice lanț al unei mpo poate fi extins la un lanț maximal.

Este bine cunoscut că toate aceste principii (PBO, PMH, PMH', LZ, LZ') sunt echivalente între ele și echivalente cu Axioma alegerii (pentru detalii indicăm [207]).

1.5.5. Numere cardinale

Noțiunea de număr cardinal fost introdusă de Cantor în 1878 într-o variantă ce pornea de la premisa existenței mulțimii tuturor mulțimilor [24] (a se vedea și [61, 179]).

³⁴Acest principiu, într-o formă oarecum diferită, apare întâi la C. Kuratowski în 1922 [110]. M. Zorn este primul care, în 1935 [231], “afirmă” că acest principiu este echivalent cu Axioma alegerii, dar fără a da nici o demonstrație. Mai târziu, N. Bourbaki în 1939 [18] și J.W. Tukey în 1940 [214] s-au referit la acest principiu ca fiind “Lema lui Zorn”, ceea ce a făcut ca acest rezultat să fie cunoscut ulterior sub această denumire, dar poate fi întâlnit însă și ca “Lema Kuratowski-Zorn” [102].

³⁵Ipoteza teoremei se cere a fi satisfăcută pentru orice lanț, deci și pentru lanțul vid. În acest caz, orice element al mulțimii M este majorant al lanțului vid, și astfel M trebuie să fie nevidă.

Într-un astfel de caz, echipotența devine relație de echivalență pe această mulțime, și atunci cardinalul mulțimii A este definit ca fiind clasa de echivalență a lui A indușă de echipotență³⁶. Așa cum am văzut, teoria creată de Cantor este contradictorie, iar în sistemul ZFC nu există mulțimea tuturor mulțimilor. Ca urmare, noțiunea de cardinal trebuie introdusă în cu totul altă variantă. Trebuie să remarcăm însă că oricum am introduce noțiunea de cardinal, aceasta poate fi considerată corespunzătoare noțiunii intuitive de cardinal dacă putem demonstra că, pentru orice două mulțimi A și B , are loc

$$(*) \quad A \sim B \Leftrightarrow |A| = |B|$$

($|A|$ denotă cardinalul mulțimii A).

În sistemul ZFC noțiunea de cardinal poate fi introdusă în cel puțin două moduri. O primă variantă constă în considerarea noțiunii de cardinal ca noțiune primitivă, așa cum este și apartenența, urmată de introducerea unei noi axiome care să asigure $(*)$ (și care poate fi chiar $(*)$)³⁷. O a doua variantă se bazează pe noțiunea de ordinal în conjuncție cu Axioma alegerii (varianta von Neumann din 1928 [158]): cardinalul mulțimii x este definit ca fiind cel mai mic ordinal echipotent cu x ; Axioma alegerii va asigura faptul că orice mulțime are un (unic) cardinal.

În cele ce urmează vom aborda cardinalii prin cea de a două variantă, sub incidență Axiomei alegerii (echivalent, a Principiului bunei ordonări). Vom începe secțiunea cu o lemă cu caracter tehnic ce va asigura consistența definiției noțiunii de cardinal.

Lema 1.5.5.1. Pentru orice mulțime A există un unic ordinal α echipotent cu A , ce nu este echipotent cu nici un alt ordinal β mai mic decât el.

Demonstrație. Corolarul 1.5.4.1 asigură existența unui ordinal γ echipotent cu A . Mulțimea

$$A(\gamma) = \{\beta \mid \beta \text{ ordinal} \wedge \beta \leq \gamma \wedge \beta \sim A\}$$

este nevidă și are cel mai mic element, fie acesta α (Corolarul 1.5.2.1(3)). Este ușor de văzut că α este unicul ordinal ce satisfac lema. \square

Definiția 1.5.5.1.

- (1) Numim *numărul cardinal* al mulțimii A sau *cardinalul* mulțimii A , cel mai mic ordinal α echipotent cu A .
- (2) Numim *număr cardinal* sau *cardinal*, orice ordinal ce este număr cardinal al unei mulțimi.

³⁶Cantor a utilizat notația \bar{A} pentru cardinalul mulțimii A pentru a indica două nivele de abstractizare. Primul indică o abstractizare față de natura particulară a elementelor, în timp ce al doilea față de ordinea lor.

³⁷O astfel de abordare apare pentru prima dată la A. Tarski [202], și multe dintre tratatele existente asupra teoriei mulțimilor încep studiul cardinalilor într-o astfel de variantă. Avantajul constă în simplitatea noțiunilor implicate și ușurința în manipulare, permitând totuși dezvoltarea aritmeticii elementare a cardinalilor. O astfel de abordare nu utilizează Axioma alegerii, dar pe baza celorlalte axiome ale teoriei mulțimilor nu se poate demonstra că numărul cardinal al unei mulțimi este o mulțime.

Cardinalul mulțimii A , a cărui existență și unicitate este asigurată de Lema 1.5.5.1, va fi notat prin $|A|$.

Propoziția 1.5.5.1. Fie α un ordinal. Atunci următoarele afirmații sunt echivalente:

- (1) α este cardinal;
- (2) α nu este echipotent cu nici un ordinal $\beta < \alpha$ ³⁸;
- (3) α este cardinalul mulțimii α (adică, $\alpha = |\alpha|$).

Demonstrație. Dacă α este cardinal, atunci există o mulțime A astfel încât α este cel mai mic ordinal echipotent cu A . De aici urmează că α nu poate fi echipotent cu nici un ordinal $\beta < \alpha$. Deci (1) implică (2).

Dacă presupunem (2), atunci relația $\alpha \sim \alpha$ conduce la $\alpha = |\alpha|$. Deci, (2) implică (3).

Faptul că (3) implică (1) urmează direct de la Definiția 1.5.5.1(2). \square

Observația 1.5.5.1. ω este număr cardinal, dar $\omega + 1$ nu este deoarece $\omega \sim \omega + 1$ și $\omega < \omega + 1$. Similar, $\omega + n$ nu este număr cardinal, pentru orice $n \in \mathbb{N}$ cu $n \geq 1$.

Propoziția 1.5.5.2. Au loc următoarele proprietăți:

- (1) $|\alpha| \leq \alpha$, pentru orice ordinal α ;
- (2) $|(|A|)| = |A|$, pentru orice mulțime A ;
- (3) $|n| = n$, pentru orice $n \in \mathbb{N}$. Deci orice număr natural este număr cardinal;
- (4) ω este număr cardinal.

Demonstrație. (1) urmează direct de la definirea noțiunii de cardinal, iar (2) de la Propoziția 1.5.5.1. Pentru (3) și (4) se folosește Propoziția 1.5.5.1 și faptul că nici un număr natural n și nici ω nu pot fi echipotente cu submulțimi proprii ale lor. \square

Următoarea propoziție, a cărei demonstrație este imediată, stabilește legătura dintre echipotența mulțimilor și egalitatea cardinalilor asociați.

Propoziția 1.5.5.3. Pentru orice două mulțimi A și B , $A \sim B$ dacă și numai dacă $|A| = |B|$.

Numerele cardinale sunt numere ordinate, și astfel putem considera relația “ $<$ ” pe ele. Echipotența mulțimilor (“ \sim ”) are drept corespondent egalitatea cardinalilor (“ $=$ ”), în timp ce existența unei injecții stricte între mulțimi (“ \prec ”) va avea drept corespondent inegalitatea cardinalilor (“ $<$ ”).

³⁸Ordinalii α ce nu sunt echipotenți cu ordinalii $\beta < \alpha$ sunt numiți *ordinali inițiali*. Ca urmare, această propoziție ne spune că orice număr cardinal este ordinal inițial și reciproc. Astfel, numerele cardinale pot fi definite ca fiind ordinali inițiali, fără a face apel la noțiunea de “cardinal al unei mulțimi”.

Ca și ordinalii, cardinalii pot fi clasificați în *cardinali finiți* și *cardinali infiniți*. Cei finiți sunt exact ordinalii finiți (numerele naturale). Există deci o mulțime a tuturor cardinalilor finiți, ea fiind ω . Mai mult, ω este cel mai mic cardinal infinit, nu orice ordinal infinit este și cardinal (a se vedea Observația 1.5.5.1) și nu există o mulțime a tuturor cardinalilor infiniți.

Propoziția 1.5.5.4. Fie A și B două mulțimi. Atunci $A \prec B$ dacă și numai dacă $|A| < |B|$.

Demonstrație. Dacă $A \prec B$, atunci $\neg(A \sim B)$ și există o injecție f de la A la B . Mulțimea $f(A)$ va fi submulțime proprie a mulțimii B . Are loc:

$$|A| = |f(A)| \leq |B|.$$

Ca urmare a faptului că A și B nu sunt echipotente, obținem că inegalitatea precedentă este strictă, adică $|A| < |B|$.

Reciproc, dacă $|A| < |B|$, atunci A și B nu pot fi echipotente. În plus, are loc $A \sim |A| \subset |B| \sim B$, adică $A \prec B$. \square

Cei mai mic ordinal echivalent cu \mathbb{N} este ω . Ca urmare, ω este cardinalul mulțimii \mathbb{N} . În mod ușual, acest cardinal se renotează prin \aleph_0 . Mulțimile \mathbb{Z} și \mathbb{Q} au același cardinal, \aleph_0 , iar mulțimea \mathbb{R} are cardinalul 2^{\aleph_0} . *Ipoteza continuului* formulată de Cantor în 1878 [24] afirmă că între \aleph_0 și 2^{\aleph_0} nu mai există nici un alt cardinal. Acceptând această afirmație, 2^{\aleph_0} devine succesorul imediat al lui \aleph_0 , iar în acest context el se notează prin \aleph_1 .

Capitolul 2

Elemente de teoria numerelor

În acest capitol vom prezenta câteva elemente de bază de teoria numerelor, necesare înțelegерii corecte a conceptelor ce vor urma. Pentru detalii, acolo unde este cazul, cititorul interesat este îndrumat către monografii standard, cum ar fi [192, 191, 83, 176], sau către [210], unde se poate găsi o colecție de algoritmi de teoria numerelor, împreună cu studiile de complexitate aferente.

2.1. Divizibilitate. Numere prime

Notăm prin $|a|$ *modulul* sau valoarea absolută a numărului $a \in \mathbb{Z}$. Adică

$$|a| = \begin{cases} a, & \text{dacă } a \geq 0 \\ -a, & \text{altfel.} \end{cases}$$

Teorema 2.1.1. (Teorema împărțirii cu rest)

Pentru orice două numere întregi a și b cu $b \neq 0$, există $q, r \in \mathbb{Z}$ astfel încât $a = bq + r$ și $0 \leq r < |b|$. În plus, q și r sunt unicele numere întregi cu aceste proprietăți.

Demonstrație. Considerăm întâi cazul $b > 0$. Fie A mulțimea

$$A = \{a - bq \mid q \in \mathbb{Z}\} \cap \mathbb{N}.$$

A este nevidă, deoarece dacă $a < 0$, atunci $a - ba \in A$, iar dacă $a \geq 0$, atunci $a \in A$. Fiind submulțime de numere naturale, A va avea un cel mai mic element; fie acesta r . Atunci, r se poate scrie $r = a - bq$, unde $q \in \mathbb{Z}$. Vom arăta că q și r astfel determinate satisfac teorema. Prin definiție, $r \geq 0$. Vom arăta că $r < b$. Dacă presupunem, prin contradicție, că $r \geq b$, atunci numărul $r - b$ este în A , deoarece $r - b \geq 0$ și 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 și r se obține cu ușurință, prin contradicție. În

adevăr, să presupunem că există $q, r, q', r' \in \mathbf{Z}$ astfel încât $a = bq + r$, $a = bq' + r'$, $0 \leq r < b$ și $0 \leq r' < b$. Dacă $q = q'$ ($r = r'$), atunci urmează imediat că $r = r'$ ($q = q'$). Ca urmare, presupunem că $q \neq q'$ și $r \neq r'$. Fie, de exemplu, $q < q'$. Atunci relația $bq + r = bq' + r'$ conduce la

$$r' = r - b(q' - q).$$

Cum $r < b$ și $q' - q > 0$, obținem $r' < 0$, ceea ce reprezintă o contradicție. Deci $q = q'$ și $r = r'$.

Cazul $b < 0$ se obține din precedentul. În primul rând, observăm că $-b > 0$, și atunci există unice q' și r' cu $a = (-b)q' + r'$ și $0 \leq r' < (-b)$. Atunci, alegând $q = -q'$ și $r = r'$, deducem că $a = bq + r$ și $0 \leq r < |b|$. Unicitatea numerelor q și r se obține ca în cazul precedent. \square

Numerele q și r din Teorema împărțirii cu rest se numesc *câțul*, respectiv, *restul* împărțirii lui a la b . Ele se mai notează prin $a \text{ div } b$, respectiv, $a \text{ mod } b$.

Definiția 2.1.1. Relația 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 numește *relația 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 $a \nmid b$.

Observăm că dacă $a \neq 0$, atunci $a|b$ dacă și numai dacă $b \text{ mod } a = 0$. Dacă $a|b$ și a este diferit de $-1, 1, -b$ și de b , atunci vom spune că a este divizor propriu al lui b . Direct de la definiție obținem următoarea propoziție.

Propoziția 2.1.1. Fie $a, b, c \in \mathbf{Z}$. Atunci au loc următoarele proprietăți:

- (1) 0 divide doar 0 ;
- (2) a divide 0 și a ;
- (3) 1 divide a ;
- (4) $a|b$ dacă și numai dacă $|a| = b$;
- (5) dacă $a|b$ și $b|c$, atunci $a|c$;
- (6) dacă $a|b + c$ și $a|b$, atunci $a|c$;
- (7) dacă $a|b$, atunci $ac|bc$. Reciproc, dacă $c \neq 0$ și $ac|bc$, atunci $a|b$;
- (8) dacă $a|b$ și $a|c$, atunci $a|\beta b + \gamma c$, pentru orice $\beta, \gamma \in \mathbf{Z}$;

(9) dacă $a|b$ și $b \neq 0$, atunci $|a| \leq |b|$. Dacă în plus a este divizor propriu al lui b , atunci $1 < |a| < |b|$.

Definiția 2.1.2. Un număr natural $n \geq 2$ este numit *prim* dacă singurii lui divizori pozitivi sunt 1 și n .

Altfel spus, numerele prime sunt numere $n \geq 2$ ce nu au divizori proprii. Numerele $n \geq 2$ ce au divizori proprii sunt numite *composte* sau *compozite*.

Definiția 2.1.3. Fie $a_1, \dots, a_m \in \mathbf{Z}$, unde $m \geq 2$. Spunem că a_1, \dots, a_m sunt *prime între ele* sau *relativ prime* sau *coprime* dacă singurii divizori comuni ai acestor numere sunt 1 și -1 .

Vom nota $(a_1, \dots, a_m) = 1$ pentru a specifica faptul că a_1, \dots, a_m sunt relativ prime (această notație va fi justificată în secțiunea următoare). Observăm că 0 și 1 sunt relativ prime. De asemenea, orice două numere, dintre care unul este par și celălalt impar, sunt prime între ele.

Următoarea teoremă este crucială în stabilirea multor proprietăți în care intervine conceptul de numere relativ prime.

Teorema 2.1.2. Fie $m \geq 2$ și $a_1, \dots, a_m \in \mathbf{Z}$. Atunci $(a_1, \dots, a_m) = 1$ dacă și numai dacă există $\alpha_1, \dots, \alpha_m \in \mathbf{Z}$ astfel încât $\alpha_1 a_1 + \dots + \alpha_m a_m = 1$.

Demonstrație. Fie $a_1, \dots, a_m \in \mathbf{Z}$, unde $m \geq 2$.

Dacă presupunem că există $\alpha_1, \dots, \alpha_m \in \mathbf{Z}$ astfel încât $\alpha_1 a_1 + \dots + \alpha_m a_m = 1$, atunci a_1, \dots, a_m nu pot avea un divizor comun d diferit de 1 și -1 , deoarece atunci d ar divide suma $\alpha_1 a_1 + \dots + \alpha_m a_m$, și deci și pe 1 . Ca urmare, a_1, \dots, a_m sunt relativ prime.

Reciproc, presupunem că a_1, \dots, a_m sunt relativ prime. Considerăm mulțimea

$$A = \{\alpha_1 a_1 + \dots + \alpha_m a_m \mid \alpha_1, \dots, \alpha_m \in \mathbf{Z}\} \cap \mathbf{N}.$$

Această mulțime este nevidă și conține elemente diferite de 0 (aceasta rezultă cu ușurință considerând, de exemplu, $\alpha_i = a_i$, pentru orice i , și 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 $d = \alpha_1 a_1 + \dots + \alpha_m a_m$. Vom arăta că $d|a_i$ pentru orice i , ceea ce va implica $d = 1$, încheind demonstrația teoremei.

Fie $1 \leq i \leq m$. În baza teoremei împărțirii cu rest, există unice q_i și r_i astfel încât

$$a_i = dq_i + r_i \quad \text{și} \quad 0 \leq r_i < d.$$

Atunci:

$$\begin{aligned} r_i &= a_i - dq_i \\ &= a_i - (\alpha_1 a_1 + \dots + \alpha_m a_m) q_i \\ &= (1 - q_i \alpha_i) a_i + \sum_{j \neq i} (-q_i \alpha_j) a_j \\ &\geq 0, \end{aligned}$$

ceea ce arată că $r_i \in A$. Conform alegерii lui d și a faptului că $r_i < d$, urmează că $r_i = 0$. Aceasta conduce însă la $d|a_i$. Ca urmare, $d = 1$. \square

Corolarul 2.1.1. Fie $a_1, \dots, a_m, b \in \mathbf{Z}$, unde $m \geq 2$. Dacă $(b, a_i) = 1$, pentru orice $1 \leq i \leq m$, atunci $(b, a_1 \cdots a_m) = 1$.

Demonstrație. Vom demonstra corolarul pentru $m = 2$, cazul general obținându-se prin simplă inducție.

Conform Teoremei 2.1.2, există $\alpha_1, \alpha_2, \beta_1$ și β_2 astfel încât $\alpha_1 a_1 + \beta_1 b = 1$ și $\alpha_2 a_2 + \beta_2 b = 1$. 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 $(b, a_1 a_2) = 1$. \square

Corolarul 2.1.2. Fie $a_1, \dots, a_m, b \in \mathbf{Z}$, unde $m \geq 2$. Dacă numerele a_1, \dots, a_m sunt prime între ele două câte două și fiecare din ele divide b , atunci produsul lor divide b .

Demonstrație. Ca și în cazul corolarului precedent vom face demonstrația doar pentru $m = 2$.

Deoarece $(a_1, a_2) = 1$, există α_1 și α_2 astfel încât $\alpha_1 a_1 + \alpha_2 a_2 = 1$, iar de la $a_1 | b$ și $a_2 | b$ urmează că există β_1 și 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 $a_1 a_2 | b$.

Pentru completarea inducție va fi necesară utilizarea Corolarului 2.1.1. \square

Corolarul 2.1.3. Fie $a_1, \dots, a_m, b \in \mathbf{Z}$, unde $m \geq 2$. Dacă b este prim cu a_1 și divide produsul $a_1 \cdots a_m$, atunci b divide produsul $a_2 \cdots a_m$.

Demonstrație. Deoarece $(b, a_1) = 1$, urmează că există numerele α și 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 $b | a_2 \cdots a_m$. \square

Corolarul 2.1.4. Fie $a_1, \dots, a_m, p \in \mathbf{Z}$, unde $m \geq 2$. Dacă p este prim și divide produsul $a_1 \cdots a_m$, atunci există i astfel încât p divide a_i .

Demonstrație. Presupunem, prin contradicție, că $p \nmid a_i$, pentru orice i . Atunci, p este prim cu oricare din numerele a_i , iar de la Corolarul 2.1.1 obținem că are loc $(p, a_1 \cdots a_m) = 1$, ceea ce contrazice $p | a_1 \cdots a_m$. \square

Fie $n \geq 2$ un număr natural. Numim *descompunere* a lui n orice secvență finită de numere naturale

$$n_1, \dots, n_k \quad (k \geq 1)$$

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 secvenței face, în general, ca descompunerea să nu fie unică. Însă astfel de permute sunt irelevante și, ca urmare, prin descompunere a numărului natural $n \geq 2$ vom înțelege orice secvență de perechi de numere naturale

$$(n_1, e_1), \dots, (n_k, e_k) \quad (k \geq 1)$$

astfel încât:

- $2 \leq n_1 < \dots < n_k$;
- $e_i > 0$, 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 menționarea numărului k este irelevantă sau se subîntelege din context).

Cu această nouă definiție a descompunerii putem vorbi de descompuneri distincte ale aceluiași număr ca fiind descompuneri pentru care secvențele corespunzătoare nu coincid. De exemplu, $20 = 4 \cdot 5$ și $20 = 2^2 \cdot 5$ sunt descompuneri distincte ale lui 20. Dacă în descompunerea de mai sus numerele n_1, \dots, n_k sunt prime, atunci descompunerea lui n va fi numită *descompunere în factori primi*.

Are loc:

Teorema 2.1.3. (Teorema fundamentală a aritmeticii)

Orice număr natural $n \geq 2$ poate fi descompus în mod unic în factori primi (unicitatea fiind înțeleasă așa cum a fost specificat mai sus).

Demonstrație. Existența unei descompuneri în factori primi a oricărui număr natural $n \geq 2$ se obține cu ușurință prin inducție, luând în calcul cele două posibilități asupra lui n : n este prim sau n nu este prim. În 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 inducțivă.

Pentru unicitate vom face apel din nou la inducție și, în plus, la Corolarul 2.1.4. Să presupunem că n admite două descompuneri în factori primi, $n = p_1^{e_1} \cdots p_s^{e_s}$ și $n = q_1^{g_1} \cdots q_t^{g_t}$. Dacă $\sum_{i=1}^s e_i = \sum_{i=1}^t g_i = 1$, atunci obținem imediat că $p_1 = q_1$. Altfel, dacă de exemplu $\sum_{i=1}^t g_i > 1$, relația $p_1|n = q_1^{g_1} \cdots q_t^{g_t}$ conduce la existența unui i astfel încât $p_1|q_i$ (Corolarul 2.1.4). Dar aceasta este posibil numai dacă $p_1 = q_i$. Simplificând cele două descompuneri ale lui n , prima prin p_1 și a doua prin q_i , obținem un nou număr $n' < n$ și două descompuneri ale lui, pentru care putem aplica ipoteza inductivă. \square

Teorema 2.1.4. Există o infinitate de numere prime.

Demonstrație. Presupunem, prin contradicție, că există doar un număr finit de numere prime, fie acestea p_1, \dots, 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, \dots, p_n . Atunci, în baza Teoremei 2.1.3, el este divizibil prin unul din aceste numere. Să presupunem că $a = p_i d$, unde $1 \leq i \leq n$ și $d \geq 2$. 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; am ajuns astfel la o contradicție. \square

Fie p_n al n -lea număr prim, pentru orice $n \geq 1$ ($p_1 = 2$). Deoarece nu se cunoaște o formulă de determinare efectivă a numărului p_n , studiul distribuției numerelor prime joacă un rol foarte important în teoria numerelor. Prin *distribuția numerelor prime* înțelegem, intuitiv, modul în care aceste numere sunt repartizate pe axa numerelor naturale. În principal, studiul distribuției numerelor prime se face prin intermediul funcției π definită pentru orice număr natural $n \geq 2$ prin

$$\pi(n) = |\{p \in \mathbb{N} | p \leq n \wedge p \text{ prim}\}|.$$

Următoarea teoremă, “intuită” de Gauss în 1801¹ dar demonstrată abia în 1896 de matematicianul francez Jacques Hadamard și, independent, de matematicianul belgian Charles-Jean de la Vallée-Poussin², estimează această funcție prin intermediul funcției $n/\ln n$ definită pentru orice număr natural $n \geq 2$ (prin \ln s-a notat funcția logaritm natural). Demonstrația ei depășește cu mult cadrul lucrării noastre. Pentru detalii cititorul este îndrumat către [192].

Teorema 2.1.5. (Teorema numerelor prime)

$$\lim_{n \rightarrow \infty} \frac{\pi(n)}{\frac{n}{\ln n}} = 1.$$

¹Mentionată în cartea sa “Disquisitiones Arithmeticae” publicată în 1801 ([66] este o traducere revizuită a acesteia în limba engleză). În 1999, editura Amarcord din Timișoara a publicat o traducere a acestei cărți sub titlul “Cercetări aritmetice”.

²O demonstrație mai simplă a fost propusă de Landau în 1903.

Vom mai scrie $\pi(n) \sim \frac{n}{\ln n}$ și vom spune că funcțiile $\pi(n)$ și $\frac{n}{\ln n}$ sunt *echivalente asimptotic*³. Tabelul de mai jos prezintă câteva valori ale funcției 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	50847534

Corolarul 2.1.5. $\lim_{n \rightarrow \infty} \frac{\pi(n)}{n \ln n} = 1$.

Acest corolar ne spune că putem aproxima p_n prin $n \ln n$, pentru n suficient de mare⁴.

Teorema 2.1.5 este de importanță uriașă în studiul numerelor prime oferind o aproximare asimptotică pentru $\pi(n)$. Ulterior, Rosser și Schoenfeld [178] au rafinat acest rezultat obținând aproximări mai precise. Următoarele două teoreme sunt datorate acestora.

Teorema 2.1.6. 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 2.1.7. Pentru orice număr natural $n \geq 17$ are loc

$$\pi(n) > \frac{n}{\ln n},$$

și pentru orice $n \geq 2$ are loc

$$\pi(n) < 1.25506 \frac{n}{\ln n}.$$

Cititorul poate compara cele două aproximări și utiliza, de la caz la caz, pe cea mai bună. Ca o simplă aplicație a acestor rezultate, ne propunem să estimăm numărul de numere prime cu 100 de cifre. Pornind de la observația că 10^{100} și 10^{99} nu sunt numere prime, putem estima numărul de numere prime cu 100 de cifre prin

$$\begin{aligned} \pi(10^{100}) - \pi(10^{99}) &\approx \frac{10^{100}}{100 \ln 10} - \frac{10^{99}}{99 \ln 10} \\ &= \frac{10^{99}}{\ln 10} \left(\frac{1}{10} - \frac{1}{99} \right) \\ &> 0.39 \cdot 10^{98} \\ &\approx 4 \cdot 10^{97} \end{aligned}$$

(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 fizieni ca fiind între 10^{79} și 10^{81} .

³A nu se confunda cu notația “ $A \sim B$ ” utilizată pentru a desemna echipotența mulțimilor A și B .

⁴În 1939, J.B. Rosser [177] a stabilit inegalitatea $p_n > n \ln n$, pentru orice $n \geq 1$.

2.2. Cel mai mare divizor comun

Lema 2.2.1. Fie a_1, \dots, 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 $1 \leq i \leq m$.

Demonstrație. Fie D_i mulțimea tuturor divizorilor numărului a_i , $1 \leq i \leq m$. Atunci mulțimea $\bigcap_{i=1}^m D_i$ este nevidă (conține cel puțin pe 1) și finită (conform ipotezei, cel puțin o mulțime D_i este finită). Cel mai mare element al acestei mulțimi satisface lema. \square

Numărul d din Lema 2.2.1 se numește *cel mai mare divizor comun* al numerelor a_1, \dots, a_m . El se notează prin $cmmdc(a_1, \dots, a_m)$ sau, atunci când nu există pericol de confuzie, prin (a_1, \dots, a_m) . Putem spune că numerele a_1, \dots, a_m sunt relativ prime dacă și numai dacă $(a_1, \dots, a_m) = 1$, ceea ce justifică notația adoptată în secțiunea anterioară.

Propoziția 2.2.1. Fie a_1, \dots, a_m numere întregi nu toate 0, unde $m \geq 2$. Atunci:

- (1) $(0, a_1, \dots, a_m) = (a_1, \dots, a_m)$;
- (2) $(0, a_1) = |a_1|$, cu condiția $a_1 \neq 0$;
- (3) $(a_1, a_2) = (a_2, a_1 \bmod a_2)$, cu condiția $a_2 \neq 0$.

Demonstrație. (1) și (2) urmează imediat de la definiții.

Pentru (3), dacă scriem $a_1 = a_2q + r$ conform teoremei împărțirii cu rest, unde $0 \leq r < a_2$, atunci observăm că orice divizor comun al numerelor a_1 și a_2 este divizor comun al numerelor a_2 și r , și reciproc. Ca urmare, $(a_1, a_2) = (a_2, r)$. \square

Teorema 2.2.1. Fie a_1, \dots, a_m numere întregi nu toate 0, unde $m \geq 2$. Atunci există numerele întregi $\alpha_1, \dots, \alpha_m$ astfel încât

$$(a_1, \dots, a_m) = \alpha_1 a_1 + \dots + \alpha_m a_m.$$

Demonstrație. Dacă $d = (a_1, \dots, a_m)$, atunci există a'_1, \dots, a'_m astfel încât $a_i = da'_i$, pentru orice i . În plus, $(a'_1, \dots, a'_m) = 1$. Afirmația din teoremă se obține atunci cu ușurință de la Teorema 2.1.2. \square

Corolarul 2.2.1. Fie a_1, \dots, 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, \dots, a_m dacă și numai dacă au loc următoarele proprietăți:

- (i) $d|a_i$, pentru orice $1 \leq i \leq m$;
- (ii) $(\forall d' \in \mathbf{N})((\forall 1 \leq i \leq m)(d'|a_i) \Rightarrow d'|d)$.

Demonstrație. Dacă $d = (a_1, \dots, a_m)$, atunci are loc (i). În plus, d se poate scrie în forma $d = \alpha_1 a_1 + \dots + \alpha_m a_m$, unde $\alpha_1, \dots, \alpha_m \in \mathbf{Z}$. Atunci orice divizor $d' \in \mathbf{N}$ al numerelor a_i va fi divizor și al lui d . Deci are loc (ii).

Reciproc, dacă d este un număr natural ce satisfacă (i) și (ii), orice alt divizor comun $d' \in \mathbf{N}$ al numerelor a_i va satisface $d' \leq d$ (de la (ii) și Propoziția 2.1.1(9)). Deci $d = (a_1, \dots, a_m)$. \square

Corolarul 2.2.1 ne spune că proprietatea de a fi maximal în raport cu divizibilitatea este aceeași cu cea de a fi maximal în raport cu ordinea uzuală pe numere naturale, ambele considerate între divizorii comuni ai numerelor a_1, \dots, a_m nu toate 0.

Se poate formula și o reciprocă a Teoremei 2.2.1, astfel:

“dacă există numerele întregi $\alpha_1, \dots, \alpha_m$ astfel încât $\alpha_1 a_1 + \dots + \alpha_m a_m$ este număr natural și $\alpha_1 a_1 + \dots + \alpha_m a_m | a_i$, pentru orice i , atunci $(a_1, \dots, a_m) = \alpha_1 a_1 + \dots + \alpha_m a_m$.”

În adevăr, orice divizor comun al numerelor a_1, \dots, a_m este divizor al oricărei sume $\alpha_1 a_1 + \dots + \alpha_m a_m$. Dacă o astfel de sumă este divizor al numerelor a_1, \dots, a_m , urmează că ea este cel mai mare divizor comun al lor (în baza Corolarului 2.2.1).

Corolarul 2.2.2. Fie a_1, \dots, a_m numere întregi nu toate 0, unde $m \geq 2$. Atunci, pentru orice $b \in \mathbf{Z}$, ecuația

$$a_1 x_1 + \dots + a_m x_m = b,$$

în necunoscutele x_1, \dots, x_m , are soluții în \mathbf{Z} dacă și numai dacă $(a_1, \dots, a_m) | b$.

Demonstrație. Dacă ecuația $a_1 x_1 + \dots + a_m x_m = b$ are soluții în \mathbf{Z} , fie $\alpha_1, \dots, \alpha_m$ o astfel de soluție, atunci orice divizor comun al numerelor a_1, \dots, a_m va fi divizor al numărului $a_1 \alpha_1 + \dots + a_m \alpha_m$, și deci al lui b . Ca urmare, $(a_1, \dots, a_m) | b$.

Reciproc, dacă $d = (a_1, \dots, a_m) | b$, atunci există numerele k și $\alpha_1, \dots, \alpha_m$ astfel încât $b = kd$ și $d = a_1 \alpha_1 + \dots + a_m \alpha_m$. Este clar atunci că $x_i = k \alpha_i$, pentru orice $1 \leq i \leq m$, este soluție a ecuației $a_1 x_1 + \dots + a_m x_m = b$. \square

Lema 2.2.2. Fie a_1, \dots, 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 $1 \leq i \leq m$.

Demonstrație. Aceasta este similară Lemei 2.2.1. \square

Numărul b din Lema 2.2.2 este multiplu comun al numerelor a_1, \dots, a_m și, exceptând pe 0 care este și el multiplu comun al acestor numere, b este cel mai mic număr natural cu această proprietate. Dacă unul din numerele a_1, \dots, a_m este 0, atunci 0 este unicul multiplu comun al acestora (deoarece 0 divide doar pe 0).

Ca urmare, vom defini *cel mai mic multiplu comun* al numerelor a_1, \dots, a_m ca fiind numărul b din Lema 2.2.2, dacă aceste numere sunt nenule, și 0, altfel. Cel mai mic multiplu comun al numerelor a_1, \dots, a_m se notează prin $cmmmc(a_1, \dots, a_n)$ sau, atunci când nu există pericol de confuzie, prin $[a_1, \dots, a_n]$.

Teorema 2.2.2. Fie a_1, \dots, a_m numere întregi, unde $m \geq 2$. Atunci un număr natural b este cel mai mic multiplu comun al numerelor a_1, \dots, a_m dacă și numai dacă au loc următoarele proprietăți:

- (i) $a_i|b$, pentru orice $1 \leq i \leq m$;
- (ii) $(\forall b' \in \mathbb{N})((\forall 1 \leq i \leq m)(a_i|b') \Rightarrow b|b')$.

Demonstrație. Teorema se verifică cu ușurință dacă cel puțin unul din numerele a_1, \dots, a_m este 0. Să presupunem în continuare că toate aceste numere sunt diferite de 0.

Fie $b = [a_1, \dots, a_m]$. Atunci $b > 0$ și 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 împărțirii cu rest deducem că există unice numerele q și r astfel încât $b' = bq + r$ și $0 \leq r < b$. Atunci $r = b' - bq$, de unde obținem 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, obținem $r = 0$, ceea ce ne arată că $b|b'$. Deci, are loc (ii).

Reciproc, presupunem că b este un număr natural ce satisfacă (i) și (ii). De la (ii) și Propoziția 2.1.1(9) urmează că b este nenul și este cel mai mic număr natural ce satisfacă (i). Deci, $b = [a_1, \dots, a_m]$. \square

Teorema 2.2.3. Fie a și b două numere naturale, nu ambele 0. Atunci are loc $ab = (a, b)[a, b]$.

Demonstrație. Dacă a sau b este 0, atunci teorema este trivial satisfăcută. Să presupunem că a și b sunt neneule. Fie $d = (a, b)$. Atunci, există a_1 și b_1 astfel încât $a = da_1$, $b = db_1$ și $(a_1, b_1) = 1$. Vom arăta că $[a, b] = da_1b_1$, ceea ce va încheia demonstrația.

Observăm întâi că $a|da_1b_1$ și $b|da_1b_1$. Dacă c este un multiplu comun al numerelor a și b , atunci există x și y astfel încât $c = ax = da_1x$ și $c = by = db_1y$. Egalitatea $a_1x = b_1y$, combinată cu $(a_1, b_1) = 1$, conduce la $b_1|x$, de unde urmează că $da_1b_1|c$. Deci $[a, b] = da_1b_1$. \square

Ca o consecință imediată a acestei teoreme obținem:

Corolarul 2.2.3. Cel mai mic multiplu comun a două numere naturale relativ prime este egal cu produsul numerelor.

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 puțin unul nenul. Fie deci $a \geq b \geq 0$. Dacă $a = b$ sau $b = 0$, atunci $(a, b) = a$. Dacă $a > b > 0$ și scriem $a = bq + r$, unde $0 \leq r < b$, atunci $(a, b) = (b, r)$ (în baza Propoziției 2.2.1(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 împărțirilor succesive:

$$\begin{aligned} r_{-1} &= r_0q_1 + r_1, & 0 < r_1 < r_0 \\ r_0 &= r_1q_2 + r_2, & 0 < r_2 < r_1 \\ &\dots \\ r_{n-2} &= r_{n-1}q_n + r_n, & 0 < r_n < r_{n-1} \\ r_{n-1} &= r_nq_{n+1} + r_{n+1}, & r_{n+1} = 0, \end{aligned}$$

unde $r_{-1} = a$ și $r_0 = b$. Atunci, conform celor menționate mai sus, obținem:

$$(a, b) = (r_{-1}, r_0) = (r_0, r_1) = \dots = (r_n, r_{n+1}) = (r_n, 0) = r_n.$$

Să facem câteva observații importante asupra secvenței de împărțiri de mai sus:

- numărul de împărțiri 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 complexității algoritmului lui Euclid necesită estimarea lui n . În 1845, Gabriel Lamé a avut ideea de a compara resturile r_i cu termenii sirului 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 $r_n \geq 1 = F_2$ și

$$r_{n-1} = r_nq_{n+1} \geq 2r_n \geq 2 = F_3.$$

Dacă presupunem că $r_{i+2} \geq F_{n-i}$ și $r_{i+1} \geq F_{n-i+1}$, atunci

$$r_i = r_{i+1}q_{i+2} + r_{i+2} \geq r_{i+1} + r_{i+2} \geq F_{n-i+1} + F_{n-i} = F_{n-i+2},$$

pentru orice $i = n-2, \dots, 0, -1$. De aici obținem $b = r_0 \geq F_{n+2}$.

Fie $R = (1 + \sqrt{5})/2$. Prin simplă inducție matematică putem arăta că are loc $F_2 = R^0$ și $F_{i+2} > R^i$, pentru orice $i \geq 1$. Obținem 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 $\log_{10} R = 0.208 \dots > 1/5$. Dacă numărul b necesită k cifre în scriere zecimală, atunci $b < 10^k$. Combinând cu inegalitatea de mai sus obținem $n < 5k$, ceea ce conduce la $n + 1 \leq 5k$. Am obținut astfel:

⁵A fost prezentat de Euclid în volumul VI al lucrării lui, *Elements*.

Teorema 2.2.4. (G. Lamé, 1845)

Fie $a > b > 0$ numere naturale. Atunci numărul de împărțiri necesare algoritmului lui Euclid pentru determinarea celui mai mare divizor comun al numerelor a și b nu depășește de 5 ori numărul de cifre din scrierea zecimală a lui b .

Utilizând acum inegalitatea $a \geq F_{n+3} > R^{n+1}$, obținem $n + 1 < \log_R a$, ceea ce conduce la:

Teorema 2.2.5. 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 $\lfloor \log_R N \rfloor - 1$ pași.

Estimarea de mai sus a numărului de pași, atunci când avem de aplicat algoritmul lui Euclid, este suficient de bună prin aceea că aplicarea acestui algoritm numerelor F_{n+3} și F_{n+2} , ce satisfac $a \geq F_{n+3}$ și $b \geq F_{n+2}$, necesită de asemenea tot $n + 1$ pași de împărțire:

$$\begin{aligned} F_{n+3} &= 1 \cdot F_{n+2} + F_{n+1}, & 0 < F_{n+1} < F_{n+2} \\ F_{n+2} &= 1 \cdot F_{n+1} + F_n, & 0 < F_n < F_{n+1} \\ \dots \\ F_4 &= 1 \cdot F_3 + F_2, & 0 < F_2 < F_3 \\ F_3 &= 2 \cdot F_2. \end{aligned}$$

Algoritmul lui Euclid poate fi descris astfel:

Euclid

```
input:  $a, b \in \mathbf{Z}$  nu ambele 0;
output:  $(a, b)$ ;
begin
1. while  $b \neq 0$  do
   begin
2.      $r := a \text{ mod } b$ ;  $a := b$ ;  $b := r$ ;
   end
3.  $(a, b) := |a|$ 
end.
```

Așa cum am văzut, cel mai mare divizor comun a două numere a și b poate fi exprimat ca o combinație liniară a acestora, $(a, b) = \alpha a + \beta b$. Există multe situații în care suntem interesăți în a determina (algoritmic) o astfel de combinație liniară. Dacă analizăm secvența de împărțiri de mai sus, prin care se determină (a, b) , constatăm următoarele:

$$\begin{aligned} r_1 &= a - bq_1 &= 1 \cdot a + (-q_1) \cdot b \\ r_2 &= b - r_1 q_2 &= (-q_2) \cdot a + (1 + q_1 q_2) \cdot b \\ r_3 &= r_1 - r_2 q_3 &= (1 + q_2 q_3) \cdot a + (-q_1 - q_3 - q_1 q_2 q_3) \cdot b \\ \dots \end{aligned}$$

Adică, odată cu determinarea restului (la un pas), putem determina și o combinație liniară a acestuia (în funcție de a și b). Ceea ce ne rămâne de făcut este de a găsi o metodă elegantă de exprimare a unei astfel de combinații liniare a restului în funcție de combinațiile liniare de la pașii anteriori. Dacă fiecare element x ce intervine în secvența de împărțiri de mai sus îi asociem un vector $V_x = (\alpha, \beta)$ ce furnizează o combinație liniară (în funcție de a și b) a lui x , adică $x = \alpha a + \beta b$, atunci o combinație liniară a resturilor se poate determina prin:

$$\begin{array}{lll} V_a &= (1, 0) \\ V_b &= (0, 1) \\ 1. \quad a &= bq_1 + r_1 & V_{r_1} = V_a - q_1 V_b \\ 2. \quad b &= r_1 q_2 + r_2 & V_{r_2} = V_b - q_2 V_{r_1} \\ 3. \quad r_1 &= r_2 q_3 + r_3 & V_{r_3} = V_{r_1} - q_3 V_{r_2} \\ \dots & & \\ n. \quad r_{n-2} &= r_{n-1} q_n + r_n & V_{r_n} = V_{r_{n-2}} - q_n V_{r_{n-1}} \\ n+1. \quad r_{n-1} &= r_n q_{n+1} & \end{array}$$

În acest mod putem determina atât (a, b) cât și o combinație liniară (de a și b) a acestuia. Algoritmul pe care l-am obținut se numește *algoritmul extins al lui Euclid*. Corectitudinea lui se demonstrează imediat în baza a ceea ce a fost menționat mai sus, iar complexitatea acestuia este aceeași cu algoritmului lui Euclid. Mai precis, la fiecare pas, pe lângă o împărțire se fac două înmulțiri (complexitatea unei înmulțiri fiind aceeași cu unei împărțiri) și două scăderi (complexitatea unei scăderi fiind liniară în raport cu lungimea maximă a reprezentării binare a operanzilor).

Algoritmul extins al lui Euclid este următorul:

EuclidExt

```
input:  $a, b \in \mathbf{Z}$  nu ambele 0;
output:  $(a, b)$  și  $V = (\alpha, \beta)$  astfel încât  $(a, b) = \alpha a + \beta b$ ;
begin
1.  $V_0 := (1, 0)$ ;
2.  $V_1 := (0, 1)$ ;
3. while  $b \neq 0$  do
   begin
4.      $q := a \text{ div } b$ ;  $r := a \text{ mod } b$ ;  $a := b$ ;  $b := r$ ;
   end
5.      $V := V_0$ ;  $V_0 := V_1$ ;  $V_1 := V - qV_0$ ;
   end
6.  $(a, b) := |a|$ ;
7.  $V := V_0$ ;
end
```

Possibilitatea determinării algoritmice a unei combinații liniare a celui mai mare divizor comun a două numere a și b conduce la posibilitatea determinării algoritmice a unei soluții a ecuației $ax + by = c$, în ipoteza $(a, b)|c$ (dacă acastă relație nu

este satisfăcută, atunci ecuația nu are soluții). În adevăr, fie α și β astfel încât $\alpha a + \beta b = (a, b)$, și fie c' astfel încât $c = (a, b)c'$. Atunci

$$c'\alpha a + c'\beta b = (a, b)c' = c,$$

ceea ce ne arată că $x = c'\alpha$ și $y = c'\beta$ constituie o soluție a ecuației $ax + by = c$.

Algoritmul de determinare de soluții pentru astfel de ecuații este următorul.

Ecuatie_Diofantica

input: $a, b, c \in \mathbf{Z}$ astfel încât nu ambele a și b sunt 0;

output: "nu are soluții întregi", dacă $ax + by = c$ nu are soluții întregi, și o soluție, altfel;

```

begin
1.   calculează  $(a, b) = \alpha a + \beta b$  cu EuclidExt;
2.   if  $(a, b)|c$ 
        then begin
3.            $c' := c/(a, b);$ 
4.            $x := \alpha c'; y := \beta c';$ 
        end
5.   else "nu are soluții întregi"
end.

```

La pasul 1 în algoritmul Ecuatie_Diofantica se înțelege că se calculează (a, b) și o combinație liniară a acestuia. Complexitatea algoritmului Ecuatie_Diofantica este exact complexitatea algoritmului lui Euclid.

Analizând algoritmul lui Euclid observăm că fracția a/b , unde $b \neq 0$, poate fi scrisă astfel:

$$\frac{a}{b} = q_1 + \frac{r_1}{b} = q_1 + \frac{1}{\frac{b}{r_1}} = q_1 + \frac{1}{q_2 + \frac{r_2}{r_1}} = q_1 + \frac{1}{q_2 + \frac{1}{q_3 + \dots + \frac{1}{q_{k+1}}}}$$

Ultimul termen din acest sir de egalități poartă denumirea de *fracție continuă asociată* a/b și se mai notează prin $[q_1, \dots, q_{k+1}]$ (a nu se confunda cu cel mai mic multiplu comun).

Definiția 2.2.1. Se numește *fracție continuă finită* orice număr $[q_1, \dots, q_n]$ de forma

$$q_1 + \frac{1}{q_2 + \frac{1}{q_3 + \dots + \frac{1}{q_n}}},$$

unde q_1, \dots, q_n sunt numere reale ce satisfac $q_i > 0$, pentru orice $2 \leq i \leq n$.

Dacă o fracție continuă finită $[q_1, \dots, q_n]$ are proprietatea că $q_1, \dots, q_n \in \mathbf{Z}$, atunci ea se mai numește *fracție continuă finită simplă*.

Este ușor de văzut că orice fracție continuă finită simplă definește un număr rațional. Combinând aceasta cu observația de mai sus, conform căreia orice fracție a/b definește o fracție continuă finită simplă, obținem:

Teorema 2.2.6. Un număr este rațional dacă și numai dacă se poate reprezenta ca o fracție continuă finită simplă.

Ceea ce trebuie să remarcăm este că reprezentarea unui număr rațional ca fracție continuă finită simplă nu este unică. În adevăr, este ușor de vazut că are loc

$$[q_1, \dots, q_n] = \begin{cases} [q_1, \dots, q_{n-1}, q_n - 1, 1], & \text{dacă } q_n > 1, \\ [q_1, \dots, q_{n-1} + 1], & \text{dacă } q_n = 1, \end{cases}$$

ceea ce ne arată că orice număr rațional are măcar două reprezentări ca fracție continuă. Însă se poate ușor arăta că orice număr rațional are exact două reprezentări ca fracție continuă finită simplă, și acestea sunt de forma de mai sus.

Fie $[q_1, \dots, q_n]$ o fracție continuă finită simplă ce reprezintă numărul rațional a/b . Fracțiile continue finite simple $[q_1, \dots, q_i]$, unde $1 \leq i \leq n$, se numesc *convergențele fracției continue* $[q_1, \dots, q_n]$. Dacă notăm prin a_i/b_i numărul rațional ce are reprezentarea $[q_1, \dots, q_i]$ ca fracție continuă finită simplă, atunci obținem următoarele proprietăți.

Propoziția 2.2.2. Fie $[q_1, \dots, q_n]$ o fracție continuă finită simplă ce reprezintă fracția a/b .

(1) Numerele raționale a_i/b_i verifică relațiile de recurență:

- $a_1 = q_1$ și $b_1 = 1$;
- $a_2 = q_2 q_1 + 1$ și $b_2 = q_2$;
- $a_i = q_i a_{i-1} + a_{i-2}$ și $b_i = q_i b_{i-1} + b_{i-2}$, pentru orice $3 \leq i \leq n$.

(2) Au loc relațiile:

$$(2.1) \quad a_i b_{i-1} - a_{i-1} b_i = (-1)^{i-1}, \text{ pentru orice } 2 \leq i \leq n;$$

$$(2.2) \quad \frac{a_i}{b_i} - \frac{a_{i-1}}{b_{i-1}} = \frac{(-1)^{i-1}}{b_i b_{i-1}}, \text{ pentru orice } 2 \leq i \leq n;$$

$$(2.3) \quad \frac{a_i}{b_i} - \frac{a_{i-2}}{b_{i-2}} = \frac{(-1)^i q_i}{b_i b_{i-2}}, \text{ pentru orice } 3 \leq i \leq n.$$

$$(3) \quad (a_i, b_i) = 1, \text{ pentru orice } 1 \leq i \leq n.$$

$$(4) \quad \frac{a_1}{b_1} < \frac{a_3}{b_3} < \dots < \frac{a}{b} \leq \dots < \frac{a_4}{b_4} < \frac{a_2}{b_2}.$$

Demonstrație. (1) se obține prin inducție matematică, iar (2) urmează de la (1). Pentru (3) vom folosi (2.1).

Fie $d = (a_i, b_i)$, unde $i \geq 2$ (pentru $i = 1$, $b_1 = 1$, și deci $(a_1, b_1) = 1$). Atunci $d|a_i b_{i-1}$ și $d|a_{i-1} b_i$. Ca urmare, $d|a_i b_{i-1} - a_{i-1} b_i$, ceea ce conduce la $d|(-1)^{i-1}$. Deci $d = 1$.

Pentru (4) se folosește (2.2) și (2.3). \square

2.3. Congruențe

În multe situații suntem interesați în considerarea restului împărțirii unui număr întreg la un alt număr întreg numit *modul*. De exemplu, determinarea zilei săptămânii (luni, marți etc.) ce va fi pe o anumită dată a anului face apel la împărțirea zilelor în grupe de câte 7 și considerarea restului. Numărarea obiectelor unei mulțimi face iarăși apel la împărțirea acestora în grupe, cel mai adesea de câte 10, și apoi numărarea acestora (care, la rândul lor, pot fi numărate prin repetarea procedeului de împărțire în grupe). În secolul al IV-lea, autorul chinez Sun Tzu Suan Ching formula următoarea problemă:

“Avem un număr de obiecte, dar nu știm 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 ?”.

Probleme ca cele menționate mai sus fac apel la conceptul matematic de *congruență modulară*.

Definiția 2.3.1. Fie m un număr întreg. Relația binară $\equiv_m \subseteq \mathbf{Z} \times \mathbf{Z}$ data prin

$$a \equiv_m b \Leftrightarrow m|(a - b),$$

pentru orice $a, b \in \mathbf{Z}$, este numita *relația de congruență modulo m* sau *congruență modulo m*.

Atunci când $a \equiv_m b$ vom spune că a și b sunt *congruente modulo m* și vom mai nota aceasta prin $a \equiv b \text{ mod } m$.

Următoarele proprietăți pot fi obținute cu ușurință de la definiția congruențelor modulare și de la proprietățile relației de divizibilitate.

Propoziția 2.3.1. Fie a, b, c, d, m și m' numere întregi și $f : \mathbf{Z} \rightarrow \mathbf{Z}$ o funcție polinomială cu coeficienți întregi. Atunci au loc următoarele proprietăți:

(1) \equiv_m este relație de echivalență pe \mathbf{Z} ;

(2) dacă $m \neq 0$, atunci $a \equiv_m b$ dacă și numai dacă $a \text{ mod } m = b \text{ mod } m$;

- (3) dacă $a \equiv_m b$, atunci $(a, m) = (b, m)$;
- (4) (reguli de combinare)
 - dacă $a \equiv_m b$ și $c \equiv_m d$, atunci $a + c \equiv_m b + d$, $a - c \equiv_m b - d$, $ac \equiv_m bd$ și $f(a) \equiv_m f(b)$;
- (5) (reguli de simplificare)
 - (a) dacă $ac \equiv_{mc} bc$ și $c \neq 0$, atunci $a \equiv_m b$;
 - (b) dacă $ac \equiv_m bc$ și $d = (m, c)$, atunci $a \equiv_{m/d} b$;
 - (c) dacă $ac \equiv_m bc$ și $(m, c) = 1$, atunci $a \equiv_m b$;
- (6) (reguli de descompunere și de compunere)
 - (a) dacă $a \equiv_{mm'} b$, atunci $a \equiv_m b$ și $a \equiv_{m'} b$;
 - (b) dacă $a \equiv_m b$ și $a \equiv_{m'} b$, atunci $a \equiv_{[m, m']} b$;
 - (c) dacă $a \equiv_m b$, $a \equiv_{m'} b$ și $(m, m') = 1$, atunci $a \equiv_{mm'} b$.

Demonstrație. Vom demonstra doar (6b). Presupunem că au loc relațiile $a \equiv_m b$ și $a \equiv_{m'} b$. Prima relație conduce la $m|(a - b)$, iar a doua la $m'|(a - b)$. De la acestea, în baza Teoremei 2.2.2, urmează $[m, m']|(a - b)$. \square

Vom nota prin \mathbf{Z}_m mulțimea claselor de echivalență induse de \equiv_m (clasa de echivalență a lui $a \in \mathbf{Z}$ fiind notată prin $[a]_m$). Vom face în cele ce urmează câteva observații importante asupra acestor clase de echivalență:

- deoarece un număr întreg se divide la m dacă și numai dacă se divide la $-m$, deducem că relațiile de congruență modulo m și $-m$ coincid. Ca urmare, putem considera numai relații de congruență modulo m pentru care $m \geq 0$;
- în cazul $m = 0$, $a \equiv b \text{ mod } m$ dacă și numai dacă $a = b$. Deci orice element din \mathbf{Z} induce o clasă de echivalență formată doar din el;
- în cazul $m \geq 1$, mulțimea \mathbf{Z}_m are m elemente. În adevăr, oricare două numere distincte dintre numerele $0, \dots, m - 1$ nu sunt congruente modulo m deoarece diferența lor este diferită de 0 și strict mai mică, în valoare absolută, decât m . Ca urmare, numerele $0, \dots, m - 1$ sunt în clase de echivalență diferite. În plus, în fiecare clasă de echivalență indusă de un element $n \in \mathbf{Z}$ se găsește unul din cele m elemente de mai sus (în baza teoremei împărțirii cu rest).

Așadar, relația de congruență modulo m împarte mulțimea \mathbf{Z} în m clase de echivalență pentru care putem alege numerele $0, \dots, m - 1$ drept reprezentanți de clasă, acestea fiind exact resturile posibile ale împărțirii numerelor întregi la m . Din acest motiv, clasele de echivalență induse de \equiv_m mai sunt numite și *clase de echivalență modulo m* sau *clase de resturi modulo m*. Uneori ele mai sunt notate prin $0, \dots, m - 1$.

Pe mulțimea \mathbf{Z}_m introducem următoarele operații:

- \oplus , operație binară dată prin $[a]_m \oplus [b]_m = [a + b]_m$, pentru orice $a, b \in \mathbf{Z}$;
- $[0]_m$, operație 0-ară;
- \ominus , operație unară dată prin $\ominus[a]_m = [-a]_m$, pentru orice $a \in \mathbf{Z}$;
- \otimes , operație binară dată prin $[a]_m \otimes [b]_m = [ab]_m$, pentru orice $a, b \in \mathbf{Z}$;
- $[1]_m$, operație 0-ară.

Cu acestea, structura $(\mathbf{Z}_m, \oplus, \ominus, [0]_m)$ devine grup ciclic comutativ, iar 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 (a se vedea Capitolul 5) în sensul că are doar un singur element, $[0]_1 = [1]_1$, care este element neutru atât pentru operația notată aditiv cât și pentru operația notată multiplicativ.

Scăderea în inelul \mathbf{Z}_m se definește prin $[a]_m \ominus (\ominus[b]_m)$, 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 împărțirii cu rest, orice număr întreg a se poate scrie în forma $a = qm + r$, unde $q, r \in \mathbf{Z}$ și $0 \leq r < m$. Determinarea lui r va fi numită *reducere modulo m* a lui a sau, în general, *reducere modulară* a lui a .

Inelul \mathbf{Z}_0 este izomorf cu inelul \mathbf{Z} și, ca urmare, vom evita cazul $m = 0$ orientând studiile cu precădere asupra lui \mathbf{Z}_m cu $m \geq 1$.

Considerând mulțimea $\mathbf{Z}'_m = \{0, \dots, m-1\}$ înzestrată cu operațiile

- $+'$, dată prin $a +' b = (a + b) \text{ mod } m$, pentru orice $a, b \in \mathbf{Z}'_m$;
- 0, ca operație 0-ară;
- $-'$, dată prin $-'a = m - a \text{ mod } m$, pentru orice $a \in \mathbf{Z}'_m$;
- \cdot' , dată prin $a \cdot' b = a \cdot b \text{ mod } m$, pentru orice $a, b \in \mathbf{Z}'_m$;
- 1, ca operație 0-ară (0 și 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 $\mathbf{Z}_m = \{0, \dots, m-1\}$ și operațiile \oplus , \ominus și \otimes prin $+$, $-$, respectiv, \cdot (aceasta din urmă fiind omisă, atunci când nu există pericol de confuzie). Atragem însă atenția că aceste operații, privite în \mathbf{Z}_m , sunt echivalente cu corespondentele lor în \mathbf{Z} la care se adaugă și reducerea modulară (a se vedea definițiile operațiilor $+'$, $-'$ și \cdot').

Mentionăm că nu orice element $a \in \mathbf{Z}_m$ are un invers multiplicativ⁶ (de exemplu, $2 \in \mathbf{Z}_6$), dar atunci când există el este unic. De asemenea, \mathbf{Z}_m poate avea divizori ai lui 0⁷ (de exemplu, $2 \cdot 3 = 0$ în \mathbf{Z}_6).

⁶Învers relativ la operația de înmulțire.

⁷Elemente diferite de 0 al căror produs este 0.

Să vedem ce condiții trebuie să satisfacă un element $a \in \mathbf{Z}_m$ pentru a avea un invers multiplicativ. Observăm că au loc echivalențele

$$(\exists x \in \mathbf{Z}_m)(ax \equiv 1 \text{ mod } m) \Leftrightarrow (\exists x, y \in \mathbf{Z})(ax - my = 1) \Leftrightarrow (a, m) = 1$$

(ultima echivalență urmează de la Corolarul 2.2.2).

Ca urmare, elementele din \mathbf{Z}_m care admit inversi multiplicativi sunt exact acele elemente care sunt prime cu m . Fie \mathbf{Z}_m^* mulțimea 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 unităților* inelului \mathbf{Z}_m . Este ușor de văzut că $\mathbf{Z}_1^* = \{0\} = \mathbf{Z}_1$, caz în care $1 = 0$ (clasele de echivalență modulo 1 induse de 0 și 1 coincid)⁸, iar grupul unităților inelului \mathbf{Z}_{26} are 12 elemente; acestea și inversele lor sunt următoarele:

$$1^{-1} = 1, 3^{-1} = 9, 5^{-1} = 21, 7^{-1} = 15, 11^{-1} = 19, 17^{-1} = 23, 25^{-1} = 25.$$

Algoritmul extins al lui Euclid ne permite determinarea inversului multiplicativ modulo m al lui a . În adevăr, dacă $(a, m) = 1$, atunci, cu ajutorul algoritmului extins al lui Euclid, putem determina α și β astfel încât $\alpha a + \beta m = 1$. De aici urmează cu ușurință că $\alpha \text{ mod } m$ este inversul multiplicativ modulo m al lui a .

Invers_Multiplicativ

```
input:   m ≥ 1 și a ∈ Zm;
output: a⁻¹ modulo m, dacă (a, m) = 1, și "a⁻¹ nu există", altfel;
begin
1.   calculează (a, m) = aa + βm cu EuclidExt;
2.   if (a, m) = 1 then a⁻¹ := α mod m else "a⁻¹ nu există"
end.
```

La pasul 1 în algoritmul Invers_Multiplicativ se înțelege că se calculează (a, m) și o combinație liniară a acestuia. Complexitatea acestui algoritm este exact complexitatea algoritmului lui Euclid.

2.4. Funcția lui Euler

Reamintim că $\mathbf{Z}_m^* = \{a \in \mathbf{Z}_m | (a, m) = 1\}$, pentru orice $m \geq 1$. Funcția ϕ ce asociază fiecarui număr $m \geq 1$ cardinalul mulțimii \mathbf{Z}_m^* este numită *funcția lui Euler*. Vom

⁸În multe tratate de teoria numerelor, \mathbf{Z}_m^* se introduce ca fiind mulțimea numerelor strict pozitive ce nu depășesc m și care sunt prime cu m . În această varianta, \mathbf{Z}_1^* este mulțimea formată doar din 1 (fiind aceeași cu \mathbf{Z}_2^*); pentru $m > 1$, această definiție produce aceeași mulțime \mathbf{Z}_m^* ca și definiția mai sus adoptată. Cum diferența dintre aceste două abordări diferă doar din punct de vedere al mulțimii \mathbf{Z}_1^* , vom prefera să mergem pe varianta deja adoptată prin care \mathbf{Z}_m^* este grupul unităților inelului \mathbf{Z}_m , pentru orice $m \geq 1$.

fi interesați în cele ce urmează de determinarea unei formule de evaluare a acestei funcții. Observăm întâi că $\phi(1) = 1$ și $\phi(p) = p - 1$, pentru orice număr prim p .

Teorema 2.4.1. Fie $m, m' \geq 1$ numere prime între ele și $f : \mathbf{Z}_m \times \mathbf{Z}_{m'} \rightarrow \mathbf{Z}_{mm'}$ data prin $f(a, a') = (ma' + m'a) \text{ mod } mm'$, pentru orice $a \in \mathbf{Z}_m$ și $a' \in \mathbf{Z}_{m'}$. Atunci:

- (1) funcția f este bijectie;
- (2) restricția funcției f la $\mathbf{Z}_m^* \times \mathbf{Z}_{m'}^*$ stabilește o bijecție între această mulțime și $\mathbf{Z}_{mm'}^*$.

Demonstrație. (1) Mulțimile $\mathbf{Z}_m \times \mathbf{Z}_{m'}$ și $\mathbf{Z}_{mm'}$ sunt același număr de elemente, și anume mm' . Ca urmare, este suficient de arătat că f este funcție 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 obținem $m(a' - b') \equiv_{mm'} m'(b - a)$ ce conduce la (Propoziția 2.3.1(6a)) $m(a' - b') \equiv_m m'(b - a)$ și $m(a' - b') \equiv_{m'} m'(b - a)$. Prima relație combinată cu $(m, m') = 1$ produce $a \equiv_m b$. Cum $a, b \in \mathbf{Z}_m$, urmează $a = b$. Similar, a doua relație produce $a' = b'$. Deci, f este injectivă.

(2) În baza rezultatului de la (1) este suficient de arătat că are loc:

- (a) dacă $a \in \mathbf{Z}_m^*$ și $a' \in \mathbf{Z}_{m'}^*$, atunci $(ma' + m'a) \text{ mod } mm'$ este în $\mathbf{Z}_{mm'}^*$;
- (b) orice element din $\mathbf{Z}_{mm'}^*$ este de forma $(ma' + m'a) \text{ mod } mm'$, unde $a \in \mathbf{Z}_m^*$ și $a' \in \mathbf{Z}_{m'}^*$.

Vom demonstra (a) prin contradicție. Adică vom presupune că $ma' + m'a \text{ mod } mm'$ și mm' nu sunt relativ prime. Deci ele vor avea un divizor comun $d > 1$. Este clar că d este divizor comun și pentru numerele $ma' + m'a$ și mm' .

Cum d divide mm' și $(m, m') = 1$, 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 $d|ma' + m'a$, obținem $d|m'a$ care, combinată cu $(d, m') = 1$, conduce la $d|a$. Dar atunci $(a, m) > 1$, ceea ce este o contradicție. Deci $ma' + m'a \text{ mod } mm' \in \mathbf{Z}_{mm'}^*$.

Demonstrăm acum (b). Fie $b \in \mathbf{Z}_{mm'}^*$. De la (1) urmează că există $a \in \mathbf{Z}_m$ și $a' \in \mathbf{Z}_{m'}$ astfel încât $b = ma' + m'a \text{ mod } mm'$. Vom arăta că $(m, a) = 1$ și $(m', a') = 1$. Presupunem, prin contradicție, că $(m, a) > 1$. Fie $d = (m, a)$. Atunci d este divizor comun pentru $ma' + m'a$ și mm' , ceea ce ne arată că d este divizor comun pentru b și mm' , contrazicând $(b, mm') = 1$.

În cazul $(m', a') > 1$ se raționează similar. Deci (b) este demonstrată. \square

Corolarul 2.4.1. Au loc următoarele proprietăți:

- (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

$$\phi(a) = \prod (p_i^{e_i} - p_i^{e_i-1}).$$

Demonstrație. (1) Dacă $(a, b) = 1$, atunci \mathbf{Z}_{ab}^* și $\mathbf{Z}_a^* \times \mathbf{Z}_b^*$ sunt izomorfe (conform Teoremei 2.4.1(2)). Deci cele două mulțimi au același cardinal. Adică, $\phi(ab) = \phi(a)\phi(b)$.

(2) În baza proprietății de la (1) și 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 și orice număr natural $e \geq 1$.

Fie p și e ca mai sus. Numerele din \mathbf{Z}_{p^e} ce nu sunt prime cu p^e sunt exact multiplii lui p . Aceștia sunt $0 \cdot p, 1 \cdot p, 2 \cdot p, \dots, (p^{e-1} - 1) \cdot p$. Ca urmare,

$$\phi(p^e) = p^e - p^{e-1},$$

ceea ce încheie demonstrația. \square

Teorema 2.4.2. (Teorema lui Euler)

Fie $m \geq 1$. Atunci, $a^{\phi(m)} \equiv 1 \text{ mod } m$, pentru orice $a \in \mathbf{Z}_m^*$.

Demonstrație. Fie $a_1, \dots, a_{\phi(m)}$ o enumerare a elementelor mulțimii \mathbf{Z}_m^* . Fie $a \in \mathbf{Z}_m^*$. Atunci $aa_1, \dots, aa_{\phi(m)}$ este, de asemenea, o enumerare a elementelor mulțimii \mathbf{Z}_m^* (pentru orice $i \neq j$, aa_i și 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, relația de mai sus conduce la $a^{\phi(m)} \equiv 1 \text{ mod } m$. \square

Corolarul 2.4.2. Fie $m \geq 1$. Atunci $a^{\phi(m)} \equiv 1 \text{ mod } m$, pentru orice $a \in \mathbf{Z}$ cu $(a, m) = 1$.

Demonstrație. Dacă $a \in \mathbf{Z}_m^*$, atunci corolarul urmează direct de la Teorema 2.4.2. Altfel, se utilizează Teorema împărțirii cu rest și se aplică restului Teorema 2.4.2 (restul și m sunt prime între ele). \square

Corolarul 2.4.3. (Teorema lui Fermat)

Dacă p este un număr prim, atunci $a^{p-1} \equiv 1 \text{ mod } p$, pentru orice $a \in \mathbf{Z}$ cu $p \nmid a$.

Demonstrație. Dacă $p \nmid a$, atunci $(a, p) = 1$. Corolarul urmează atunci de la Corolarul 2.4.2 și faptul că $\phi(p) = p - 1$. \square

Corolarul 2.4.3 poate fi formulat echivalent astfel.

Corolarul 2.4.4. Dacă p este un număr prim, atunci $a^p \equiv a \text{ mod } p$, pentru orice $a \in \mathbf{Z}$.

Demonstrație. Dacă $p \mid a$, atunci $a^p \equiv_p 0 \equiv_p a$. Altfel, $a^{p-1} \equiv 1 \text{ mod } p$ care, combinată cu $a \equiv a \text{ mod } p$, conduce la $a^p \equiv a \text{ mod } p$. \square

Dacă presupunem că p este prim și $p \nmid a$, atunci relația $a^p \equiv a \text{ mod } p$ conduce, în baza Propoziției 2.3.1(5c), la $a^{p-1} \equiv 1 \text{ mod } p$. Deci, Corolarul 2.4.4 implică Corolarul

2.4.3. Cum în demonstrarea Corolarului 2.4.4 s-a utilizat Corolarul 2.4.3, deducem că afirmațiile din cele două corolare sunt echivalente.

Teorema 2.4.3. Fie $m \geq 1$. Atunci, $\sum_{d|m} \phi(d) = m$ (d este subînțeles ca fiind divizor pozitiv al lui m deoarece funcția ϕ este definită numai pentru numere strict pozitive).

Demonstrație. Fie $A = \{1, \dots, m\}$ și $A_d = \{a \in A | (a, m) = d\}$, pentru orice $1 \leq d \leq m$. Este clar că $A_d \cap A_{d'} = \emptyset$, pentru orice $d \neq d'$, și $A = \bigcup_{d=1}^m A_d$. Ca urmare,

$$m = |A| = \sum_{d=1}^m |A_d|.$$

Fie $1 \leq d \leq m$. Dacă $d \nmid m$, atunci $A_d = \emptyset$. Dacă $d|m$, atunci $A_d \neq \emptyset$ și

$$|A_d| = |\{a \in A | (a, m) = d\}| = |\{a' | 1 \leq a' \leq m/d, (a', m/d) = 1\}| = \phi(m/d).$$

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 demonstrația teoremei. \square

2.5. Ecuații congruențiale

Ecuațiile congruențiale, similare ecuațiilor clasice, joacă un rol deosebit de important în aritmetică, structura grupurilor ciclice, criptografie etc. Vom prezenta mai jos câteva elemente de bază asupra acestora.

Definiția 2.5.1. Fie $f(x) = a_n x^n + \dots + a_0$ un polinom cu coeficienți întregi și $m \geq 1$.

- (1) Spunem că f are *gradul n modulo m* dacă $a_n \not\equiv 0 \pmod{m}$.
- (2) Spunem că $c \in \mathbf{Z}$ este *rădăcină modulo m a lui f* sau că este *soluție* a ecuației $f(x) \equiv 0 \pmod{m}$ dacă $f(c) \equiv 0 \pmod{m}$.

Ecuațiile de forma $f(x) \equiv 0 \pmod{m}$ vor fi numite *ecuații congruențiale*. Dacă gradul polinomului f este n modulo m , atunci vom mai spune că ecuația congruențială este de *grad n* sau *ordin n*. Ecuațiile congruențiale de grad 1 vor mai fi numite și *ecuații congruențiale liniare*.

O primă ecuație congruențială a fost implicit întâlnită în Teorema 2.4.2,

$$x^{\phi(m)} - 1 \equiv 0 \pmod{m}.$$

Conform Teoremei 2.4.2, această ecuație are $\phi(m)$ soluții distincte (ne-congruente) modulo m (vom spune mai simplu că această ecuație are $\phi(m)$ soluții modulo m).

Să ne întreptăm atenția spre ecuații congruențiale liniare $ax + b \equiv 0 \pmod{m}$ în necunoscuta x , unde $a, b \in \mathbf{Z}$. Conform definiției congruenței modulo m , a determină o soluție a acestei ecuații revine la a determina un cuplu (x, y) de numere întregi astfel încât $ax + (-m)y = -b$. Apelând la Corolarul 2.2.2, această ultimă ecuație admite soluție (în x și y) dacă și numai dacă $(a, m)|b$. Am obținut astfel următorul rezultat important.

Teorema 2.5.1. Fie $a, b, m \in \mathbf{Z}$ cu $m \geq 1$. Atunci, ecuația $ax \equiv b \pmod{m}$ are soluții în \mathbf{Z} dacă și numai dacă $(a, m)|b$. În plus, dacă această ecuație are soluții, atunci ea are exact (a, m) soluții în \mathbf{Z}_m , ce sunt de forma

$$(x_0 + im/(a, m)) \pmod{m},$$

unde x_0 este o soluție (arbitră dar fixată) a acestei ecuații și $0 \leq i < (a, m)$.

Demonstrație. Conform observației de mai sus, ne rămâne de demonstrat doar partea a două a acestei teoreme.

Presupunem că ecuația $ax \equiv b \pmod{m}$ are soluții și, fie x_0 o soluție a ei. Vom arăta că numerele din teoremă sunt soluții distincte două către două ale ecuației și, reciproc, orice soluție a ecuației este de forma menționată în teoremă.

Fie d cel mai mare divizor comun al numerelor a și m . Prin simplă verificare deducem că $(x_0 + im/d) \pmod{m}$ este soluție a acestei ecuații, pentru orice $i \in \{0, \dots, d-1\}$, și orice două astfel de soluții sunt distincte.

Fie $c \in \mathbf{Z}_m$ o soluție a ecuației $ax \equiv b \pmod{m}$. Relațiile $ac \equiv b \pmod{m}$ și $ax_0 \equiv b \pmod{m}$ conduc la $ac \equiv ax_0 \pmod{m}$, iar în baza Propoziției 2.3.1(5b) obținem $c \equiv x_0 \pmod{m/d}$. Ca urmare, există i astfel încât $c = (x_0 + im/d) \pmod{m}$. Deoarece

$$c = (x_0 + im/d) \pmod{m} = (x_0 + (i \pmod{m})m/d) \pmod{m},$$

deducem că numărul i poate fi ales satisfăcând $0 \leq i \leq d-1$.

Deci orice soluție din \mathbf{Z}_m a ecuației $ax \equiv b \pmod{m}$ este de forma

$$(x_0 + im/(a, m)) \pmod{m},$$

unde $0 \leq i < d$. \square

Corolarul 2.5.1. Fie $a, m \in \mathbf{Z}$ cu $a \neq 0$ și $m \geq 1$. Atunci următoarele afirmații sunt echivalente:

- (1) ecuația $ax \equiv 1 \pmod{m}$ are soluție în \mathbf{Z}_m , și 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 .

Soluțiile modulo m ale ecuațiilor $ax \equiv b \pmod{m}$, atunci când $(a, m)|b$, se obțin astfel. Fie $d = (a, m)$ și $\alpha, \beta \in \mathbf{Z}$ astfel încât $a\alpha + m\beta = d$ (α și β determinate cu ajutorul algoritmului extins al lui Euclid). Considerând $b = db'$, relația de mai sus conduce la $aab' + m\beta b' = db'$, de unde deducem că are loc $aab' \equiv b \pmod{m}$. Deci, $aab' \pmod{m}$ este soluție în \mathbf{Z}_m a ecuației $ax \equiv b \pmod{m}$. Celelalte soluții în \mathbf{Z}_m se obțin pe baza relației din Teorema 2.5.1. Algoritmul astfel rezultat este următorul:

Ecuatie_Grad1

```

input:  a,b,m ∈ Z cu m ≥ 1;
output: toate soluțiile modulo m ale ecuației ax ≡ b mod m;
begin
1.   calculează (a, m) := aa + βm cu EuclidExt;
2.   if (a, m)|b
        then begin
3.           b' := b/(a, m);
4.           x0 := ab';
5.           for i := 0 to (a, m) - 1 do
                print((x0 + im/(a, m)) mod m);
            end
6.   else "ecuația nu are soluții"
end.
```

La pasul 1 în algoritmul Ecuatie_Grad1 se înțelege că se calculează (a, m) și o combinație liniară a acestuia. Complexitatea algoritmului Ecuatie_Grad1 este exact complexitatea algoritmului lui Euclid.

Teorema 2.5.2. (Teorema lui Lagrange)

Orice ecuație congruențială de grad n modulo p , unde p este un număr prim, are cel mult n soluții (ne-congruente) modulo p .

Demonstrație. Fie p un număr prim. Vom face demonstrația prin inducție matematică după gradul ecuației.

Fie $f(x) = a_1x + a_0$ un polinom de gradul 1 modulo p . Ca urmare, $(a_1, p) = 1$, iar Teorema 2.4.1 ne spune că ecuația $f(x) \equiv 0 \pmod{p}$ are exact o soluție modulo p .

Presupunem afirmația din teoremă adevărată pentru ecuații congruențiale modulo p de grad $n \geq 1$, și fie $f(x) = a_{n+1}x^{n+1} + a_nx^n + \dots + a_0$ un polinom de gradul $n+1$ modulo p . Presupunem, prin contradicție, că ecuația $f(x) \equiv 0 \pmod{p}$ are cel puțin $n+2$ soluții (ne-congruente) modulo p , și fie c_0, \dots, c_{n+1} soluții 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 $a_{n+1} \not\equiv 0 \pmod{p}$, deducem că g are gradul n modulo p .

Arătăm acum că c_1, \dots, c_{n+1} sunt rădăcini (ne-congruente) modulo p ale lui g , ceea ce va constitui o contradicție cu ipoteza inductivă, și astfel afirmația noastră va fi falsă. Fie $1 \leq i \leq n+1$. Deoarece $f(c_i) \equiv 0 \pmod{p}$, obținem

$$f(c_i) - f(c_0) = (c_i - c_0)g(c_i) \equiv 0 \pmod{p}.$$

c_i și c_0 nu sunt congruente modulo p , și atunci relația de mai sus conduce la $g(c_i) \equiv 0 \pmod{p}$. Deci c_i este rădăcină modulo p a lui g . Cum c_1, \dots, c_{n+1} nu sunt congruente modulo p (două căte două), deducem că g ar avea cel puțin $n+1$ rădăcini (ne-congruente) modulo p . Ca urmare, afirmația făcută este falsă, și astfel teorema este demonstrată. \square

În Teorema 2.5.2, dacă nu se cere ca p să fie număr prim, atunci concluzia acesteia ar putea să nu fie adevărată (a se vedea Exemplul 2.6.1(2)).

Corolarul 2.5.2. Fie p un număr prim și d un divizor al lui $p-1$. Atunci ecuația $x^d \equiv 1 \pmod{p}$ are d soluții (ne-congruente) modulo p .

Demonstrație. Fie $p-1 = de$. Atunci

$$x^{p-1} - 1 = (x^d - 1)(x^{d(e-1)} + \dots + x^d + 1) = (x^d - 1)g(x).$$

Conform teoremei lui Euler, $x^{p-1} - 1$ are exact $p-1$ rădăcini (ne-congruente) modulo p , iar relația 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 2.5.2). Combinând cu relația de mai sus, deducem că $x^d - 1$ trebuie să aibă cel puțin $(p-1) - d(e-1) = d$ rădăcini (ne-congruente) modulo p . Aplicând Teorema 2.5.2 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 modulo p . \square

2.6. Teorema chineză a resturilor

Problema lui Sun Tzu Suan Ching, menționată la începutul Secțiunii 2.3, poate fi formalizată astfel: determinați $x \in \mathbf{Z}$ astfel încât acesta să verifice simultan congruențele

$$\begin{cases} x \equiv 2 \pmod{3} \\ x \equiv 3 \pmod{5} \\ x \equiv 2 \pmod{7} \end{cases}$$

Ca urmare, suntem conduși la rezolvarea unor sisteme de ecuații congruențiale liniare. Menționăm încă de la început că astfel de sisteme pot să aibă sau să nu aibă soluție.

De exemplu, sistemul de mai sus admite soluția $x = 23$, dar sistemul

$$\begin{cases} x \equiv 0 \pmod{3} \\ x \equiv 1 \pmod{6} \end{cases}$$

nu are soluții (altfel, ar exista $\alpha, \beta \in \mathbf{Z}$ astfel încât $3\alpha = x$ și $6\beta = x - 1$, ceea ce ar conduce la $3\alpha - 6\beta = 1$ care nu admite soluții în α și β deoarece $3 \nmid 1$).

Să considerăm cazul unui sistem cu trei ecuații,

$$\begin{cases} x \equiv b_1 \pmod{m_1} \\ x \equiv b_2 \pmod{m_2} \\ x \equiv b_3 \pmod{m_3}, \end{cases}$$

și să analizăm posibilitățile de a obține soluții pentru acesta.

În primul rând, putem porni de la ideea determinării unei soluții x_i pentru fiecare ecuație și de a combina aceste soluții în una singură x astfel încât, prin reducere modulo m_i , x să devină soluție pentru a i -a ecuație. O variantă naturală de a combina x_1 , x_2 și x_3 ar fi prin

$$x = m_2m_3x_1 + m_1m_3x_2 + m_1m_2x_3.$$

Atunci, x este soluție a primei ecuații dacă și numai dacă

$$m_2m_3x_1 \equiv b_1 \pmod{m_1}.$$

Ca urmare, x_1 ales la început nu trebuie să fie soluție a primei ecuații a sistemului, ci el trebuie să fie soluție a ecuației

$$m_2m_3x \equiv b_1 \pmod{m_1},$$

ceea ce este posibil dacă și numai dacă $(m_2m_3, m_1) | b_1$.

În mod similar, x_2 trebuie să fie soluție a ecuației

$$m_1m_3x \equiv b_2 \pmod{m_2},$$

ceea ce este posibil dacă și numai dacă $(m_1m_3, m_2) | b_2$, iar x_3 trebuie să fie soluție a ecuației

$$m_1m_2x \equiv b_3 \pmod{m_3},$$

ceea ce este posibil dacă și numai dacă $(m_1m_2, m_3) | b_3$.

Acstea observații conduc la următorul rezultat foarte important.

Teorema 2.6.1. (Teorema chineză a resturilor)

Fie $k \geq 1$ un număr natural și m_1, \dots, m_k numere întregi prime între ele două câte două. Atunci, pentru orice $b_1, \dots, b_k \in \mathbf{Z}$, sistemul de ecuații

$$\begin{cases} x \equiv b_1 \pmod{m_1} \\ \dots \\ x \equiv b_k \pmod{m_k} \end{cases}$$

admete o unică soluție modulo $m_1 \cdots m_k$.

Demonstrație. Fie $m = m_1 \cdots m_k$. Considerăm $c_i = m/m_i$, 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 și cu c_i . În baza Teoremei 2.5.2, ecuația

$$c_i x \equiv b_i \pmod{m_i}$$

admete soluții, pentru orice $1 \leq i \leq k$; fie x_i o astfel de soluție.

Este imediat de verificat că $x = c_1x_1 + \cdots + c_kx_k$ este soluție a sistemului, iar $x \pmod{m}$ este soluție în \mathbf{Z}_m .

Vom arăta că $y \equiv x \pmod{m}$, pentru orice altă soluție y a sistemului. Fie y o soluție a sistemului. Deoarece $y \equiv b_i \pmod{m_i}$ și $x \equiv b_i \pmod{m_i}$, pentru orice $1 \leq i \leq k$, deducem $y \equiv x \pmod{m_i}$, pentru orice $1 \leq i \leq k$. Acestea, combinate cu $(m_i, m_j) = 1$, pentru orice $i \neq j$, conduc la $y \equiv x \pmod{m}$. \square

Unica soluție $x \in \mathbf{Z}_m$ 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 2.6.1. Fie $k \geq 1$ un număr natural și $a_1, \dots, a_k, m_1, \dots, m_k$ numere întregi astfel încât $(a_i, m_i) = 1$ și $(m_i, m_j) = 1$, pentru orice $1 \leq i, j \leq k$ cu $i \neq j$. Atunci, pentru orice $b_1, \dots, b_k \in \mathbf{Z}$, sistemul de ecuații

$$\begin{cases} a_1x \equiv b_1 \pmod{m_1} \\ \dots \\ a_kx \equiv b_k \pmod{m_k} \end{cases}$$

admete o unică soluție modulo $m_1 \cdots m_k$.

Demonstrație. Deoarece $(a_i, m_i) = 1$, pentru orice $1 \leq i \leq k$, deducem că sistemul

$$\begin{cases} a_1x \equiv b_1 \pmod{m_1} \\ \dots \\ a_kx \equiv b_k \pmod{m_k} \end{cases}$$

este echivalent cu sistemul

$$\begin{cases} x \equiv a_1^{-1}b_1 \pmod{m_1} \\ \dots \\ x \equiv a_k^{-1}b_k \pmod{m_k} \end{cases}$$

ce admite soluție unică modulo $m_1 \cdots m_k$ în baza Teoremei chineză a resturilor. \square

Teorema 2.6.2. Fie $k \geq 1$ un număr natural și $b_1, \dots, b_k, m_1, \dots, m_k$ numere întregi.

Atunci sistemul de ecuații

$$\begin{cases} x \equiv b_1 \pmod{m_1} \\ \dots \\ x \equiv b_k \pmod{m_k} \end{cases}$$

admite soluții dacă și numai dacă $b_i \equiv b_j \pmod{m_i, m_j}$, pentru orice $1 \leq i, j \leq k$ cu $i \neq j$. În plus, dacă acest sistem admite soluții, atunci aceasta este unică modulo $[m_1, \dots, m_k]$.

Demonstrație. Vom demonstra teorema pentru cazul particular $k = 2$.

Dacă sistemul de mai sus admite o soluție, fie aceasta $x = a$, atunci

$$a \equiv b_1 \pmod{m_1} \text{ și } a \equiv b_2 \pmod{m_2}.$$

De aici urmează că

$$a \equiv b_1 \pmod{(m_1, m_2)} \text{ și } a \equiv b_2 \pmod{(m_1, m_2)},$$

ceea ce conduce la $b_1 \equiv b_2 \pmod{(m_1, m_2)}$.

Reciproc, presupunem că $(m_1, m_2) | b_1 - b_2$. Orice soluție a primei ecuații este de forma $x = b_1 + m_1y$, unde $y \in \mathbf{Z}$. Cerința ca o astfel de soluție să verifice și cea de a doua ecuație conduce la problema existenței unui $y \in \mathbf{Z}$ astfel încât

$$b_1 + m_1y \equiv b_2 \pmod{m_2}$$

sau, altfel spus, la existența unei soluții (în y) a ecuației

$$m_1y \equiv b_1 - b_2 \pmod{m_2}.$$

Cum $(m_1, m_2) | b_1 - b_2$, deducem că această ecuație admite soluții (în y). Ca urmare, sistemul admite soluții.

Să presupunem acum că sistemul admite soluții și să arătăm că orice două soluții sunt congruente modulo $[m_1, m_2]$. Fie deci a și a' două soluții. Din faptul că acestea trebuie să verifice prima ecuație obținem $a \equiv a' \pmod{m_1}$. Similar, $a \equiv a' \pmod{m_2}$. Aceste două relații conduc la $a \equiv a' \pmod{[m_1, m_2]}$, în baza Propoziției 2.3.1(6b). \square

O aplicație importantă a teoremei chineze a resturilor constă în determinarea numărului de soluții ale ecuațiilor de forma $f(x) \equiv 0 \pmod{m_1 \cdots m_k}$, unde f este un polinom cu coeficienți întregi, iar m_1, \dots, m_k sunt numere naturale prime între ele două câte două.

Teorema 2.6.3. Fie f un polinom cu coeficienți întregi și m_1, \dots, m_k numere naturale prime între ele două câte două, unde $k \geq 2$. Atunci, un număr $a \in \mathbf{Z}$ este soluție a ecuației

$$f(x) \equiv 0 \pmod{m_1 \cdots m_k},$$

dacă și numai dacă a este soluție a fiecărei ecuații

$$f(x) \equiv 0 \pmod{m_i},$$

$1 \leq i \leq k$. În plus, numărul soluțiilor în $\mathbf{Z}_{m_1 \cdots m_k}$ ale ecuației

$$f(x) \equiv 0 \pmod{m_1 \cdots m_k}$$

este egal cu produsul numerelor de soluții în \mathbf{Z}_{m_i} ale ecuațiilor

$$f(x) \equiv 0 \pmod{m_i},$$

unde $1 \leq i \leq k$.

Demonstrație. Este clar că $a \in \mathbf{Z}$ este soluție a ecuației

$$f(x) \equiv 0 \pmod{m_1 \cdots m_k}$$

dacă și numai dacă este soluție a ecuațiilor

$$f(x) \equiv 0 \pmod{m_i},$$

$1 \leq i \leq k$.

Fie a_i soluție în \mathbf{Z}_{m_i} a ecuației

$$f(x) \equiv 0 \pmod{m_i},$$

$1 \leq i \leq k$. Sistemul

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ \dots \\ x \equiv a_k \pmod{m_k} \end{cases}$$

admete o unică soluție în $\mathbf{Z}_{m_1 \cdots m_k}$ (conform Teoremei chineze a resturilor). Mai mult, se verifică ușor că aceasta este soluție a ecuației

$$f(x) \equiv 0 \pmod{m_1 \cdots m_k}.$$

Ca urmare, orice k -uplu de soluții $(a_1, \dots, a_k) \in \mathbf{Z}_{m_1} \times \dots \times \mathbf{Z}_{m_k}$ a ecuațiilor

$$f(x) \equiv 0 \pmod{m_i},$$

$1 \leq i \leq k$, conduce la o unică soluție în $\mathbf{Z}_{m_1 \cdots m_k}$ a ecuației

$$f(x) \equiv 0 \pmod{m_1 \cdots m_k},$$

și reciproc, pentru orice soluție a a acestei ecuații, $a \pmod{m_i}$ este soluție în \mathbf{Z}_{m_i} a ecuației

$$f(x) \equiv 0 \pmod{m_i},$$

$i \leq i \leq k$. Aceasta încheie demonstrația celei de a doua părți a teoremei. \square

Exemplul 2.6.1.

- (1) Fie p un număr prim impar. Ecuația $x^2 \equiv 1 \pmod{p}$ are exact 2 soluții în \mathbf{Z}_p , și anume $x = 1$ și $x = p - 1$. În adevară,

$$\begin{aligned} x^2 \equiv 1 \pmod{p} &\Leftrightarrow p|x^2 - 1 \\ &\Leftrightarrow p|(x-1)(x+1) \\ &\Leftrightarrow p|x-1 \text{ sau } p|x+1 \\ &\Leftrightarrow x \equiv 1 \pmod{p} \text{ sau } x \equiv -1 \pmod{p}. \end{aligned}$$

- (2) Fie p_1, \dots, p_k numere prime distințe, unde $k \geq 2$. Atunci ecuația

$$x^2 \equiv 1 \pmod{p_1 \cdots p_k}$$

are exact 2^k soluții în $\mathbf{Z}_{p_1 \cdots p_k}$.

În secțiunea 4.5 vom adăuga noi rezultate asupra rezolvării ecuațiilor de forma $x^n \equiv 1 \pmod{m}$ și $x^n \equiv -1 \pmod{p}$, unde $n, m \geq 1$, iar p este număr prim.

2.7. Reziduozație pătratică

În această secțiune ne vom îndrepta atenția asupra rezolvării congruențelor pătratice

$$ax^2 + bx + c \equiv 0 \pmod{p},$$

unde $a, b, c \in \mathbf{Z}$, iar p este un număr prim.

Problema cu care ne vom confrunta aici este de a decide dacă discriminantul congruenței, $\Delta = b^2 - 4ac \pmod{p}$, este sau nu un "pătrat perfect" modulo p . Această problemă, mult diferită de cazul real, conduce la un studiu foarte interesant și elegant numit *reziduozație pătratică*. Ceea ce este însă și mai interesant este că reziduozația pătratică are aplicații consistente în criptografie. Vom puncta câteva repede de acest fel pe parcursul expunerii și invităm cititorul să consulte [34, 212] pentru detalii suplimentare.

2.7.1. Congruențe pătratice

Începem această sub-secțiune prin studiul congruenței pătratice anunțate mai sus.

Propoziția 2.7.1.1. Fie $p > 2$ un număr prim și $a, b, c \in \mathbf{Z}$ astfel încât $(a, p) = 1$. Atunci, dacă notăm $\Delta = b^2 - 4ac \pmod{p}$, congruența pătratică

$$ax^2 + bx + c \equiv 0 \pmod{p}$$

satisfacă exact una din următoarele proprietăți:

1. dacă există $y \in \mathbf{Z}$ cu $p \nmid y$ și $\Delta \equiv y^2 \pmod{p}$, atunci congruența are exact două soluții în \mathbf{Z}_p , ce sunt de forma $(-b \pm y)(2a)^{-1} \pmod{p}$;
2. dacă $\Delta \equiv 0 \pmod{p}$, atunci congruența are exact o soluție în \mathbf{Z}_p , ce este de forma $-b(2a)^{-1} \pmod{p}$;

3. nu are nicio soluție, altfel.

Demonstrație. Ipoteza $p > 2$ și $(a, p) = 1$ ne permite să înmulțim congruența $ax^2 + bx + c \equiv 0 \pmod{p}$ cu $4a$ fără a modifica numărul și natura soluțiilor ei. În plus, dacă adunăm și scădem b^2 la membrul stâng al acesteia obținem forma echivalentă

$$(2ax + b)^2 \equiv \Delta \pmod{p}.$$

Concluziile propoziției se deduc atunci în mod imediat. \square

Observația 2.7.1.1. Propoziția 2.7.1.1 ne arată cum se rezolvă congruențele pătratice modulo $p > 2$. În cazul $p = 2$, rezolvarea acestor congruențe se reduce la rezolvarea unor congruențe liniare deoarece $x^2 \equiv x \pmod{2}$.

Observația 2.7.1.2. Folosind teorema chineză a resturilor, și în special Teorema 2.6.3, putem rezolva congruențe pătratice în care modulul este produs de numere prime între ele două câte două. De exemplu, pentru rezolvarea congruenței

$$ax^2 + bx + c \equiv 0 \pmod{pq},$$

unde p și q sunt numere prime distincte, putem determina soluțiile congruenței modulo p și modulo q în mod separat, după care combinăm soluțiile două câte două prin teorema chineză a resturilor.

Fie, de exemplu, $a = 2$, $b = 3$, $c = -5$, $p = 7$ și $q = 11$. Congruența

$$2x^2 + 3x - 5 \equiv 0 \pmod{7}$$

are o soluție în \mathbf{Z}_7 , și anume $x_0 = 1$. Congruența

$$2x^2 + 3x - 5 \equiv 0 \pmod{11}$$

are două soluții în \mathbf{Z}_{11} , și anume $x_1 = 1$ și $x_2 = 3$. Combinând x_0 și x_1 prin teorema chineză a resturilor obținem

$$\begin{cases} x = 1 \pmod{7} \\ x = 1 \pmod{11} \end{cases}$$

care conduce la unica soluție $x_{0,1} = 22$ în \mathbf{Z}_{88} . În mod similar, dacă combinăm x_0 cu x_2 vom obține o unică soluție $x_{0,2} = 36$ în \mathbf{Z}_{88} .

Astfel, $2x^2 + 3x - 5 \equiv 0 \pmod{88}$ are exact 2 soluții în \mathbf{Z}_{88} , și anume 22 și 36.

Observația 2.7.1.3. Schema lui Cocks de criptare bazată pe identitate [34, 212] folosește congruențe pătratice de tipul celor din Observația 2.7.1.2, în care factorizarea modulului pq este cunoscută doar de generatorul de chei private necesare decriptării. Cum factorizarea este o problemă pentru care nu se cunoaște, la momentul actual, niciun algoritm deterministic de complexitate timp polinomială, utilizarea de numere prime mari p și q conferă securitate schemei (pentru mai multe detalii indicăm cititorului consultarea referințelor de mai sus).

2.7.2. Reziduuri pătratice și simbolul Legendre

Constatăm că, în Propoziția 2.7.1.1, este necesar să decidem dacă Δ este sau nu un “pătrat perfect” modulo p . Vom dezvolta în continuare o metodă prin care putem decide aceasta în mod eficient, atâtă timp cât lucrăm cu numere prime p . Extensia acestui rezultat la numere compuse se “confruntă” cu problema factorizării care, la momentul actual, nu poate fi ocolită (detalii vor fi date mai departe).

Definiția 2.7.2.1. Fie $p > 2$ un număr prim. Un număr $a \in \mathbf{Z}$ nedivizibil prin p este numit *reziduu pătratic* modulo p dacă există $x \in \mathbf{Z}$ astfel încât $a \equiv x^2 \pmod{p}$.

Înregii ce nu sunt reziduuri pătratice se mai numesc și *ne-reziduuri pătratice*. Reziduurile pătratice au fost obiect de studiu al unor matematicieni precum Fermat, Euler, Lagrange, Legendre și Gauss. Terminologia de “reziduu pătratic” a fost introdusă de Gauss în 1801.

Definiția 2.7.2.1 nu ia în considerare cazul $p = 2$ și nici cazul în care a este divizibil prin p pentru a evita situații triviale de tipul următor:

- dacă acceptăm $p = 2$ în Definiția 2.7.2.1 dar păstrăm cerința de nedivizibilitate prin 2, atunci numerele impare sunt reziduuri pătratice modulo 2, în timp ce numerele pare sunt ne-reziduuri pătratice modulo 2;
- dacă acceptăm $p = 2$ în Definiția 2.7.2.1 dar renunțăm la cerința de divizibilitate prin 2, atunci toate numerele întregi sunt reziduuri pătratice.

Statutul de reziduu pătratic modulo p al unui întreg poate fi decis doar pe baza restului modulo p al acestuia.

Propoziția 2.7.2.1. Fie $p > 2$ un număr prim și $a, b \in \mathbf{Z}$. Dacă a și b sunt congruente modulo p , atunci a este reziduu pătratic modulo p dacă și numai dacă b este reziduu pătratic modulo p .

Demonstrație. Este clar ca relația $a \equiv b \pmod{p}$ conduce la $a \equiv x^2 \pmod{p}$ dacă și numai dacă $b \equiv x^2 \pmod{p}$, pentru orice $x \in \mathbf{Z}$. \square

Propoziția 2.7.2.1 conduce la faptul că a este reziduu pătratic modulo p dacă și numai dacă $a \pmod{p}$ este reziduu pătratic modulo p . În plus, dacă $a \in \mathbf{Z}_p^*$ este reziduu pătratic modulo p atunci există $x \in \mathbf{Z}_p^*$ astfel încât $a \equiv x^2 \pmod{p}$. Ca urmare, studiul reziduozițăii pătratice modulo p poate fi restrâns la \mathbf{Z}_p^* . Fie atunci mulțimile:

- $QR_p = \{a \in \mathbf{Z}_p^* \mid a$ este reziduu pătratic modulo $p\}$;
- $QNR_p = \{a \in \mathbf{Z}_p^* \mid a$ este ne-reziduu pătratic modulo $p\}$.

Un prim rezultat interesant ne arată că aceste două mulțimi au același număr de elemente.

Propoziția 2.7.2.2. Pentru orice număr prim $p > 2$ are loc $|QR_p| = |QNR_p| = \frac{p-1}{2}$.

Demonstrație. Fie $p > 2$ un număr prim și $m = \frac{p-1}{2}$. Putem scrie atunci

$$\mathbf{Z}_p^* = \{1, \dots, m, p-m, \dots, p-1\}$$

Observăm că $i^2 \equiv_p (p-i)^2$, pentru orice $1 \leq i \leq m$. Prinț-o foarte simplă analiză se deduce că mulțimea $\{i^2 \pmod{p} \mid i \in \mathbf{Z}_p^*\}$ are exact m elemente, ce sunt exact reziduurile pătratice din \mathbf{Z}_p^* . Ca urmare, ea este QR_p . Urmează atunci că $QNR_p = \mathbf{Z}_p^* - QR_p$, ce are și ea exact m elemente. \square

Inversabilitatea modulară păstrează reziduozițatea.

Propoziția 2.7.2.3. Fie $p > 2$ un număr prim și $a \in \mathbf{Z}_p^*$. Atunci, $a \in QR_p$ dacă și numai dacă $a^{-1} \in QR_p$ (a^{-1} este inversul lui a în \mathbf{Z}_p^*).

Demonstrație. Dacă $a \in QR_p$ și $a \equiv x^2 \pmod{p}$ cu $x \in \mathbf{Z}_p^*$, atunci $a^{-1} \equiv (x^{-1})^2 \pmod{p}$. \square

Este util de știut cum se păstrează reziduozițatea prin înmulțirea reziduurilor și ne-reziduurilor pătratice între ele.

Propoziția 2.7.2.4. Fie $p > 2$ un număr prim și $a, b \in \mathbf{Z}_p^*$.

- (1) Dacă $a, b \in QR_p$, atunci $(ab \pmod{p}) \in QR_p$.
- (2) Dacă $a \in QR_p$ și $b \in QNR_p$, atunci $(ab \pmod{p}) \in QNR_p$.
- (3) Dacă $a, b \in QNR_p$, atunci $(ab \pmod{p}) \in QR_p$.

Demonstrație. (1) Dacă $a \equiv x^2 \pmod{p}$ și $b \equiv y^2 \pmod{p}$, atunci $(ab \pmod{p}) \equiv (xy)^2 \pmod{p}$.

(2) Fie $a \in QR_p$ și $b \in QNR_p$. Dacă presupunem că $ab \pmod{p}$ este reziduu pătratic modulo p , fie acesta $r \in QR_p$, atunci $b \equiv ra^{-1} \pmod{p}$. Această ultimă relație constituie o contradicție deoarece b este ne-reziduu în timp ce ra^{-1} este reziduu pătratic modulo p . Ca urmare, $(ab \pmod{p}) \in QNR_p$.

(3) Fie $a, b \in QNR_p$ și r_1, \dots, r_m toate reziduurile pătratice modulo p din \mathbf{Z}_p^* , unde $p = 2m + 1$. Se verifică imediat că au loc proprietățile:

- $QNR_p = \{ar_i \pmod{p} \mid 1 \leq i \leq m\}$;
- $ab \not\equiv ar_i \pmod{p}$, pentru orice $1 \leq i \leq m$.

Ca urmare, presupunerea $(ab \pmod{p}) \in QNR_p$ conduce la o contradicție. Deci, $(ab \pmod{p}) \in QR_p$. \square

Vom discuta în cele ce urmează un criteriu fundamental de caracterizare a reziduozițăii pătratice propus de Euler în 1748 (a se vedea [118]). Pornim de la observația

că teorema lui Fermat asigură faptul că $a^{p-1} \equiv 1 \pmod{p}$, pentru orice număr prim p și orice întreg a nedivizibil prin p . Să presupunem că p este impar, $p = 2m + 1$, unde $m \geq 1$. Atunci, relația $a^{p-1} \equiv 1 \pmod{p}$ conduce la $p|a^m - 1$ sau $p|a^m + 1$. În plus, constatăm că nu pot avea loc ambele relații deoarece s-ar obține $p|2$ ceea ce contrazice alegerea lui p . Reziduozațitatea pătratică este ceea ce diferențiază între aceste două posibilități. Pentru a vedea acest lucru reamintim întâi un rezultat de caracterizare a primalității.

În 1770 [218], Edward Waring menționa o caracterizare interesantă a numerelor prime, nedemonstrată și datorată lui John Wilson. Un an mai târziu, Joseph Louis Lagrange propunea o demonstrație a acesteia [112]. Este vorba de ceea ce numim astăzi ca fiind *teorema lui Wilson*.

Teorema 2.7.2.1. (Teorema lui Wilson)

Un număr natural $n > 1$ este prim dacă și numai dacă $(n - 1)! \equiv -1 \pmod{n}$.

Demonstrație. Presupunem că $n > 1$ este prim. Atunci, orice i , $1 < i < n - 1$, admite un unic invers modulo n (în \mathbf{Z}_n^*). În plus, inversul lui i modulo n nu poate fi i (congruența $i^2 \equiv_n 1$ admite soluțiile $i = 1$ și $i = n - 1$), nu poate fi 1 și nici $n - 1$. Ca urmare, numerele de la 2 la $n - 2$ pot fi grupate două câte două astfel încât produsul numerelor din fiecare grupă să fie congruent cu 1 modulo n . Aceasta ne arată că $(n - 1)! \equiv n - 1 \pmod{n}$, de la care urmează $(n - 1)! \equiv -1 \pmod{n}$.

Reciproc, presupunem că $n > 1$ și $(n - 1)! \equiv -1 \pmod{n}$ dar n nu este prim. Atunci, n admite un factor d cu $1 < d < n$. Cum $d|n$ și $n|(n - 1)! + 1$, obținem $d|(n - 1)! + 1$. De la aceasta urmează $d|1$ deoarece $d|(n - 1)!$; contradicție. \square

Putem acum prezenta criteriul lui Euler de caracterizare a reziduozației pătratice.

Teorema 2.7.2.2. (Criteriul lui Euler)

Fie $p = 2m + 1$ un număr prim și $a \in \mathbf{Z}$ nedivizibil prin p , unde $m \geq 1$. Atunci, $a^m \equiv 1 \pmod{p}$ dacă și numai dacă a este reziduu pătratic modulo p .

Demonstrație. Fie $p = 2m + 1$ un număr prim și $a \in \mathbf{Z}$ nedivizibil prin p , unde $m \geq 1$.

Presupunem întâi că $a^m \equiv 1 \pmod{p}$ dar a nu satisfacă $a \equiv x^2 \pmod{p}$ pentru niciun $x \in \mathbf{Z}$. Deoarece p este prim, pentru orice i cu $1 \leq i \leq n - 1$, există un unic j , $1 \leq j \leq n - 1$, astfel încât $ij \equiv a \pmod{p}$ (congruență liniară $ij \equiv a \pmod{p}$, în necunoscută j , are soluție unică în \mathbf{Z}_p^*). În plus, presupunerea făcută asupra lui a arată că $j \not\equiv i \pmod{p}$. Mai mult, dacă $i_1 \not\equiv i_2 \pmod{p}$ atunci numerele j_1 și j_2 corespunzătoare (cu proprietatea de mai sus) sunt distincte (dacă ar coincide, s-ar deduce cu ușurință că $i_1 = i_2$). Atunci, grupând două câte două numerele i și j cu proprietatea $ij \equiv a \pmod{p}$, obținem:

$$(p - 1)! \equiv \underbrace{a \cdots a}_{m \text{ ori}} \pmod{p}$$

Teorema 2.7.2.1 conduce atunci la $a^m \equiv -1 \pmod{p}$; contradicție.

Reciproc, dacă există x cu $a \equiv x^2 \pmod{p}$, atunci $p \nmid x$ (altfel, p ar divide a). Ca urmare, teorema lui Fermat conduce la $a^m \equiv_p x^{2m} \equiv_p x^{p-1} \equiv_p 1$. \square

Observația 2.7.2.1. Fie $p > 2$ un număr prim și $a \in \mathbf{Z}$ nedivizibil prin p . Teorema 2.7.2.2 arată că:

1. a este reziduu pătratic modulo p dacă și numai dacă $a^{(p-1)/2} \equiv 1 \pmod{p}$;
2. a este ne-reziduu pătratic modulo p dacă și numai dacă $a^{(p-1)/2} \equiv -1 \pmod{p}$.

Criteriul lui Euler ne permite să decidem, pentru orice întreg a dacă acesta este sau nu reziduu pătratic modulo un număr prim $p > 2$. Exponențierea modulară poate fi realizată în timp cubic în raport cu lungimea reprezentării binare a datelor, care este $\log_2 p$ în cazul nostru (a se vedea [210] precum și Secțiunea 2.8). Ca urmare, aceasta va fi și complexitatea procedurii de decizie oferite de criteriul lui Euler. Însă, putem obține o procedură de decizie mai eficientă. Pentru aceasta avem nevoie să dezvoltăm anumite proprietăți prin care să putem decide mai eficient asupra reziduozației pătratice. Începem prin adoptarea unei notații convenabile de lucru cu reziduozațiatea pătratică, și anume *simbolul Legendre* introdus de Adrien Marie Legendre în 1798 [117] (a se vedea și [118]).

Definiția 2.7.2.2. Fie p un număr prim impar. *Simbolul Legendre* al lui $a \in \mathbf{Z}$ modulo p , notat $\left(\frac{a}{p}\right)$, este definit prin

$$\left(\frac{a}{p}\right) = \begin{cases} 0, & \text{dacă } p|a \\ 1, & \text{dacă } a \text{ este reziduu pătratic modulo } p \\ -1, & \text{dacă } a \text{ este ne-reziduu pătratic modulo } p \end{cases}$$

Mai puțin frecvent, simbolul Legendre poate fi întâlnit sub notația $J_p(a)$ sau $(a|p)$.

Observația 2.7.2.2. Fie $p > 2$ un număr prim. Atunci, Teorema 2.7.2.2 conduce la

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p},$$

pentru orice $a \in \mathbf{Z}$.

Corolarul 2.7.2.1. Fie $p > 2$ un număr prim. Atunci,

$$\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}} = \begin{cases} 1, & \text{dacă } p \equiv 1 \pmod{4} \\ -1, & \text{dacă } p \equiv 3 \pmod{4} \end{cases}$$

Demonstrație. Folosind Observația 2.7.2.2, este suficient să remarcăm că $\frac{p-1}{2}$ este par dacă și numai dacă $p \equiv 1 \pmod{4}$. \square

Un alt criteriu foarte util în studiul reziduozației pătratice este cel propus de Gauss în 1808 [64] (poate fi găsit și în [65] la pagina 457).

Teorema 2.7.2.3. (Criteriul lui Gauss)

Fie $p = 2m + 1$ un număr prim și $a \in \mathbf{Z}$ nedivizibil prin p , unde $m \geq 1$. Atunci, $\left(\frac{a}{p}\right) = (-1)^t$, unde

$$t = |\{i \in \{1, \dots, m\} | ia \bmod p > p/2\}|.$$

Demonstrație. Fie $p = 2m + 1$ un număr prim și $a \in \mathbf{Z}$ nedivizibil prin p , unde $m \geq 1$. Putem scrie

$$\mathbf{Z}_p^* = \{1, \dots, m, p-m, \dots, p-1\}$$

Pentru orice i și j cu $i, j \in \{1, \dots, m\}$ și $i \neq j$, au loc proprietățile $ia \not\equiv ja \bmod p$ și $ia \not\equiv 0 \bmod p$. Ca urmare,

$$(1a)(2a) \cdots (ma) \equiv_p (-1)^t 1 \cdot 2 \cdots m,$$

unde t este numărul din enunțul teoremei. Deci,

$$a^m \cdot m! \equiv_p (-1)^t \cdot m!$$

Deoarece p nu divide $m!$, congruența de mai sus conduce la $a^m \equiv (-1)^t \bmod p$, de la care obținem concluzia teoremei folosind criteriul lui Euler. \square

Corolarul 2.7.2.2. Fie $p > 2$ un număr prim. Atunci,

$$\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}} = \begin{cases} 1, & \text{dacă } p \equiv \pm 1 \bmod 8 \\ -1, & \text{dacă } p \equiv \pm 3 \bmod 8 \end{cases}$$

Demonstrație. Vom folosi criteriul lui Gauss și notăm

$$t = |\{i \in \{1, \dots, m\} | 2i \bmod p > p/2\}|,$$

unde $p = 2m + 1$. Fie, de asemenea, $p = 8k + r$, unde $k \in \mathbf{Z}$ și $r \in \{1, 3, 5, 7\}$. Observația fundamentală de la care pornim constă în aceea că nu suntem interesați în a determina valoarea lui t , ci doar paritatea lui.

Pentru orice $1 \leq i \leq m$, $2i \bmod p = 2i$. Cerința $2i \bmod p > p/2$ este echivalentă atunci cu

$$2k + \frac{r}{4} < i < 4k + \frac{r}{2}$$

Conform celor spuse mai sus, dorim să cunoaștem dacă numărul de valori i ce satisfac această inegalitate dublă este par sau impar. Nu vom pierde nimic din concluzia finală dacă studiem, în funcție de valorile lui r , paritatea numărului de întregi i ce satisfac

$$\frac{r}{4} < i < \frac{r}{2}$$

Constatăm că pentru $r = 1$ și $r = 7$ numărul de astfel de întregi i este par, iar pentru celelalte două valori ale lui r este impar. Deci, t este par în cazul $p \equiv \pm 1 \bmod 8$ și impar în cazul $p \equiv \pm 3 \bmod 8$. Aceasta conduce la concluzia corolarului. \square

Criteriul lui Gauss nu are valoare algoritmică așa cum are criteriul lui Euler. Însă, din punct de vedere teoretic, el permite obținerea unor formule de calcul pentru simbolul Legendre, așa cum este formula din corolarul anterior, ceea ce nu s-ar fi putut obține, cel puțin la fel de ușor, prin criteriul lui Euler.

Ultimul rezultat fundamental asupra reziduozații pătratice, pe care îl prezentăm fără demonstrație, este *legea reciprocității pătratice*.

Teorema 2.7.2.4. (Legea reciprocității pătratice)

Fie $p, q > 2$ numere prime distințte. Atunci,

$$\left(\frac{q}{p}\right) \left(\frac{p}{q}\right) = (-1)^{\frac{p-1}{2} \cdot \frac{q-1}{2}} = \begin{cases} \left(\frac{p}{q}\right), & \text{dacă } p \equiv 1 \bmod 4 \text{ sau } q \equiv 1 \bmod 4 \\ -\left(\frac{p}{q}\right), & \text{dacă } p, q \equiv 3 \bmod 4 \end{cases}$$

Rezultatul din Teorema 2.7.2.4 a fost propus independent de Euler și Legendre, dar fără demonstrație. Aceasta a fost obținută mai târziu, de către Gauss [66].

Putem acum aduna la un loc regulile de bază pentru calculul simbolului Legendre a unui întreg.

Teorema 2.7.2.5. (Reguli de calcul a simbolului Legendre)

Date $p, q > 2$ numere prime și $a, b \in \mathbf{Z}$, au loc următoarele proprietăți:

$$(1) \quad \text{Dacă } a \equiv b \bmod p \text{ atunci } \left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$$

$$(2) \quad \left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right) \left(\frac{b}{p}\right)$$

$$(3) \quad \left(\frac{1}{p}\right) = 1$$

$$(4) \quad \left(\frac{-1}{p}\right) = \begin{cases} 1, & \text{dacă } p \equiv 1 \bmod 4 \\ -1, & \text{dacă } p \equiv 3 \bmod 4 \end{cases}$$

$$(5) \quad \left(\frac{2}{p}\right) = \begin{cases} 1, & \text{dacă } p \equiv \pm 1 \bmod 8 \\ -1, & \text{dacă } p \equiv \pm 3 \bmod 8 \end{cases}$$

$$(6) \quad \left(\frac{q}{p}\right) = \begin{cases} \left(\frac{p}{q}\right), & \text{dacă } p \equiv 1 \bmod 4 \text{ sau } q \equiv 1 \bmod 4 \\ -\left(\frac{p}{q}\right), & \text{dacă } p \equiv q \equiv 3 \bmod 4 \end{cases}$$

Exemplul 2.7.2.1. Presupunem că dorim să vedem dacă 7 este reziduu pătratic modulo 59. Folosind regulile din Teorema 2.7.2.5 obținem

$$\left(\frac{7}{59}\right) = -\left(\frac{59}{7}\right) = -\left(\frac{3}{7}\right) = \left(\frac{7}{3}\right) = \left(\frac{1}{3}\right) = 1$$

ce arată că, în adevăr, 7 este reziduu pătratic modulo 59.

Așa cum se va discuta în secțiunea următoare, complexitatea de calcul a simbolului Legendre obținută prin utilizarea regulilor din Teorema 2.7.2.5 este cel mult pătratică în raport cu dimensiunea reprezentării operanzilor. Reamintim că, prin criteriul lui Euler complexitatea era cubică.

2.7.3. Simbolul Jacobi

In 1837, Gustav Jacobi a generalizat simbolul Legendre după cum urmează [98].

Definiția 2.7.3.1. Fie $n > 0$ un număr impar și $a \in \mathbb{Z}$. *Simbolul Jacobi al lui a modulo n*, notat $(\frac{a}{n})$, este definit prin

$$\left(\frac{a}{n}\right) = \begin{cases} 1, & \text{dacă } n=1 \\ \left(\frac{a}{p_1}\right)^{e_1} \cdots \left(\frac{a}{p_k}\right)^{e_k}, & \text{altfel.} \end{cases}$$

unde $n = p_1^{e_1} \cdots p_k^{e_k}$ este descompunerea în factori primi a lui n .

Este un bun exercițiu pentru cititor să verifice că toate proprietățile din teorema 2.7.2.5 se păstrează prin trecere la simbolul Jacobi. Datorită importanței lor, preferăm să rescriem Teorema 2.7.2.5 pentru simbolul Jacobi.

Teorema 2.7.3.1. (Reguli de calcul a simbolului Jacobi)

Date $n, m > 0$ numere impare și $a, b \in \mathbb{Z}$, au loc următoarele proprietăți:

1. Dacă $a \equiv b \pmod{p}$ atunci $(\frac{a}{n}) = (\frac{b}{n})$
2. $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$
3. $(\frac{1}{n}) = 1$
4. $(\frac{-1}{n}) = \begin{cases} 1, & \text{dacă } n \equiv 1 \pmod{4} \\ -1, & \text{dacă } n \equiv 3 \pmod{4} \end{cases}$
5. $(\frac{2}{n}) = \begin{cases} 1, & \text{dacă } n \equiv \pm 1 \pmod{8} \\ -1, & \text{dacă } n \equiv \pm 3 \pmod{8} \end{cases}$
6. $(\frac{m}{n}) = \begin{cases} (\frac{n}{m}), & \text{dacă } n \equiv 1 \pmod{4} \text{ sau } m \equiv 1 \pmod{4} \\ -(\frac{n}{m}), & \text{dacă } n \equiv m \equiv 3 \pmod{4} \end{cases}$

Complexitatea de calcul a simbolului Jacobi este aceeași ca și în cazul simbolului Legendre.

Conceptul de reziduu pătratic se extinde la moduli compuși în mod natural. Astfel, un întreg a relativ prim cu n este reziduu pătratic modulo n dacă există $x \in \mathbb{Z}$ astfel

încât $a \equiv x^2 \pmod{n}$. Următoarea teoremă face legătura dintre reziduozație relativ la un modul prim și reziduozație relativ la un modul compus.

Teorema 2.7.3.2. Fie $n > 2$ un număr impar cu descompunerea în factori primi $n = p_1^{e_1} \cdots p_k^{e_k}$. Atunci, un întreg a relativ prim cu n este reziduu pătratic modulo n dacă și numai dacă a este reziduu pătratic modulo p_i , pentru orice i cu $1 \leq i \leq k$.

Demonstrație. Fie n și a ca în enunțul teoremei.

Este clar că dacă a este reziduu pătratic modulo n atunci a este reziduu pătratic modulo p_i , pentru orice i cu $1 \leq i \leq k$.

Reciproc, presupunem că a este reziduu pătratic modulo p_i , pentru orice i cu $1 \leq i \leq k$. Vom parcurge 2 pași.

Primul pas este de a face trecerea de la reziduu pătratic modulo p^e la reziduu pătratic modulo p^{e+1} , unde $p > 2$ este număr prim și $e \geq 1$. Deci, să presupunem că un întreg b relativ prim cu p^e este reziduu pătratic modulo p^e . Atunci, există $x, \alpha \in \mathbb{Z}$ astfel încât

$$b = x^2 + \alpha p^e.$$

Pentru orice $y \in \mathbb{Z}$ are loc:

$$(x + yp^e)^2 \equiv b + (2yx - \alpha)p^e \pmod{p^{e+1}}.$$

Dacă y poate fi ales astfel încât $2yx \equiv \alpha \pmod{p}$, atunci $b \equiv (x + yp^e)^2 \pmod{p^{e+1}}$, ceea ce înseamnă că b este și reziduu pătratic modulo p^{e+1} . Însă, $2yx \equiv \alpha \pmod{p}$ este o congruență liniară ce are exact o soluție y în \mathbb{Z}_p deoarece $(2x, p) = 1$ (a se vedea Teorema 2.5.1). Ca urmare b este reziduu pătratic modulo p^{e+1} .

Acest prim pas ne arată că, în baza ipotezei, a este reziduu pătratic modulo $p_i^{e_i}$, pentru orice i cu $1 \leq i \leq k$.

Pasul al doilea constă în utilizarea teoremei chineze a resturilor pentru a deduce concluzia finală. Mai exact, în urma primului pas am obținut că ecuația congruențială $a \equiv x^2 \pmod{p_i^{e_i}}$ are soluții în x , pentru orice i . Aplicând atunci teorema chineză a resturilor vom obține o soluție a congruenței $a \equiv x^2 \pmod{p_1^{e_1} \cdots p_k^{e_k}}$ (a se vedea Teorema 2.6.3). Deci, a este reziduu pătratic modulo n . \square

Simbolul Legendre este un indicator ce caracterizează complet reziduozațiea pătratică a unui întreg modulo un număr prim $p > 2$. Nu același lucru se întâmplă cu simbolul Jacobi. De exemplu, dacă p și q sunt numere prime impare și $(\frac{q}{pq}) = 1$, unde a este un întreg relativ prim cu pq , atunci $(\frac{a}{p}) = 1 = (\frac{a}{q})$ sau $(\frac{a}{p}) = -1 = (\frac{a}{q})$. Ca urmare, $(\frac{q}{pq}) = 1$ ne spune că a poate fi reziduu pătratic modulo p și modulo q (deci și modulo pq), dar poate să nu fie reziduu pătratic nici modulo p și nici q . Însă, dacă $(\frac{q}{pq}) = -1$, atunci a nu este reziduu pătratic modulo pq .

Studiul reziduozației pătratice modulo un întreg compus este extrem de interesant și util în informatică. Indicăm [212] pentru o introducere în acest domeniu. Exten-

sia la reziduuri de ordin înalt este un alt subiect care a atras atenția cercetătorilor informaticieni, în special din punct de vedere al criptografiei.

2.8. Complexitatea operațiilor

2.8.1. Ordine de mărime

Analiza eficienței algoritmilor nu este întotdeauna un lucru simplu, fiind adesea foarte dificil de determinat timpul exact de execuție al unui algoritm. În astfel de situații suntem forțați în determinarea unei aproximări a timpului de execuție și, frecvent, putem determina doar aproximări asymptotice.

În această secțiune vom prezenta principalele *ordine de mărime* prin intermediul cărora vom putea discuta despre comportarea asymptotică a algoritmilor. Reamintim că prin \mathbf{R}_+ (\mathbf{R}_+^*) s-a notat multimea numerelor reale pozitive (strict pozitive).

Fie g o funcție de la \mathbf{N} la \mathbf{R}_+ . Considerăm următoarele mulțimi:

$$\begin{aligned}\mathcal{O}(g) &= \{f : \mathbf{N} \rightarrow \mathbf{R}_+ | (\exists c \in \mathbf{R}_+^*) (\exists n_0 \in \mathbf{N}) (\forall n \geq n_0) (f(n) \leq cg(n))\} \\ \Omega(g) &= \{f : \mathbf{N} \rightarrow \mathbf{R}_+ | (\exists c \in \mathbf{R}_+^*) (\exists n_0 \in \mathbf{N}) (\forall n \geq n_0) (cg(n) \leq f(n))\} \\ \Theta(g) &= \{f : \mathbf{N} \rightarrow \mathbf{R}_+ | (\exists c_1, c_2 \in \mathbf{R}_+^*) (\exists n_0 \in \mathbf{N}) (\forall n \geq n_0) \\ &\quad (c_1 g(n) \leq f(n) \leq c_2 g(n))\} \\ o(g) &= \{f : \mathbf{N} \rightarrow \mathbf{R}_+ | (\forall c \in \mathbf{R}_+^*) (\exists n_0 \in \mathbf{N}) (\forall n \geq n_0) (f(n) \leq cg(n))\}\end{aligned}$$

Definiția 2.8.1.1. Fie f și g funcții de la \mathbf{N} la \mathbf{R}_+ , și $X \in \{\mathcal{O}, \Omega, \Theta, o\}$. Spunem că f este de ordinul X al lui g , și notăm $f(n) = X(g(n))$, dacă $f \in X(g)$.

Notația “ \mathcal{O} ” a fost introdusă de Paul Bachmann în 1894 [4], fiind apoi popularizată intens de Edmund Landau [114, 115], în timp ce notația “ o ” îi este datorată lui Landau [114].⁹

Intuitiv, “ $f(n) = \mathcal{O}(g(n))$ ” înseamnă că f nu crește mai repede, din punct de vedere asymptotic, decât g (eventual multiplicată printr-o constantă). Atragem atenția asupra notației “ $f(n) = \mathcal{O}(g(n))$ ”; ea nu trebuie gândită ca o egalitate, ci ca apartenența funcției f la mulțimea $\mathcal{O}(g)$. Citorul s-ar putea arăta nedumerit, și pe bună dreptate, de adoptarea a încă unei notații (cea prin “ $=$ ”) atâtă timp cât notația prin “ \in ” este clară și corectă din punct de vedere formal. Adoptarea acestei noi notații este datorată faptului că aceasta este încetătenită în rândul matematicienilor și informaticienilor.

⁹Toate acestea notații pot fi considerate într-un cadru mai general, cel al funcțiilor definite pe \mathbf{R} cu valori în \mathbf{R} . Pentru necesitățile noastre, varianta deja considerată este suficientă.

Evident, o notație de genul $f(n) \neq \mathcal{O}(g(n))$ înseamnă că f nu este de ordinul \mathcal{O} al lui g .

Următoarea propoziție, a cărei demonstrație urmează cu ușurință de la definiții, prezintă câteva din proprietățile de bază ale ordinelor de mărime.

Propoziția 2.8.1.1. Fie f, g, h și k funcții de la \mathbf{N} la \mathbf{R}_+ . Atunci, au loc următoarele proprietăți:

- (1) $f(n) = \mathcal{O}(f(n))$;
- (2) dacă $f(n) = \mathcal{O}(g(n))$ și $g(n) = \mathcal{O}(h(n))$, atunci $f(n) = \mathcal{O}(h(n))$;
- (3) $f(n) = \mathcal{O}(g(n))$ dacă și numai dacă $g(n) = \Omega(f(n))$;
- (4) $f(n) = \Theta(g(n))$ dacă și numai dacă $f(n) = \mathcal{O}(g(n))$ și $f(n) = \Omega(g(n))$;
- (5) dacă $f(n) = \mathcal{O}(h(n))$ și $g(n) = \mathcal{O}(k(n))$, atunci $(f \cdot g)(n) = \mathcal{O}(h(n)k(n))$ și $(f + g)(n) = \mathcal{O}(\max\{h(n), k(n)\})$;
- (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 și numai dacă $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$.

Următoarele inegalități sunt foarte utile în stabilirea ordinelor de mărime pentru diverse funcții:

- (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 și 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 obține:

- dacă f este un polinom de grad k cu coeficienți reali pentru care funcția polinomială asociată (notată tot prin f) ia valori reale pozitive, atunci $f(n) = \Theta(n^k)$.

În adevăr, fie $f(x) = a_0 + a_1x + \dots + a_kx^k$ cu $a_k \neq 0$. Dacă $f(n) \geq 0$ pentru orice număr natural n , atunci putem scrie

$$0 \leq \frac{f(n)}{n^k} = a_k + \underbrace{\frac{a_{k-1}}{n} + \dots + \frac{a_0}{n^k}}_A.$$

Cum A tinde la zero pentru n tînzând la infinit, a_k nu poate fi negativ. Cum $a_k \neq 0$, deducem $a_k > 0$.

Acum putem arăta că $f(n) = \Theta(n^k)$:

- $f(n) \leq n^k(|a_0| + \dots + |a_k|)$, pentru orice $n > 0$. Deci, dacă alegem $c_2 = |a_0| + \dots + |a_k| > 0$ vom obține $f(n) \leq c_2 n^k$, pentru orice $n > 0$;
- deoarece $a_k > 0$, există c_1 și $n_0 > 0$ astfel încât

$$0 < c_1 \leq \frac{f(n)}{n^k} = a_k + \underbrace{\frac{a_{k-1}}{n} + \dots + \frac{a_0}{n^k}}_A.$$

pentru orice $n \geq n_0$.

Ca urmare, $f(n) = \Theta(n^k)$;

- pentru orice constantă reală $c > 1$, $\log_c n = \Theta(\log n)$ (\log reprezintă funcția logaritm în baza 2);
- pentru orice număr real ϵ cu $0 < \epsilon < 1$, $\log n = \mathcal{O}(n^\epsilon)$;
- pentru orice număr natural $k \geq 1$, $\log^k n = \mathcal{O}(n)$;
- $n! = \Omega(2^n)$ și $n! = o(n^n)$ ¹⁰;
- $\log(n!) = \Theta(n \log n)$;
- dacă $f : \mathbf{N} \rightarrow \mathbf{R}_+$ este o funcție 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 atenția asupra unor situații de genul $4^n \neq \mathcal{O}(2^n)$ ¹¹.

Dacă \mathcal{A} și \mathcal{B} sunt mulțimi de funcții ca cele definite mai sus ($\mathcal{O}(g)$ etc.), iar f este o funcție de la \mathbf{N} la \mathbf{R}_+ , atunci vom nota:

- $f + \mathcal{A} = \{f + g | g \in \mathcal{A}\}$;

¹⁰De fapt, observăm că $\lim_{n \rightarrow \infty} \frac{n!}{\left(\frac{n}{e}\right)^n \sqrt{2\pi n}} = 1$, ceea ce ne arată că $\left(\frac{n}{e}\right)^n \sqrt{2\pi n}$ constituie o și mai bună aproximare asymptotică pentru $n!$. Astfel, este interesant de remarcat că, pentru $n = 100$, $e^{\frac{1}{1201}} \approx 1.00083299$ și $e^{\frac{1}{1200}} \approx 1.00083368$. Ca urmare, formula lui Stirling oferă margini superioare și inferioare foarte apropiate pentru aproximarea lui $n!$ (pentru $n \geq 100$ diferența dintre acestea este mai mică decât 10^{-6}).

¹¹Dacă presupunem, prin contradicție, că $4^n = \mathcal{O}(2^n)$, atunci există o constantă reală $c > 0$ și 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.

- $\mathcal{A} + \mathcal{B} = \{f + g | f \in \mathcal{A}, g \in \mathcal{B}\}$;
- $f\mathcal{A} = \{f \cdot g | g \in \mathcal{A}\}$. Dacă f este funcția constantă c , atunci vom scrie $c\mathcal{A}$ în loc de $f\mathcal{A}$;
- $\mathcal{AB} = \{fg | f \in \mathcal{A}, g \in \mathcal{B}\}$;
- $\mathcal{O}(\mathcal{A}) = \bigcup_{f \in \mathcal{A}} \mathcal{O}(f)$.

Egalitatea $\mathcal{A} = \mathcal{B}$ va fi înțeleasă prin inclusiune (pentru orice funcție f , dacă $f \in \mathcal{A}$, atunci $f \in \mathcal{B}$). Această ultimă convenție este în strânsă legătură cu convenția deja adoptată ($f(n) = \mathcal{O}(g(n))$, de exemplu). Astfel, $\mathcal{O}(f(n)) = \mathcal{O}(g(n))$ ne spune că orice funcție ce este de ordinul \mathcal{O} al lui f este și de ordinul \mathcal{O} al lui g (dar nu în mod necesar și invers), iar $f(n) + \mathcal{O}(g(n)) = \mathcal{O}(h(n))$ ne spune că suma dintre f și o funcție de ordinul \mathcal{O} al lui g este o funcție de ordinul \mathcal{O} al lui h . De exemplu,

$$\frac{1}{3}n^3 + \mathcal{O}(n^2) = \mathcal{O}(n^3).$$

Putem accepta și notații de genul $f(n) = g(n) + \mathcal{O}(h(n))$ pentru a specifica faptul că f este suma dintre g și o funcție de ordinul \mathcal{O} al lui h (sau, altfel spus, f este un element al mulțimii $g + \mathcal{O}(h)$).

Următoarea propoziție urmează cu ușurință de la definiții (dar atragem încă o dată atenția asupra faptului că egalitatea dintre mulțimile noastre de funcții desemnează de fapt inclusiune de la stânga la dreapta).

Propoziția 2.8.1.2. Fie f și g funcții de la \mathbf{N} la \mathbf{R}_+ și $c \in \mathbf{R}_+$. Atunci au loc următoarele proprietăți:

- (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) $\mathcal{O}(f(n)g(n)) = f(n)\mathcal{O}(g(n))$.

2.8.2. Timpul de execuție al unui algoritm

Estimarea “timpului” necesar execuției unui algoritm (calculării unei funcții) 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 ¹²¹³. Menționăm 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 $0 \leq i < k$, și $n_{k-1} \neq 0$. Această reprezentare este complet determinată de secvența de numere

$$(n_{k-1}, \dots, n_0)_b,$$

motiv pentru care această secvență 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 subînțeleasă din context), n_{k-1} se numește *cifra cea mai semnificativă*, iar n_0 *cifra cea mai puțin 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*.

Relația dintre n și k este dată prin

$$k = \lfloor \log_b n \rfloor + 1$$

(cu convenția $\log_b 0 = 0$). Ca urmare, funcția $f(n)$ ce furnizează lungimea reprezentării în baza b a lui n satisface relația $f(n) = \Theta(\log_b n)$.

Atunci când $b \leq 10$, notația în baza b va fi simplificată la $(n_{k-1} \cdots n_0)_b$. Putem face această simplificare și pentru $10 < b \leq 35$ dacă folosim, de exemplu, alfabetul englez și renotăm numerele 10, 11 etc. prin A, B etc., în această ordine. Astfel, $(1A2)_{16}$ constituie aceeași reprezentare în baza 16 ca și secvența $(1, 10, 2)_{16}$. Atunci când vom lucra cu baza 2 sau 10 vom folosi o nouă simplificare, obținută prin eliminarea parantezelor și a indicelui bazei. Cifrele utilizate pentru scrierea unui număr în baza 2 sunt numite *cifre binare* sau *biți*. *Secvențele binare* sunt secvențe de biți. Uneori este convenabil să completăm la stânga cu zerouri reprezentarea în baza b a lui n . Ne vom referi la șirurile obținute ca fiind tot reprezentarea în baza b a lui n (această convenție are caracter pur tehnic).

Presupunem că cititorul este familiarizat cu operațiile uzuale de adunare, scădere, înmulțire și împărțire cu numere scrise într-o bază b . Acestea se realizează prin repetarea de un număr finit de ori a următoarelor operații considerate primitive:

- compararea a două cifre ale bazei;
- adunare, scădere și înmulțire a 2 cifre ale bazei, luând în considerare și transportul, având drept răspuns o cifră a bazei și un transport;

¹²O bază de numerație este un număr natural $b \geq 2$. Numerele i , cu $0 \leq i < b$, sunt numite *cifre bazei b* .

¹³Reprezentarea internă a datelor în calculator se face utilizând codificarea binară a acestora. Operațiile cu date sunt astfel convertite în operații cu șiruri binare (deplasare la stânga sau la dreapta cu o poziție, adunare de șiruri binare etc.). Ca urmare, complexitatea executării de către calculator a unor operații cu anumite tipuri de date se rezumă la complexitatea realizării unor operații cu șiruri binare, aceasta din urmă fiind "măsurată" în raport cu lungimea șirurilor.

- împărțirea unui număr format din două cifre ale bazei la o cifră a bazei, având drept răspuns un cât și un rest (ambele fiind cifre ale bazei).

Regulile de bază folosite în evaluarea complexității unui algoritm sunt:

- complexitatea unei structuri repetitive (de tip *for*, *while*, *until*) este dată de complexitatea testului la care se adaugă complexitatea maximă a corpului structurii, înmulțind totul cu numărul de repetări ale acestuia;
- complexitatea unei structuri secvențiale 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 operațiilor de bază, cum ar fi adunarea, scăderea, înmulțirea etc. Vom urma cu precădere [210], unde cititorul poate găsi detalii complete asupra algoritmilor discutați sumar mai jos și a complexității acestora.

Adunarea și scăderea a două numere cu reprezentare binară pe cel mult k biți pot fi realizate în complexitate timp $\mathcal{O}(k)$.

Înmulțirea realizată în variantă "școlărească" necesită 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 înmulțirea realizată prin transformata Fourier discretă este eficientă doar pentru valori mari ale lui k (de obicei, cel puțin 1000), în practică, algoritmul Karatsuba se dovedește de preferat.

Împărțirea realizată prin algoritmul uzual de împărțire necesită complexitate timp $\mathcal{O}(k^2)$. Ea poate fi realizată mult mai eficient utilizând un rezultat care spune că, la un pas de împărțire, o cifră a câtului se poate determina prin cel mult două încercări. Astfel, determinarea câtului q și a restului r al împărțirii lui a la b se poate face în complexitate timp $\mathcal{O}((\log b)(\log q))$. Există și o variantă recursivă a împărțirii, bazată oarecum pe ideea ce stă la baza algoritmului Karatsuba, ce necesită $\mathcal{O}(k^{\log 3} + k \log k)$.

O analiză simplistă a algoritmului lui Euclid ne arată că acesta are complexitatea $\mathcal{O}(k^3)$. Însă o analiză atentă conduce la un rezultat mult mai bun. Împărțirea lui r_i la r_{i+1} cu obținerea câtului q_{i+2} și a restului r_{i+2} , utilizând notațiile din Secțiunea 2.2, se poate realiza în $\mathcal{O}((\log r_{i+1})(\log q_{i+2}))$. Atunci

$$\begin{aligned} \sum_{i=-1}^{n-1} (\log r_{i+1})(\log q_{i+2}) &\leq (\log b) \sum_{i=-1}^{n-1} \log q_{i+2} \\ &= (\log b)(\log q_1 \cdots q_{n+1}) \end{aligned}$$

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 biți, atunci complexitatea algoritmului lui Euclid este $\mathcal{O}(k^2)$.

Există și alți algoritmi de calcul al celui mai mare divizor comun a două numere. Cea mai eficientă soluție este de complexitate $\mathcal{O}(k^2 / \log k)$.

Exponențierea modulară, adică calculul lui $a^n \text{ mod } m$, necesită complexitate timp $\mathcal{O}(k^3)$, presupunând că a , n și m se reprezintă pe cel mult k biți. Există multe soluții pentru această problemă, depinzând de diverse particularități ale exponentului sau modulului, soluții de complexitate mult mai bună decât cea menționată mai sus. Toate acestea pot fi găsite în [210].

Simbolul Jacobi al lui a modulo n se poate calcula folosind criteriul lui Euler în complexitate timp $\mathcal{O}(\log^3 n)$ deoarece este necesară doar o exponențiere modulară. Însă, dacă aplicăm regulile din Teorema 2.7.3.2 obținem un algoritm cu o comportare similară algoritmului lui Euclid. La fiecare pas se reduce a modulo n cu rezultatul tot în a , după care ori se aplică o formulă de determinare a simbolului Jacobi ori se schimbă rolurile lui a și n între ele. Oricare din aceste operații necesită, în cel mai defavorabil caz, determinarea restului prin împărțire la 4 sau 8. Ca urmare, algoritmul astfel descris are complexitatea timp $\mathcal{O}((\log a)(\log n))$.

Atunci când pentru o problemă nu se cunoaște nici un algoritm de complexitate timp polinomială, iar algoritmii existenți sunt impracticabili pentru valori rezonabile ale datelor de intrare, vom spune că problema este *dificilă* sau *greă* sau *intractabilă*. De exemplu, factorizarea numerelor este o problemă dificilă. Pentru a înțelege ce înseamnă aceasta, aducem la cunoștința cititorului următorul rezultat. Pe data de 9 mai 2005, o echipă a Agenției Federale Germane pentru securitatea informației, compusă din F. Bahr, M. Boehm, J. Franke și T. Kleinjung, a anunțat factorizarea unui număr de 200 de cifre, număr cunoscut sub denumirea de RSA-200. Acest număr face parte dintr-o selecție de numere propuse de compania americană *RSA Security*, numere ce au exact 2 factori primi și care sunt considerate ca fiind dificil de factorizat (“pietre de încercare” pentru problema factorizării). Echipa germană a utilizat pentru factorizarea acestui număr o rețea 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 sfârșitul anului 2003 și factorizarea s-a încheiat cu succes în mai 2005.

O altă poveste interesantă asupra factorizării numerelor mari este aceea a numărului RSA-768 pe care cititorul o poate găsi în [104].

Capitolul 3

Semigrupuri și monoizi

Așa cum spune L.E. Dickson în 1905 în “On Semigroups and the General Isomorphism Between Infinite Groups”, terminologia de “semigrup” ar fi fost utilizată pentru prima dată de Monsieur l’Abbé J.A. Séquier în cartea sa “Eléments de la théorie des groupes abstraits”, publicată în 1904 la Paris. Dickson, în lucrarea menționată mai sus, introduce explicit conceptul de semigrup ca fiind o structură asociativă și cu simplificare, iar în 1916, O.J. Schmidt utilizează din plin semigrupuri în cartea sa “Abstract Group Theory”.

Studiile consistente pe teoria semigrupurilor și monoizilor încep prin lucrarea lui Suschkewitsch din 1928 [200] și continuă prin cea a lui Rees din 1940 [170] și apoi cea a lui Dubreil din 1941 [48]. Începând cu perioada anilor 1940, teoria semigrupurilor se dezvoltă mult ca urmare a identificării multor structuri de tip semigrup în diverse domenii ale matematicii și, apoi, ca urmare a aplicațiilor acestora în domenii noi, cum ar fi informatica.

Printre lucrările standard ce au marcat trajectoria teoriei semigrupurilor până în zilele noastre menționăm [201, 128, 32, 33, 167, 92, 113, 89]. Evident, există multe alte tratate de teoria semigrupurilor și aplicații ale acestora.

În acest capitol ne propunem o trecere în revistă a unor concepte de bază de teoria semigrupurilor, concepte ce au importanță majoră în informatică. Drept urmare, accentul va cădea pe aplicații, mai exact pe teoria codurilor de lungime variabilă care beneficiază din plin de teoria semigrupurilor.

3.1. Definiții și exemple

În Secțiunea 1.4.4 s-au introdus conceptele de semingrup și monoid ca fiind algebrelle cu proprietăți particulare. Astfel, un *semigrup* este o algebră $\mathbf{S} = (S, \cdot)$ în care \cdot este o operație binară asociativă, iar un *monoid* este o algebră $\mathbf{M} = (M, \cdot, e)$ în care \cdot este

o operație binară asociativă iar e , numită *unitatea monoidului* și notată uneori și prin 1_M , este o operație nulară ce verifică relația

$$e \cdot a = a \cdot e = a,$$

pentru orice $a \in M$.

Toate conceptele de bază asupra semigrupurilor și monoizilor, introduse în Secțiunea 1.4.4, vor fi utilizate în cele ce urmează. Ca urmare, pentru conceptele și rezultatele ce sunt menționate aici, dar nu sunt explicate, cititorul este îndrumat către respectiva secțiune. Semnul operației binare, atât pentru semigrupuri cât și pentru monoizi, va fi omis de cele mai multe ori (dar vom avea grijă să nu fie generate ambiguități prin aceasta).

Dacă un monoid (M, \cdot, e) , vom nota prin (S_M, \cdot) semigrupul obținut din acest monoid prin eliminarea unității și restricționarea operației binare la $S_M = M - \{e\}$ (așa cum se vede, restricția acestei operații o notăm tot prin \cdot).

Fie (S, \cdot) un semigrup, $A, B \subseteq S$ și $a \in S$. Următoarele notații vor fi utilizate intens:

1. $AB = \{ab | a \in A, b \in B\}$;
2. $A^1 = A$ și $A^{n+1} = A^n A$, pentru orice $n \geq 1$;
3. $aA = \{a\}A$ și $Aa = A\{a\}$;
4. $aS^1 = aS \cup \{a\}$, $S^1 a = Sa \cup \{a\}$ și $S^1 a S^1 = SaS \cup Sa \cup aS \cup \{a\}$. Menționăm explicit că acestea notații vor fi relevante numai pentru semigrupuri fără unitate și nu trebuie confundate cu notații de tipul $A^1 a$ etc.;
5. $a^1 = a$ și $a^{n+1} = a^n a$, pentru orice $n \geq 1$.

Aceste notații se extind în mod natural și la monoizi. În plus, pentru acesteia vom adopta și notațiile $A^0 = \{e\}$ și $a^0 = e$, unde e este unitatea monoidului în cauză.

Evident, putem combina aceste notații uzând și de asociativitatea operației binare a semigrupului sau monoidului. De exemplu, putem scrie $AxBc$ pentru

$$AxBc = \{axbc | a \in A, b \in B, c \in C\}.$$

Exemplul 3.1.1.

- (1) Fiecare din mulțimile **N**, **Z**, **Q**, **R** și **C** formează monoid comutativ de ordin ∞ cu operația de adunare ca operație binară și 0 ca operație nulară.
- (2) Fiecare din mulțimile **N**, **Z**, **Q**, **R** și **C** formează monoid comutativ de ordin ∞ cu operația de înmulțire ca operație binară și 1 ca operație nulară.
- (3) **R**₊, mulțimea numerelor reale pozitive, formează monoid comutativ de ordin ∞ cu operația binară $\max\{x, y\}$, ce asociază maximum numerelor x și y , și 0 ca operație nulară.

- (4) Mulțimea tuturor funcțiilor de la o mulțime A la ea însăși formează monoid cu operația de compunere și operația nulară 1_A (funcția identică pe A). Dacă A este finită, atunci ordinul acestui monoid este finit.

Definiția 3.1.1. Fie (S, \cdot) un semigrup și $z \in S$.

- (1) z se numește *element zero la stânga* sau *zero la stânga* al semigrupului dacă are loc $za = z$, pentru orice $a \in S$.
- (2) z se numește *element zero la dreapta* sau *zero la dreapta* al semigrupului dacă are loc $az = z$, pentru orice $a \in S$.
- (3) z se numește *element zero* sau *zero* al semigrupului dacă z este zero la stânga și la dreapta al semigrupului.

Este clar că dacă un semigrup are un zero la stânga și unul la dreapta, atunci aceștia coincid. Ca urmare, un semigrup poate avea cel mult un zero.

Definiția 3.1.1 se extinde în mod natural și la monoizi. În acest caz, dacă un monoid (M, \cdot, e) are doar un element, atunci unitatea lui este și zero. Evident, acesta este un caz cu totul izolat.

Un semigrup (monoid) ce are un zero (la stânga, la dreapta) va fi numit *semigrup (monoid) cu zero (la stânga, la dreapta)*.

Într-o structură algebrică cu unitate (monoid, grup etc.), inversul unui element a poate fi definit într-un mod natural ca fiind un element a' ce satisfacă $a \cdot a' = a' \cdot a = e$, unde e este unitatea. Într-un semigrup fără unitate, această definiție nu funcționează. Însă ideea principală surprinsă mai sus este că elementul a este compus cu a' , care apoi, compus cu a , produce e .

Definiția 3.1.2. Fie $\mathbf{S} = (S, \cdot)$ un semigrup și $a, b \in S$. b se numește *invers* al lui a dacă are loc $aba = a$ și $bab = b$.

Într-un semigrup, un element poate să nu aibă nici un invers sau poate avea mai mult decât unul. Un element ce admite cel puțin un invers este numit *inversabil*.

Definiția 3.1.3. Fie $\mathbf{M} = (M, \cdot, e)$ un monoid și $a, b \in M$.

- (1) b se numește *invers la stânga* al lui a dacă are loc $ba = e$.
- (2) b se numește *invers la dreapta* al lui a dacă are loc $ab = e$.
- (3) b se numește *invers* al lui a dacă b este invers al lui a atât la stânga, cât și la dreapta.

Remarcăm că Definiția 3.1.3(3) este în concordanță cu Definiția 3.1.2. Orice invers în sensul Definiției 3.1.3(3) este invers și în sensul Definiției 3.1.2.

Un element care admite cel puțin un invers (la stânga, la dreapta) într-un monoid este numit element *inversabil (la stânga, la dreapta)*.

Este clar că dacă un element are un invers, atunci acesta este unic. Din acest motiv, inversul unui element a , atunci când există, se mai notează și prin a^{-1} .

Mulțimea tuturor elementelor inversabile într-un monoid \mathbf{M} formează grup în raport cu operațiile induse de \mathbf{M} . El se notează prin $U(\mathbf{M})$ și se numește *grupul unităților monoidului \mathbf{M}* .

Definiția 3.1.4. Fie (S, \cdot) un semigrup. Un element $a \in S$ este numit *element idempotent* dacă $aa = a$.

Mulțimea elementelor idempotente ale unui semigrup (S, \cdot) se notează prin $E(S, \cdot)$. Dat $a \in E(S, \cdot)$, notăm prin H_a mulțimea

$$H_a = \{b \in S \mid ba = b = ab \wedge (\exists b' \in S)(bb' = a = b'b)\}$$

și o numim *grupul elementului idempotent* a . Terminologia este justificată pentru că, în adevăr, H_a formează grup cu operația binară indusă, operația nulară a și o operație unară care se deduce cu ușurință din definiția mulțimii H_a (remarcăm că proprietatea lui a de a fi idempotent face ca a să fie element al acestei mulțimi).

Atât terminologia de element idempotent, cât și notațiile corespunzătoare, se extind în mod natural la monoizi. În cadrul monoizilor, unitatea acestora este element idempotent.

Definiția 3.1.5. Fie $\mathbf{S} = (S, \cdot)$ un semigrup și $a \in S$.

- (1) a se numește *cu simplificare la stânga* dacă relația $ab = ac$ implică $b = c$, pentru orice $b, c \in S$.
- (2) a se numește *cu simplificare la dreapta* dacă relația $ba = ca$ implică $b = c$, pentru orice $b, c \in S$.
- (3) a se numește *cu simplificare* dacă el este atât cu simplificare la stânga, cât și la dreapta.
- (4) Semigrupul \mathbf{S} este numit *cu simplificare (la stânga, la dreapta)* dacă orice element al lui este cu simplificare (la stânga, la dreapta).

Definiția 3.1.5 se extinde în mod natural și la monoizi. Extensia la grupuri este irelevantă deoarece existența inversului implică automat simplificare la stânga și la dreapta.

Exemplul 3.1.2. (Monoidul funcțiilor de la o mulțime la ea însăși)

Fie A o mulțime și $M(A)$ ($M_0(A)$, $M_1(A)$) mulțimea tuturor funcțiilor (surjective, injective) de la A la A . Atunci:

- (1) $M(A)$, în raport cu compunerea funcțiilor, formează monoid cu simplificare la stânga prin funcții injective și cu simplificare la dreapta prin funcții surjective (conform Corolarului 1.2.5.1);

(2) $M_0(A)$ și $M_1(A)$, în raport cu compunerea funcțiilor, sunt submonoizi ai monoidului $M(A)$;

(3) $M_0(A) \cap M_1(A)$, în raport cu compunerea funcțiilor, formează grup.

Atragem atenția asupra faptului că dacă $A = \emptyset$, atunci $M(A)$ conține o singură funcție, și anume funcția vidă, care este atât injectivă cât și surjectivă.

Definiția 3.1.6. Fie $\mathbf{S} = (S, \cdot)$ un semigrup și $I \subseteq S$ o submulțime nevidă.

- (1) I este numită *ideal stâng* sau *ideal la stânga* al semigrupului \mathbf{S} dacă $SI \subseteq I$.
- (2) I este numită *ideal drept* sau *ideal la dreapta* al semigrupului \mathbf{S} dacă $IS \subseteq I$.
- (3) I este numită *ideal* al semigrupului \mathbf{S} dacă I este atât ideal stâng, cât și drept, al semigrupului \mathbf{S} .
- (4) I este numită *ideal (stâng/la stânga, drept/la dreapta) generat de $A \subseteq S$* , unde A este nevidă, dacă I este intersecția tuturor idealelor (stângi, drepte) ce includ A .
- (5) I este numită *ideal principal (stâng/la stânga, drept/la dreapta) generat de $a \in S$* dacă I este ideal (stâng, drept) generat de $\{a\}$.

Idealul principal (stâng, drept) generat de un element $a \in S$ este exact S^1aS^1 (S^1a, aS^1); el se mai notează și prin $I(a)$ ($L(a), R(a)$).

Definiția 3.1.6 precum și notațiile corespunzătoare se extind în mod natural și la monoizi.

3.2. Relațiile lui Green

Relațiile lui Green, introduse de J.A. Green în 1951 [78], constituie un instrument de bază în clarificarea structurii algebrice a semigrupurilor, în înțelegerea profundă a multor clase importante de semigrupuri.

Conceptul de ideal principal (stâng, drept) conduce în mod natural la considerarea următoarelor relații binare într-un semigrup S :

1. $\mathcal{L} \subseteq S \times S$ dată prin: două elemente $a, b \in S$ sunt în relația \mathcal{L} dacă și numai dacă generează același ideal principal stâng. Altfel spus, $a \mathcal{L} b$ dacă și numai dacă $S^1a = S^1b$;
2. $\mathcal{R} \subseteq S \times S$ dată prin: două elemente $a, b \in S$ sunt în relația \mathcal{R} dacă și numai dacă generează același ideal principal drept. Altfel spus, $a \mathcal{R} b$ dacă și numai dacă $aS^1 = bS^1$;

3. $\mathcal{I} \subseteq S \times S$ dată prin: două elemente $a, b \in S$ sunt în relația \mathcal{I} dacă și numai dacă generează același ideal principal. Altfel spus, $a \mathcal{I} b$ dacă și numai dacă $S^1 a S^1 = S^1 b S^1$.

Deoarece egalitatea pe $\mathcal{P}(S)$ este relație de echivalență, obținem că \mathcal{L}, \mathcal{R} și \mathcal{I} sunt relații de echivalență pe S .

Notăm $\mathcal{H} = \mathcal{L} \cap \mathcal{R}$, care este relație de echivalență pe S deoarece este intersecția a două relații de echivalență, și $\mathcal{D} = \mathcal{L} \vee \mathcal{R}$, care este supremumul mulțimii $\{\mathcal{L}, \mathcal{R}\}$ în laticea completă a relațiilor de echivalență pe S și, deci, este relație de echivalență pe S . Am obținut astfel 5 relații de echivalență pe S numite *relațiile lui Green* sau *echivalențele lui Green*.

Evident, aceste relații pot fi definite și pentru monoizi.

Diagrama Hasse din Figura 3.1 ne arată ordinea parțială între aceste relații binare (s-a inclus atât ι_S , diagonala lui S , cât și ω_S , relația totală pe S).

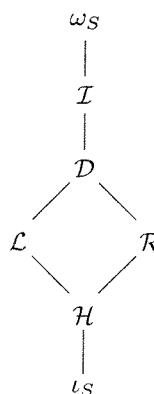


Figura 3.1: Relațiile lui Green

Propoziția 3.2.1. Fie (S, \cdot) un semigrup și $a, b \in S$. Atunci au loc următoarele proprietăți:

- (1) $a \mathcal{L} b$ dacă și numai dacă $a = b$ sau există $x, y \in S$ astfel încât $a = xb$ și $b = ya$;
- (2) $a \mathcal{R} b$ dacă și numai dacă $a = b$ sau există $x, y \in S$ astfel încât $a = bx$ și $b = ay$;
- (3) \mathcal{L} este congruență la dreapta, iar \mathcal{R} este congruență la stânga;
- (4) $\mathcal{L} \circ \mathcal{R} = \mathcal{R} \circ \mathcal{L} = \mathcal{L} \vee \mathcal{R} = \mathcal{D}$;
- (5) dacă S este comutativ, atunci $\mathcal{H} = \mathcal{L} = \mathcal{R} = \mathcal{D} = \mathcal{I}$;
- (6) dacă orice element din S este element zero la dreapta, atunci $\iota_S = \mathcal{H} = \mathcal{L}$ și $\mathcal{R} = \mathcal{D} = \omega_S$;

- (7) dacă S este grup, atunci $\mathcal{H} = \omega_S$;
- (8) H_e coincide cu clasa de echivalență a lui e relativ la \mathcal{H} , pentru orice element $e \in E(S, \cdot)$.

Demonstrație. (1), (2), (3) și (5) urmează imediat de la definiții, iar (8) rămâne în seama cititorului.

(4) Fie $a, b \in S$ cu proprietatea $a(\mathcal{L} \circ \mathcal{R})b$. Atunci există $c \in S$ astfel încât $a \mathcal{L} c \mathcal{R} b$. Dacă $a = c$ sau $c = b$, atunci se obține imediat $a(\mathcal{R} \circ \mathcal{L})b$. Să presupunem acum că există $x, y, u, v \in S$ cu $a = xc$, $c = ya$, $c = bu$ și $b = cv$.

Considerăm $d = xcv$ și arătăm că are loc $a \mathcal{R} d \mathcal{L} b$. Au loc relațiile

$$d = xcv = av$$

și

$$du = xcvu = xbu = xc = a,$$

care conduc la $a \mathcal{R} d$. Similar,

$$d = xcv = xb$$

și

$$yd = yxcv = yav = cv = b,$$

care conduc la $d \mathcal{L} b$.

Că urmare, $\mathcal{L} \circ \mathcal{R} \subseteq \mathcal{R} \circ \mathcal{L}$. În mod similar se stabilește și cealaltă inclusiune.

Pentru a stabili egalitatea $\mathcal{L} \circ \mathcal{R} = \mathcal{R} \circ \mathcal{L} = \mathcal{D}$, observăm că $\mathcal{L} \circ \mathcal{R}$ include \mathcal{L} și \mathcal{R} . Cum $\mathcal{L} \circ \mathcal{R}$ este relație de echivalență, urmează că $\mathcal{D} \subseteq \mathcal{L} \circ \mathcal{R}$ (pentru că \mathcal{D} este cea mai mică relație de echivalență ce include \mathcal{L} și \mathcal{R}). Pe de altă parte, \mathcal{D} trebuie să includă $\mathcal{L} \circ \mathcal{R}$, și deci obținem $\mathcal{D} = \mathcal{L} \circ \mathcal{R}$.

(6) De la definiții urmează $\iota_S \subseteq \mathcal{H} \subseteq \mathcal{L}$. Dacă $a \mathcal{L} b$, conform ipotezei,

$$\{a\} = S^1 a = S^1 b = \{b\},$$

cea ce arată că $a \iota_S b$. Deci $\mathcal{L} \subseteq \iota_S$, stabilind astfel egalitățile $\iota_S = \mathcal{H} = \mathcal{L}$. În mod similar se stabilesc și celealte două egalități: inclusiunile $\mathcal{R} \subseteq \mathcal{D} \subseteq \omega_S$ urmează de la definiții, iar ipoteza conduce la

$$aS^1 = S = bS^1,$$

pentru orice $a, b \in S$. Ca urmare, $\omega_S \subseteq \mathcal{R}$.

(7) Date $a, b \in S$, are loc $a = (ab^{-1})b$ și $b = (ba^{-1})a$. Deci $a \mathcal{L} b$. În mod similar se arată că are loc $a \mathcal{R} b$, și deci $a \mathcal{H} b$. Ca urmare, $\omega_S \subseteq \mathcal{H}$. Cum $\mathcal{H} \subseteq \omega_S$, obținem $\omega_S = \mathcal{H}$. \square

3.3. Clase remarcabile de semigrupuri și monoizi

Vom prezenta în această secțiune câteva clase foarte importante de semigrupuri și monoizi, împreună cu câteva rezultate asupra acestora.

3.3.1. Monoizi de cuvinte

Cuvintele și operațiile pe mulțimea acestora au fascinat oamenii din cele mai vechi timpuri. Probabil că nu se poate stabili o dată precisă de la care s-a început un studiu sistematic al cuvintelor. Din punct de vedere informatic ne vom limita la a lăua în considerare momentul Axel Thue, 1906 [205]. De atunci și până astăzi, teoria cuvintelor finite s-a dezvoltat atât cantitativ, cât și calitativ, căpătând o structură algebrică, cea de monoid liber generat, și beneficiind de proprietăți combinatoriale diverse. Nu de puține ori s-a dovedit că raționamentul pe cuvinte și limbaje (mulțimi de cuvinte) este destul de dificil. Este suficient să menționăm celebrele probleme puse de Axel Thue în 1906 și 1912 [205, 206] relativ la cuvinte libere de patrat sau cub sau celebra problemă a corespondenței a două liste de cuvinte pusă de către Emil Post în 1946 [168]. Cuvintele și limbajele pot fi, și sunt, folosite cu scop de amuzament sau de verificare a iștețimii dar, mai presus de acestea, ele au o importanță deosebită în multe domenii, printre care cel lingvistic și informatic. Tratări moderne în cadrul sintaxei și al semanticii limbajelor de programare, construcției de compilatoare, teoriei codurilor, procesării de imagini, construcției de editoare de texte etc. necesită cunoașterea unor proprietăți combinatoriale ale acestora.

Definiția 3.3.1.1. Se numește *alfabet* orice mulțime nevidă.

Elementele unui alfabet se numesc *litere* sau *simboluri*.

Exemplul 3.3.1.1. Mulțimile

- $\Sigma_1 = \{a, b, c\}$,
- $\Sigma_2 = \{0, 1, 2, 3\}$,
- $\Sigma_3 = \{\text{begin, end, if, then, else, while, do}\}$ și
- $\Sigma_4 = \{0, 1, 10, 100\}$

sunt alfabete.

În alegerea unui alfabet este util, din rațuni tehnice, să considerăm indivizibilitatea literelor alfabetului în raport cu literele același alfabet. Aceasta deoarece vom omite frecvent simbolul operației de concatenare ("alipire") a cuvintelor, și atunci fără presupunerea de mai sus ar putea apărea probleme de ambiguitate a unicității scrierii

unui cuvânt. De exemplu, considerând alfabetul Σ_4 din exemplul anterior, secvența (cuvântul) 100 poate fi gândită ca fiind alcătuită în trei moduri distincte:

$$100, (1)(0)(0), (10)(0)$$

(s-au utilizat parantezele pentru a descrie modul de descompunere a acestei secvențe).

Definiția 3.3.1.2. Fie Σ un alfabet și $k \geq 0$ un număr natural. Se numește *cuvânt de lungime k peste Σ* orice funcție w de la $\{1, \dots, k\}$ la Σ (în cazul $k = 0$, mulțimea $\{1, \dots, k\}$ este considerată mulțimea vidă). Ca urmare, în acest caz w este funcția vidă. Numărul natural k se numește *lungimea cuvântului w* și se notează prin $|w|$.

Cuvântul $w : \emptyset \rightarrow \Sigma$ este numit și *cuvântul vid sau nul*. În mod ușual el se notează prin λ_Σ sau λ , atunci când Σ se subînțelege din context, sau 1_{Σ^*} (această ultimă notație va fi clară când vom discuta despre structura algebrică a mulțimii cuvintelor peste Σ).

Conform presupunerii că elementele unui alfabet Σ nu pot fi divizate în elemente ale lui Σ , putem scrie cuvintele nevide peste Σ ca secvențe de forma

$$w = w(1) \cdots w(k),$$

unde $w(1), \dots, w(k)$ sunt scrise unele după altele fără spații sau alte simboluri între ele. Mulțimea tuturor cuvintelor de lungime $k \geq 0$ peste Σ va fi notată prin Σ^k , iar mulțimea tuturor cuvintelor peste Σ prin Σ^* . Observăm că $\Sigma^0 = \{\lambda\}$ și $\Sigma^* = \bigcup_{k \geq 0} \Sigma^k$. Mulțimea cuvintelor nevide peste Σ va fi notată prin Σ^+ , adică $\Sigma^+ = \Sigma^* - \{\lambda\}$.

Din rațuni tehnice, cuvintele de lungime 1 vor fi identificate cu elementele alfabetului Σ , adică $\Sigma^1 = \Sigma$. Vom avea deci $|a| = 1$, pentru orice $a \in \Sigma$.

Definiția 3.3.1.3. Spunem că două cuvinte w_1 și w_2 peste același alfabet sunt *egale* și notăm $w_1 = w_2$, dacă ele au aceeași lungime și

$$(\forall i)(1 \leq i \leq |w_1| = |w_2| \Rightarrow w_1(i) = w_2(i)).$$

Constatăm că λ nu poate fi egal decât cu λ .

Evident, egalitatea pe mulțimea cuvintelor peste un alfabet este o relație reflexivă, simetrică și tranzitivă.

Pe mulțimea cuvintelor peste un alfabet Σ introducem o operație binară foarte importantă, *concatenarea*. Ea va fi operația-cheie în lucrul pe cuvinte.

Definiția 3.3.1.4. Fie Σ un alfabet. Operația binară $\cdot : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$ dată prin

$$w_1 \cdot w_2 : \{i | 1 \leq i \leq |w_1| + |w_2|\} \rightarrow \Sigma,$$

unde

$$(w_1 \cdot w_2)(i) = \begin{cases} w_1(i), & \text{dacă } 1 \leq i \leq |w_1| \\ w_2(i - |w_1|), & \text{altfel,} \end{cases}$$

pentru orice i , se numește operația de *concatenare* a cuvintelor peste Σ .

Este clar că $\lambda \cdot \lambda = \lambda$ și $\lambda \cdot w = w \cdot \lambda = w$, pentru orice cuvânt nevid w .

Concatenarea este o operație binară asociată implicit cu alfabetul. Concatenarea a două cuvinte produce un nou cuvânt a căruia lungime este suma lungimilor celor două cuvinte. Notațional, ori de câte ori nu este pericol de confuzie, vom omite semnul operației de concatenare dintre elemente. Astfel, în loc de $w_1 \cdot w_2$ vom scrie $w_1 w_2$.

Structura algebrică a mulțimilor Σ^+ și Σ^* este dată de următoarea teoremă a cărei demonstrație este lăsată în seama cititorului.

Teorema 3.3.1.1. Fie Σ un alfabet. Atunci:

- (1) (Σ^+, \cdot) este semigrup generat de Σ ;
- (2) $(\Sigma^*, \cdot, \lambda)$ este monoid generat de Σ ;
- (3) $(\Sigma^*, \cdot, \lambda)$ este monoid cu simplificare;
- (4) funcția $l : \Sigma^* \rightarrow \mathbf{N}$ dată prin $l(w) = |w|$, pentru orice $w \in \Sigma^*$, este homomorfism de la monoidul $(\Sigma^*, \cdot, \lambda)$ la monoidul $(\mathbf{N}, +, 0)$, ce satisfacă $l^{-1}(0) = \{\lambda\}$;
- (5) grupul unităților monoidului $(\Sigma^*, \cdot, \lambda)$ este trivial.

Punctul 2 al acestei teoreme ne arată că λ este element neutru al monoidului cuvintelor peste Σ , ceea ce justifică și notația alternativă 1_{Σ^*} ce poate fi utilizată pentru a desemna cuvântul vid peste Σ .

Următoarea teoremă reprezintă baza studiului ecuațiilor peste Σ^* (teorema fiind de fapt validă în orice monoid liber generat, aşa cum se va vedea ulterior).

Teorema 3.3.1.2. (Teorema lui Levi)

Fie x, y, u și v cuvinte peste un alfabet Σ astfel încât $xy = uv$.

- (1) Dacă $|x| < |u|$, atunci există un unic $z \in \Sigma^*$ astfel încât $u = xz$.
- (2) Dacă $|x| = |u|$, atunci $x = u$ și $y = v$.
- (3) Dacă $|x| > |u|$, atunci există un unic $z \in \Sigma^*$ astfel încât $x = uz$.

Demonstrația acestei teoreme poate fi ușor făcută de cititor uzând eventual de reprezentarea grafică a cuvintelor xy și uv . De exemplu, în cazul $|x| < |u|$, reprezentarea grafică ar putea fi cea din Figura 3.2.

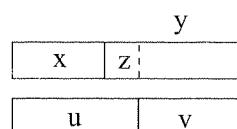


Figura 3.2: Teorema lui Levi

Definiția 3.3.1.5. Fie Σ un alfabet și $u, v \in \Sigma^*$.

- (1) Spunem că u este *prefix* sau *stâng* al lui v dacă există $w \in \Sigma^*$ astfel încât $v = uw$.
- (2) Spunem că u este *sufix* sau *drept* al lui v dacă există $w \in \Sigma^*$ astfel încât $v = wu$.
- (3) Spunem că u este *subcuvânt* al lui v dacă există $x, y \in \Sigma^*$ astfel încât $v = xuy$.

Vom nota prin $\text{Pref}(u)$ ($\text{Suf}(u)$, $\text{Sub}(u)$) mulțimea tuturor prefixelor (sufixelor, subcuvintelor) lui u . Observăm că pentru orice $u \in \Sigma^*$ are loc:

- $\{\lambda, u\} \subseteq \text{Pref}(u) \cap \text{Suf}(u)$;
- $\text{Pref}(u) \cup \text{Suf}(u) \subseteq \text{Sub}(u)$

(prima incluziune are loc prin egalitate dacă și numai dacă $u \in \Sigma \cup \{\lambda\}$, iar cea de a doua dacă și numai dacă $|u| \leq 2$).

Este adesea util de considerat ordinea pe cuvinte indușă de proprietatea de prefix (sufix, subcuvânt). Vom nota această ordine prin \leq_{pref} ($\leq_{\text{suf}}, \leq_{\text{sub}}$). De exemplu, $x \leq_{\text{pref}} y$ dacă și numai dacă x este prefix al lui y . Este ușor de verificat că acestea sunt ordini parțiale pe mulțimea cuvintelor.

Următorul rezultat este consecință directă a Teoremei lui Levi.

Corolarul 3.3.1.1. Fie x, y, u și v cuvinte peste un alfabet Σ . Dacă $xy = uv$, atunci $x \leq_{\text{pref}} u$ sau $u \leq_{\text{pref}} x$.

Propoziția 3.3.1.1. Fie x, y, u și v cuvinte peste un alfabet Σ . Dacă $x \leq_{\text{pref}} u$ și $y \leq_{\text{pref}} uv$, atunci $x \leq_{\text{pref}} y$ sau $y \leq_{\text{pref}} x$.

Demonstrație. Deoarece $x \leq_{\text{pref}} u$, există $x' \in \Sigma^*$ astfel încât $u = xx'$. Analog, există $y' \in \Sigma^*$ astfel încât $uv = yy'$. Obținem atunci $xx'v = uv = yy'$. Aplicând acum Corolarul 3.3.1.1 egalități $xx'v = yy'$, deducem $x \leq_{\text{pref}} y$ sau $y \leq_{\text{pref}} x$. \square

În unele situații este util de considerat ordini parțiale pe elementele alfabetelor. Aceste ordini parțiale induc, în mod natural, ordini parțiale pe mulțimea cuvintelor peste alfabetul în cauză.

Definiția 3.3.1.6.

- (1) Se numește *alfabet ordonat* orice cuplu (Σ, \prec) format dintr-un alfabet Σ și o ordine totală \prec pe acest alfabet.
- (2) Fie (Σ, \prec) un alfabet ordonat. Relația binară $\leq_{(\Sigma, \prec)}$ pe Σ^* dată prin

$$x \leq_{(\Sigma, \prec)} y$$

dacă și numai dacă

- $x \leq_{pref} y$ sau
- există $u, v, w \in \Sigma^*$ și $a, b \in \Sigma$ astfel încât $x = uav$, $y = ubw$ și $a \prec b$,

pentru orice $x, y \in \Sigma^*$, se numește *relația de ordine (direct) lexicografică* pe (Σ, \prec) .

Atunci când ordinea totală pe Σ este subînțeleasă din context, vom simplifica notația $\leq_{(\Sigma, \prec)}$ la \leq_Σ . Atributul "direct" se referă la faptul că verificarea coincidenței sirurilor x și y se face de la începutul lor. Dacă verificarea s-ar face de la sfârșitul lor, ordinea lexicografică ar primi atributul "invers". Rămâne în seama cititorului de a da definiția formală a *ordinii invers lexicografice*. Atâtă timp cât atributul "direct" sau "invers" nu este specificat, vom înțelege implicit că ordinea este direct lexicografică.

Următoarea propoziție justifică terminologia de "ordine" și stabilește câteva proprietăți elementare ale ordinii lexicografice.

Propoziția 3.3.1.2. Fie (Σ, \prec) un alfabet ordonat. Atunci au loc următoarele proprietăți:

- (1) $\leq_{(\Sigma, \prec)}$ este ordine totală pe Σ^* ;
- (2) $\leq_{(\Sigma, \prec)}$ este compatibilă la stânga cu concatenarea (adică $x \prec y$ implică $zx \prec zy$, pentru orice $z \in \Sigma^*$).

Demonstrația acestei propoziții este lăsată în seama cititorului. Atragem atenția asupra faptului că ordinea lexicografică nu este, în general, compatibilă la dreapta cu concatenarea.

Ordinea lexicografică este foarte utilă, dar există situații în care se preferă alte tipuri de ordonări totale. Să considerăm, de exemplu, un alfabet $\Sigma = \{a, b\}$ cu două simboluri și cu ordonarea $a \prec b$. Ordinea lexicografică așază cuvintele ca în Figura 3.3(a).

Dacă dorim să prelucrăm cuvintele unul câte unul în ordine lexicografică, atunci constatăm că ab nu poate fi atins după un număr finit de pași. Dacă însă utilizăm ordinea lexicografică pentru a enumera cuvintele în varianta: întâi λ , apoi toate cuvintele de lungime 1 în ordine lexicografică, apoi toate cuvintele de lungime 2 în ordine lexicografică etc., atunci orice cuvânt va fi atins după un număr finit de pași. Pentru alfabetul de mai sus, acest nou tip de ordonare arată ca în Figura 3.3(b).

Rămâne în seama cititorului să definească formal această nouă ordine totală a cuvintelor peste un alfabet ordonat.

În teoria limbajelor formale, submulțimile $L \subseteq \Sigma^*$ se numesc *limbaje (peste Σ)*. Notațiile din Secțiunea 3.1 se aplică, evident, și limbajelor sub operația de concatenare. Astfel, dacă L_1 și L_2 sunt limbaje, iar w este un cuvânt, atunci $L_1 L_2 = \{uv | u \in L_1, v \in L_2\}$, $w L_1 = \{wu | u \in L_1\}$ și $L_1 w = \{uw | u \in L_1\}$.

Dacă un limbaj $L \subseteq \Sigma^*$, prin $\text{alph}(L)$ notăm mulțimea

$$\text{alph}(L) = \{a \in \Sigma | (\exists u, v)(uav \in L)\}.$$

λ	λ
a	a
aa	b
aaa	aa
...	...
$aaab$	bb
aab	aaa
ab	aab
...	...
b	bbb
ba	$aaaa$
baa	$aaab$
...	...

a) b)

Figura 3.3: a) Ordine lexicografică; b) Ordine lexicografică pe cuvinte de aceeași lungime

$\text{alph}(L)$ reprezintă alfabetul minimal peste care poate fi definit limbajul L .

Definiția 3.3.1.7. Fie Σ un alfabet și $L_1, L_2 \subseteq \Sigma^*$.

- (1) Spunem că produsul $L_1 L_2$ este *neambiguu* dacă pentru orice $w \in L_1 L_2$ există două cuvinte unice $u \in L_1$ și $v \in L_2$ astfel încât $w = uv$.
- (2) Spunem că L_1 este *independent la concatenare* dacă $L_1 \cap \bigcup_{n \geq 2} L_1^n = \emptyset$.

3.3.2. Semigrupuri de transformări

În teoria semigrupurilor, funcțiile de la o mulțime A la ea însăși mai sunt numite și *transformări* pe A , iar monoidul $M(A)$ se numește atunci *monoidul transformărilor* pe A , notat și prin $\mathcal{T}(A)$. Orice subsemigrup (submonoid) al lui $\mathcal{T}(A)$ se numește, în acest context, *semigrup (monoid) de transformări* pe A . Homomorfismele de la un semigrup (monoid) \mathbf{S} la un semigrup (monoid) de transformări peste o mulțime A sunt numite în mod ușual *reprezentări* ale lui \mathbf{S} prin transformări peste A ; monomorfismele cu aceeași proprietate sunt numite *reprezentări fidele* ale lui \mathbf{S} .

O subclasa foarte importantă de transformări o constituie cea a translațiilor.

Definiția 3.3.2.1. Fie (S, \cdot) un cuplu format dintr-o mulțime nevidă S și o operație binară \cdot pe ea.

- (1) Se numește *translație la stânga* pe (S, \cdot) orice funcție $f : S \rightarrow S$ pentru care are loc

$$(\forall x, y \in S)(f(x \cdot y) = f(x) \cdot y).$$

- (2) Se numește *translație la dreapta* pe (S, \cdot) orice funcție $f : S \rightarrow S$ pentru care are loc

$$(\forall x, y \in S)(f(x \cdot y) = x \cdot f(y)).$$

Vom nota prin $\mathcal{L}(S, \cdot)$ ($\mathcal{R}(S, \cdot)$) mulțimea translațiilor la stânga (dreapta) pe (S, \cdot) (atrăgând atenția cititorului să nu confundă aceste notări cu relațiile lui Green \mathcal{L} și \mathcal{R}). În raport cu compunerea, acestea formează submonoizi ai monoidului $\mathcal{T}(S)$ (funcția identică este atât translație la stânga cât și la dreapta).

Fie (S, \cdot) un cuplu ca în definiția anterioară, $a \in S$ și $\alpha_a, \beta_a : S \rightarrow S$ funcțiile date prin

$$\alpha_a(x) = a \cdot x \text{ și } \beta_a(x) = x \cdot a,$$

pentru orice $x \in S$.

Propoziția 3.3.2.1. Fie (S, \cdot) un cuplu format dintr-o mulțime nevidă S și o operație binară \cdot pe ea.

- (1) Dacă \cdot este asociativă, atunci α_a este translație la stânga, iar β_a este translație la dreapta, pentru orice $a \in S$. Dacă în plus S admite unitate la stânga (dreapta) în raport cu \cdot , atunci orice translație la stânga (dreapta) este de forma α_a (β_a).

- (2) \cdot este asociativă dacă și numai dacă are loc

$$(\forall a, b \in S)(\alpha_{a \cdot b} = \alpha_a \circ \alpha_b).$$

- (3) \cdot este asociativă dacă și numai dacă are loc

$$(\forall a, b \in S)(\beta_{a \cdot b} = \beta_b \circ \beta_a).$$

Demonstrație. (1) Presupunem că \cdot este asociativă. Pentru orice $a, x, y \in S$ are loc

$$\alpha_a(x \cdot y) = a \cdot (x \cdot y) = (a \cdot x) \cdot y = \alpha_a(x) \cdot y,$$

ceea ce arată că α_a este translație la stânga. Similar pentru β_a .

Să presupunem suplimentar că S admite unitate la stânga în raport cu \cdot , fie e un astfel de element, și fie f este o translație la stânga. Atunci, are loc

$$f(x) = f(e \cdot x) = f(e) \cdot x = \alpha_{f(e)}(x),$$

pentru orice $x \in S$. Alegând $a = f(e)$, obținem $f = \alpha_a$. Similar pentru cazul în care S admite unitate la dreapta în raport cu \cdot și f este translație la dreapta.

- (2) Să presupunem că \cdot este asociativă. Atunci, pentru orice $a, b, x \in S$, are loc

$$\alpha_{a \cdot b}(x) = (a \cdot b) \cdot x = a \cdot (b \cdot x) = \alpha_a(\alpha_b(x)),$$

ceea ce arată că $\alpha_{a \cdot b} = \alpha_a \circ \alpha_b$.

Reciproc, fie $x, y, z \in S$. Deoarece $x \cdot (y \cdot z) = \alpha_x(\alpha_y(z))$ și $(x \cdot y) \cdot z = \alpha_{x \cdot y}(z)$, ipoteza conduce la $x \cdot (y \cdot z) = (x \cdot y) \cdot z$, și deci \cdot este asociativă.

(3) este similară lui (2). \square

Corolarul 3.3.2.1. Orice semigrup (monoid) este izomorf cu un semigrup (monoid) de transformări.

Demonstrație. Fie (M, \cdot, e) un monoid. Considerăm monoidul de transformări $\mathcal{T}(M)$ și submonoidul acestuia $T = \{\alpha_a | a \in M\}$. Evident, T este submonoid al monoidului $\mathcal{T}(M)$ deoarece

$$\alpha_a \circ \alpha_b = \alpha_{ab} \in T,$$

iar $\alpha_e = 1_M$ satisfacă

$$\alpha_a \circ \alpha_e = \alpha_e \circ \alpha_a = \alpha_a.$$

Arătăm că funcția $h : M \rightarrow T$ dată prin $h(a) = \alpha_a$, pentru orice $a \in M$, este izomorfism. Este clar că ea este funcție surjectivă.

Fie $a, b \in M$ astfel încât $h(a) = h(b)$. Atunci $\alpha_a(x) = \alpha_b(x)$, pentru orice $x \in M$. Alegând $x = e$ obținem $a = b$, ceea ce arată că h este funcție injectivă.

Faptul că h este homomorfism urmează imediat de la Propoziția 3.3.2.1.

Pentru cazul semigrupurilor este suficient să observăm că orice semigrup poate fi extins la un monoid prin adăugarea unui nou element cu rolul de unitate. \square

Definiția 3.3.2.2. Fie \mathbf{S} un semigrup, $f \in \mathcal{L}(\mathbf{S})$ și $g \in \mathcal{R}(\mathbf{S})$. Spunem că f și g sunt legate dacă, pentru orice $x, y \in S$, are loc $x \cdot f(y) = g(x) \cdot y$.

Pentru orice $a \in S$, α_a și β_a sunt legate deoarece

$$x \cdot \alpha_a(y) = x \cdot a \cdot y = \beta_a \cdot y,$$

pentru orice x și y .

Mulțimea

$$H(\mathbf{S}) = \{(f, g) | f \in \mathcal{L}(\mathbf{S}), g \in \mathcal{R}(\mathbf{S}), f \text{ și } g \text{ legate}\}$$

se numește *înfășurătoarea de translație* a lui \mathbf{S} . Această mulțime, în raport cu operația

$$(f_1, g_1)(f_2, g_2) = (f_2 \circ f_1, g_1 \circ g_2),$$

formează semigrup (ceea ce se verifică ușor).

Mulțimea

$$H_0(\mathbf{S}) = \{(\alpha_a, \beta_a) | a \in S\}$$

este subsemigrup al lui $H(\mathbf{S})$, iar funcția $h : S \rightarrow H_0(\mathbf{S})$ dată prin $h(a) = (\alpha_a, \beta_a)$, pentru orice $a \in S$, este homomorfism; h este izomorfism dacă și numai dacă are loc

$$(\forall a, b \in S)(\alpha_a = \alpha_b \wedge \beta_a = \beta_b \Rightarrow a = b),$$

sau, echivalent, dacă și numai dacă are loc

$$(\forall a, b \in S)(\forall x \in S)(ax = bx \wedge xa = xb) \Rightarrow a = b.$$

Un semigrup ce satisfac această ultimă relație se numește *slab reductiv*.

Propoziția 3.3.2.2. Fie $\mathbf{S} = (S, \cdot)$ un semigrup, $a \in S$, $f \in \mathcal{L}(\mathbf{S})$ și $g \in \mathcal{R}(\mathbf{S})$. Atunci au loc relațiile

$$f \circ \alpha_a = \alpha_{f(a)} \text{ și } g \circ \beta_a = \beta_{g(a)}.$$

Dacă f și g sunt legate, atunci au loc și relațiile

$$\alpha_a \circ f = \alpha_{g(a)} \text{ și } \beta_a \circ g = \beta_{f(a)}.$$

Demonstrație. Vom demonstra doar prima relație, restul rămânând în grija cititorului. Pentru orice $x \in S$ are loc

$$(f \circ \alpha_a)(x) = f(\alpha_a(x)) = f(a \cdot x) = f(a) \cdot x = \alpha_{f(a)},$$

ceea ce stabilește relația $f \circ \alpha_a = \alpha_{f(a)}$. \square

Direct de la Propoziția 3.3.2.2 obținem:

Corolarul 3.3.2.2. Fie $\mathbf{S} = (S, \cdot)$ un semigrup. Atunci $H_0(\mathbf{S})$ este ideal în $H(\mathbf{S})$.

Demonstrație. Pentru orice $a \in S$ au loc relațiile

$$(\alpha_a, \beta_a)(f, g) = (f \circ \alpha_a, \beta_a \circ g) = (\alpha_{f(a)}, \beta_{f(a)}),$$

ceea ce demonstrează că $H_0(\mathbf{S})$ este ideal drept în $H(\mathbf{S})$. În mod similar se arată că $H_0(\mathbf{S})$ este ideal stâng în $H(\mathbf{S})$. \square

3.3.3. Semigrupuri și monoizi ciclici

Semigrupurile și monoizii ciclici constituie o altă clasă foarte importantă de semigrupuri și monoizi. Reamintim (Sectiunea 1.4.4.2) că un semigrup (monoid) este numit *ciclic* dacă poate fi generat de un singur element al său. Semigrupul $(\mathbb{N} - \{0\}, +)$ este ciclic generat de 1.

Dacă \mathbf{S} este un semigrup și $a \in S$, atunci putem scrie

$$\langle a \rangle_{\mathbf{S}} = \{a, a^2, \dots, a^n, \dots\}.$$

În cazul monoizilor, dacă $\mathbf{M} = (M, \cdot, e)$ este monoid și $a \in M$, atunci putem scrie

$$\langle a \rangle_{\mathbf{M}} = \{e = a^0, a, a^2, \dots, a^n, \dots\}.$$

Ne vom concentra în principal pe semigrupuri ciclice, dar rezultatele prezentate pot fi translate cu ușurință la monoizi ciclici.

Teorema 3.3.3.1. Fie a un element al unui semigrup (S, \cdot) . Atunci, are loc exact una din următoarele două proprietăți:

- (1) puterile lui a sunt distințe două câte două și semigrupul ciclic generat de a este izomorf cu semigrupul ciclic comutativ $(\mathbb{N} - \{0\}, +)$;
- (2) există două numere naturale strict pozitive m și r astfel încât:
 - (a) $a^m = a^{m+r}$;
 - (b) $a^{m+u} = a^{m+v}$ dacă și numai dacă $u \equiv v \pmod r$, pentru orice $u, v \in \mathbb{N}$;
 - (c) $\langle a \rangle = \{a, a^2, \dots, a^{m+r-1}\}$ conține exact $m+r-1$ elemente;
 - (d) mulțimea $K(a) = \{a^m, \dots, a^{m+r-1}\}$ este subgrup ciclic al lui $\langle a \rangle$.

Demonstrație. Dată secvența a, a^2, \dots , există două cazuri de analizat.

Cazul 1: Pentru orice $x, y \geq 1$, dacă $x \neq y$, atunci $a^x \neq a^y$. În acest caz este trivial de verificat că funcția $f : \langle a \rangle \rightarrow \mathbb{N} - \{0\}$ dată prin $f(a^n) = n$, pentru orice $n \geq 1$, este un izomorfism de la $\langle a \rangle$ la $(\mathbb{N} - \{0\}, +)$. Aceasta stabilește (1).

Cazul 2: Există $x, y \geq 1$ cu $x \neq y$ și $a^x = a^y$. Mulțimea

$$\{x \geq 1 | (\exists y > x)(a^x = a^y)\}$$

este nevidă și, fiind submulțime de numere naturale are un cel mai mic element. Fie acesta m . Considerăm acum mulțimea

$$\{y \geq 1 | a^m = a^{m+y}\}$$

care este nevidă, având astfel un cel mai mic element. Fie acesta r . Ca urmare, $a^m = a^{m+r}$, ceea ce demonstrează (2a).

În baza proprietății de la (a) obținem $a^{m+u} = a^{m+(u \pmod r)}$, pentru orice u . Ca urmare, dacă $a^{m+u} = a^{m+v}$, atunci $a^{m+(u \pmod r)} = a^{m+(v \pmod r)}$, pentru orice u și v . Dacă am presupune, de exemplu, că $u \pmod r < v \pmod r$, atunci, compunând egalitatea de mai sus cu $a^{r-(u \pmod r)}$ și folosind relația de la (a), am obțin

$$a^m = a^{m+(v \pmod r)-(u \pmod r)}$$

care, în baza alegerii lui r ar conduce la o contradicție. Deci, $u \equiv v \pmod r$.

Reciproc, dacă $u \equiv v \pmod r$, atunci există $q \in \mathbb{N}$ astfel încât $u = v + qr$ sau $v = u + qr$. Oricare din cele două cazuri conduce la $a^{m+u} = a^{m+v}$, ceea ce demonstrează (2b).

(2c) urmează imediat în baza discuției de mai sus. Să ne ocupăm de (2d). Pentru orice $a^{m+x}, a^{m+y} \in K(a)$,

$$a^{m+x} a^{m+y} = a^{m+(m+x+y) \pmod r} \in K(a).$$

Pentru a determina unitatea lui $K(a)$, care este de forma a^{m+z} cu $0 \leq z < r$, considerăm egalitatea $a^{m+x} a^{m+z} = a^{m+x}$, unde $0 \leq x < r$, care în baza proprietății de

la (2b) este echivalentă cu $m + z \equiv 0 \pmod{r}$. Ca urmare, dacă alegem z astfel încât $0 \leq z < r$ și $m + z \equiv 0 \pmod{r}$, atunci a^{m+z} este unitatea lui $K(a)$ (este ușor de văzut că un astfel de z există întotdeauna). În manieră similară se arată că orice element din $K(a)$ are un invers în $K(a)$. Deci, $K(a)$ cu operația binară indușă de semigrupul S și cu operațiile nulară și unară specificate ca mai sus, formează grup.

Pentru a stabili ciclicitatea lui $K(a)$ va trebui să arătăm că există g cu proprietățile:

- $0 \leq g < r$;
- $a^{k(m+g)} \in K(a)$, pentru orice $k \geq 1$;
- pentru orice $a^{m+x} \in K(a)$ există $k \geq 1$ astfel încât $a^{k(m+g)} = a^{m+x}$.

Este ușor de văzut că dacă alegem g astfel încât $m + g \equiv 1 \pmod{r}$, atunci toate aceste proprietăți vor fi satisfăcute. \square

Numărul $m \geq 1$ din Teorema 3.3.3.1 se numește *indexul lui a în semigrupul* (S, \cdot) , iar r , *perioada lui a*. Atunci când $S = \langle a \rangle$, vom spune că m este indexul, iar r perioada lui (S, \cdot) . Ordinul lui a (a se vedea Secțiunea 1.4.4.2), atunci când este finit, verifică relația:

$$\text{ordin} = \text{index} + \text{perioadă} - 1.$$

Este clar că două semigrupuri ciclice finite sunt izomorfe dacă și numai dacă au același index și aceeași perioadă.

Date două numere naturale $m, r \geq 1$, există semigrupuri ciclice de index m și perioadă r ? Răspunsul este afirmativ. Considerând funcția

$$f : \{1, \dots, m+r\} \rightarrow \{1, \dots, m+r\}$$

dată prin

$$f = \begin{pmatrix} 1 & 2 & \cdots & m & \cdots & m+r-1 & m+r \\ 2 & 3 & \cdots & m+1 & \cdots & m+r & m+1 \end{pmatrix}$$

(adică, $f(1) = 2$, $f(2) = 3$ etc.), este ușor de verificat că această funcție, cu operația de compunere a funcțiilor, generează un semigrup ciclic de index m și perioadă r .

Cum orice două semigrupuri ciclice de index m și perioadă r sunt izomorfe, putem vorbi de *semigrupul ciclic de index m și perioadă r*, notat generic prin $C(m, r)$.

Definiția 3.3.3.1. Un semigrup (monoid) este numit *periodic* dacă fiecare element al lui are ordin finit.

Evident, semigrupurile finite (monoizii finiți) sunt periodice (periodici).

Propoziția 3.3.3.1. Pentru orice semigrup (monoid) periodic și orice element a al acestuia există $n \in \mathbb{N}$ astfel încât a^n este idempotent.

Demonstrație. Pentru orice element a al unui semigrup (monoid) periodic, $K(a)$ din demonstrația Teoremei 3.3.3.1 este grup. Deci el conține un element unitate care este de forma a^n . Evident, a^n este idempotent. \square

Conform Propoziției 3.3.3.1, orice semigrup (monoid) finit conține cel puțin un element idempotent.

Propoziția 3.3.3.2. Pentru orice semigrup (monoid) periodic are loc $\mathcal{D} = \mathcal{I}$.

Demonstrație. Fie $a, b \in S$ astfel încât $a \mathcal{I} b$. Vom arăta că există c astfel încât $a \mathcal{L} c \mathcal{R} b$, ceea ce va arăta că $a (\mathcal{L} \circ \mathcal{R}) b$ și, deci, $\mathcal{I} \subseteq \mathcal{D}$.

Relația $a \mathcal{I} b$ conduce la $SaS \cup Sa \cup aS \cup \{a\} = SbS \cup Sb \cup bS \cup \{b\}$. Există acum mai multe cazuri de analizat. Vom considera doar cazul în care a se poate scrie în forma $a = ubv$, iar b în forma $b = xay$, unde $x, y, u, v \in S$ (celelalte cazuri sunt similare acestuia). Atunci

$$a = ubv = (ux)a(yv) = (ux)ubv(yv) = (ux)^2a(yv)^2 = \dots$$

și

$$b = xay = (xu)b(vy) = (xu)xay(vy) = (xu)^2b(vy)^2 = \dots$$

Cum semigrupul este periodic, va exista m astfel încât $(ux)^m$ este idempotent (Propoziția 3.3.3.1). Fie $c = xa$. Atunci

$$a = (ux)^m a(yv)^m = (ux)^m (ux)^m a(yv)^m = (ux)^m a = (ux)^{m-1}uc,$$

ceea ce arată că are loc $a \mathcal{L} c$.

Fie n astfel încât $(vy)^n$ este idempotent. Observăm că are loc $cy = xay = b$ și

$$\begin{aligned} c &= xa = x(ux)^{n+1}a(yv)^{n+1} = (xu)^{n+1}xay(vy)^n v \\ &= (xu)^{n+1}b(vy)^{2n}v = (xu)^{n+1}b(vy)^{n+1}(vy)^{n-1}v \\ &= b(vy)^{n-1}v, \end{aligned}$$

ceea ce arată că are loc $c \mathcal{R} b$.

Ca urmare, $\mathcal{I} \subseteq \mathcal{D}$. Cum incluziunea $\mathcal{D} \subseteq \mathcal{I}$ urmează de la definiții, propoziția este demonstrată. \square

3.3.4. Semigrupuri regulate și inverse

Elementele regulate așa cum vor fi prezentate mai jos au fost considerate de către von Neumann în studii asupra inelelor [157]. Se dovedește că regularitatea este în strânsă legătură cu inversabilitatea și ambele admit caracterizări naturale și interesante prin intermediul relațiilor lui Green.

Definiția 3.3.4.1.

- (1) Un element a al unui semigrup (monoid) este numit *regulat* dacă există x astfel încât $a = axa$.
- (2) Un semigrup (monoid) este numit *regulat* dacă fiecare element al său este regulat.

Exemplul 3.3.4.1. Multimea tuturor funcțiilor injective de la o mulțime A la ea însăși, cu operația de compunere, formează monoid regulat.

Teorema 3.3.4.1. Fie (S, \cdot) un semigrup. Atunci, următoarele afirmații sunt echivalente:

- (1) (S, \cdot) este semigrup regulat;
- (2) pentru orice $a \in S$, există un element $e \in E(S, \cdot)$ astfel încât $a \mathcal{L} e$;
- (3) pentru orice $a \in S$, există un element $e \in E(S, \cdot)$ astfel încât $a \mathcal{R} e$.

Demonstrație. Presupunem că are loc (1) și demonstrăm (2). Fie $a \in S$. Atunci, există x astfel încât $a = axa$. Considerăm $e = xa$. Relațiile $e = xa$ și $a = axa = ae$ ne arată că are loc $a \mathcal{L} e$. În plus, $ee = xaxa = xa = e$. Deci e este idempotent.

Presupunem că are loc (2) și demonstrăm (1). Fie $a \in S$ și e un element idempotent astfel încât $a \mathcal{L} e$. Atunci există $x, y \in S$ astfel încât $e = xa$ și $a = ye$. Relația $a = ye$ conduce la

$$ae = yee = ye = a,$$

care, combinată cu $e = xa$, conduce la

$$a = ae = axa,$$

ceea ce arată că a este regulat.

În mod similar se arată că (1) și (3) sunt echivalente. \square

Definiția 3.3.4.2. Un semigrup este numit *invers* dacă fiecare element al său are un unic invers.

Exemplul 3.3.4.2. Multimea tuturor funcțiilor injective de la o mulțime A la ea însăși, cu operația de compunere, formează semigrup invers.

Teorema 3.3.4.2. Fie (S, \cdot) un semigrup. Atunci următoarele afirmații sunt echivalente:

- (1) (S, \cdot) este semigrup invers;
- (2) (S, \cdot) este regulat și orice două elemente idempotente comută;
- (3) pentru orice $a \in S$ există un unic element $e \in E(S, \cdot)$ și un unic element $g \in E(S, \cdot)$ astfel încât $a \mathcal{L} e$ și $a \mathcal{R} g$.

Demonstrație. Arătăm că (1) implică (2). Direct de la definiții, dacă S este semigrup invers, atunci el este și regulat.

Fie $e, f \in E(S, \cdot)$. Există atunci un element invers x pentru ef . Deci,

$$efxe = ef \text{ și } xefx = x.$$

Atunci

$$(fxe)(fxe) = f(xefx)e = fxe,$$

ceea ce ne arată că fxe este idempotent și, ca urmare, este element invers al lui însuși. Pe de altă parte,

$$(ef)(fxe)(ef) = ef^2xe^2f = ef xef = ef$$

și

$$(fxe)(ef)(fxe) = fxe^2f^2xe = fxe fxe = fxe,$$

care înseamnă că ef este element invers pentru fxe . Unicitatea inversului conduce atunci la $ef = fxe$, de la care urmează $efe = fxe^2 = fxe$. Aceste relații furnizează $efe = ef$. Schimbând rolurile lui e și f , obținem $efe = fe$, care, combinată cu relația precedentă, conduce la $ef = fe$. Deci orice două elemente idempotente comută.

Vom arăta acum că (2) implică (3). Pentru orice $a \in S$, Teorema 3.3.4.1 implică existența a două elemente e și g , așa cum este în (2), dar exceptând unicitatea acestora. Vom arăta, de exemplu, că e este unic (demonstrația va fi similară și pentru g). Presupunem că, dat un element $a \in S$, există două elemente idempotente, e și f , astfel încât $a \mathcal{L} e$ și $a \mathcal{L} f$. Atunci, $e \mathcal{L} f$ și, $e = f$ sau există $x, y \in S$ astfel încât $xe = f$ și $yf = e$. În cel de-al doilea caz, faptul că e și f comută conduce la

$$e = yf = yff = ef = fe = xee = xe = f,$$

stabilind astfel unicitatea elementului e .

Presupunem acum că are loc (3) și arătăm că are loc (1). Fie $a \in S$. Există atunci un unic element idempotent e astfel încât $a \mathcal{L} e$. Ca urmare, Teorema 3.3.4.1 ne spune că S este regulat. Deci există x astfel încât $a = axa$. Verificăm că elementul xax este un invers al lui a . În adevăr,

$$a(xax)a = axaxa = axa = a$$

și

$$(xax)a(xax) = x(axa)xax = xaxax = x(axa)x = xax.$$

Presupunem că b este un alt invers al lui a . Este trivial de văzut că are loc $a \mathcal{L} xaxa$ și $a \mathcal{L} ba$. Conform ipotezei, $xaxa = ba$. În mod similar, utilizând relația \mathcal{R} , obținem $xax = cb$. Aceste două egalități conduc la

$$xax = (xax)a(xax) = (xax)a(xax) = baxax = bab = b,$$

stabilind astfel unicitatea inversului lui a . \square

3.4. Semigrupuri și monoizi liberi

Semigrupurile și monoizii liberi au proprietatea fundamentală că elementele lor, exceptând unitatea, atunci când este vorba de monoizi, se pot scrie în mod unic sub formă de “produs (finit) de generatori”. Astfel de structuri au importanță matematică intrinsecă și, pe de altă parte, proprietatea de a fi liber generate le face extrem de utile în multe domenii, iar în informatică în mod deosebit.

3.4.1. Definiții. Exemple. Proprietăți de bază

Definiția 3.4.1.1.

- (1) Un semigrup (S, \cdot) este numit *liber generat* de $X \subseteq S$ dacă orice element al său se scrie în mod unic ca produs (finit) de generatori.
- (2) Un semigrup (S, \cdot) este numit *liber generat* dacă există $X \subseteq S$ care să genereze liber S .
- (3) Un monoid (M, \cdot, e) este numit *liber generat* dacă semigrupul (S_M, \cdot) este liber generat.

Vom simplifica adesea terminologia de “semigrup (monoid) liber generat” la cea de “semigrup (monoid) liber”.

O mulțime X ce generează liber un semigrup sau un monoid se mai numește *multime de generatori liberi* pentru semigrupul sau monoidul respectiv.

Scrierea unică a unui element a ca produs de generatori din X poate fi formalizată prin

- există unice $x_1, \dots, x_n \in X$ astfel încât $a = x_1 \cdots x_n$

sau

- pentru orice $x_1, \dots, x_n, y_1, \dots, y_m \in X$, dacă $a = x_1 \cdots x_n = y_1 \cdots y_m$, atunci $n = m$ și $x_i = y_i$, pentru orice $1 \leq i \leq n$.

Observația 3.4.1.1. Un semigrup (monoid) poate avea mai mult de o mulțime de generatori. De exemplu, $(\mathbf{Z}, +)$ poate fi generat de $\{-1, 1\}$ sau de $\{-3, 2\}$, dar nici una din aceste mulțimi nu generează liber acest semigrup.

Există semigrupuri libere? Răspunsul este pozitiv. Pentru orice mulțime nevidă, semigrupul X^+ al cuvintelor peste X este semigrup liber generat de X și, ca urmare, X^* este monoid liber generat de X .

Proprietatea de unicitate a scrierii elementelor unui semigrup (monoid) liber, exceptând unitatea în cazul monoizilor, face ca orice funcție definită pe mulțimea de

generatori liberi ai semigrupului (monoidului) și cu valori într-un semigrup (monoid) arbitrar să poată fi extinsă la un unic homomorfism definit pe întregul semigrup (monoid).

Teorema 3.4.1.1. Fie (S, \cdot) un semigrup liber generat de $X \subseteq S$ și $f : X \rightarrow S'$, unde (S', \circ) este un semigrup arbitrar. Atunci există un unic homomorfism h de la S la S' ce extinde f (adică, $h(x) = f(x)$, pentru orice $x \in X$).

Demonstrație. Definim h prin $h(x_1 \cdots x_n) = f(x_1) \circ \cdots \circ f(x_n)$, pentru orice $x_1, \dots, x_n \in X$. Se arată cu ușurință că h este homomorfism, extinde f și este unicul cu aceste proprietăți. \square

Evident, Teorema 3.4.1.1 poate fi reformulată și pentru monoizi. Diagrama din Figura 3.4 furnizează o imagine grafică a proprietății din această teoremă.

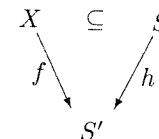


Figura 3.4: Proprietatea din Teorema 3.4.1.1

Corolarul 3.4.1.1. Orice semigrup (monoid) liber este izomorf cu un semigrup (monoid) de cuvinte.

Demonstrație. Ne vom ocupa doar de cazul semigrupurilor. Fie (S, \cdot) un semigrup liber generat de X . Considerăm semigrupul X^+ și funcția $f : X \rightarrow X^+$ dată prin $f(x) = x$, pentru orice $x \in X$. Conform Teoremei 3.4.1.1, această funcție se extinde la un unic homomorfism $h : S \rightarrow X^+$. Este trivial de arătat că h este bijecție, ceea ce încheie demonstrația. \square

Corolarul 3.4.1.2. Orice semigrup (monoid) este izomorf cu un semigrup (monoid) cât al unui semigrup (monoid) de cuvinte.

Demonstrație. Ne vom ocupa doar de cazul semigrupurilor. Fie (S, \cdot) un semigrup și $f : S \rightarrow S$ funcția dată prin $f(a) = a$, pentru orice $a \in S$. Conform Teoremei 3.4.1.1, f se extinde la un unic homomorfism $h : S^+ \rightarrow S$. Deoarece h este epimorfism, $S^+/ker(h)$ va fi izomorf cu S , în baza Teoremei 1.4.4.2. \square

Să considerăm un semigrup (S, \cdot) generat de o submulțime A a sa. Obiectele semigrupului pot fi structuri destul de complexe și, uneori este util să folosim o altă mulțime de obiecte prin care să ne referim la elementele mulțimii A care generează semigrupul. Dacă X este o astfel de mulțime și $\varphi : X \rightarrow S$ este funcția care asociază nume din X elementelor din A ($\varphi(X) = A$), atunci putem spune că X este mulțime de generatori pentru (S, \cdot) prin intermediul funcției φ .

Definiția 3.4.1.2. Fie (S, \cdot) un semigrup, X o mulțime și $\varphi : X \rightarrow S$ o funcție. Spunem că X este o *mulțime de φ -generatori (liberi) pentru (S, \cdot)* sau că X generează (liber) (S, \cdot) relativ la $\varphi : X \rightarrow S$ dacă $\varphi(X)$ generează (liber) (S, \cdot) .

Teorema 3.4.1.1 poate fi acum extinsă astfel:

Teorema 3.4.1.2. Fie (S, \cdot) un semigrup, X o mulțime și $\varphi : X \rightarrow S$ o funcție. Dacă X generează liber (S, \cdot) relativ la $\varphi : X \rightarrow S$, atunci pentru orice semigrup (S', \circ) și orice funcție $f : X \rightarrow S'$, există un unic homomorfism $h : S \rightarrow S'$ ce satisfac $f = h \circ \varphi$.

Demonstrația acestei teoreme urmează în linii mari aceeași idee ca a Teoremei 3.4.1.1. Diagrama din Figura 3.5 furnizează o imagine grafică a proprietății din această teoremă. Similaritatea cu diagrama din Figura 3.4 nu este de loc întâmplătoare.

$$\begin{array}{ccc} X & \xrightarrow{\varphi} & S \\ f \searrow & & \downarrow h \\ & & S' \end{array}$$

Figura 3.5: Proprietatea din Teorema 3.4.1.2

Teorema 3.4.1.1 este caz particular al Teoremei 3.4.1.2, considerând φ ca fiind funcția incluziune.

Proprietatea din Teorema 3.4.1.2 este frecvent numită *proprietatea de universalitate a semigrupului (S, \cdot) relativ la $\varphi : X \rightarrow S$* . Adică, un semigrup (S, \cdot) satisfac proprietatea de universalitate relativ la $\varphi : X \rightarrow S$ dacă pentru orice semigrup (S', \circ) și orice funcție $f : X \rightarrow S'$ există un unic homomorphism $h : S \rightarrow S'$ ce satisfac $f = h \circ \varphi$.

Propoziția 3.4.1.1. Dacă un semigrup (S, \cdot) satisfac proprietatea de universalitate relativ la o funcție $\varphi : X \rightarrow S$, atunci φ este injectivă.

Demonstrație. Fie (S', \cdot) un semigrup și $f : X \rightarrow S'$ astfel încât f este injectivă (oricând se poate găsi un astfel de semigrup și o astfel de funcție). Proprietatea de universalitate asigură existența unui unic homomorfism $h : S \rightarrow S'$ astfel încât $f = h \circ \varphi$.

Dacă φ nu ar fi injectivă, atunci este ușor de văzut că egalitatea $f = h \circ \varphi$ nu ar putea avea loc. \square

Proprietatea de universalitate este definitorie pentru conceptul de semigrup liber în sensul că orice semigrup ce satisfac această proprietate este liber. Vom arăta aceasta în cele ce urmează.

Propoziția 3.4.1.2. Dacă un semigrup (S_1, \cdot) satisfac proprietatea de universalitate relativ la $\varphi_1 : X_1 \rightarrow S$, un semigrup (S_2, \circ) satisfac proprietatea de universalitate

relativ la $\varphi_2 : X_2 \rightarrow S$, iar X_1 și X_2 sunt echipotente, atunci semigrupurile (S_1, \cdot) și (S_2, \circ) sunt izomorfe.

Demonstrație. Fie $f : X_1 \rightarrow X_2$ o funcție bijectivă. Considerăm diagramele din Figura 3.6, unde h_1 și h_2 sunt unicele homomorfisme asigurate de proprietatea de universalitate ce satisfac $\varphi_2 \circ f = h_1 \circ \varphi_1$ și $\varphi_1 \circ f^{-1} = h_2 \circ \varphi_2$. Aceste relații conduc

$$\begin{array}{ccc} X_1 & \xrightarrow{\varphi_1} & S_1 \\ \varphi_2 \circ f \searrow & & \downarrow h_1 \\ & & S_2 \\ \varphi_1 \circ f^{-1} \swarrow & & \downarrow h_2 \\ X_2 & \xrightarrow{\varphi_2} & S_2 \end{array}$$

Figura 3.6: Proprietatea de universalitate pentru (S_1, \cdot) și (S_2, \circ)

la:

$$\varphi_2 = (h_1 \circ \varphi_1) \circ f^{-1} = h_1 \circ (\varphi_1 \circ f^{-1}) = h_1 \circ (h_2 \circ \varphi_2) = (h_1 \circ h_2) \circ \varphi_2,$$

iar h_1 și h_2 sunt unicele care satisfac această relație finală. Pe de altă parte, $\varphi_2 = 1_{S_2} \circ \varphi_2$. Unicitatea homomorfismelor h_1 și h_2 relativ la proprietatea de mai sus conduce la $h_1 \circ h_2 = 1_{S_2}$.

În mod similar se arată că are loc $h_2 \circ h_1 = 1_{S_1}$ care, combinată cu relația de mai sus, conduce la faptul că h_1 și h_2 sunt inverse unul altuia. Deci ele sunt izomorfisme și, ca urmare, S_1 și S_2 sunt izomorfe. \square

Corolarul 3.4.1.3. Dacă un semigrup (S, \cdot) satisfac proprietatea de universalitate relativ la $\varphi : X \rightarrow S$, atunci au loc următoarele proprietăți:

- (1) (S, \cdot) și X^+ sunt izomorfe;
- (2) X generează liber (S, \cdot) relativ la $\varphi : X \rightarrow S$.

Demonstrație. (2) urmează direct de la (1), așa că ne vom ocupa de (1). Fie $\varphi' : X \rightarrow X^+$ dată prin $\varphi'(x) = x$, pentru orice $x \in X$. Este ușor de văzut că X^+ satisfac proprietatea de universalitate relativ la φ' . Atunci S și X^+ vor fi izomorfe în baza Propoziției 3.4.1.2. \square

În multe lucrări conceptual de semigrup liber generat este introdus prin intermediul proprietății de universalitate. Există avantaje pentru o astfel de abordare, dar acestea sunt esențiale doar în cazul structurilor algebrice mai complexe și în special în cazul algebrelor universale, așa cum se va vedea în Capitolul 8. În cazul semigrupurilor și monoizilor credem că este de preferat abordarea deja aleasă, deoarece este mult mai naturală și ușor de asimilat.

3.4.2. Rezultate de caracterizare

Vom prezenta în această secțiune câteva rezultate de bază de caracterizare a monoizilor liberi (ce sunt valabile și pentru semigrupuri libere).

Observația 3.4.2.1. Dacă M este un monoid, atunci orice mulțime de generatori ai lui M trebuie să includă $S_M - S_M^2$. Însă este posibil ca $S_M - S_M^2$ să nu fie suficientă pentru a genera monoidul M . De exemplu, mulțimea numerelor cardinale $M = \{\gamma | 0 \leq \gamma \leq \aleph_0\}$ formează monoid cu operația de adunare. Însă $S_M - S_M^2 = \{1\}$, care nu generează M (\aleph_0 nu poate fi scris ca sumă finită de elemente din $S_M - S_M^2$, iar \aleph_0 nu este în $S_M - S_M^2$, deoarece $\aleph_0 = \aleph_0 + 1$).

Propoziția 3.4.2.1. Fie M un monoid liber. Atunci unica mulțime de generatori liberi ai monoidului M este $S_M - S_M^2$.

Demonstrație. Fie X o mulțime ce generează liber M . Dacă $X = \emptyset$, atunci $M = \{1_M\}$, și deci $S_M = \emptyset$. Ca urmare, $X = S_M - S_M^2$.

Să presupunem acum că $X \neq \emptyset$. Pentru orice $x \in X$, ori $x \in S_M - S_M^2$, ori $x \in S_M^2$. Vom arăta că x nu poate fi în S_M^2 . Dacă presupunem, prin contradicție, că x ar fi în această mulțime, atunci x s-ar scrie în forma $x = s_1 s_2$, cu $s_1, s_2 \in S_M$. Deoarece X generează liber M , s_1 și s_2 se pot scrie unic ca produs de elemente din X . Ca urmare, x are două scrieri distincte ca produs de elemente din X , una fiind x , iar cealaltă fiind cea rezultată prin înlocuirea lui s_1 și s_2 cu unicele lor scrieri ca produse de elemente din X , ceea ce contrazice cu caracterul liber al monoidului M . Am obținut astfel inclusiunea $X \subseteq S_M - S_M^2$ care, combinată cu $S_M - S_M^2 \subseteq X$ (Observația 3.4.2.1), conduce la $X = S_M - S_M^2$. \square

Definiția 3.4.2.1. Un monoid M este numit *echidivizibil* dacă are loc

$$xy = uv \Rightarrow (\exists z \in M)((x = uz \wedge v = zy) \vee (u = xz \wedge y = zv)),$$

pentru orice $x, y, u, v \in M$.

Echidivizibilitatea poate fi reprezentată grafic ca în Figura 3.7.

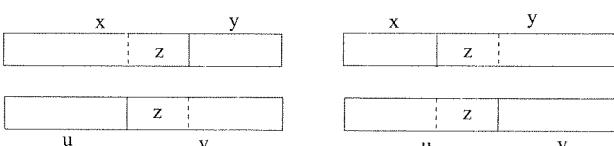


Figura 3.7: Proprietatea de echidivizibilitate

Observația 3.4.2.2. Echidivizibilitatea nu este, în general, echivalentă cu simplificarea¹. Monoidul din Observația 3.4.2.1 este echidivizibil, dar nu este cu simplificare deoarece, de exemplu,

$$\aleph_0 + 1 = \aleph_0 + 2,$$

dar $1 \neq 2$.

Monoidul multiplicativ al numerelor naturale nenule este cu simplificare, dar nu este echidivizibil, deoarece egalitatea $2 \cdot 3 = 3 \cdot 2$ nu conduce la existența nici unui număr natural z astfel încât $2 = 3 \cdot z$ sau $3 = 2 \cdot z$.

Teorema 3.4.2.1. Fie M un monoid. M este liber dacă și numai dacă M este echidivizibil și $\bigcap_{n \geq 1} S_M^n = \emptyset$.

Demonstrație. Să presupunem că M este liber. Atunci $S_M - S_M^2$ este unica mulțime ce generează liber monoidul M .

Vom arăta că M este echidivizibil. Fie $x, y, u, v \in M$ astfel încât $xy = uv$. Atunci, x, y, u și v pot fi scrise unic ca produse de elemente din $S_M - S_M^2$,

$$\begin{aligned} x &= a_1 \cdots a_n \\ y &= b_1 \cdots b_m \\ u &= c_1 \cdots c_k \\ v &= d_1 \cdots d_l, \end{aligned}$$

unde $n, m, k, l \geq 0$ (în cazul $n = 0$ considerăm $x = 1_M$ și similar pentru celelalte cazuri), iar de la relația $xy = uv$ urmează

$$a_1 \cdots a_n b_1 \cdots b_m = c_1 \cdots c_k d_1 \cdots d_l.$$

Avem de analizat acum 3 cazuri: $n > k$, $n = k$ și $n < k$. În cazul $n > k$ ($n = k$, $n < k$), elementul $z = a_{k+1} \cdots a_n$ ($z = 1_M$, $z = c_{n+1} \cdots c_k$) asigură proprietatea de echidivizibilitate.

Să arătăm acum că $\bigcap_{n \geq 1} S_M^n = \emptyset$. Dacă $M = \{1_M\}$, atunci $S_M = \emptyset$, ceea ce conduce la relația de mai sus. Dacă $M \neq \{1_M\}$, atunci, pentru orice $x \in S_M$, există un unic $n \geq 1$ astfel încât $x \in S_M^n - S_M^{n+1}$. Ca urmare, $\bigcap_{n \geq 1} S_M^n = \emptyset$.

Reciproc, presupunem că M este echidivizibil și $\bigcap_{n \geq 1} S_M^n = \emptyset$. Fără a restrânge generalitatea, putem presupune că $S_M \neq \emptyset$.

Arătăm că $S_M - S_M^2$ generează M . Fie $x \in S_M$. Atunci există $n \geq 1$ astfel încât $x \in S_M^n - S_M^{n+1}$ (altfel, $\bigcap_{n \geq 1} S_M^n$ ar fi nevidă). Ca urmare, există elementele $a_1, \dots, a_n \in S_M - S_M^2$ astfel încât $x = a_1 \cdots a_n$. Deci $S_M - S_M^2$ generează M .

Vom arăta acum că $S_M - S_M^2$ generează liber M . Dacă presupunem că există un element diferit de 1_M ce admite două scrieri

$$a_1 \cdots a_k = b_1 \cdots b_l$$

¹Așa cum vom vedea în această secțiune, monoizii liberi sunt atât echidivizibili, cât și cu simplificare.

cu $a_1, \dots, a_k, b_1, \dots, b_l \in S_M - S_M^2$, atunci considerând $y = a_2 \cdots a_k$ și $v = b_2 \cdots b_l$, echidivizibilitatea va conduce la existența unui element $z \in M$ astfel încât $a_1 = b_1 z$ și $v = zy$, sau $b_1 = a_1 z$ și $y = zv$. În oricare dintre aceste două cazuri deducem că $z = 1_M$, ceea ce va conduce la $a_1 = b_1$ și $y = v$. Repetăm raționamentul cu $y = v$ și, după un număr de pași, obținem unul dintre următoarele cazuri:

- $k = l$ și $a_i = b_i$, pentru orice $1 \leq i \leq k$, sau
- $k < l$, $a_i = b_i$, pentru orice $1 \leq i \leq k$, și $1_M = b_{k+1} \cdots b_l$ (cazul $k > l$, care este și el posibil, este similar acestuia).

Vom arăta că nu poate avea loc $k < l$ (și, similar, $k > l$), ceea ce va încheia demonstrația teoremei. În adevăr, dacă presupunem că are loc $k < l$, atunci $k+1 < l$, deoarece altfel am obține $b_{k+1} = 1_M$, ceea ce ar contrazice alegerea lui b_{k+1} . Ca urmare, $b_{k+1} = (b_{k+1} \cdots b_l)^n b_{k+1} \in S_M^n$, pentru orice $n \geq 1$, ceea ce contrazice ipoteza $\bigcap_{n \geq 1} S_M^n = \emptyset$. \square

Corolarul 3.4.2.1. (Teorema lui Levi)

Fie M un monoid. M este liber dacă și numai dacă M este echidivizibil și există un homomorfism l de la M la monoidul $(\mathbb{N}, +)$ astfel încât $l^{-1}(0) = \{1_M\}$.

Demonstrație. Să presupunem că M este monoid liber. Conform Teoremei 3.4.2.1, este suficient de arătat că există un homomorfism de la M la monoidul $(\mathbb{N}, +)$ astfel încât $l^{-1}(0) = \{1_M\}$.

Fie $l' : S_M - S_M^2 \rightarrow \mathbb{N}$ dată prin $l'(x) = 1$, pentru orice $x \in S_M - S_M^2$. Deoarece M este liber generat de $S_M - S_M^2$, funcția l' se poate extinde la un unic homomorfism l de la M la monoidul $(\mathbb{N}, +)$. Atunci este trivial de verificat că, pentru orice $x \in M$, $l(x) = 0$ dacă și numai dacă $x = 1_M$.

Reciproc, fie l un homomorfism ca în enunțul corolarului. Pentru a arăta că M este liber este suficient de arătat că $\bigcap_{n \geq 1} S_M^n = \emptyset$. Putem presupune că $S_M \neq \emptyset$.

Fie $x \in S_M$ și $k = l(x)$. Vom arăta că $x \notin S_M^{k+1}$. În adevăr, dacă presupunem că $x \in S_M^{k+1}$, atunci x se poate scrie în forma

$$x = x_1 \cdots x_{k+1},$$

cu $x_i \in S_M$, pentru orice $1 \leq i \leq k+1$. Deoarece $l^{-1}(0) = \{1_M\}$, deducem că $l(x_i) \geq 1$, pentru orice $1 \leq i \leq k+1$, de unde urmează

$$l(x) = l(x_1) + \cdots + l(x_{k+1}) \geq k+1,$$

ceea ce contrazice $l(x) = k$. Deci, $x \notin S_M^{k+1}$.

Ca urmare, pentru fiecare $x \in S_M$ există $k \geq 1$ astfel încât $x \notin S_M^{k+1}$. Deci, $\bigcap_{n \geq 1} S_M^n = \emptyset$. \square

Fie M un monoid, $A \subseteq S_M$ și $x = a_1 \cdots a_n \in M$, unde $a_i \in A$, pentru orice i . Elementele

$$1_M, a_1, a_1 a_2, \dots, a_1 \cdots a_n$$

vor fi numite *factori stângi ai lui x în raport cu A* .

Corolarul 3.4.2.2. (Teorema Dubreil-Jacotin)

Fie M un monoid. M este liber dacă și numai dacă M este echidivizibil, cu simplificare, are grupul unităților trivial și orice element are un număr finit de factori stângi distincți în raport cu $S_M - S_M^2$.

Demonstrație. Presupunem că monoidul M este liber. Atunci, de la Teorema 3.4.2.1 urmează că M este echidivizibil. Pentru a arăta că M este cu simplificare la stânga considerăm $x, u, v \in M$ și presupunem că are loc $xu = xv$ și $x \neq 1_M$. Cum M este liber, x, u și v pot fi scrise în mod unic în forma $x = a_1 \cdots a_n$, $u = b_1 \cdots b_m$ și $v = c_1 \cdots c_l$, unde $a_i, b_j, c_k \in S_M - S_M^2$, pentru orice i, j și k (în cazul $m = 0$ considerăm $u = 1_M$, și similar pentru $l = 0$). Relația $xu = xv$ conduce la

$$a_1 \cdots a_n b_1 \cdots b_m = a_1 \cdots a_n c_1 \cdots c_l$$

care, în baza proprietății lui M de a fi liber furnizează $m = l$ și $b_j = c_j$, pentru orice j . Deci, $u = v$.

În mod similar se arată că M este cu simplificare la dreapta.

Să arătăm acum că M are grupul unităților trivial. Dacă presupunem că există $x, y \in M$ cu $x \neq 1_M$ și $xy = yx = 1_M$, atunci $y \neq 1_M$. Acum, putem constata cu ușurință că x admite două scrieri distințe ca produse de elemente din $S_M - S_M^2$ și anume, x și xyx , ceea ce reprezintă o contradicție.

Fie $a_1, \dots, a_n \in S_M - S_M^2$, unde $n \geq 1$. Factorii stângi ai lui $x = a_1 \cdots a_n$ în raport cu $S_M - S_M^2$ sunt

$$1_M, a_1, a_1 a_2, \dots, a_1 \cdots a_n,$$

și aceștia sunt distincți doi câte doi (acest fapt se poate verifica ușor folosind proprietatea monoidului M de a fi liber).

Reciproc, presupunem că M este un monoid echidivizibil, cu simplificare, are grupul unităților trivial și fiecare element al lui are un număr finit de factori stângi distincți în raport cu $S_M - S_M^2$. Conform Teoremei 3.4.2.1, este suficient de arătat că $\bigcap_{n \geq 1} S_M^n = \emptyset$.

Fără a restrânge generalitatea putem considera că $S_M \neq \emptyset$. Presupunem, prin contradicție, că $\bigcap_{n \geq 1} S_M^n \neq \emptyset$. Fie $x \in \bigcap_{n \geq 1} S_M^n$. Pentru orice $k \geq 2$, $x \in S_M^k$. Deci x se poate scrie în forma

$$x = x_1 \cdots x_k,$$

cu $x_1, \dots, x_k \in S_M$. Pentru orice $1 \leq i \leq k$, $x_1 \cdots x_i$ este factor stâng al lui x . Vom arăta că oricare doi astfel de factori stângi sunt distincți. Fie i și j cu $1 \leq i < j \leq k$. Presupunem, prin contradicție, că $x_1 \cdots x_i = x_1 \cdots x_j$. Proprietatea de simplificare conduce la

$$x_{i+1} \cdots x_j = 1_M.$$

Deoarece x_{i+1} nu poate fi 1_M , urmează că $i + 1 < j$. Fie $y = x_{i+2} \cdots x_j$. Relația de mai sus poate fi rescrisă în forma

$$x_{i+1}y = 1_M,$$

de unde urmează

$$yx_{i+1}y = y$$

care, prin simplificare la dreapta, conduce la

$$yx_{i+1} = 1_M.$$

Deci, x_{i+1} este unitate a lui M ; contradicție (grupul unităților lui M este trivial).

Ca urmare, orice doi factori stângi ai lui x de forma $x_1 \cdots x_i$ și $x_1 \cdots x_j$, unde $1 \leq i < j \leq k$, sunt distincți.

Deoarece $x \in S_M^k$, pentru orice $k \geq 2$, deducem că x nu poate avea un număr finit de factori stângi distincți, ceea ce constituie o contradicție. Deci $\bigcap_{n \geq 1} S_M^n = \emptyset$. \square

Încheiem secțiunea prin câteva considerații generale asupra monoizilor liberi. Așa cum am văzut, orice monoid liber M are următoarele proprietăți:

1. unică mulțime de generatori liberi a lui M este $S_M - S_M^2$;
2. M este echidivizibil;
3. M este cu simplificare;
4. grupul unităților lui M este trivial;
5. orice element $x \in M$ are un număr finit de factori stângi distincți;
6. există un homomorfism l de la M la monoidul aditiv al numerelor naturale astfel încât, pentru orice $x \in M$, $l(x) = 0$ dacă și numai dacă $x = 1_M$.

În plus, orice monoid liber este izomorf cu un monoid de cuvinte. Aceasta ne permite ca multe studii asupra monoizilor liberi să fie realizate direct pe monoizi de cuvinte, fără a pierde din generalitate.

Dacă M este un monoid liber, atunci homomorfismul $l : M \rightarrow \mathbf{N}$ dat prin $l(1_M) = 0$ și $l(a) = 1$, pentru orice $a \in S_M - S_M^2$, se numește *homomorfismul (funcția) lungime* a elementelor monoidului. În loc de $l(x)$ se notează frecvent $|x|$ (această notație este în concordanță deplină cu notația adoptată pentru lungimea cuvintelor peste un alfabet). De fapt, elementele monoidului liber M pot fi gândite ca fiind cuvinte peste $S_M - S_M^2$.

3.4.3. Submonoizi liberi

Una dintre întrebările naturale ce se pun asupra submonoizilor unui monoid M este următoarea: dacă M este liber, urmează că orice submonoid al lui M este liber? Răspunsul la această întrebare este, în general, negativ.

Observația 3.4.3.1. Fie A o mulțime nevidă și $a \in A$.

- (1) Mulțimea

$$M' = \{a^n \mid n \geq 0\}$$

formează submonoid liber (generat de a) al monoidului liber A^* .

- (2) Mulțimea

$$M'' = \{a^n \mid n \geq 0, n \neq 1\}$$

formează submonoid al monoidului A^* , generat de $S_{M''} - S_{M''}^2 = \{a^2, a^3\}$ (aceasta urmează de la faptul că orice număr natural $n \geq 2$ se poate scrie în forma $n = 3\alpha + 2\beta$, unde $\alpha, \beta \geq 0$). Însă, M'' nu este liber generat de $S_{M''} - S_{M''}^2$, deoarece, de exemplu, a^7 se poate scrie

$$a^7 = a^2a^2a^3 = a^2a^3a^2 = a^3a^2a^2$$

(de fapt, M'' nu este liber, deoarece altfel unică mulțime de generatori liberi ar trebui să fie $S_{M''} - S_{M''}^2$).

Lema 3.4.3.1. Fie M' un submonoid al unui monoid liber M . Atunci M' este generat de $S_{M'} - S_{M'}^2$.

Demonstrație. Fie M un monoid liber și M' un submonoid al lui M . Conform Corolarului 3.4.2.1, există un homomorfism $l : M \rightarrow \mathbf{N}$ cu proprietatea $l^{-1}(0) = \{1_M\}$. Fie $w \in M' - \{1_M\}$. Prin inducție după $l(w) \geq 1$ vom arăta că w se poate scrie ca produs (finit) de elemente din $S_{M'} - S_{M'}^2$. Dacă presupunem proprietatea adevărată pentru toate elementele $u \in M' - \{1_M\}$ cu $l(u) < l(w)$, atunci avem de analizat următoarele două cazuri:

- (a) $w \in S_{M'} - S_{M'}^2$, caz în care proprietatea este satisfăcută de w ;
- (b) $w \in S_{M'}^2$. Atunci există $u, v \in S_{M'}$ astfel încât $w = uv$. Deoarece

$$l(w) = l(u) + l(v)$$

și $l^{-1}(0) = \{1_M\}$, urmează $l(u), l(v) < l(w)$. Dar atunci, ipoteza inducțivă conduce la faptul că u și v se pot scrie ca produs de elemente din $S_{M'} - S_{M'}^2$, ceea ce ne arată că w are această proprietate.

Ca urmare, M' este generat de $S_{M'} - S_{M'}^2$. \square

Observația 3.4.3.2. În Lema 3.4.3.1, dacă M nu este liber, concluzia poate să nu se păstreze. În adevăr, considerând monoidul aditiv al numerelor cardinale mai mici decât \aleph_1 ,

$$M = \{\gamma \mid 0 \leq \gamma \leq \aleph_1\},$$

care nu este liber, atunci constatăm că

$$M' = \{\gamma \mid 0 \leq \gamma \leq \aleph_0\}$$

este submonoid al lui M , ce nu este generat de $S_{M'} - S_{M'}^2$.

Observația 3.4.3.3. Fie M' un submonoid al unui monoid liber M . Dacă M' nu este liber, atunci există $a_1, \dots, a_m, b_1, \dots, b_n \in S_{M'} - S_{M'}^2$ astfel încât

$$(1) \quad a_1 \cdots a_m = b_1 \cdots b_n \text{ și}$$

$$(2) \quad a_1 \neq b_1.$$

În locul relației (2) se poate presupune chiar mai mult, și anume:

$$(2') \quad |a_1 \cdots a_i| \neq |b_1 \cdots b_j|, \text{ pentru orice } 1 \leq i \leq m \text{ și } 1 \leq j \leq n \text{ cu } i \neq m \text{ sau } j \neq n$$

(altfel, simplificăm la dreapta în M). În reprezentarea grafică din Figura 3.8, (2') ne spune că nici o “tăietură” a lui u , exceptând-o pe cea inițială și finală, nu coincide cu o tăietură a lui v .

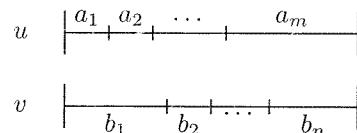


Figura 3.8: u și v nu au tăieturi comune

Teorema 3.4.3.1. (Criteriul lui Schützenberger)

Fie A o mulțime nevidă și M un submonoid al monoidului A^* . Atunci următoarele afirmații sunt echivalente:

- (1) M este liber;
- (2) $(\forall w \in A^+)(Mw \cap M \neq \emptyset \wedge M \cap wM \neq \emptyset \Rightarrow w \in M)$;
- (3) $(\forall w \in A^+)(Mw \cap M \cap wM \neq \emptyset \Rightarrow w \in M)$.

Demonstrație. Presupunem că M este liber și arătăm că are loc (2). Fie $w \in A^+$ cu $Mw \cap M \neq \emptyset$ și $M \cap wM \neq \emptyset$. Există atunci $x_1, x_2 \in M$ astfel încât $x_1w \in M$ și $wx_2 \in M$. Ca urmare, $x_1wx_2 \in M$. Egalitatea

$$x_1(wx_2) = (x_1w)x_2,$$

în baza proprietății de echidivizibilitate a monoidului M , conduce la existența unui element $z \in M$ astfel încât $x_1 = (x_1w)z$ sau $x_1w = x_1z$.

Dacă $x_1 = (x_1w)z$, atunci prin simplificare la stânga deducem $wz = 1_M = 1_{A^*}$, care conduce la faptul că w și z sunt unități ale lui A^* , ceea ce constituie o contradicție ($w \in A^+$ și singura unitate a lui A^* este 1_{A^*}). Deci relația $x_1 = (x_1w)z$ nu poate avea loc, ceea ce conduce la faptul că are loc $x_1w = x_1z$. Dar atunci, prin simplificare la stânga (în A^*) deducem $w = z$, de unde urmează $w \in M$ deoarece $z \in M$. Am obținut astfel (2).

Faptul că (2) implică (3) este imediat.

Presupunem că are loc (3) și demonstrăm (1). De la Lema 3.4.3.1 urmează că M este generat de $S_M - S_M^2$. Presupunem prin contradicție că M nu este liber. Atunci, există $a_1, b_1 \in S_M - S_M^2$ și $u, v \in M$ astfel încât

$$a_1u = b_1v$$

și $a_1 \neq b_1$ (a se vedea Observația 3.4.3.3). Proprietatea de echidivizibilitate a lui A^* conduce la existența unui element $z \in A^*$ astfel încât $a_1 = b_1z$ sau $b_1 = a_1z$. Elementul z nu poate fi 1_{A^*} deoarece $a_1 \neq b_1$. Vom arăta că $z \in M$, ceea ce va contrazice faptul că $a_1, b_1 \in S_M - S_M^2$.

Considerăm egalitatea $a_1 = b_1z$ (pentru cealaltă egalitate se raționează similar acesteia). Atunci, $b_1zu = b_1v$ care, prin simplificare la stânga (în A^*), conduce la $zu = v$, de unde urmează că

$$z(ua_1) = va_1 = (vb_1)z.$$

Cum $ua_1, va_1, vb_1 \in M$, ipoteza de la (3) este satisfăcută, și deci $z \in M$. Dar atunci, relația $a_1 = b_1z$ contrazice $a_1 \in S_M - S_M^2$. \square

Observația 3.4.3.4. Afirmațiile (2) și (3) din Teorema 3.4.3.1 sunt echivalente în orice submonoid M al unui monoid de cuvinte A^* . În adevăr, este trivial de verificat că (2) implică (3).

Reciproc, dacă $w \in A^+$ astfel încât $Mw \cap M \neq \emptyset$ și $M \cap wM \neq \emptyset$, atunci există $x_1, x_2 \in M$ astfel încât $x_1w, wx_2 \in M$. Deducem atunci că $wx_2x_1w \in M$. Dar,

$$wx_2x_1w = (wx_2x_1)w = w(x_2x_1w),$$

ceea ce arată că $Mw \cap M \cap wM \neq \emptyset$ de la care, în baza lui (3), urmează $w \in M$. Deci (3) implică (2).

3.5. Aplicații: coduri de lungime variabilă

Conceptul de cod apare pentru prima dată în teoria comunicației inițiată de Claude Shannon în perioada anilor 1948 [189], în strânsă legătură cu conceptele de detecție

și corecție a erorilor ce pot apărea în transmisia informației pe diverse canale de comunicație. Codurile studiate au fost cele *bloc* și cele *instantanee* (numite și *prefix*)². În 1949 Kraft realizează un studiu asupra codurilor instantanee și stabilește o inegalitate pe care orice cod instantaneu trebuie să o satisfacă, inegalitate ce poartă denumirea de *inegalitatea lui Kraft* [106]. Trei ani mai târziu, Huffman arată cum pot fi construite coduri instantanee de lungime medie minimă, numite astăzi *coduri Huffman*, ce au importanță majoră în teoria compresiei informației [95].

Codurile de lungime variabilă au fost introduse de Schützenberger în 1955 [187], acesta fiind condus către ele de investigații algebrice asupra acestei teorii. Inegalitatea lui Kraft referitoare la coduri instantanee a fost generalizată de McMillan în 1956 [144] la întreaga clasă a codurilor de lungime variabilă. În 1959 Gilbert și Moore reconsideră codurile de lungime variabilă dintr-un punct de vedere oarecum diferit de cel al lui Schützenberger [69].

Abordarea noastră va fi făcută la modul general prin prisma submonoizilor liberi generați, urmând ca apoi să ne apropiem de aplicațiile majore ale acestei teorii.

3.5.1. Definiții. Exemple. Proprietăți de bază

Intuitiv, un cod este o mulțime C de cuvinte (peste un alfabet) cu proprietatea că orice produs de elemente din C poate fi unic factorizat (ca produs de elemente din C). Altfel spus, pentru orice $w \in C^+$ există exact o descompunere

$$w = c_1 \cdots c_n,$$

unde $c_i \in C$, pentru orice $1 \leq i \leq n$.

Exemplul 3.5.1.1. Fie $C = \{a, ab, ba\}$ o mulțime de cuvinte peste $A = \{a, b\}$. C nu este cod, deoarece cuvântul aba poate fi factorizat în două moduri distincte ca produs de elemente din C (vom folosi parantezele rotunde pentru a specifica descompunerea):

$$aba = (ab)a = a(ba).$$

Exemplul 3.5.1.2. Fie $C = \{a, bb, aab, bab\}$ o mulțime de cuvinte peste $A = \{a, b\}$. C este cod. În adevăr, dacă w a fost obținut ca produs de elemente din C , atunci w poate fi unic factorizat în produs de elemente din C , “citindu-l” de la dreapta la stânga. Dacă w se termină cu a , adică $w = w'a$, atunci separăm a ca fiind ultimul cuvânt cod din w și continuăm cu decodificarea lui w' . Dacă w se termină cu b , adică $w = w'b$, atunci analizăm ultima literă din w' . Dacă aceasta este b , adică $w = w''bb$, atunci separăm bb ca fiind ultimul cuvânt cod din w și continuăm cu decodificarea lui w'' . Altfel, dacă $w = w''ab$, vom analiza ultima literă din w'' . Dacă aceasta este a , adică $w = w'''aab$, atunci separăm aab ca fiind ultimul cuvânt cod din w și continuăm

²Pentru detalii se pot consulta [131, 127, 3, 38, 143].

cu decodificarea lui w''' ; altfel, $w = w'''bab$, separăm bab ca fiind ultimul cuvânt cod din w și continuăm cu decodificarea lui w''' .

Este clar că procedeul descris mai sus se termină și furnizează unică descompunere a lui w (dacă acesta a fost obținut ca produs de elemente din C). Deci C este cod.

Atragem atenția că analiza lui C de la stânga la dreapta este mult mai dificilă. Considerând un cuvânt de forma $w = aababu$ obținut ca produs de elemente din C , analiza lui de la stânga la dreapta nu poate fi realizată în mod determinist, aşa cum se poate vedea mai jos:

$$(a)(a)(bab)u, \quad (aab)abu, \quad (aab)(a)bu.$$

Nici una din aceste descompuneri nu poate fi considerată corectă încă de la început; este nevoie a se citi (analiza) întregul cuvânt w pentru a se decide care descompunere este corectă.

Definiția 3.5.1.1. Fie A o mulțime nevidă. Se numește *cod* peste A orice submulțime nevidă $C \subseteq A^+$ ce generează liber un submonoid al lui A^* .

Dacă C este cod peste A , atunci submonoidul liber generat de C va fi notat prin C^* ; el este format din mulțimea tuturor produselor de elemente din C , inclusiv și unitatea lui A^* . Prin C^+ vom nota mulțimea $C^+ = C^* - \{1_{A^*}\}$.

Elementele codului C vor fi numite *cuvinte cod*, iar secvențele de forma $c_1 \cdots c_n$, unde $c_i \in C$, pentru orice $1 \leq i \leq n$, vor fi numite *secvențe de cod*.

Observația 3.5.1.1. Fie A o mulțime nevidă.

(1) Dacă C este cod peste A , atunci orice element $w \in C^+$ se scrie unic ca produs de elemente din C , $w = c_1 \cdots c_n$, unde $c_i \in C$, pentru orice $1 \leq i \leq n$. Altfel spus, pentru orice $w \in C^+$, dacă

$$u_1 \cdots u_m = w = v_1 \cdots v_n,$$

unde $u_1, \dots, u_m, v_1, \dots, v_n \in C$, atunci $n = m$ și $u_i = v_i$, pentru orice i .

Reciproc, dacă C este o submulțime nevidă a mulțimii A^+ ce are proprietatea că, pentru orice $u_1, \dots, u_m, v_1, \dots, v_n \in C$, egalitatea

$$u_1 \cdots u_m = v_1 \cdots v_n$$

conduce la $n = m$ și $u_i = v_i$ pentru orice $1 \leq i \leq m$, atunci C este cod peste A .

(2) Conceptul de cod poate fi introdus și în următoarea variantă. O submulțime nevidă $C \subseteq A^+$ este cod peste A dacă, pentru orice $u_1, \dots, u_m, v_1, \dots, v_n \in C$, egalitatea

$$u_1 \cdots u_m = v_1 \cdots v_n$$

conduce la $u_1 = v_1$. Cititorul poate verifica ușor că această definiție este echivalentă cu cea precedentă.

- (3) Orice submulțime nevidă a unui cod peste A este cod peste A .
(4) Dacă $|A| = 1$, atunci $C \subseteq A^+$ este cod peste A dacă și numai dacă $|C| = 1$.

Exemplul 3.5.1.3. Fie A o mulțime nevidă.

- (1) O mulțime $C \subseteq A^+$ este numită *mulțime prefix* dacă are loc:

$$(\forall u, v \in C)(\exists z \in A^+)(u = vz).$$

Mulțimile prefix nevide sunt coduri, numite *coduri prefix* sau *coduri instantane*³.

- (2) O mulțime $C \subseteq A^+$ este numită *mulțime sufîx* dacă are loc:

$$(\forall u, v \in C)(\exists z \in A^+)(u = zv).$$

Mulțimile sufîx nevide sunt coduri, numite *coduri sufîx*.

- (3) O mulțime prefix și sufîx este numită *mulțime biprefix*. Mulțimile biprefix nevide sunt numite *coduri biprefix*.
(4) O mulțime $C \subseteq A^+$ este numită *mulțime bloc* dacă are loc:

$$(\exists k \geq 1)(\forall c \in C)(|c| = k)$$

Mulțimile bloc nevide sunt numite *coduri bloc*. Numărul k de mai sus se numește *lungimea codului*.

Definiția 3.5.1.2. Fie A o mulțime nevidă și C un cod peste A .

- (1) C este numit *cod binar* dacă $|A| = 2$.
(2) C este numit *cod finit* dacă C este mulțime finită.
(3) Spunem că C are *distribuția de lungime* ($k_i | i \geq 1$), unde $(k_i | i \geq 1)$ este o secvență infinită de numere naturale nu toate 0, dacă $|C \cap A^i| = k_i$, pentru orice $i \geq 1$.

Observația 3.5.1.2.

- (1) Uzual, codurile binare se consideră peste mulțimea $A = \{0, 1\}$.
(2) Distribuția de lungime a unui cod ne spune câte cuvinte de lungime i conține codul, pentru orice $i \geq 1$. Dacă C este un cod finit, atunci distribuția de lungime a lui va fi notată ca o secvență finită

$$(k_i | 1 \leq i \leq n),$$

unde n este lungimea maximă a cuvintelor codului C .

³Terminologia de "cod instantaneu" provine de la faptul că decodificarea secvențelor cod obținute prin codificare cu astfel de coduri se poate face prin citirea acestora de la stânga la dreapta și separarea primului cuvânt cod întâlnit. Spunem, într-un astfel de caz, că decodificarea este instantanee.

- (3) Codurile bloc sunt coduri finite, ele conținând cuvinte de o același lungime.
(4) În unele lucrări, conceptul de cod este introdus prin ceea ce corespunde în lucrarea noastră conceptului de cod bloc. Atunci, pentru coduri aşa cum au fost introduse mai sus se folosește terminologia de *cod de lungime variabilă*.

Exemplul 3.5.1.4. Codul ASCII pe 8 biți este cod bloc de lungime 8 peste $\{0, 1\}$.

Observația 3.5.1.3. Codurile fiind mulțimi de cuvinte, putem defini *concatenarea (produsul)* a două coduri ca fiind concatenarea celor două mulțimi de cuvinte. Însă trebuie să observăm că produsul a două coduri nu este întotdeauna cod. În adevăr, $C_1 = \{a, ba\}$ și $C_2 = \{a, ab\}$ sunt coduri, dar $C_1 C_2 = \{a^2, a^2b, ba^2, ba^2b\}$ nu este cod deoarece a^2ba^2 admite două descompuneri distincte.

Produsul a două coduri bloc este întotdeauna cod bloc.

Propoziția 3.5.1.1. Dacă C este cod peste o mulțime nevidă A atunci, pentru orice $k \geq 1$, C^k este cod peste A .

Demonstrație. Fie $k \geq 1$ și $u_1, \dots, u_m, v_1, \dots, v_n \in C^k$ astfel încât

$$u_1 \cdots u_m = v_1 \cdots v_n.$$

Atunci, pentru orice $1 \leq i \leq m$ și $1 \leq j \leq n$, putem scrie $u_i = u_i^1 \cdots u_i^k$ și $v_j = v_j^1 \cdots v_j^k$, unde $u_i^p, v_j^p \in C$, pentru orice $1 \leq p \leq k$. Egalitatea de mai sus conduce la

$$u_1^1 \cdots u_1^k \cdots u_m^1 \cdots u_m^k = v_1^1 \cdots v_1^k \cdots v_n^1 \cdots v_n^k.$$

Deoarece C este cod, obținem $m = n$ și $u_i^p = v_i^p$, pentru orice $1 \leq i \leq m$ și $1 \leq p \leq k$, de unde urmează $u_i = v_i$, pentru orice i . Deci C^k este cod. □

Teorema 3.5.1.1. Fie A și $C \subseteq A^+$ mulțimi nevide. Atunci C este cod peste A dacă și numai dacă există o mulțime Σ și un homomorfism injectiv $h : \Sigma^* \rightarrow A^*$ astfel încât $C = h(\Sigma)$.

Demonstrație. Să presupunem că C este cod peste A . Fie Σ o mulțime echipotentă cu C și $f : \Sigma \rightarrow C$ o bijecție. Considerăm $g : \Sigma \rightarrow A^*$ dată prin $g(a) = f(a)$, pentru orice $a \in \Sigma$. Funcția g este injectivă și poate fi extinsă la un unic homomorfism de la Σ^* la A^* ; fie h această extensie. Este clar că $C = h(\Sigma)$, și astfel ne rămâne de arătat că h este homomorfism injectiv.

Fie $u, v \in \Sigma^*$ astfel încât $h(u) = h(v)$. Atunci u și v pot fi scrise unic în forma $u = a_1 \cdots a_m$ și $v = b_1 \cdots b_n$ cu $a_i, b_j \in \Sigma$, pentru orice $1 \leq i \leq m$ și $1 \leq j \leq n$. Relația $h(u) = h(v)$ conduce atunci la

$$f(a_1) \cdots f(a_m) = f(b_1) \cdots f(b_n),$$

de unde, în baza faptului că C este cod și f este bijecție, obținem $n = m$ și $a_i = b_i$, pentru orice $1 \leq i \leq n$. Deci, h este homomorfism injectiv.

Reciproc, fie Σ o mulțime și $h : \Sigma^* \rightarrow A^*$ un homomorfism injectiv astfel încât $C = h(\Sigma)$. Vom arăta că C este cod.

Fie $u_1, \dots, u_m, v_1, \dots, v_n \in C$ astfel încât

$$u_1 \cdots u_m = v_1 \cdots v_n,$$

și fie $a_1, \dots, a_m, b_1, \dots, b_n \in \Sigma$ astfel încât $h(a_i) = u_i$ și $h(b_j) = v_j$, pentru orice $1 \leq i \leq m$ și $1 \leq j \leq n$. Relația de mai sus conduce atunci la

$$h(a_1 \cdots a_m) = h(b_1 \cdots b_n),$$

de unde, în baza faptului că h este homomorfism injectiv, obținem

$$a_1 \cdots a_m = b_1 \cdots b_n.$$

Faptul că Σ^* este monoid liber generat de Σ conduce direct la $m = n$ și $a_i = b_i$, pentru orice $1 \leq i \leq n$. Adică, C este cod peste A . \square

Observația 3.5.1.4.

(1) În implicația directă a Teoremei 3.5.1.1 este crucial faptul că C este cod. Altfel, unica extensie a funcției g , chiar dacă g este injecție, poate să nu fie injecție. În adevăr, considerând $C = \{a, ab, ba\}$, despre care știm că nu este cod, și $\Sigma = \{0, 1, 2\}$, funcția $g : \Sigma \rightarrow \{a, b\}^*$ dată prin

$$g(0) = a, \quad g(1) = ab \text{ și } g(2) = ba$$

este injecție, dar unica extensie a ei $h : \Sigma^* \rightarrow \{a, b\}^*$ nu este injecție deoarece

$$h(10) = aba = h(02).$$

(2) Fie Σ și A două mulțimi nevide. Atunci, pentru orice homomorfism injectiv $h : \Sigma^* \rightarrow A^*$, mulțimea $h(\Sigma)$ este cod peste A (în baza Teoremei 3.5.1.1). Ca urmare, putem defini codurile peste A ca fiind homomorfisme injective ce iau valori în A^* . De fapt vom prefera să spunem că un homomorfism injectiv h ca mai sus este un *cod peste A* sau, mai precis, că este o *codificare a mulțimii Σ peste A*.

(3) Caracterizarea codurilor prin homomorfisme injective permite introducerea conceptului de *compunere de coduri* ca fiind compunere de homomorfisme. Cum compunerea a două homomorfisme injective este homomorfism injectiv, obținem că, atunci când este posibil, compunerea a două coduri este cod (spre deosebire de produsul a două coduri).

Exemplul 3.5.1.5. Fie $h : \{0, 1, 2\}^* \rightarrow \{a, b\}^*$ dată prin

$$\begin{array}{c|ccccc} & 0 & 1 & 2 \\ \hline h & a & ba & bb \end{array}$$

și $g : \{A, B, C, D, E\}^* \rightarrow \{0, 1, 2\}^*$ dată prin

$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline g & 0 & 01 & 11 & 2 & 21 \end{array}$$

Se verifică faptul că h și g sunt injective și, deci, sunt coduri. Atunci, $h \circ g$ este codul

$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline h \circ g & a & aba & baba & bb & bbba \end{array}$$

Definiția 3.5.1.3. Un cod C peste o mulțime A este numit *cod maximal* dacă nu există nici un cod C' peste A astfel încât $C \subset C'$.

Observația 3.5.1.5. Dacă C este cod maximal peste A , atunci $\text{alph}(C) = A$ (altfel, dacă ar exista $a \in A - \text{alph}(C)$, $C' = C \cup \{a\}$ ar fi cod ce ar extinde strict codul C).

Exemplul 3.5.1.6.

(1) Fie A o mulțime nevidă și $n \geq 2$. Atunci A^n este cod maximal peste A . În adevăr, A^n este mulțime bloc, deci este cod. Pe de altă parte, pentru orice $w \in A^+$, cuvântul $w^{|w|n}$ (a cărei lungime este multiplu de n) se poate scrie ca produs de cuvinte de lungime n (prin simpla împărțire a acestui cuvânt, de la stânga la dreapta, în blocuri de lungime n). Deci codului A^n nu i se mai poate adăuga nici un nou cuvânt cu păstrarea proprietății de cod, ceea ce ne spune că A^n este maximal.

(2) Codul ASCII pe 8 biți este cod maximal.

3.5.2. Rezultate de caracterizare

Un cod C peste A generează liber un submonoid al monoidului A^* . O primă tentativă de caracterizare a codurilor ar fi prin intermediul criteriului lui Schützenberger:

O submulțime nevidă $C \subseteq A^+$ este cod dacă și numai dacă are loc:

$$(\forall w \in A^+)(C^*w \cap C^* \neq \emptyset \wedge C^* \cap wC^* \neq \emptyset \Rightarrow w \in C^*).$$

Această caracterizare ar funcționa dacă afirmația “ C cod” ar fi echivalentă cu afirmația “ C^* liber”. Însă aceste două afirmații nu sunt echivalente. Dacă C este cod, atunci C^* este liber generat de C , dar dacă C^* este liber atunci nu rezultă că C este cod. De fapt, C este cod dacă și numai dacă C^* este liber generat de C sau, dacă și numai dacă C^* este liber și $S_{C^*} - S_{C^*}^2 = C$. Cum inclusiunea $S_{C^*} - S_{C^*}^2 \subseteq C$ este întotdeauna satisfăcută, urmează că C este cod dacă și numai dacă C^* este liber și $C \subseteq S_{C^*} - S_{C^*}^2$. Însă relația $C \subseteq S_{C^*} - S_{C^*}^2$ este echivalentă cu faptul că nu există elemente din C care să poată fi scrise ca produse de elemente din C (altfel spus, C este independentă la concatenare).

Teorema 3.5.2.1. (Criteriul lui Schützenberger pentru coduri)

Fie A și $C \subseteq A^+$ mulțimi nevide. C este cod dacă și numai dacă au loc următoarele două proprietăți:

- (1) C este independentă la concatenare;
- (2) $(\forall w \in A^+)(C^*w \cap C^* \neq \emptyset \wedge C^* \cap wC^* \neq \emptyset \Rightarrow w \in C^*)$.

Proprietatea (2) din Teorema 3.5.2.1 poate fi înlocuită, în mod echivalent, cu

$$(2') (\forall w \in A^+)(C^*w \cap C^* \cap wC^* \neq \emptyset \Rightarrow w \in C^*)$$

Criteriul lui Schützenberger nu are aplicabilitate practică dar, în schimb, are o mare importanță teoretică.

Vom prezenta în continuare un rezultat de caracterizare ce poate fi utilizat în practică în cazul codurilor finite.

Să presupunem că $C \subseteq A^+$ este o submulțime nevidă și am fi interesați în a testa proprietatea de cod a ei pornind direct de la definiție, adică de la proprietatea de unică decodificare a oricărei secvențe de elemente peste C . Fie deci $w \in C^+$. Avem de analizat următoarele două cazuri:

1. există un unic $c \in C$ astfel încât $w = cw_1$ (Figura 3.9). În acest caz reținem c



Figura 3.9: Delimitare unică a primului cuvânt cod

și continuăm decodificarea lui w_1 ;

2. există $c_1, c_2 \in C$ astfel încât $c_1 \neq c_2$, $w = c_1w_1$ și $w = c_2w_2$. Fără a restrâng generalitatea putem presupune că $|c_1| < |c_2|$; fie $x \in A^+$ astfel încât $c_2 = c_1x$ (Figura 3.10). Este clar că dacă $x \in C$, atunci C nu este cod (C nu este independentă la concatenare). Ca urmare, putem considera mulțimea

$$C_1 = \{x \in A^+ \mid \exists c \in C : cx \in C\}.$$

Dacă $C \cap C_1 \neq \emptyset$, atunci C nu este cod.

Asupra lui $x \in C_1$ avem de analizat următoarele două cazuri (presupunând că $x \notin C$):

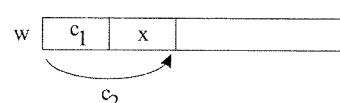


Figura 3.10: Cuvinte cod distincte dar comparabile prin relația \leq_{pref}

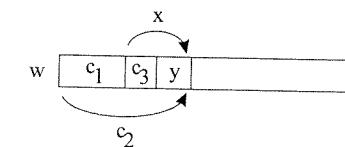


Figura 3.11: Separarea de noi cuvinte cod din x

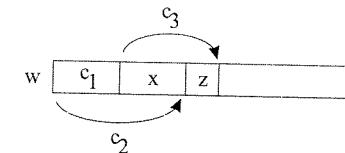


Figura 3.12: Completarea lui x până la un cuvânt cod

- (a) există $c_3 \in C$ astfel încât $x = c_3y$, unde $y \in A^+$ (Figura 3.11). Dacă $y \in C$, atunci C nu este cod (C nu este independentă la concatenare). Ca urmare, putem considera mulțimea

$$C'_2 = \{y \in A^+ \mid \exists c \in C : cy \in C_1\}.$$

Dacă $C \cap C'_2 \neq \emptyset$, atunci C nu este cod;

- (b) există $z \in A^+$ astfel încât $xz \in C$ (Figura 3.12). Dacă $z \in C$, atunci C nu este cod (produsul c_2z mai admite încă o descompunere în elemente din C , și anume $c_2z = c_1xz = c_1c_3$). Ca urmare, putem considera mulțimea

$$C''_2 = \{z \in A^+ \mid \exists c \in C_1 : cz \in C\}.$$

Dacă $C \cap C''_2 \neq \emptyset$, atunci C nu este cod.

Fie $C_2 = C'_2 \cup C''_2$. Atunci cazurile (a) și (b) pot fi grupate în unul singur exprimat prin: dacă $C \cap C_2 \neq \emptyset$, atunci C nu este cod.

Discuția de mai sus ne conduce la a considera secvența de mulțimi

- $C_1 = \{x \in A^+ \mid (\exists c \in C)(cx \in C)\},$
- $C_{i+1} = \{x \in A^+ \mid ((\exists c \in C)(cx \in C_i) \vee (\exists c \in C_i)(cx \in C))\},$ pentru orice $i \geq 1$.

Lema 3.5.2.1. Fie A și $C \subseteq A^+$ mulțimi nevide. Atunci, pentru orice $i \geq 1$ și orice $x \in C_i$, $C^*x \cap C^* \neq \emptyset$.

Demonstrație. Vom demonstra afirmația din lema prin inducție după $i \geq 1$.

Pentru $i = 1$, dacă $x \in C_1$, atunci există $c \in C$ astfel încât $cx \in C$. Ca urmare, $Cx \cap C \neq \emptyset$. Deoarece $Cx \subseteq C^*x$ și $C \subseteq C^*$, urmează $C^*x \cap C^* \neq \emptyset$.

Fie $i \geq 1$. Presupunem afirmația din lema adevărată pentru orice $y \in C_i$. Fie $x \in C_{i+1}$. Atunci avem de analizat următoarele două cazuri:

(a) există $c \in C$ astfel încât $cx \in C_i$. În baza ipotezei inducitive, $C^*cx \cap C^* \neq \emptyset$. Cum $C^*c \subseteq C^*$, deducem că are loc $C^*x \cap C^* \neq \emptyset$;

(b) există $c \in C_i$ astfel încât $cx \in C$. Conform ipotezei inducitive, $C^*c \cap C^* \neq \emptyset$, de la care urmează $C^*cx \cap C^*x \neq \emptyset$. Cum $cx \in C$, obținem $C^*cx \subseteq C^*$, și deci $C^* \cap C^*x \neq \emptyset$.

Ca urmare, $C^*x \cap C^* \neq \emptyset$. \square

Vom prezenta acum cel mai vechi rezultat de caracterizare a proprietății de cod, rezultat datorat lui Sardinas și Patterson [182].

Teorema 3.5.2.2. (Teorema Sardinas-Patterson)

Fie A și $C \subseteq A^+$ mulțimi nevide. Atunci C este cod dacă și numai dacă $C \cap C_i = \emptyset$, pentru orice $i \geq 1$.

Demonstrație. Presupunem că C este cod, dar există $i \geq 1$ astfel încât $C \cap C_i \neq \emptyset$. Vom analiza următoarele cazuri posibile.

Cazul 1: $i = 1$. În acest caz, relația $C \cap C_1 \neq \emptyset$ conduce la existența unui cuvânt cod $c_1 \in C$ astfel încât $c_1 \in C_1$. De aici urmează că există $c \in C$ astfel încât $cc_1 \in C$, ceea ce contrazice faptul că C este cod, deci independent la concatenare.

Cazul 2: $i > 1$. Fie $c_i \in C \cap C_i$. Deoarece $c_i \in C_i$ și $i > 1$, deducem că avem de analizat următoarele două cazuri:

(a) există $c_{i-1} \in C_{i-1}$ astfel încât $c_{i-1}c_i \in C$. Ca urmare, $C \cap c_{i-1}C \neq \emptyset$, și deci $C^* \cap c_{i-1}C^* \neq \emptyset$. Combinând această relație cu Lema 3.5.2.1 și cu Criteriul lui Schützenberger pentru coduri, obținem $c_{i-1} \in C^*$. Cum $c_{i-1} \in C_{i-1}$, deducem $c_{i-1} \in C^+$. Dar atunci, relația $c_{i-1}c_i \in C$ ne arată că C nu este mulțime independentă la concatenare, contrazicând faptul că C este cod;

(b) există $c_{i-1} \in C$ astfel încât $c_{i-1}c_i \in C_{i-1}$.

Deoarece cazul (a) nu poate avea loc, urmează că are loc (b). Adică există $c_{i-1} \in C$ astfel încât $c_{i-1}c_i \in C_{i-1}$. Inductiv, pe baza unui raționament similar celui de mai sus, deducem că există $c_{i-2}, \dots, c_1 \in C$ astfel încât

$$c_{i-2}c_{i-1}c_i \in C_{i-2}, \dots, c_1 \dots c_{i-2}c_{i-1}c_i \in C_1.$$

Însă ultima relație conduce la existența unui element $c \in C$ astfel încât

$$cc_1 \dots c_{i-2}c_{i-1}c_i \in C,$$

ceea ce contrazice faptul că C este cod.

Reciproc, presupunem că $C \cap C_i = \emptyset$ pentru orice $i \geq 1$, dar C nu este cod. Există atunci $u_1, \dots, u_m, v_1, \dots, v_n \in C$ astfel încât

$$u_1 \dots u_m = v_1 \dots v_n$$

și

$$|u_1 \dots u_i| \neq |v_1 \dots v_j|,$$

pentru orice $1 \leq i \leq m$ și $1 \leq j \leq n$ cu $i \neq m$ sau $j \neq n$ (Observația 3.4.3.3).

Considerând egalitatea de mai sus în A^* , echidivizibilitatea conduce la existența unui element $z_1 \in A^*$ astfel încât $u_1 = v_1 z_1$ sau $v_1 = u_1 z_1$. În primul rând observăm că $z_1 \in A^+$ (altfel, am avea $u_1 = v_1$). Cum $u_1, v_1 \in C$, oricare din relațiile $u_1 = v_1 z_1$ și $v_1 = u_1 z_1$ conduce la $z_1 \in C_1$. Deoarece $C \cap C_1 = \emptyset$, urmează $z_1 \notin C$. Să presupunem că are loc $u_1 = v_1 z_1$. Atunci

$$v_1 z_1 \dots u_m = v_1 \dots v_n$$

de unde, prin simplificare la stânga, obținem

$$z_1 u_2 \dots u_m = v_2 \dots v_n.$$

Utilizând iarăși echidivizibilitatea în A^* , există $z_2 \in A^*$ astfel încât $z_1 = v_2 z_2$ sau $v_2 = z_1 z_2$. Acum observăm că $z_2 \neq 1_{A^*}$ deoarece, altfel, z_1 ar fi în C . Din oricare din cele două cazuri obținem $z_2 \in C_2$ de unde, în baza relației $C \cap C_2 = \emptyset$, deducem că $z_2 \notin C$. Dacă $z_1 = v_2 z_2$, atunci obținem

$$z_2 u_2 \dots u_m = v_3 \dots v_n,$$

iar dacă $v_2 = z_1 z_2$, atunci obținem

$$u_2 \dots u_m = z_2 v_3 \dots v_n.$$

Continuând acest procedeu, în baza presupunerii

$$|u_1 \dots u_i| \neq |v_1 \dots v_j|,$$

pentru orice i și j cu $i \neq m$ sau $j \neq n$, deducem că există $k \geq 1$ și $z_k \in C_k - C$ astfel încât ori $z_k = v_n$ ori $u_m = z_k$, ceea ce constituie o contradicție. \square

Corolarul 3.5.2.1. Fie A și $C \subseteq A^+$ mulțimi nevide. Dacă C este finită, atunci există un algoritm de decizie a proprietății de cod a mulțimii C .

Demonstrație. Fie $n = \max\{|c| \mid c \in C\}$. Numărul n poate fi determinat efectiv deoarece C este finită. Nu este dificil de observat că, pentru orice $i \geq 1$ și orice $x \in C_i$, $|x| \leq n$. Ca urmare, mulțimile C_i sunt submulțimi ale mulțimii tuturor cuvintelor de lungime cel mult n peste $\text{alph}(C)$. Deoarece $\text{alph}(C)$ este finită, mulțimea tuturor cuvintelor de lungime cel mult n peste $\text{alph}(C)$ va fi finită. Atunci, urmează că există $i, j \geq 1$ cu $j < i$ și $C_i = C_j$.

Aceste observații permit elaborarea următorului algoritm de decizie a proprietății de cod a mulțimii C :

Algoritmul Sardinas-Patterson

```

input:  $C \subseteq A^+$  finită;
output:  $\text{cod}(C) = 1$ , dacă  $C$  este cod, și  $\text{cod}(C) = 0$ , altfel;
begin
   $C_1 := \{x \in A^+ | (\exists c \in C)(cx \in C)\}$ ;
  if  $C \cap C_1 \neq \emptyset$  then  $\text{cod}(C) := 0$ 
  else
    begin
       $i := 1$ ;
       $cont := 1$ ;
      while  $cont = 1$  do
        begin
           $i := i + 1$ ;
           $C_i := \{x \in A^+ | (\exists c \in C_{i-1})(cx \in C) \vee (\exists c \in C)(cx \in C_{i-1})\}$ ;
          if  $C \cap C_i \neq \emptyset$ 
            then begin  $\text{cod}(C) := 0$ ;  $cont := 0$  end
          else if  $(\exists j < i)(C_i = C_j)$ 
            then begin  $\text{cod}(C) := 1$ ;  $cont := 0$  end;
        end;
      end;
    end;
end.

```

Este clar că algoritmul descris mai sus se oprește întotdeauna și decide dacă C are sau nu proprietatea de cod. \square

Exemplul 3.5.2.1. Fie $C = \{a^2, a^2b, ba^2, ba^2b\}$. Dacă aplicăm algoritmului Sardinas-Patterson mulțimii C obținem:

- $C_1 = \{b\}$;
- $C_2 = \{a^2, a^2b\}$.

Cum $C \cap C_2 \neq \emptyset$, deducem că C nu este cod.

Exemplul 3.5.2.2. Fie $C = \{ab, ab^n, b^m a\}$, unde $n, m \geq 1$. Aplicând algoritmul Sardinas-Patterson mulțimii C , obținem $C_1 = \{b^{n-1}\}$. Vom analiza următoarele două cazuri:

1. $n = 1$. În acest caz $C_1 = \emptyset$, ceea ce conduce imediat la faptul că C este cod, indiferent de valoarea lui m ;

2. $n \neq 1$. Atunci vom analiza următoarele 3 subcazuri:

- (a) $m < n - 1$. În acest caz $C_2 = \emptyset$, ceea ce conduce imediat la faptul că C este cod;
- (b) $m > n - 1$. În acest caz, $C_2 = \{b^{m-n+1}a\}$ și $C_3 = \emptyset$, de unde urmează că C este cod;

(c) $m = n - 1$. În acest caz, $C_2 = \{a\}$, $C_3 = \{b, b^n\}$, $C_4 = \{b^{m-1}a\}$ și $C_5 = \emptyset$, ceea ce arată că C este cod.

Ca urmare, C este cod.

3.5.3. Măsura unui cod

Sunt rare cazurile în care limbajele pot fi caracterizate prin cantități numerice asociate lor. Un astfel de caz rar îl constituie clasa codurilor de lungime variabilă. Vom arăta, de exemplu, că o mulțime de cuvinte (un limbaj) este cod maximal dacă și numai dacă indicatorul de cod al ei ia valoarea 1.

Pentru a face prezentarea cât mai facilă vom reaminti întâi câteva elemente de teoria seriilor cu termeni pozitivi (pentru detalii recomandăm [166]).

O serie de numere reale este un cuplu $((a_n)_{n \geq 0}, (S_n)_{n \geq 0})$ format din două siruri de numere reale, în care cel de-al doilea sir este sirul sumelor parțiale asociat primului sir. În mod ușual, seria de mai sus se notează simplificat prin $\sum_{n \geq 0} a_n$. Dacă sirul sumelor parțiale este convergent (are limită finită), atunci seria este numită *convergentă*, iar limita acestui sir se numește *suma seriei*. Altfel, seria este numită *divergentă*. Vom mai scrie $\sum_{n \geq 0} a_n = s$ pentru a specifica faptul că limita sirului sumelor parțiale a seriei $\sum_{n \geq 0} a_n$ este s .

Dacă unei serii i se adaugă sau suprimă un număr finit de termeni, atunci natura ei nu se schimbă; suma se modifică însă.

În cazul *seriilor cu termeni pozitivi*, adică $a_n \geq 0$ pentru orice $n \geq 0$, condiția necesară și suficientă ca seria $\sum_{n \geq 0} a_n$ să fie convergentă este ca sirul sumelor parțiale să fie majorat. În plus, pentru astfel de serii, schimbarea ordinii termenilor nu afectează natura seriei (convergentă sau divergentă), iar în caz de convergență suma seriei nu se schimbă. Ca urmare, putem scrie $\sum_{n \in \mathbb{N}} a_n$ în loc de $\sum_{n \geq 0} a_n$, gândind “ordinea de sumare” ca nefiind în mod necesar ordinea uzuială pe \mathbb{N} (ordinea de sumare se reflectă în construcția sirului sumelor parțiale). Mai mult, dacă A este o mulțime cel mult numărabilă iar x_a este un număr real pozitiv, pentru orice $a \in A$, atunci prin $\sum_{a \in A} x_a$ vom nota suma uzuială în cazul în care A este finită, sau seria $\sum_{n \geq 0} x_{f(n)}$, unde f este o bijecție arbitrară dar fixată de la \mathbb{N} la A . În baza celor menționate mai sus, modul de alegere a bijecției f nu schimbă natura, iar atunci când este cazul, suma seriei.

Fie $\sum_{a \in A} x_a$ și $\sum_{b \in B} y_b$ două serii cu termeni pozitivi, iar $f : A \rightarrow B$ o funcție cu proprietatea $x_a = y_{f(a)}$, pentru orice $a \in A$. Dacă f este injectivă, atunci $\sum_{a \in A} x_a \leq \sum_{b \in B} y_b$, iar dacă f este bijectivă atunci inegalitatea de mai sus este satisfăcută prin egalitate.

Produsul a două serii $\sum_{a \in A} x_a$ și $\sum_{b \in B} y_b$ este seria în care figurează toate produsele $x_a y_b$ cu $a \in A$ și $b \in B$. Un procedeu convenabil pentru scrierea acestor produse în forma unui sir este procedeul Cauchy, sugerat de înmulțirea polinoamelor. În cazul seriilor cu termeni pozitivi, datorită faptului că ordinea termenilor în serie

nu afectează natura și suma acesteia (atunci când este cazul), vom denota produsul celor două serii prin $\sum_{a \in A, b \in B} x_a y_b$.

Este bine cunoscut faptul că produsul (Cauchy) a două serii convergente poate să nu fie serie convergentă. Însă dacă produsul este serie convergentă atunci suma seriei produs este produsul sumelor celor două serii. În cazul serilor cu termeni pozitivi, dacă două astfel de serii sunt convergente, atunci produsul este serie convergentă și, ca urmare, suma seriei produs este produsul sumelor serilor. Formal,

$$\left(\sum_{a \in A} x_a \right) \left(\sum_{b \in B} y_b \right) = \sum_{a \in A, b \in B} x_a y_b,$$

pentru orice două serii cu termeni pozitivi convergente $\sum_{a \in A} x_a$ și $\sum_{b \in B} y_b$.

Cu această scurtă introducere în teoria seriilor cu termeni pozitivi putem trece la a aborda conceptul de măsură a unui cod și demonstra câteva dintre proprietățile de bază ale acestuia.

Definiția 3.5.3.1. Fie A o mulțime nevidă și cel mult numărabilă. Se numește *distribuție de probabilitate* pe A orice funcție π de la A la mulțimea numerelor reale pozitive \mathbf{R}_+ pentru care are loc $\sum_{a \in A} \pi(a) = 1$.

În tot ceea ce urmează, atunci când vom vorbi despre distribuții de probabilitate pe o mulțime A vom înțelege că A este nevidă și cel mult numărabilă.

Distribuțiile de probabilitate $\pi : A \rightarrow \mathbf{R}_+$ se extind la homomorfisme de la A^* la monoidul multiplicativ al numerelor reale pozitive, în mod ușor; extensia va fi notată tot prin π ⁴.

Observația 3.5.3.1. Fie π o distribuție de probabilitate pe A .

- (1) Direct de la definiție urmează că are loc $\pi(a) \in [0, 1]$, pentru orice $a \in A$.
- (2) Pentru a justifica necesitatea extensiei distribuțiilor la monoizi liberi, vom anticipa un concept care va fi discutat în detaliu în Secțiunea 3.5.5. Se numește *sursă de informații* orice cuplu $IS = (A, \pi)$, unde A este o mulțime nevidă și cel mult numărabilă, numită *alfabetul sursei*, iar π este o distribuție de probabilitate pe A . O sursă de informații IS trebuie gândită ca un dispozitiv “black-box” ce are capacitatea de a emite informații prin emiterea de simboluri $a \in A$. Probabilitatea cu care este emis un simbol $a \in A$ este $\pi(a)$. Proprietatea de homomorfism,

$$\pi(a_1 \cdots a_n) = \pi(a_1) \cdots \pi(a_n),$$

pentru orice $a_1, \dots, a_n \in A$, o interpretăm prin aceea că probabilitatea emiterii unui simbol este independentă de simbolurile emise anterior.

Definiția 3.5.3.2. O distribuție de probabilitate π pe A spunem că este *pozitivă* dacă $\pi(a) > 0$, pentru orice $a \in A$.

⁴În [8], astfel de extensii sunt numite *distribuții Bernoulli* pe A^* .

Exemplu 3.5.3.1. Dacă A este o mulțime finită și nevidă, atunci funcția

$$\pi(a) = \frac{1}{|A|},$$

pentru orice $a \in A$, este o distribuție de probabilitate pozitivă pe A . Această distribuție se numește *distribuția uniformă* pe A .

Fie π o distribuție de probabilitate pe A . Vom extinde π la submulțimi $L \subseteq A^*$ considerând

- $\pi(\emptyset) = 0$, și
- $\pi(L) = \sum_{u \in L} \pi(u)$, pentru orice submulțime nevidă L a mulțimii A^* .

Atragem atenția asupra faptului că $\sum_{u \in L} \pi(u)$ reprezintă seria cu termeni pozitivi indexată după mulțimea (cel mult numărabilă) L . Ordinea termenilor, aşa cum a fost menționat la începutul acestei secțiuni, nu contează.

Numărul $\pi(L)$ este numit *măsura lui L relativ la distribuția de probabilitate π* . Atunci când π este distribuția uniformă, $\pi(L)$ se mai numește și *indicatorul de cod al lui L* .

Propoziția 3.5.3.1. Fie π o distribuție de probabilitate pe A . Atunci au loc următoarele proprietăți:

- (1) $\pi(A^n) = 1$, pentru orice $n \geq 1$;
- (2) $\pi(\bigcup_{i \in I} L_i) \leq \sum_{i \in I} \pi(L_i)$, pentru orice familie $(L_i | i \in I)$ cel mult numărabilă de submulțimi ale lui A^* . În plus, dacă există $i \in I$ astfel încât $\pi(L_i) = \infty$, atunci $\pi(\bigcup_{i \in I} L_i) = \infty$, iar dacă familia $(L_i | i \in I)$ este formată din mulțimi disjuncte două câte două, atunci inegalitatea este satisfăcută prin egalitate;
- (3) $\pi(L_1 L_2) \leq \pi(L_1) \pi(L_2)$, pentru orice $L_1, L_2 \subseteq A^*$. În plus, dacă produsul $L_1 L_2$ este neambiguu, atunci inegalitatea este satisfăcută prin egalitate;
- (4) $\pi(L^*) \leq \sum_{n \geq 0} \pi(L^n) \leq \sum_{n \geq 0} \pi(L)^n$, pentru orice $L \subseteq A^*$. În plus, dacă $\pi(L) = \infty$, atunci $\pi(L^*) = \infty$.

Demonstrație. (1) Vom demonstra proprietatea prin inducție după $n \geq 1$. Pentru $n = 1$, proprietatea urmează direct de la Definiția 3.5.3.1. Dacă presupunem proprietatea adevărată pentru $n \geq 1$, atunci:

$$\begin{aligned} \sum_{u \in A^{n+1}} \pi(u) &= \sum_{v \in A^n, a \in A} \pi(va) \\ &= \sum_{v \in A^n, a \in A} \pi(v)\pi(a) \\ &= (\sum_{v \in A^n} \pi(v))(\sum_{a \in A} \pi(a)) \\ &= \sum_{v \in A^n} \pi(v) \\ &= 1 \end{aligned}$$

(ultima egalitate urmează în baza ipotezei inducitive).

- (2) rămâne în grija cititorului.
- (3) Utilizând (2) obținem:

$$\begin{aligned}\pi(L_1 L_2) &= \pi(\bigcup_{u \in L_1, v \in L_2} \{uv\}) \\ &\leq \sum_{u \in L_1, v \in L_2} \pi(uv) \\ &= \sum_{u \in L_1, v \in L_2} \pi(u)\pi(v) \\ &= (\sum_{u \in L_1} \pi(u))(\sum_{v \in L_2} \pi(v)) \\ &= \pi(L_1)\pi(L_2).\end{aligned}$$

Dacă produsul $L_1 L_2$ este neambiguu, atunci familia $(\{uv\} | u \in L_1, v \in L_2)$ este disjunctă, iar (2) conduce la faptul că inegalitatea va fi satisfăcută prin egalitate.

(4) urmează de la (2) și (3). \square

Propoziția 3.5.3.2. Fie π o distribuție de probabilitate pe A .

(1) Pentru orice cod C peste A au loc următoarele proprietăți:

- (1.1) $\pi(C^n) = \pi(C)^n$, pentru orice $n \geq 1$;
- (1.2) $\pi(C^*) = \sum_{n \geq 0} \pi(C^n)$. În particular, $\pi(C^*) < \infty$ dacă și numai dacă $\pi(C) < 1$.

(2) Dacă π este pozitivă iar $C \subseteq A^+$ este o submulțime nevidă pentru care $\pi(C)$ este finită și $\pi(C^n) = \pi(C)^n$, pentru orice $n \geq 1$, atunci C este cod.

Demonstrație. (1.1) Deoarece C este cod, există o bijecție între C^n și produsul cartezian al lui C cu el însuși de n ori, $C \times \dots \times C$. Ca urmare,

$$\begin{aligned}\pi(C^n) &= \sum_{c_1 \dots c_n \in C^n} \pi(c_1 \dots c_n) \\ &= \sum_{(c_1, \dots, c_n) \in C \times \dots \times C} \pi(c_1 \dots c_n) \\ &= \sum_{(c_1, \dots, c_n) \in C \times \dots \times C} \pi(c_1) \dots \pi(c_n) \\ &= \pi(C)^n.\end{aligned}$$

(1.2) Familia $(C^n | n \geq 0)$ este formată din mulțimi disjuncte două câte două. Ca urmare, Propoziția 3.5.3.1(2) și (1.1) conduc la:

$$\pi(C^*) = \pi(\bigcup_{n \geq 0} C^n) = \sum_{n \geq 0} \pi(C^n) = \sum_{n \geq 0} \pi(C)^n.$$

Dacă $\pi(C^*)$ este convergentă dar $\pi(C)$ ar fi mai mare sau egal cu 1, atunci seria $\sum_{n \geq 0} \pi(C)^n$ ar fi divergentă, ceea ce constituie o contradicție.

Reciproc, dacă $\pi(C) < 1$ atunci sirul sumelor parțiale al seriei $\sum_{n \geq 0} \pi(C)^n$ are limită finită, ceea ce înseamnă că seria este convergentă.

Deci $\pi(C^*) < \infty$ dacă și numai dacă $\pi(C) < 1$.

(2) Presupunem, prin contradicție, că C nu este cod. Există atunci un cuvânt $w \in C^*$ ce admite cel puțin două factorizări distințe. Fie

$$w = c_1 \dots c_m = c'_1 \dots c'_n$$

două astfel de factorizări, unde $c_1, \dots, c_m, c'_1, \dots, c'_n \in C$.

Cuvântul $w = uu$ va avea și el cel puțin două factorizări distințe, printre care

$$w = c_1 \dots c_m c'_1 \dots c'_n = c'_1 \dots c'_n c_1 \dots c_m.$$

Atunci

$$\begin{aligned}\pi(C)^{m+n} &= \sum_{(c'_1, \dots, c'_{m+n}) \in C \times \dots \times C} \pi(c''_1) \dots \pi(c''_{m+n}) \\ &= \sum_{(c''_1, \dots, c''_{m+n}) \in C \times \dots \times C} \pi(c''_1 \dots c''_{m+n}) \\ &\geq \pi(C^{m+n}) + \pi(w)\end{aligned}$$

(ultima inegalitate urmează de la faptul că w are cel puțin două descompuneri distincte de lungime $m+n$ și de la Propoziția 3.5.3.1(3)). Combinând această relație cu ipoteza și relația $\pi(C^{m+n}) = \pi(C)^{m+n}$, obținem $\pi(w) = 0$, ceea ce contrazice faptul că π este pozitivă. \square

Teorema 3.5.3.1. Fie C un cod peste o mulțime nevidă și cel mult numărabilă A . Atunci, pentru orice distribuție de probabilitate π pe A , are loc $\pi(C) \leq 1$.

Demonstrație. Considerăm întâi cazul în care C este cod și lungimile cuvintelor lui sunt mărginite de un număr natural arbitrar dar fixat. Presupunem deci că există $k \geq 1$ astfel încât

$$C \subseteq A \cup A^2 \cup \dots \cup A^k$$

(atragem atenția asupra faptului că C nu este în mod necesar finit deoarece A poate fi infinită). Atunci, pentru orice număr natural $j \geq 1$,

$$C^j \subseteq A \cup A^2 \cup \dots \cup A^{jk}.$$

În baza Propoziției 3.5.3.1(1) obținem

$$\pi(C^j) \leq jk,$$

iar de la Propoziția 3.5.3.2(1.1) urmează

$$\pi(C)^j \leq jk,$$

pentru orice $j \geq 1$.

Acum, dacă am presupune $\pi(C) > 1$, relația de mai sus ar conduce la o contradicție (ea trebuie să fie satisfăcută de orice număr natural $j \geq 1$, ceea ce ar implica faptul că o funcție exponențială de bază strict mai mare decât 1 este majorată peste tot de o funcție liniară). Deci $\pi(C) \leq 1$.

Fie acum cazul în care C este un cod arbitrar peste A . Considerăm mulțimile

$$C_n = \{c \in C \mid |c| \leq n\},$$

pentru orice $n \geq 1$. Conform primei etape a demonstrației,

$$\pi(C_n) \leq 1,$$

pentru orice $n \geq 1$, ceea ce ne arată că sirul sumelor parțiale a seriei $\pi(C)$ este majorat de 1. Deci, $\pi(C) \leq 1$. \square

Corolarul 3.5.3.1. Fie C un cod peste o mulțime nevidă și cel mult numărabilă A . Dacă există o distribuție pozitivă de probabilitate pe A , π , astfel încât $\pi(C) = 1$, atunci C este cod maximal peste A .

Demonstrație. Presupunem că există o distribuție de probabilitate pozitivă pe A , π , astfel încât $\pi(C) = 1$, dar C nu este maximal. Atunci, există un cod C' peste A cu $C \subset C'$. Putem presupune că C' conține exact un cuvânt în plus față de C , fie acesta w . Atunci

$$1 = \pi(C) < \pi(C) + \pi(w) = \pi(C') \leq 1,$$

ceea ce constituie o contradicție. Deci C este maximal. \square

Teorema 3.5.3.2. Fie A o mulțime finită și nevidă și $d = (d_n)_{n \geq 1}$ o secvență de numere naturale, nu toate 0. Atunci există un cod instantaneu peste A cu distribuția de lungime d dacă și numai dacă are loc

$$\sum_{n \geq 1} d_n \frac{1}{|A|^n} \leq 1.$$

Demonstrație. Dacă există un cod instantaneu peste A cu distribuția de lungime d , atunci inegalitatea de mai sus este satisfăcută, ea fiind caz particular a Teoremei 3.5.3.1 (cazul unei distribuții uniforme).

Reciproc, să presupunem că secvența d satisfacă inegalitatea din teoremă. Vom arăta că există o secvență de mulțimi $(C_n \mid n \geq 1)$ astfel încât:

- (i) $C_n \subseteq A^n$, pentru orice $n \geq 1$;
- (ii) $|C_n| = d_n$, pentru orice $n \geq 1$;
- (iii) $\bigcup_{i \leq n} C_i$ este cod instantaneu, pentru orice $n \geq 1$.

Observăm că există C_1 care să satisfacă (i), (ii) și (iii) deoarece d_1 satisfacă

$$d_1 \leq |A| - \sum_{n \geq 2} d_n \frac{1}{|A|^{n-1}},$$

de unde urmează $d_1 \leq |A|$ (care ne spune că se pot alege d_1 elemente din A care să formeze un cod instantaneu).

Presupunem că există mulțimile C_1, \dots, C_n care verifică (i), (ii) și (iii). Vom arăta că mulțimea A^{n+1} are cel puțin d_{n+1} cuvinte care nu au ca prefixe nici unul din cuvintele mulțimii $C_1 \cup \dots \cup C_n$. Ca urmare, va exista C_{n+1} care să verifice, împreună cu C_1, \dots, C_n , relațiile de la (i), (ii) și (iii).

Numărul de cuvinte din A^{n+1} care nu au ca prefixe nici unul din cuvintele mulțimii $C_1 \cup \dots \cup C_n$ este dat de:

$$\begin{aligned} |A|^{n+1} - |C_1||A|^n - \dots - |C_n||A|^1 &= |A|^{n+1} - d_1|A|^n - \dots - d_n|A|^1 \\ &= |A|^{n+1}(1 - d_1 \frac{1}{|A|} - \dots - d_n \frac{1}{|A|^n}) \\ &\geq |A|^{n+1} \sum_{i \geq n+1} d_i \frac{1}{|A|^i} \\ &= d_{n+1} + |A|^{n+1} \sum_{i \geq n+2} d_i \frac{1}{|A|^i} \\ &\geq d_{n+1} \end{aligned}$$

(prima inegalitate utilizează ipoteza). Aceasta încheie demonstrația teoremei. \square

Această teoremă a fost demonstrată pentru prima dată de Kraft în 1949 [106] pentru cazul codurilor instantanee finite. Inegalitatea din această teoremă este întâlnită în mod frecvent sub denumirea de *inegalitatea lui Kraft*. În 1956, McMillan [144] a arătat că această inegalitate este satisfăcută de orice cod finit, motiv pentru care ea mai poate fi întâlnită și sub denumirea de *inegalitatea Kraft-McMillan*. Teorema 3.5.3.1 generalizează inegalitatea Kraft-McMillan la coduri și distribuții de probabilitate arbitrare.

3.5.4. Coduri Huffman

În 1952, David Huffman [95] a propus o clasă de coduri optime pentru codificarea datelor, optimalitatea fiind înțeleasă ca număr minim de biți necesari codificării unui caracter al textului sursă. Aceste coduri sunt cunoscute astăzi drept *coduri Huffman*. Ele au constituit tehnică principală de compresie a datelor până prin jurul anilor 1977, când au fost propuse noi tehnici de compresie ce produc rată de compresie sub *entropie*.

Definiția 3.5.4.1. Se numește *sursă de informație* orice cuplu $S = (A, \pi)$, unde A este o mulțime nevidă și cel mult numărabilă, numită *alfabetul sursei de informație*, iar π este o distribuție de probabilitate pe A .

Pe parcursul acestei secțiuni vom considera numai *surse de informație finite*, adică surse de informație ce au un alfabet finit și a căror distribuție de probabilitate este pozitivă (a se vedea Definiția 3.5.3.2).

Definiția 3.5.4.2. Se numește *codificare* a sursei de informație $S = (A, \pi)$ orice codificare a alfabetului sursei S .

Definiția 3.5.4.3. Fie $S = (A, \pi)$ o sursă de informație și $h : A \rightarrow \Sigma^*$ un homomorfism. Se numește *lungimea medie* a homomorfismului h relativ la sursa de informație S numărul real $L_h(S)$ dat prin

$$L_h(S) = \sum_{a \in A} |h(a)|\pi(a).$$

Dacă h este o codificare a sursei de informație S , atunci lungimea medie $L_h(S)$ ne furnizează numărul mediu de simboluri necesare codificării prin h a unui simbol al sursei (fiind o medie ponderată a lungimilor cuvintelor cod asociate simbolurilor sursei de informație).

Exemplul 3.5.4.1. Fie S sursa de informație dată prin

A	a	b	c	d	e	f
π	0,4	0,2	0,1	0,1	0,1	0,1

și codificarea h

A	a	b	c	d	e	f
h	0	10	1101	1110	1100	1111

Lungimea medie a acestei codificări este 2,4, ceea ce ne spune că, prin această codificare, sunt necesari în medie 2,4 biți pentru a codifica un simbol al sursei.

Definiția 3.5.4.4. Se numește *cod Huffman* al unei surse de informație orice codificare prefix binară de lungime medie minimă a respectivei surse de informație (minimalitatea fiind înțeleasă în clasa codurilor prefix binare ale sursei de informație în cauză).

Fără a restrângе generalitatea, codurile Huffman vor fi considerate numai peste alfabetul $\{0, 1\}$.

Problema fundamentală care se pune acum este de a stabili dacă există sau nu coduri Huffman asociate surselor de informație și, dacă există, cum pot fi construite astfel de coduri. Vom răspunde la această întrebare introducând întâi conceptul de redusă a unei surse de informație.

Definiția 3.5.4.5. Fie $S = (A, \pi)$ o sursă de informație cu $n \geq 3$ simboluri. Se numește *redusă* a sursei de informație S orice sursă de informație $S' = (A', \pi')$ ce satisfacă:

1. $A' = (A - \{a, b\}) \cup \{c\}$, unde $a, b \in A$ sunt două simboluri distincte de probabilitate minimă în S (adică, $\pi(a) \geq \pi(b) \geq \pi(c)$, pentru orice $d \neq a, b$), iar c este un simbol nou;
2. $\pi'(x) = \pi(x)$, pentru orice $x \in A - \{a, b\}$, și $\pi'(c) = \pi(a) + \pi(b)$.

Este ușor de văzut că definiția de mai sus este consistentă în sensul că redusa unei surse de informație este sursă de informație.

O sursă de informație poate avea mai multe reduse, depinzând de simbolul nou introdus (ceea ce nu este relevant) și de simbolurile reduse (pot exista mai mult de 2 simboluri cu probabilitate minimă).

Vom adopta următoarele notații relativ la surse de informație:

- orice sursă de informație va fi scrisă în formă

A	a_1	a_2	\cdots	a_{n-1}	a_n
π	p_1	p_2	\cdots	p_{n-1}	p_n

cu $p_1 \geq p_2 \geq \cdots \geq p_{n-1} \geq p_n$;

- orice redusă a unei surse de informație cum e cea de mai sus va fi scrisă în formă

A'	a_1	a_2	\cdots	a_{n-2}	$a_{n-1,n}$
π'	p_1	p_2	\cdots	p_{n-2}	$p_{n-1,n}$

cu $p_{n-1,n} = p_{n-1} + p_n$ (simbolurile eliminate sunt a_{n-1} și a_n , iar cel nou introdus este $a_{n-1,n}$).

În raport cu o ordonare fixată a simbolurilor sursei de informație S și cu notația adoptată pentru simbolul nou introdus, putem spune că S' este unică redusă a sursei de informație S . Din acest punct de vedere ne vom referi adesea la “redusa unei surse de informație” construită aşa cum este prezentat mai sus.

Orice sursă de informație cu două simboluri admite drept cod Huffman codul ce asociază 0 unui simbol și 1 celuilalt. Dacă considerăm o sursă de informație cu trei simboluri,

$$S = (\{a_1, a_2, a_3\}, \pi),$$

atunci o idee naturală de a construi coduri Huffman pentru ea este următoarea:

- reducem sursa de informație S la o sursă de informație cu două simboluri $S' = (\{a_1, a_{2,3}\}, \pi')$ (presupunând $\pi(a_1) \geq \pi(a_2) \geq \pi(a_3)$);
- considerăm un cod Huffman h' pentru S' ;
- codul h dat prin $h(a_1) = h'(a_1)$, $h(a_2) = h'(a_{2,3})0$ și $h(a_3) = h'(a_{2,3})1$ este cod Huffman pentru S (ceea ce se poate verifica imediat).

Acest rezultat poate fi generalizat la surse de informație arbitrară.

Teorema 3.5.4.1. Au loc următoarele proprietăți:

- (1) orice sursă de informație admite coduri Huffman;
- (2) dacă S' este redusa unei surse de informație S cu $n \geq 3$ simboluri și h' este cod Huffman al sursei S' , atunci codificarea h a sursei de informație S , dată prin

$$h(x) = \begin{cases} h'(x), & \text{dacă } x \notin \{a_{n-1}, a_n\} \\ h'(x)0, & \text{dacă } x = a_{n-1} \\ h'(x)1, & \text{dacă } x = a_n, \end{cases}$$

este cod Huffman al sursei S .

Demonstrație. (1) Fie S o sursă de informație cu n simboluri și h un cod ce asociază fiecărui simbol al ei o secvență binară de lungime n (un astfel de cod există întotdeauna, fiind un subcod al codului bloc de lungime n). Atunci $L_h(S) = n$. Există un număr finit de coduri prefix binare pentru S ce au lungimea medie cel mult n . Ca urmare, orice cod dintre acestea, ce minimizează lungimea medie, este cod Huffman pentru S .

(2) Vom face demonstrația în trei etape.

Etapa 1: Arătăm că S admite coduri Huffman h ce satisfac

$$|h(a_1)| \leq \dots \leq |h(a_n)|.$$

Conform cu (1), S admite coduri Huffman, și fie h' un astfel de cod. Fie $i < n$. Dacă $p_i > p_{i+1}$, atunci $|h'(a_i)| \leq |h'(a_{i+1})|$ (altfel, am interschimba codul lui a_i cu cel al lui a_{i+1} și am obține un cod prefix binar h'' ce ar avea lungimea medie mai mică decât a lui h' , ceea ce ar constitui o contradicție). Dacă $p_i = p_{i+1}$ și $|h'(a_i)| > |h'(a_{i+1})|$, atunci interschimbăm codul lui a_i cu cel al lui a_{i+1} și obținem un nou cod h'' ce are exact aceeași lungime medie ca și h' .

Realizând această transformare pentru orice $i < n$, obținem un cod Huffman h pentru S ce satisfac proprietatea de mai sus.

Etapa 2: Arătăm că S admite un cod Huffman h ce satisfac $|h(a_{n-1})| = w0$ și $|h(a_n)| = w1$ (sau invers), unde w este o secvență binară.

Conform etapei 1, S admite un cod Huffman h' ce satisfac

$$|h'(a_1)| \leq \dots \leq |h'(a_n)|.$$

Fie $|h'(a_n)| = wb$, unde $b \in \{0, 1\}$. Considerăm funcția h'' dată prin:

- $h''(a_i) = h'(a_i)$, pentru orice $1 \leq i \leq n-1$;
- $h''(a_n) = w$.

Atunci $L_{h''}(S) < L_{h'}(S)$, ceea ce arată că h'' nu este cod prefix (altfel el ar avea lungimea medie mai mică decât cea a unui cod Huffman). Cum h' este cod prefix, deducem că singura posibilitate pentru care h'' nu este cod prefix constă în aceea că

există $i < n$ astfel încât w este prefix în $h'(a_i)$. Atunci $h'(a_i) = wb'$, unde $b' \in \{0, 1\}$ și $b' \neq b$. Sirul de inegalități $|h'(a_i)| \leq \dots \leq |h'(a_n)|$ conduce la

$$|h'(a_i)| = \dots = |h'(a_n)|.$$

Aceasta ne arată că putem interschimba cuvintele cod $h'(a_i)$ și $h'(a_{n-1})$, obținând un nou cod prefix h ce are aceeași lungime medie ca și h' . Deci h este cod Huffman ce satisfac cerințele acestei etape.

Etapa 3: Fie h' un cod Huffman al sursei S' și h codul obținut din h' ca în enunțul teoremei. Este clar că h este cod prefix. Are loc:

$$L_h(S) = L_{h'}(S') + p_{n-1} + p_n.$$

Fie \bar{h} un cod Huffman binar pentru S ce satisfac cerințele etapei 2. Construim codul \tilde{h} pentru S' prin:

- $\tilde{h}(a_i) = \bar{h}(a_i)$, pentru orice $1 \leq i \leq n-2$;
- $\tilde{h}(a_{n-1,n}) = w$.

Este ușor de văzut că \tilde{h} este cod prefix pentru S' și are loc:

$$L_{\tilde{h}}(S) = L_{\bar{h}}(S') + p_{n-1} + p_n.$$

Atunci:

$$\begin{aligned} L_h(S) &= L_{h'}(S') + p_{n-1} + p_n \\ &= L_{\bar{h}}(S) - (L_{\bar{h}}(S') - L_{h'}(S')) \\ &\leq L_{\bar{h}}(S) \end{aligned}$$

(ultima inegalitate urmează de la faptul că $L_{\bar{h}}(S') \geq L_{h'}(S')$). Cum \bar{h} este cod Huffman pentru S , deducem că are loc $L_h(S) = L_{\bar{h}}(S)$, ceea ce arată că h este cod Huffman pentru S . \square

Teorema 3.5.4.1 ne arată cum putem construi coduri Huffman asociate surselor de informație cu cel puțin trei simboluri. Datează o sursă de informație S cu $n \geq 3$ simboluri, procedăm astfel:

- ordonăm simbolurile descrescător după probabilitate, după care reducem sursa de informație S obținând o sursă de informație S' cu $n-1$ simboluri. Dacă $n-1 = 2$, atunci putem construi un cod Huffman pentru S' , de la care vom putea construi un cod Huffman pentru S ;
- dacă S' are cel puțin trei simboluri, atunci repetăm pasul anterior cu S' în locul sursei de informație S .

Vom considera un exemplu care să ilustreze aceste operații.

Exemplul 3.5.4.2. Fie S sursa de informație din tabelul de mai jos:

A	a	b	c	d	e	f
π	0,4	0,2	0,1	0,1	0,1	0,1

“Arboarele” din Figura 3.13, citit de la stânga la dreapta, descrie procesul de reducere a sursei de informație S până la o sursă de informație cu două simboluri.

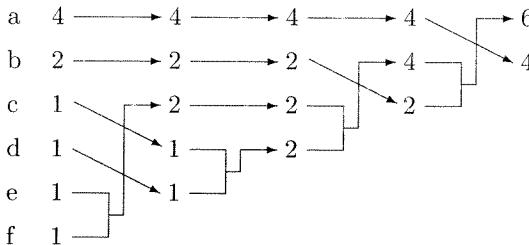


Figura 3.13: Reducerea unei surse de informație

Acum, pornind de la dreapta spre stânga, putem asocia coduri Huffman surselor de informație obținute în procesul de reducere. Vom desemna aceste coduri inscripționând arcele arborelui de mai sus, ca în Figura 3.14.

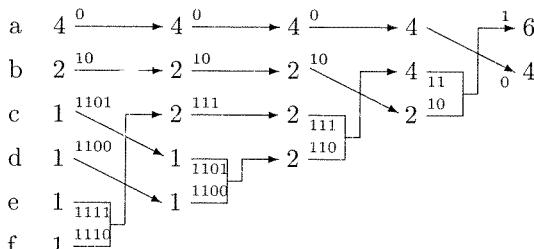


Figura 3.14: Construcția unui cod Huffman

Lungimea medie (minimă) a acestui cod este

$$4 \cdot 0,1 + 4 \cdot 0,1 + 4 \cdot 0,1 + 4 \cdot 0,1 + 2 \cdot 0,2 + 1 \cdot 0,4 = 2,4$$

Este util de remarcat că reordonarea simbolurilor nu este în general unică, ceea ce poate conduce la coduri Huffman diferite pentru aceeași sursă de informație. Ceea ce este important este că toate aceste coduri au aceeași lungime medie, care este cea minimă. De exemplu, reducerea sursei de informație S se poate face și ca în Figura 3.15. Codul Huffman obținut prin această nouă reducere nu are cuvinte cod de lungime 4, dar are cuvinte de lungime 3, spre deosebire de cazul anterior. Însă lungimea lui medie este aceeași, adică 2,4.

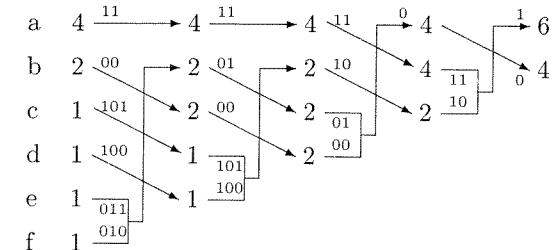


Figura 3.15: O altă reducere a aceleiași surse de informație

Deoarece codurile Huffman produc lungimi medii minime (conform definiției), este natural să ne punem problema compresiei unui text prin utilizarea unui astfel de cod. Tehnica naturală de compresie ce se poate imagina este următoarea. Fie text un text. Într-o primă etapă se parcurge textul și se evidențiază toate caracterele (simbolurile) ce apar în el, cu numărul lor de apariții (dar cel puțin una). Fie a_1, \dots, a_n aceste simboluri și $\#(a_i, \text{text})$ numărul de apariții ale caracterului a_i în textul text . Am obținut astfel o sursă de informație $S = (A, \pi)$, unde $A = \{a_1, \dots, a_n\}$ și $\pi(a_i) = \#(a_i, \text{text}) / |\text{text}|$, pentru orice $1 \leq i \leq n$. În cea de a două etapă vom asocia sursei de informație S un cod Huffman h . Atunci compresia textului text este dată prin textul $h(\text{text})$. Inevitabil, pentru orice alt cod prefix h' pentru S are loc $|h(\text{text})| \leq |h'(\text{text})|$.

Ca urmare, compresia prin coduri Huffman se realizează în două etape:

- parcurgerea textului pentru determinarea frecvenței de apariție a caracterelor în text;
- construcția codului Huffman și codificarea textului.

Deci procesul de compresie necesită două parcurgeri ale textului sursă. Pentru accelerarea procesului de compresie se poate recurge la o variantă a codificării Huffman în care textul sursă să fie scanat doar o singură dată. Ideea este foarte simplă și poate fi descrisă astfel:

- să presupunem că prefixul u al textului sursă $\text{text} = uav$ a fost parcurs și i s-a asociat codul Huffman h_u ;
- simbolul următor ce trebuie scanat, a , va fi codificat prin codul lui în h_u (deci prin $h_u(a)$), după care se va construi un cod Huffman h_{ua} pentru ua pornind de la h_u .

Pentru ca această metodă să funcționeze corect este necesar ca inițial să se pornească de la un cod Huffman asociat sevenței $a_1 a_2 \dots a_n$, presupunând că textul text este peste un alfabet $A = \{a_1, \dots, a_n\}$. Metoda descrisă mai sus poartă denumirea de *metoda Huffman adaptivă*. Se observă că, prin această metodă, un caracter a poate fi codificat diferit în funcție de poziția lui în text. Ca urmare, metoda Huffman adaptivă

nu produce un cod asociat alfabetului A . Se poate arăta că, de fapt, acestă tehnică este specifică clasei codurilor variabile în timp [208, 209].

Să ne întoarcem la metoda Huffman adaptivă și să vedem cum poate fi transformat în mod eficient codul Huffman h_u într-un cod Huffman h_{ua} . Tehnica este foarte simplă și se bazează pe faptul că frecvența de apariție a simbolului a în h_u crește cu o unitate. Aceasta face ca simbolul a să avanseze, posibil, spre început. Pentru o descriere elegantă a acestui aspect vom face apel la reprezentarea arborescentă a codurilor prefix. Frunzele acestui arbore sunt caracterele ce se codifică, iar drumul de la rădăcină la frunză furnizează codul asociat caracterului în cauză. Nodurile arborelui, etichetate cu frecvențe de apariție a simbolurilor (inclusiv a noilor simboluri obținute prin reducere) sunt ordonate crescător, de la stânga la dreapta și de la frunze spre rădăcină (pe nivele). De exemplu, codul din Figura 3.15 este reprezentat arborescent ca în Figura 3.16 (pentru fiecare nod ce nu este frunză, ramura ce conduce la descendental stâng al acestuia este considerată ca fiind etichetată cu 0, iar cealaltă cu 1; astfel, codul lui f este 010 etc.).

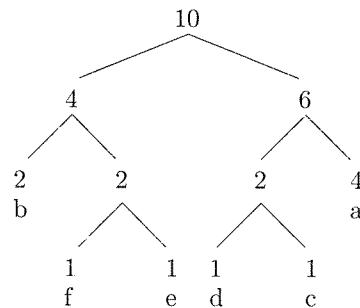


Figura 3.16: Reprezentare arborescentă a codurilor Huffman

Să presupunem că u este un text ce produce codul din Figura 3.15, acel cod fiind de fapt h_u (în notația de mai sus). Să presupunem că următorul simbol scanat este e . În vechea sursă de informație, frecvența simbolului e crește cu o unitate. Vom conveni ca noua sursă de informație să fie obținută din prima prin interschimbarea poziției lui c cu a lui e . Procedând în acest mod vom obține codul h_{ue} din Figura 3.17 ce are asociat arborele din Figura 3.18.

Interschimbarea ce am făcut-o pentru sursa inițială de informație din Figura 3.17 se reflectă și în cadrul reducerilor ce se realizează mai departe, în sensul că aceste reduceri se realizează exact după aceeași politică. Generalizând, stabilim că metoda de obținere a codului h_{ua} din codul h_u este următoarea (în termeni de arbore asociat codului):

- compară a cu succesorii din arborele asociat lui h_u (de la stânga la dreapta și de jos în sus). Dacă succesorul imediat are frecvența $k + 1$ (presupunând că frecvența lui a este k), atunci nodurile sunt în ordine. Altfel, a va trebui

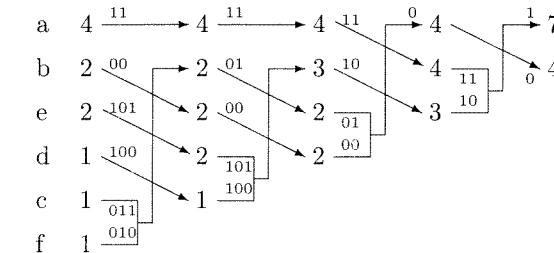


Figura 3.17: Construcția codului h_{ue}

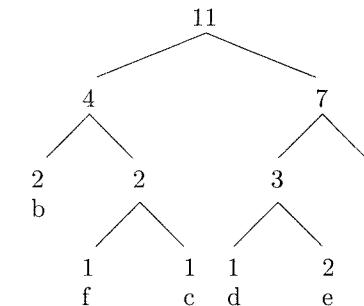


Figura 3.18: Reprezentare arborescentă a codului h_{ue}

înlocuit cu ultimul succesor ce are frecvența k sau mai mică (exceptând cazul în care succesorul este părintele lui a);

- incrementează frecvența lui a de la k la $k + 1$;
- dacă a este rădăcina arborelui, atunci transformarea este încheiată; altfel, se repetă pașii anteriori cu părintele lui a în locul lui a .

În practică se dovedește că metoda Huffman adaptivă este de preferat metodei Huffman “în doi pași”. Cum această secțiune nu are scopul de a acoperi în întregime tehnica de compresie Huffman adaptivă, cititorul interesat de detalii asupra acesteia este îndrumat către [181, 183].

3.5.5. Entropie. Limita compresiei

Conceptul de entropie, ca măsură a informației și a gradului de incertitudine, a fost introdus în 1948 de către Claude Shannon [189]. Notiunea este preluată din termodynamica, unde entropia măsoară gradul de dezordine al unui sistem fizic. Termenul “entropie” a fost utilizat pentru prima dată de Clausius în 1864.

Pentru o introducere facilă a conceptului de entropie vom urma [82, 174]. Fie $S = (\{a_i | 1 \leq i \leq n\}, (p_i | 1 \leq i \leq n))$ o sursă de informație. Dacă $p_i = 1/n$ pentru orice i , atunci *gradul de incertitudine* al apariției simbolurilor sursei este maxim (ne putem aștepta la apariția oricărui simbol al sursei, în egală măsură). Dacă $p_1 = 1$ (restul probabilităților fiind 0), atunci nu este nici o surpriză faptul că simbolul emis de sursă este numai a_1 . Ca urmare, putem spune că gradul de incertitudine este 0. Pe măsură ce p_1 scade și cresc celelalte probabilități, gradul de incertitudine crește, ajungând să fie maxim atunci când toate simbolurile au aceeași probabilitate.

Cum am defini gradul de incertitudine al apariției unui simbol a cărui probabilitate de apariție este p ? Facem întâi observația ca acest grad de incertitudine nu trebuie să depindă de simbol, ci numai de probabilitatea de apariție a acestuia. Dacă notăm prin $I(p)$ gradul de incertitudine al apariției unui astfel de simbol, atunci este natural să presupunem că $I(p)$ trebuie să satisfacă următoarele proprietăți:

1. (măsură reală pozitivă) $I(p) \geq 0$ și $I(p) = 0$ dacă și numai dacă $p = 0$ sau $p = 1$;
2. (aditivitate)⁵ $I(p_1 p_2) = I(p_1) + I(p_2)$;
3. (continuitate) $I(p)$ este funcție continuă.

Proprietatea de aditivitate conduce la $I(p^m) = mI(p)$, pentru orice $0 \leq p \leq 1$ și $m \in \mathbf{N}$. Considerând $p = q^{1/m}$, aceeași proprietate de aditivitate conduce la $I(q) = mI(q^{1/m})$, de la care urmează

$$I(q^{m/k}) = (m/k)I(q),$$

pentru orice $m, k \in \mathbf{N}$ cu $k \neq 0$. Ca urmare, funcția I se comportă ca și funcția logaritm pentru cazul puterilor raționale. Cerința de continuitate ne permite să extindem această proprietate la puteri reale și, totodată, să considerăm forma funcției $I(p)$ ca fiind

$$I(p) = \begin{cases} 0, & \text{dacă } p = 0 \\ k \log p, & \text{altfel,} \end{cases}$$

unde logaritmul este într-o bază nespecificată. Pentru a avea satisfăcută și prima cerință asupra acestei funcții, este natural să considerăm $k = -1$. Ca urmare, am fi tentați să presupunem forma funcției I ca fiind

$$I(p) = \begin{cases} 0, & \text{dacă } p = 0 \\ -\log p = \log(1/p), & \text{altfel.} \end{cases}$$

Revenind la baza logaritmului, putem spune că aceasta nu este importantă, ca urmare a proprietății de schimbare a bazei și a proprietății de aditivitate pe care o satisfacă I . Astfel, ea va fi considerată în tot ceea ce urmează ca fiind 2.

⁵Probabilitatea apariției a două simboluri, independent unul de celălalt, este produsul probabilităților de apariție a acestora.

Întrebarea care se pune acum este dacă mai există și altă funcție, să zicem $g(p)$, care să satisfacă proprietățile 1–3 de mai sus. Vom arăta că o astfel de funcție este în mod necesar de forma

$$g(p) = \begin{cases} 0, & \text{dacă } p = 0 \\ C \log(1/p), & \text{altfel,} \end{cases}$$

ceea ce ne spune că funcția I este unică până la o constantă de proporționalitate.

În adevăr, dacă $g(p)$ este o funcție ce satisfacă proprietățile 1–3, atunci urmează că are loc $g(p^m) = mg(p)$, pentru orice $m \in \mathbf{N}$, de la care obținem

$$g(p^m) - C \log(1/p^m) = m(g(p) - C \log(1/p)),$$

pentru orice constantă C și $p > 0$. Alegând $C = g(p_0)/\log(1/p_0)$ cu p_0 diferit de 0 sau 1, obținem $g(p_0^m) - C \log(1/p_0^m) = 0$, adică $g(p_0^m) = C \log(1/p_0^m)$, pentru orice $m \in \mathbf{N}$. Cum orice număr real $p \in (0, 1)$ poate fi scris în forma $p = p_0^m$ cu o alegere potrivită pentru m , deducem că are loc $g(p) = C \log(1/p)$ pentru orice p de acest tip. În plus, $g(1) = 0 = g(0)$. Aceasta ne arată că funcția g este de forma menționată mai sus și, deci, g și I diferă între ele doar printr-o constantă.

Am determinat până acum forma pe care ar trebui să o aibă funcția ce furnizează gradul de incertitudine al apariției unui simbol al unei surse. Dacă sursa S are n simboluri, $S = (\{a_i | 1 \leq i \leq n\}, (p_i | 1 \leq i \leq n))$, *gradul mediu de incertitudine*, notat $H(p_1, \dots, p_n)$, este suma ponderată a gradelor de incertitudine asociate simbolurilor sursei. Pentru a exprima în mod elegant această sumă ponderată, convenim ca $0 \cdot \log(1/0)$ să fie 0. Atunci putem scrie,

$$H(p_1, \dots, p_n) = \sum_{i=1}^n p_i \log(1/p_i).$$

Definiția 3.5.5.1. Fie S o sursă de informație cu n simboluri și distribuția de probabilitate p_1, \dots, p_n . *Entropia* sursei de informație S , notată $H(S)$, este definită ca fiind $H(S) = H(p_1, \dots, p_n)$.

Ne vom ocupa acum de studiul câtorva proprietăți de bază a entropiei. Reamintim întâi următoarea inegalitate a cărei demonstrație poate fi ușor stabilită de către cititor.

Lema 3.5.5.1. Pentru orice $b > 1$ și $x > 0$, are loc $\log_b x \leq x - 1$. Inegalitatea este satisfăcută prin egalitate dacă și numai dacă $x = 1$.

Cu ajutorul acestei leme putem stabili următoarea inegalitate cunoscută sub numele de *inegalitatea lui Gibbs*.

Lema 3.5.5.2. (Inegalitatea lui Gibbs)

Fie $b > 1$ și $p_1, q_1, \dots, p_n, q_n$ numere reale astfel încât $0 \leq p_i, q_i \leq 1$, pentru orice i ,

$\sum_{i=1}^n p_i = 1$ și $\sum_{i=1}^n q_i \leq 1$. Atunci

$$\sum_{i=1}^n p_i \log_b(1/p_i) \leq \sum_{i=1}^n p_i \log_b(1/q_i),$$

cu convențiile $0 \cdot \log(1/0) = 0$ și $p \cdot \log(1/0) = \infty$, pentru orice $p > 0$. În plus, inegalitatea este satisfăcută prin egalitate dacă și numai dacă $p_i = q_i$, pentru orice $1 \leq i \leq n$.

Demonstrație. Conform Lemei 3.5.5.1, putem scrie

$$\log_b(q_i/p_i) \leq (q_i/p_i) - 1,$$

pentru orice $p_i \neq 0$ și $q_i \neq 0$. În plus, inegalitatea este satisfăcută prin egalitate dacă și numai dacă $q_i/p_i = 1$, ceea ce este echivalent cu $p_i = q_i$.

Inegalitatea de mai sus împreună cu convențiile adoptate conduc la

$$p_i \log_b(1/p_i) \leq p_i \log_b(1/q_i) + q_i - p_i,$$

pentru orice i . Prin însumare obținem

$$\sum_{i=1}^n p_i \log_b(1/p_i) \leq \sum_{i=1}^n p_i \log_b(1/q_i) + \sum_{i=1}^n (q_i - p_i).$$

Deoarece $\sum_{i=1}^n (q_i - p_i) \leq 0$, deducem că are loc

$$\sum_{i=1}^n p_i \log_b(1/p_i) \leq \sum_{i=1}^n p_i \log_b(1/q_i),$$

inegalitatea fiind satisfăcută prin egalitate dacă și numai dacă $p_i = q_i$, pentru orice $1 \leq i \leq n$. \square

Corolarul 3.5.5.1. Pentru orice distribuție de probabilitate p_1, \dots, p_n are loc

$$0 \leq H(p_1, \dots, p_n) \leq \log n.$$

Demonstrație. Prima inegalitate urmează de la $p_i \log(1/p_i) \geq 0$, pentru orice i .

Folosind Lema 3.5.5.2 și relația

$$\log n = \sum_{i=1}^n p_i \log(1/(1/n))$$

obținem cu ușurință și cea de a doua inegalitate. \square

Corolarul 3.5.5.2. Pentru orice distribuție de probabilitate p_1, \dots, p_n au loc relațiile:

(1) $H(p_1, \dots, p_n) = 0$ dacă și numai dacă există i astfel încât $p_i = 1$;

(2) $H(p_1, \dots, p_n) = \log n$ dacă și numai dacă $p_1 = \dots = p_n = 1/n$.

Demonstrație. (1) Este clar că $\sum_{i=1}^n p_i \log(1/p_i) = 0$ dacă și numai dacă $p_i \log(1/p_i) = 0$, pentru orice i . Ca urmare, va exista i astfel încât $p_i = 1$ (nu toți p_j pot fi 0 deoarece p_1, \dots, p_n este o distribuție de probabilitate).

(2) Conform Lemei 3.5.5.2, inegalitatea

$$H(p_1, \dots, p_n) = \sum_{i=1}^n p_i \log(1/p_i) \leq \sum_{i=1}^n p_i \log(1/(1/n)) = \log n$$

este satisfăcută prin egalitate dacă și numai dacă $p_i = 1/n$, pentru orice i . \square

În cazul unei surse cu 2 simboluri cu probabilitățile p și $1-p$, funcția H , privită ca o funcție de o singură variabilă, are graficul din Figura 3.19.

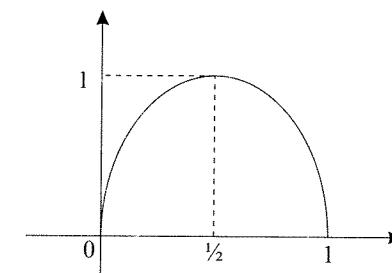


Figura 3.19: Funcția $H(p, 1-p)$

Definiția 3.5.5.2. Produsul a două surse de informație S_1 și S_2 , notat $S_1 \circ S_2$, este definit ca fiind sursa de informație

$$S_1 \circ S_2 = (\{(a_i, a_j) | 1 \leq i \leq n, 1 \leq j \leq m\}, (p_i \cdot q_j | 1 \leq i \leq n, 1 \leq j \leq m)),$$

unde $S_1 = (\{a_i | 1 \leq i \leq n\}, (p_i | 1 \leq i \leq n))$ și $S_2 = (\{b_j | 1 \leq j \leq m\}, (q_j | 1 \leq j \leq m))$.

Este ușor de văzut că produsul a două surse de informație este sursă de informație. Atunci când $S_2 = S_1$ vom scrie S_1^2 în loc de $S_1 \circ S_2$, notație ce poate fi generalizată la un număr arbitrar de surse de informație.

Teorema 3.5.5.1. Pentru orice surse de informație S_1 și S_2 are loc

$$H(S_1 \circ S_2) = H(S_1) + H(S_2).$$

Demonstrație. Au loc relațiile:

$$\begin{aligned} H(S_1 \circ S_2) &= \sum_{i,j} p_i q_j \log(1/(p_i q_j)) \\ &= \sum_{i,j} p_i q_j \log(1/(p_i) + \sum_{i,j} p_i q_j \log(1/(q_j))) \\ &= (\sum_j q_j)(\sum_i p_i \log(1/(p_i)) + (\sum_i p_i)(\sum_j q_j \log(1/(q_j))) \\ &= H(S_1) + H(S_2) \end{aligned}$$

ce demonstrează teorema. \square

Direct de la Teorema 3.5.5.1 obținem:

Corolarul 3.5.5.3. Dacă S este o sursă de informație, atunci pentru orice $k \geq 1$ are loc $H(S^k) = kH(S)$.

Teorema 3.5.5.2. (Teorema lui Shannon pentru canale fără zgomot)

Fie S o sursă de informație. Atunci au loc următoarele proprietăți:

- (1) $H(S) \leq L_h(S)$, pentru orice codificare h a sursei de informație S ;
- (2) $H(S) \leq L_h(S) < H(S) + 1$, pentru orice cod Huffman h al sursei de informație S ;
- (3) $\lim_{k \rightarrow \infty} \frac{L_{min}(S^k)}{k} = H(S)$, unde $L_{min}(S')$ reprezintă lungimea medie (minimă) a unui cod Huffman al sursei de informație S' .

Demonstrație. (1) Fie h o codificare pentru sursa de informație S cu n simboluri a_1, \dots, a_n și distribuția de probabilitate p_1, \dots, p_n . Atunci:

$$\begin{aligned} L_h(S) &= \sum_{i=1}^n p_i |h(a_i)| \\ &= \sum_{i=1}^n p_i \log 2^{|h(a_i)|} \\ &= \sum_{i=1}^n p_i \log(1/(1/2^{|h(a_i)|})). \end{aligned}$$

Deoarece $0 \leq 1/2^{|h(a_i)|} \leq 1$ și $\sum_{i=1}^n 1/2^{|h(a_i)|} \leq 1$ (conform inegalității Kraft-McMillan), în baza Lemei 3.5.5.2 urmează $H(S) \leq L_h(S)$.

(2) Este suficient de arătat că există un cod prefix h al sursei de informație S pentru care are loc $L_h(S) < H(S) + 1$.

Pentru orice i există un număr natural $d_i \neq 0$ astfel încât

$$\log(1/p_i) \leq d_i < 1 + \log(1/p_i).$$

Atunci:

$$\sum_{i=1}^n (1/2^{d_i}) \leq \sum_{i=1}^n (1/2^{\log(1/p_i)}) = \sum_{i=1}^n (1/(1/p_i)) = 1.$$

Ca urmare, conform Teoremei 3.5.3.2, există un cod prefix h ce satisfacă $|h(a_i)| = d_i$, pentru orice i . În plus:

$$\begin{aligned} L_h(S) &= \sum_{i=1}^n p_i d_i \\ &< \sum_{i=1}^n p_i (1 + \log(1/p_i)) \\ &= \sum_{i=1}^n p_i + \sum_{i=1}^n p_i \log(1/p_i) \\ &= H(S) + 1. \end{aligned}$$

(3) Conform cu (2), are loc $H(S^k) \leq L_{min}(S^k) < H(S^k) + 1$, ceea ce conduce la

$$H(S) \leq \frac{L_{min}(S^k)}{k} < H(S) + \frac{1}{k},$$

de unde obținem concluzia teoremei. \square

Observația 3.5.5.1.

- (1) Codurile de lungime variabilă se utilizează în special pentru compresia informației, fără a lua în calcul erorile care pot apărea atunci când informația este transmisă pe un canal de transmisie (ca și cum canalul nu ar altera informația ce îl parcurge). Cum canalele de transmisie se clasifică în canale cu zgomot (ce pot produce erori) și canale fără zgomot (ce nu produc erori), aceasta justifică denumirea Teorema 3.5.5.2 (a se vedea și Secțiunea 6.6).
- (2) Teorema 3.5.5.2(1) ne spune că lungimea medie a codurilor Huffman pentru o sursă de informație S nu poate scădea sub $H(S)$. Punctul (3) al aceleiași teoreme ne spune că, grupând din ce în ce mai multe simboluri ale sursei de informație și asociind acestora coduri Huffman, lungimea medie tinde la $H(S)$.
- (3) Există tehnici de compresie pentru care lungimea medie scade sub entropia sursei de informație. Pentru detalii cititorul este îndrumat către [181].

Capitolul 4

Grupuri

Teoria grupurilor, cea mai veche ramură a algebrei moderne, își are originile în lucrările lui Joseph Louis Lagrange (1736–1813), Paolo Ruffini (1765–1822) și Evariste Galois (1811–1832). Grupurile considerate de aceștia au fost în principal grupuri de rădăcini ale polinoamelor sau grupuri de permutări. Ca urmare, acestea erau grupuri finite. Interesul pentru grupuri infinite a apărut din topologie și geometrie, fiind stimulat de lucrările lui Felix Klein (1849–1925), Henri Poincaré (1854–1912) și mulți alții. Școala rusă condusă de Alexander Kurosh (1908–1971) a adus contribuții deosebite la dezvoltarea teoriei grupurilor.

Teoria grupurilor este indispensabilă informaticii ca urmare a aplicațiilor majore pe care aceasta le are. În acest capitol vom prezenta elemente de bază de teoria grupurilor și vom puncta câteva aplicații ale acesteia în criptografie.

4.1. Definiții. Exemple. Proprietăți de bază

Așa cum s-a spus în Secțiunea 1.4.4.1, un *grup* este o algebră $\mathbf{G} = (G, \cdot, ', e)$ formată dintr-o mulțime nevidă G , o operație binară \cdot , o operație unară $'$ și o operație nulară e ce satisfac următoarele proprietăți:

- (1) \cdot este asociativă;
- (2) $(\forall x \in G)(x \cdot e = e \cdot x = x);$
- (3) $(\forall x \in G)(x \cdot x' = x' \cdot x = e).$

Dacă \mathbf{G} satisfacă în plus

- (4) $(\forall x, y \in G)(x \cdot y = y \cdot x),$

Atunci **G** este numit *grup comutativ* sau *abelian*.

Elementul e (de la (2)) este numit *unitatea* lui **G** și el este unicul cu proprietatea (2) (a se vedea Exemplul 1.4.4.1(3)). Uzual el se mai notează prin 1_G sau chiar 1, dacă nu există pericol de confuzie. Pentru orice $x \in G$, x' este numit *inversul* lui x și, de asemenea, este unicul cu proprietatea (3), așa cum s-a arătat în Exemplul 1.4.4.1(3). De altfel, pentru toate concepcile de bază asupra grupurilor, cum ar fi cele de *subgrup*, *subgrup generat*, *grup ciclic*, *ordin al unui grup*, *ordin al unui element într-un grup*, *homomorfism de grupuri* și *congruențe în grupuri*, invităm cititorul să parcurgă Secțiunea 1.4.4.

Atunci când nu va fi pericol de confuzie ne vom referi la grupul **G** prin intermediul mulțimii G , așa cum am făcut și în cadrul semigrupurilor și monoizilor. Adică, vom spune simplu că “ G este grup”.

În cadrul monoizilor s-au definit puterile întregi pozitive ale elementelor acestora. Pentru grupuri putem defini și puteri negative ale elementelor. Fie (G, \cdot', e) un grup și $a \in G$. Definim:

- $a^0 = e$;
- $a^n = a^{n-1} \cdot a$, pentru orice $n \geq 1$;
- $a^{-1} = a'$, unde a' este inversul lui a ;
- $a^{-n} = (a^{-1})^n$, pentru orice $n \geq 1$,

pe care le vom numi *puterile întregi ale lui a*.

Pentru $n \geq 1$, a^n reprezintă o notație pentru a operat cu el însuși de n ori

$$\underbrace{a \cdots a}_{n \text{ ori}}$$

iar a^{-n} reprezintă o notație pentru a^{-1} operat cu el însuși de n ori

$$\underbrace{a^{-1} \cdots a^{-1}}_{n \text{ ori}}$$

Atunci când grupul este notat aditiv, notările de mai sus sunt schimbate în na și, respectiv, $-na$, întocmai ca la monoizi. Mai precis, dacă grupul este $(G, +, -, 0)$, atunci definim:

- $0a = 0$;
- $na = (n-1)a + a$, pentru orice $n \geq 1$;
- $(-1)a = -a$, unde $-a$ este inversul lui a ;
- $(-n)a = n(-a)$, pentru orice $n \geq 1$,

pe care le vom numi *multiplii întregi ai lui a*.

Pentru $n \geq 1$, na reprezintă o notație pentru

$$\underbrace{a + \cdots + a}_{n \text{ ori}}$$

iar $-na$ reprezintă o notație pentru

$$\underbrace{(-a) + \cdots + (-a)}_{n \text{ ori}}$$

Inversul unui element a într-un grup notat aditiv $(G, +, -, 0)$ se mai numește și *opusul lui a*. Este ușor de văzut că opusul lui 0 este 0, adică $-0 = 0$.

Scăderea în $(G, +, -, 0)$ este definită în mod ușor prin

$$a - b = a + (-b),$$

pentru orice $a, b \in G$.

În tot ceea ce urmează în acest capitol vom folosi cu precădere notația multiplicativă și, ca și în cazul semigrupurilor și monoizilor, vom folosi construcții de forma aA , Aa , AB etc., unde $a \in G$ și $A, B \subseteq G$. În plus, definim

$$A^{-1} = \{a^{-1} | a \in A\}.$$

Demonstrația următoarei propoziții rămâne în seama cititorului.

Propoziția 4.1.1. Fie G un grup, $a, b \in G$ și $m, n \in \mathbf{Z}$. Atunci au loc următoarele proprietăți:

- (1) $(a^{-1})^{-1} = a$;
- (2) $(ab)^{-1} = b^{-1}a^{-1}$;
- (3) $a^m a^n = a^{m+n} = a^n a^m$;
- (4) $(a^m)^n = a^{mn} = (a^n)^m$;
- (5) $a^{-m} = (a^{-1})^m = (a^m)^{-1}$.

În mod ușor în matematică, notația “ $^{-1}$ ” se utilizează pentru a desemna inversul unui obiect, care poate fi funcție, matrice etc., atunci când stim că acest invers este unic. Din acest motiv am preferat să utilizăm pentru început, în cadrul grupurilor, notația “ \cdot' ” și să justificăm întâi unicitatea elementului a' . Prin definirea puterilor negative ale unui element a , am identificat a^{-1} cu a' . Ca urmare, putem utiliza a^{-1} pentru a desemna inversul lui a fără a crea confuzii de nici un fel.

Proprietățile din Propoziția 4.1.1 pot fi transcrise cu ușurință în notație aditivă:

- $-(-a) = a$;

- $-(a + b) = -b - a$;
- $ma + na = (m + n)a = na + ma$;
- $m(na) = (mn)a = n(ma)$;
- $(-m)a = m(-a) = -(ma)$,

pentru orice $a, b \in G$ și $m, n \in \mathbf{Z}$.

Exemplul 4.1.1.

- (1) $(\mathbf{Z}, +, -, 0)$, $(\mathbf{Q}, +, -, 0)$, $(\mathbf{R}, +, -, 0)$ și $(\mathbf{C}, +, -, 0)$ sunt grupuri comutative. Ele vor fi notate simplificat prin \mathbf{Z} , \mathbf{Q} , \mathbf{R} și, respectiv, \mathbf{C} . În plus, grupul \mathbf{Z} este ciclic.
- (2) $(\mathbf{Q}^*, \cdot, ^{-1}, 1)$, $(\mathbf{R}^*, \cdot, ^{-1}, 1)$ și $(\mathbf{C}^*, \cdot, ^{-1}, 1)$ sunt grupuri comutative. Ele vor fi notate simplificat prin \mathbf{Q}^* , \mathbf{R}^* și, respectiv, \mathbf{C}^* .
- (3) Fie $n \in \mathbf{Z}$. Considerăm mulțimea $n\mathbf{Z} = \{n \cdot a | a \in \mathbf{Z}\}$ a multiplilor întregi ai lui n . În cazul $n = 1$ sau $n = -1$, $n\mathbf{Z} = \mathbf{Z}$, iar pentru $n = 0$, $n\mathbf{Z} = \{0\}$. Atunci $(n\mathbf{Z}, +, -, 0)$ este grup comutativ, iar $(n\mathbf{Z}, \cdot, 1)$ este monoid comutativ. Atunci când nu vom specifica explicit, $n\mathbf{Z}$ se va referi la grupul comutativ $(n\mathbf{Z}, +, -, 0)$.
- (4) $(\mathbf{Z}_m, +, -, 0)$ este grup ciclic comutativ, iar $(\mathbf{Z}_m^*, \cdot, ^{-1}, 1)$ este grup comutativ, pentru orice $m \geq 1$ (a se vede Secțiunea 2.3).
- (5) Fie A o mulțime. Mulțimea tuturor funcțiilor bijective de la A la A , cu operația de compunere ca operație binară, cu inversa uzuală a funcțiilor ca operație unară și cu funcția identitate pe A ca operație nulară, formează grup¹. Acest grup este comutativ dacă și numai dacă $|A| \leq 2$.

Deoarece funcțiile bijective de la A la A se mai numesc și permutări ale mulțimii A , acest grup mai poartă denumirea de *grupul permutărilor mulțimii A* sau *grupul simetric pe A* și se notează prin $Sym(A)$. Atunci când $A = \{1, \dots, n\}$, $n \geq 1$, grupul $Sym(A)$ se mai notează și prin S_n .

Din punct de vedere istoric, grupurile S_n sunt printre primele grupuri ce au fost studiate sistematic.

¹În cazul în care A este mulțimea vidă, singura funcție de la A la A este funcția vidă, care este bijectivă. Ca urmare, în acest caz avem de a face cu un grup trivial.

Prezentăm în continuare două caracterizări utile ale conceptului de grup.

Propoziția 4.1.2. Fie G un semigrup. Atunci G este grup dacă și numai dacă există $e \in G$ astfel încât:

- (1) $ea = a$, pentru orice $a \in G$;
- (2) pentru orice $a \in G$ există $a' \in G$ astfel încât $a'a = e$.

Demonstrație. Dacă G este grup, atunci unitatea lui satisfac proprietățile (1) și (2). Reciproc, presupunem că există e cu proprietățile (1) și (2). Fie $a \in G$ și a' ca la (2). Atunci

$$(aa')(aa') = a(a'a)a' =aea' = aa'.$$

Aplicând acestei relații (aa') la stânga, (2) conduce la $aa' = e$. Deci, a' este inversul lui a (atât la stânga cât și la dreapta).

Fie $a \in G$ și a' ca la (2). Are loc

$$ae = a(a'a) = (aa')a = ea = a,$$

ceea ce arată că e este element unitate. Deci G este grup. \square

Un element e cu proprietatea din Propoziția 4.1.2(1) se numește *unitate la stânga*, iar un element a' cu proprietatea din Propoziția 4.1.2(2) se numește *invers la stânga* al lui a . Ca urmare, Propoziția 4.1.2 ne spune că un semigrup este grup dacă și numai dacă el admite unitate la stânga și orice element al său admite invers la stânga.

În mod similar se pot introduce *unitatea la dreapta* și *inversul la dreapta*, iar Propoziția 4.1.2 poate fi reformulată corespunzător.

Corolarul 4.1.1. Dacă un monoid finit este cu simplificare la stânga sau la dreapta, atunci el este grup.

Demonstrație. Fie (M, \cdot, e) un monoid finit cu simplificare la dreapta (cauzul simplificării la stânga se discută în mod similar acestuia).

Unitatea monoidului M satisfac Propoziția 4.1.2(1). Fie $a \in M$. Vom arăta că există $a' \in M$ astfel încât $a'a = e$. Pentru orice $x \in M$, $xa \in M$. Mai mult, funcția $f_a : M \rightarrow M$ dată prin $f_a(x) = xa$ este injectivă datorită proprietății monoidului M de a fi cu simplificare la dreapta. Cum M este monoid finit, f_a este chiar bijectivă. Ca urmare, există x astfel încât $xa = e$. Atunci $a' = x$ va satisface proprietatea cerută, iar în baza Propoziției 4.1.2 deducem că M este grup. \square

Propoziția 4.1.3. Fie G un semigrup.

- (1) Dacă G este grup atunci, pentru orice $a, b \in G$, ecuațiile $ax = b$ și $ya = b$ au soluție unică (în necunoscuta x și, respectiv, y), atunci G este grup.
- (2) Dacă pentru orice $a, b \in G$, ecuațiile $ax = b$ și $ya = b$ au soluție (în x și, respectiv, y), atunci G este grup.

Demonstrație. (1) este trivial satisfăcută în baza existenței și unicității inversului unui element.

Să presupunem că are loc ipoteza de la (2). Vom arăta că există unitate la stânga și invers la stânga pentru orice element din G , ceea ce ne va arăta că G este grup în baza Propoziției 4.1.2.

Fie $a \in G$. Ecuația $xa = a$ admite soluții în G , și fie e o astfel de soluție. Pentru orice $b \in G$, ecuația $b = ay$ admite soluții în G . Dacă y_0 este o astfel de soluție, atunci:

$$b = ay_0 = eay_0 = eb,$$

ceea ce arată că e este unitate la stânga în G .

Existența inversului la stânga a unui element arbitrar $a \in G$ este asigurată de existența unei soluții a ecuației $e = xa$. \square

4.2. Subgrupuri. Teorema lui Lagrange

Subgrupurile au fost introduse în Secțiunea 1.4.4.2 ca fiind submulțimi ale grupurilor ce pot fi structurate ca grupuri prin restricția operațiilor grupurilor găzădă la respectivele submulțimi. Dacă H este subgrup al grupului G , atunci vom nota aceasta prin $H \leq G$.

Exemplul 4.2.1. Considerând grupurile din Exemplul 4.1.1(1)(2)(3), au loc următoarele relații:

- (1) $\mathbf{Z} \leq \mathbf{Q} \leq \mathbf{R} \leq \mathbf{C}$;
- (2) $\mathbf{Q}^* \leq \mathbf{R}^* \leq \mathbf{C}^*$;
- (3) $n\mathbf{Z} \leq \mathbf{Z}$, pentru orice $n \in \mathbf{Z}$. Mai mult, orice subgrup al lui \mathbf{Z} este de forma $n\mathbf{Z}$, cu $n \geq 0$. În adevăr, dacă S este un subgrup al lui \mathbf{Z} , atunci se verifică cu ușurință că au loc următoarele:
 - dacă $S = \{0\}$, atunci $S = 0\mathbf{Z}$;
 - dacă $S = \mathbf{Z}$, atunci $S = 1\mathbf{Z} = (-1)\mathbf{Z}$;
 - dacă $S \subset \mathbf{Z}$ și $S \neq \{0\}$, atunci $S = n\mathbf{Z} = (-n)\mathbf{Z}$, unde n este cel mai mic număr natural strict pozitiv ce aparține lui S . În adevăr, $n\mathbf{Z} \subseteq S$ deoarece $n \in S$ și S este subgrup al lui \mathbf{Z} . Pentru a arăta că $S \subseteq n\mathbf{Z}$, presupunem prin contradicție că ar exista $k \in S - n\mathbf{Z}$. Atunci, $k > n$ (din alegerea lui n), iar teorema împărțirii cu rest conduce la existența întregilor α și r astfel încât $k = \alpha n + r$ și $0 < r < n$. Cum $k - \alpha n \in S$, urmează că are loc $r \in S$. Aceasta contrazice însă alegerea lui n deoarece $0 < r < n$.

Propoziția 4.2.1. Fie (G, \cdot', e) un grup și $H \subseteq G$ nevidă. Următoarele afirmații sunt echivalente:

- (1) H este subgrup al lui G ;
- (2) $ab \in H$ și $a' \in H$, pentru orice $a, b \in H$;
- (3) $ab' \in H$, pentru orice $a, b \in H$.

Demonstrație. Este clar că (1) implică (2), iar (2) implică (3).

Să arătăm că (3) implică (1). Pentru orice $a, b \in H$ au loc relațiile:

- $aa' = e$ și $aa' \in H$ conduc la $e \in H$;
- $ea' = a'$ și $ea' \in H$ conduc la $a' \in H$;
- $a(b')' = ab$ și $a(b')' \in H$ conduc la $ab \in H$.

Deci, H este subgrup al grupului G . \square

Corolarul 4.2.1. Fie (G, \cdot', e) un grup finit. Atunci o submulțime nevidă H a lui G este subgrup al lui G dacă și numai dacă $ab \in H$, pentru orice $a, b \in H$.

Demonstrație. Dacă H este subgrup al lui G , atunci $ab \in H$, pentru orice $a, b \in H$.

Reciproc, vom arăta că pentru orice $a \in H$, a' este tot în H . Atunci, combinând aceasta cu ipoteza obținem că are loc proprietatea din Propoziția 4.2.1(2), ceea ce ne va spune că H este subgrup al lui G .

Fie $a \in H$. Conform ipotezei, $a^k \in H$, pentru orice $k \geq 1$. Cum G este finit, există $i, j \in \mathbf{N}$ cu $i \neq j$ și $a^i = a^j$. Dacă presupunem $i < j$, atunci $a^{j-i} = e$. Deci, există $m \geq 1$ cu proprietatea $a^m = e$. Mai mult, $e = a^m \in H$. Atunci

$$a^{m-1}a = aa^{m-1} = a^m = e,$$

ceea ce ne arată că $a^{m-1} = a'$ (conform unicității inversului). Cum $a^{m-1} \in H$, deducem că $a' \in H$. \square

Fie G un grup și $H \leq G$. Subgrupul H induce două relații binare pe G foarte importante, notate \sim_H și ${}_H \sim$ și date prin

$$a \sim_H b \Leftrightarrow (\exists c \in H)(b = ac)$$

și

$${}_H \sim b \Leftrightarrow (\exists c \in H)(b = ca),$$

pentru orice $a, b \in G$. $a \sim_H b$ ne spune că b se obține din a prin înmulțire la dreapta cu un element din H , iar ${}_H \sim b$ ne spune că b se obține din a prin înmulțire la stânga cu un element din H .

Propoziția 4.2.2. Fie G un grup, H un subgrup al lui G și $a, b \in G$.

- (1) $a \sim_H b$ dacă și numai dacă $a'b \in H$.
- (2) $a_H \sim b$ dacă și numai dacă $ba' \in H$.
- (3) \sim_H și $H \sim$ sunt relații de echivalență pe G .
- (4) $[a]_{\sim_H} = aH$ și $[a]_{H \sim} = Ha$.

Demonstrație. Vom demonstra una din relațiile de la (4), de exemplu prima relație, celelalte rămânând în seama cititorului.

Fie $b \in [a]_{\sim_H}$. Atunci, $a \sim_H b$, ceea ce înseamnă că există $c \in H$ cu proprietatea $b = ac$. Dar această ultimă relație este echivalentă cu $b \in aH$.

Reciproc, dacă $b \in aH$, atunci există $c \in H$ astfel încât $b = ac$, ceea ce conduce la $b \in [a]_{\sim_H}$. \square

În baza Propoziției 4.2.2 putem scrie:

$$G = \bigcup_{a \in G} Ha = \bigcup_{a \in G} aH.$$

În plus, aceste două descompuneri ale lui G sunt partiții ale lui G deoarece atât Ha cât și aH sunt clase de echivalență. Mai mult, aceste partiții au proprietăți foarte interesante, așa cum ne arată următoarea propoziție.

Propoziția 4.2.3. Fie G un grup și H un subgrup al lui.

- (1) Multimile H , aH și Ha sunt echipotente două câte două, pentru orice $a \in G$.
- (2) Multimile $\{Ha|a \in G\}$ și $\{aH|a \in G\}$ sunt echipotente.

Demonstrație. (1) Se arată cu ușurință că funcția $f : H \rightarrow aH$ dată prin $f(b) = ab$, pentru orice $b \in H$, este bijectivă. Deci H și aH sunt echipotente.

Un raționament similar se folosește și pentru multimile H și Ha .

(2) Considerăm funcția

$$f : \{Ha|a \in G\} \rightarrow \{aH|a \in G\}$$

dată prin $f(Ha) = a'H$, pentru orice $a \in G$ (a' este inversul lui a), și arătăm că este bijectivă.

Fie $a, b \in G$. Presupunem că are loc $f(Ha) = f(Hb)$. Adică $a'H = b'H$. Cum $e \in H$, obținem că există $c \in H$ astfel încât $b' = a'c$. Această relație conduce la $b = c'a$. În plus, $c' \in H$ deoarece H este subgrup al lui G . Ca urmare, $a_H \sim b$, ceea ce arată că $Ha = Hb$ (Ha și Hb fiind clase de echivalență). Deci f este injectivă.

Pentru surjectivitate este suficient să remarcăm că $f(Ha') = aH$, pentru orice $a \in G$. Deci $\{Ha|a \in G\}$ și $\{aH|a \in G\}$ sunt echipotente. \square

Ca urmare a acestei propoziții, cele două descompuneri ale lui G au același “număr” de blocuri și, pentru orice $a \in G$, blocurile aH și Ha au același “număr” de elemente care este $|H|$ (aH și Ha au același cardinal, care este $|H|$).

Când G este grup finit, numărul de blocuri în oricare din cele două partiții ale lui G se notează prin $(G : H)$ și se numește *indexul lui H în G* . Adică

$$(G : H) = |\{Ha|a \in G\}| = |\{aH|a \in G\}|.$$

Este clar atunci că are loc

$$|G| = (G : H)|H|.$$

Am obținut astfel:

Teorema 4.2.1. (Teorema lui Lagrange²)

Pentru orice grup finit G și subgrup H al său are loc $|G| = (G : H)|H|$.

Direct de la Teorema 4.2.1 obținem următorul rezultat extrem de important.

Corolarul 4.2.2. Ordinul oricărui subgrup al unui grup finit divide ordinul grupului.

4.3. Subgrupuri normale

Fie G un grup și H un subgrup al lui. În secțiunea anterioară am văzut că H induce două relații de echivalență pe G , \sim_H și $H \sim$. Clasele de echivalență în raport cu aceste relații sunt aH și, respectiv, Ha , pentru orice $a \in G$. În plus,

$$|aH| = |Ha| = |H|,$$

pentru orice $a \in G$, și cele două relații de echivalență generează exact același număr de clase de echivalență.

Dacă aceste relații de echivalență ar fi congruențe, atunci multimile cât induse de ele ar fi automat structurate ca grupuri (a se vedea Secțiunea 1.4.4.3). Însă în general aceste relații de echivalență nu sunt congruențe. O analiză atentă arată că ele pot deveni congruențe dacă coincid (această afirmație va fi demonstrată mai jos). Un subgrup H ce induce două relații de echivalență \sim_H și $H \sim$ ce coincid, se numește subgrup normal.

Definiția 4.3.1. Fie G un grup și H un subgrup al lui. H este numit *subgrup normal al lui G* dacă are loc $aH = Ha$, pentru orice $a \in G$.

Evident, cerința “ $(\forall a)(aH = Ha)$ ” este echivalentă cu oricare din cerințele “ $\sim_H = H \sim$ ”, sau “ $(\forall a)(H = aHa')$ ”, sau “ $(\forall a)(H = a'Ha)$ ”.

²Această teoremă este inspirată de studiile lui Lagrange din perioada anilor 1770 asupra rădăcinilor polinoamelor (a se vedea [225], pag. 78); prima demonstrație a ei apare, după cât se pare, în studiile lui Galois din perioada anilor 1830.

Dacă H este subgrup normal al lui G , atunci vom nota aceasta prin $H \triangleleft G$. Arătăm acum că dacă $H \triangleleft G$, atunci relația \sim_H , care este aceeași cu $_H \sim$, este congruență în grup.

Propoziția 4.3.1. Dacă H este subgrup normal al grupului G , atunci \sim_H este congruență în G .

Demonstrație. Fie $a, b, c, d \in G$ astfel încât $a \sim_H b$ și $c \sim_H d$. Va trebui să arătăm că are loc $ac \sim_H bd$ și $a' \sim_H b'$.

Relația $a \sim_H b$ conduce la existența unui element $x \in H$ cu proprietatea $b = ax$, iar relația $c \sim_H d$ conduce la existența unui element $y \in H$ cu proprietatea $d = cy$. Atunci, $bd = axcy$.

H este subgrup normal și, deci, $Hc = cH$. Ca urmare, există $z \in H$ astfel încât $xc = cz$. Combinând cu relația de mai sus obținem

$$bd = axcy = aczy = ac(zy),$$

care ne arată că $ac \sim_H bd$.

Relația $b = ax$ de mai sus conduce la $b' = x'a'$. Cum $Ha' = a'H$, există $z \in H$ astfel încât $x'a' = a'z$. Deci, $b' = a'z$, ceea ce ne arată că $a' \sim_H b'$.

Deci, \sim_H este congruență în G . \square

Propoziția 4.3.1 ne permite să definim *grupul cât (factor) G/\sim_H* care, în mod ușor în teoria grupurilor se notează prin G/H și se numește *grupul cât (factor) induș de H în G* (cu condiția ca H să fie subgrup normal în G). Operațiile acestui grup sunt date prin³:

- $(aH)(bH) = abH$, pentru orice $a, b \in G$;
- elementul neutru este $eH = H$;
- $(aH)' = a'H$, pentru orice $a \in G$.

Prima teoremă de izomorfism din Secțiunea 1.4.4.3 poate fi reformulată și completată pentru cazul grupurilor, astfel (a se revedea definiția homomorfismelor de grupuri din Exemplul 1.4.4.4(3)).

Teorema 4.3.1. (Prima teoremă de izomorfism)

- (1) Fie G_1 și G_2 grupuri și $h : G_1 \rightarrow G_2$ un epimorfism. Atunci $G_1/\ker(h)$ și G_2 sunt izomorfe.

³În notație aditivă, clasele de echivalență sunt de forma $a + H$, cu $a \in G$. Operațiile grupului cât vor fi:

- $(a + H) + (b + H) = (a + b) + H$, pentru orice $a, b \in G$;
- elementul neutru este $0 + H = H$;
- $-(a + H) = (-a) + H$, pentru orice $a \in G$.

- (2) Fie G un grup și $H \triangleleft G$. Atunci funcția $h : G \rightarrow G/H$ dată prin $h(a) = aH$, pentru orice $a \in G$, este epimorfism. În plus, $\ker(h) = \sim_H$.

Demonstrație. (1) urmează direct de la Teorema 1.4.4.2.

(2) Vom demonstra doar egalitatea $\ker(h) = \sim_H$. Fie $(a, b) \in \ker(h)$. Atunci, $h(a) = h(b)$, ceea ce înseamnă $aH = bH$. Cum $e \in H$, va exista $c \in H$ astfel încât $b = ac$. Adică $a \sim_H b$. Am obținut astfel $\ker(h) \subseteq \sim_H$.

Reciproc, fie $(a, b) \in \sim_H$. Atunci, există $c \in H$ astfel încât $b = ac$. Vom demonstra egalitatea $aH = bH$ prin dublă incluziune. Fie $x \in aH$. Atunci există $y \in H$ astfel încât $x = ay$. Însă

$$x = ay = ac(c'y) = b(c'y),$$

ceea ce arată că $x \in bH$ (H fiind subgrup, $c', c'y \in H$). Deci $aH \subseteq bH$. În mod similar se obține și cealaltă inclusiune. Deci $aH = bH$, ceea ce conduce la $h(a) = h(b)$ și, astfel, la $(a, b) \in \ker(h)$. \square

Atragem atenția asupra faptului că nucleul unui homomorfism $h : G_1 \rightarrow G_2$ este definit diferit în lucrarea noastră decât în marea majoritate a lucrărilor de algebră clasică. La noi $\ker(h)$ este o relație binară (care este chiar congruență) în timp ce în alte lucrări $\ker(h)$ este multimea tuturor elementelor a căror imagine prin h este unitatea grupului G_2 (în această abordare, $\ker(h)$ devine subgrup normal al lui G_1). Evident, abordările sunt echivalente, însă o noastră are avantajul că poate fi translată la orice altă structură algebraică și face linie comună cu algebrele universale (a se vedea Secțiunile 1.4.4.3 și 8.5.4).

Prezentăm în continuare câteva proprietăți de bază ale subgrupurilor normale. Dacă $\langle H_i | i \in I \rangle$ este o familie de subgrupuri normale ale grupului G , atunci vom nota prin $\langle H_i | i \in I \rangle_G$ subgrupul

$$\langle H_i | i \in I \rangle_G = \langle \bigcup_{i \in I} H_i \rangle_G$$

(adică subgrupul generat de $\bigcup_{i \in I} H_i$). Dacă I este de forma $I = \{1, \dots, k\}$ cu $k \geq 1$, atunci vom scrie $\langle H_1, \dots, H_k \rangle_G$ în loc de $\langle H_i | i \in I \rangle_G$. Mai mult, vom elimina indicele “G” ori de câte ori acesta va fi subînțeleas din context.

Propoziția 4.3.2. Fie $(G, \cdot, ', e)$ un grup.

- (1) Intersecția oricărei familii nevide de subgrupuri normale ale lui G este subgrup normal al lui G .
- (2) Dacă $\langle H_i | i \in I \rangle$ este o familie de subgrupuri normale ale grupului G , atunci $\langle H_i | i \in I \rangle$ este subgrup normal al lui G .
- (3) Dacă $H_1 \triangleleft G$ și $H_2 \leq G$, atunci $H_1 \cap H_2 \triangleleft H_2$.
- (4) Dacă $H_1 \triangleleft G$ și $H_2 \leq G$, atunci $H_1 \triangleleft \langle H_1, H_2 \rangle$.

- (5) Dacă $H_1 \triangleleft G$ și $H_2 \leq G$, atunci $\langle H_1, H_2 \rangle = H_1H_2 = H_2H_1$.
(6) Dacă $H_1 \triangleleft G$ și $H_2 \leq G$, atunci $H_1 \triangleleft H_2H_1$.
(7) Dacă $H_1 \triangleleft G$, $H_2 \triangleleft G$ și $H_1 \leq H_2$, atunci $H_1 \triangleleft H_2$ și $H_2/H_1 \triangleleft G/H_1$.

Demonstrație. (1), (2) și (3) se obțin prin simple verificări de la definiții.

(4) urmează imediat de la faptul că H_1 este subgrup al grupului $\langle H_1, H_2 \rangle$ și subgrup normal al grupului G .

(5) Vom demonstra că are loc $\langle H_1, H_2 \rangle = H_1H_2$ (egalitatea $\langle H_1, H_2 \rangle = H_2H_1$ se obține în mod similar).

Facem întâi observația că $H_1 \subseteq H_1H_2$ și $H_2 \subseteq H_1H_2$ (deoarece unitatea grupului este element al intersecției $H_1 \cap H_2$).

Conform definiției, $\langle H_1, H_2 \rangle$ include H_1H_2 . Dacă mai arătăm că H_1H_2 este subgrup al lui G , atunci $\langle H_1, H_2 \rangle$ va coincide cu H_1H_2 în baza minimalității acestuia (el este cel mai mic subgrup ce include H_1 și H_2).

În primul rând, $e \in H_1H_2$. Fie acum $a, c \in H_1$ și $b, d \in H_2$. Vom arăta că $(ab)(cd) \in H_1H_2$. Are loc:

$$(ab)(cd) = a(bcb'b)d = a(bcb')(bd).$$

Deoarece H_1 este subgrup normal, $H_1 = bH_1b'$, ceea ce arată că $bcb' \in H_1$. Deci, $(ab)(cd) \in H_1H_2$.

Fie $a \in H_1$ și $b \in H_2$. Atunci

$$(ab)' = b'a' = (b'a'b)b' \in H_1H_2$$

deoarece H_1 este normal și, deci, $H_1 = b'H_1b$.

Ca urmare, H_1H_2 este subgrup al lui G , ceea ce încheie demonstrația.

(6) urmează de la (4) și (5).

(7) Prima parte a acestei proprietăți urmează direct de la (3) (observând că $H_1 \cap H_2 = H_1$). Pentru partea a doua, considerăm $aH_1 \in G/H_1$, unde $a \in G$, și arătăm că are loc

$$(aH_1)(H_2/H_1) = (H_2/H_1)(aH_1).$$

Fie $bH_1 \in H_2/H_1$, unde $b \in H_2$. Atunci

$$(aH_1)(bH_1) = abH_1.$$

Cum $a \in G$, $b \in H_2$ și $H_2 \triangleleft G$, există $c \in H_2$ astfel încât $ab = ca$. Atunci

$$(aH_1)(bH_1) = abH_1 = caH_1 = (cH_1)(aH_1),$$

care arată că $(aH_1)(H_2/H_1) \subseteq (H_2/H_1)(aH_1)$. În mod similar se obține și cealaltă inclusiune. \square

Cu acestea obținem încă două teoreme de izomorfism în teoria grupurilor.

Teorema 4.3.2. (A două teoreme de izomorfism)

Fie G un grup, $N \triangleleft G$ și $H \leq G$. Atunci grupurile $H/(N \cap H)$ și $(HN)/N$ sunt izomorfe.

Demonstrație. De la Propoziția 4.3.2(3)(6) urmează $N \cap H \triangleleft H$ și $N \triangleleft HN$.

Fie $h : H \rightarrow (HN)/N$ dată prin $h(a) = aN$, pentru orice $a \in H$. Arătăm că h este epimorfism:

- au loc relațiile

$$h(ab) = abN = (aN)(bN) = h(a)h(b),$$

pentru orice $a, b \in H$, ceea ce arată că h este homomorfism;

- elementele grupului căt $(HN)/N$ sunt de forma abN , unde $a \in H$ și $b \in N$. Însă, $abN = aN$. Ca urmare, $h(a) = abN$, pentru orice $a \in H$ și $b \in N$. Deci, h este surjecție.

Am obținut astfel că h este epimorfism.

Arătăm că are loc $\ker(h) = \sim_{N \cap H}$. Fie $a, b \in H$ astfel încât $a \ker(h) b$. Atunci $aN = bN$. De aici urmează că există $c \in N$ astfel încât $b = ac$. Mai mult, c trebuie să fie și în H deoarece relația $b = ac$ conduce la $c = a'b \in H$. Ca urmare, $a \sim_{N \cap H} b$, stabilind astfel inclusiunea $\ker(h) \subseteq \sim_{N \cap H}$. Inclusiunea în sens invers se obține în mod similar (utilizând și faptul că $\sim_{N \cap H} = N \cap H \sim$).

Astfel, teorema urmează de la prima teoremă de izomorfism. \square

Teorema 4.3.3. (A treia teoremă de izomorfism)

Fie G un grup, $N_1 \triangleleft G$ și $N_2 \triangleleft G$ astfel încât $N_1 \leq N_2$. Atunci $(G/N_1)/(N_2/N_1)$ și G/N_2 sunt grupuri izomorfe.

Demonstrație. De la Propoziție 4.3.2(7) urmează $N_1 \triangleleft N_2$ și $N_2/N_1 \triangleleft G/N_1$.

Fie $h : G/N_1 \rightarrow G/N_2$ dată prin $h(aN_1) = aN_2$, pentru orice $a \in G$. Ca în demonstrația Teoremei 4.3.2 se arată că h este epimorfism al cărei nucleu este $\ker(h) = \sim_{N_2/N_1}$. Atunci teorema urmează de la prima teoremă de izomorfism. \square

4.4. Grupuri ciclice

Așa cum s-a spus în Secțiunea 1.4.4.2, un grup G este *ciclic* dacă este generat de un singur element al său a . În notație aditivă, G este generat de $a \in G$ dacă

$$G = \langle a \rangle = \{na | n \in \mathbf{Z}\},$$

iar în notație multiplicativă,

$$G = \langle a \rangle = \{a^n | n \in \mathbf{Z}\}.$$

$(\mathbf{Z}, +, -, 0)$ este grup ciclic infinit (generat de 1), iar $(\mathbf{Z}_m, +, -, 0)$ este grup ciclic finit, pentru orice $m \geq 1$ (pentru $m = 1$ acest grup este generat de 0, iar pentru $m > 1$ grupul este generat de 1).

Următoarea teoremă, ce este oarecum similară Teoremei 3.3.3.1, ne furnizează informații precise asupra structurii grupurilor ciclice (folosind notația multiplicativă).

Teorema 4.4.1. Fie a un element al unui grup (G, \cdot', e) . Atunci are loc exact una din următoarele două proprietăți:

- (1) puterile întregi ale lui a sunt distințe două câte două și grupul ciclic generat de a este izomorf cu grupul ciclic comutativ $(\mathbf{Z}, +, -, 0)$;
- (2) există un număr natural strict pozitiv r astfel încât:
 - (a) $a^r = e$;
 - (b) $a^u = a^v$ dacă și numai dacă $u \equiv v \pmod r$, pentru orice $u, v \in \mathbf{Z}$;
 - (c) $\langle a \rangle = \{a^0, a^1, \dots, a^{r-1}\}$ conține exact r elemente;
 - (d) grupul $\langle a \rangle$ este izomorf cu grupul ciclic $(\mathbf{Z}_r, +, -, 0)$.

Demonstrație. Fie $a \in G$. Considerând secvența infinită

$$\dots, a^{-2}, a^{-1}, a^0 = e, a^1, a^2, \dots$$

există două cazuri.

Cazul 1: Pentru orice $i, j \in \mathbf{Z}$, dacă $i \neq j$, atunci $a^i \neq a^j$. Este clar atunci că submulțimea $\{a^n | n \in \mathbf{Z}\}$ este subgrup al grupului G și funcția $h(a^n) = n$, pentru orice $n \in \mathbf{Z}$, stabilește un izomorfism între acest grup și $(\mathbf{Z}, +, -, 0)$.

Cazul 2: Există $i, j \in \mathbf{Z}$ astfel încât $i \neq j$ și $a^i = a^j$. Să presupunem $j > i$. Atunci $a^{j-i} = e$. Ca urmare, mulțimea

$$\{x \geq 1 | a^x = e\}$$

este nevidă. Fie r cel mai mic element al ei. Este clar că $a^r = e$, ceea ce demonstrează (a).

Pentru a demonstra (b) pornim de la observația că relațiile $a^u = a^v$ și $a^r = e$ conduc la $a^{u \pmod r} = a^{v \pmod r}$. Dacă $u \pmod r$ și $v \pmod r$ ar fi diferite, atunci similar procedeului de mai sus s-ar obține un element r' ce satisfacă $1 \leq r' < r$ și $a^{r'} = e$, contrazicând astfel minimalitatea lui r .

Reciproc, dacă $u \pmod r = v \pmod r$, atunci $a^u = a^v$.

(c) urmează cu ușurință de la (a) și (b), iar (d) se bazează pe izomorfismul $h(a^i) = i$, pentru orice $0 \leq i < r$. \square

Teorema 4.4.1 ne spune că orice grup ciclic infinit este izomorf cu grupul aditiv \mathbf{Z} , și orice grup ciclic finit este izomorf cu un grup \mathbf{Z}_m , unde $m \geq 1$. Evident, orice două grupuri ciclice de același ordin sunt izomorfe.

Reamintim că *ordinul unui element* a într-un grup G , notat prin $ord_G(a)$, este definit ca fiind ordinul subgrupului generat de a (a se vedea Secțiunea 1.4.4.2).

Teorema 4.4.2. Fie (G, \cdot', e) un grup și $a \in G$ de ordin finit. Atunci au loc următoarele proprietăți:

- (1) $ord_G(a) = \min\{r \geq 1 | a^r = e\}$;
- (2) dacă G este finit, atunci $ord_G(a) \mid |G|$;
- (3) $(\forall s \in \mathbf{Z})(a^s = e \Leftrightarrow ord_G(a) \mid s)$;
- (4) dacă G este finit, atunci $a^{|G|} = e$;
- (5) $(\forall s, t \in \mathbf{Z})(a^s = a^t \Leftrightarrow s \equiv t \pmod{ord_G(a)})$;
- (6) $(\forall t \in \mathbf{Z})(ord_G(a^t) = ord_G(a)/(t, ord_G(a)))$;
- (7) dacă $ord_G(a) = r_1 r_2$ și $r_1, r_2 > 1$, atunci $ord_G(a^{r_1}) = r_2$.

Demonstrație. (1) urmează direct de la Teorema 4.4.1, iar (2) de la Teorema lui Lagrange.

Pentru (3) se aplică teorema împărțirii cu rest și proprietatea $a^{ord_G(a)} = e$, iar (4) urmează de la (2) și (3). (5) este o altă formulare a Teoremei 4.4.1(2)(b), iar (7) urmează de la (6).

Ca urmare, ne rămâne de demonstrat (6). Fie $r = ord_G(a)$ și $d = ord_G(a^t)$. Relația $(a^t)^d = e$ implică $r \mid dt$. De fapt, d este cel mai mic număr natural strict pozitiv pentru care td este multiplu de r . Însă este trivial de văzut că cel mai mic $d \geq 1$ cu această proprietate este $d = r/(t, r)$. \square

Corolarul 4.4.1. Fie (G, \cdot', e) un grup și $a, b \in G$ de ordin finit. Dacă a și b comută, iar $ord_G(a)$ și $ord_G(b)$ sunt prime între ele, atunci $ord_G(ab) = ord_G(a)ord_G(b)$.

Demonstrație. Fie $s = ord_G(a)$ și $t = ord_G(b)$ cu $(s, t) = 1$. Evident, putem presupune că $s, t > 1$ (dacă, de exemplu, $s = 1$, atunci $a = e$).

Are loc $(ab)^{st} = e$. Presupunem că st nu ar fi ordinul lui ab în G . Fie $ord_G(ab) = d$. Atunci, $d < st$. Conform Teoremei 4.4.2(3), $d \mid st$. Analizăm următoarele cazuri.

Cazul 1: $(d, s) = 1$ sau $(d, t) = 1$. Să presupunem, de exemplu, că $(d, t) = 1$. Atunci, $s = ds_1$ cu $s_1 \geq 1$. Are loc:

$$e = (ab)^{ds_1} = a^{ds_1}b^{ds_1} = a^s b^s = b^s.$$

Cum $ord_G(b) = t$, deducem că $t \mid s$, ceea ce contrazice ipoteza $(s, t) = 1$.

Cazul 2: $d = d_1 d_2$ cu $d_1, d_2 > 1$, $d_1 \mid s$ și $d_2 \mid t$. Deoarece $d < st$, are loc $d_1 < s$ sau $d_2 < t$. Să presupunem că are loc $d_2 < t$.

Fie $s = d_1 s_1$ cu $s_1 \geq 1$. Atunci:

$$e = (ab)^{ds_1} = a^{ds_1}b^{ds_1} = a^{sd_2}b^{sd_2} = b^{sd_2}.$$

Cum $ord_G(b) = t$, deducem că are loc $t \mid sd_2$, iar relația $(s, t) = 1$ conduce la $t \mid d_2$, ce contrazice $d_2 < t$.

Cum ambele cazuri (posibile) conduc la contradicții, presupunerea făcută este falsă, și astfel $\text{ord}_G(ab) = st$. \square

Teorema 4.4.3. Fie (G, \cdot', e) un grup finit și $a \in G$. Atunci, au loc următoarele proprietăți:

- (1) $G = \langle a \rangle$ dacă și numai dacă $\text{ord}_G(a) = |G|$;
- (2) a este generator pentru G dacă și numai dacă $a^{|G|/q} \neq e$, pentru orice factor prim q al lui $|G|$;
- (3) dacă a este generator al grupului G , atunci, pentru orice $t \in \mathbb{Z}$, a^t este generator al grupului G dacă și numai dacă $(t, |G|) = 1$;
- (4) dacă G este ciclic, atunci el are $\phi(|G|)$ generatori.

Demonstrație. (1) urmează de la definiții și Teorema 4.4.2(2).

(2) Să presupunem că a este generator al grupului G . Atunci $|G|$ este cel mai mic număr natural strict pozitiv ce satisfacă $a^{|G|} = e$, ceea ce ne arată că $a^{|G|/q} \neq e$, pentru orice factor prim q al lui $|G|$.

Reciproc, $\text{ord}_G(a)||G|$ (de la Teorema 4.4.2(2)). Dacă presupunem că are loc $\text{ord}_G(a) < |G|$, atunci există $t > 1$ astfel încât $|G| = \text{ord}_G(a)t$. Aceasta ne arată că există un factor prim q al lui $|G|$ astfel încât $\text{ord}_G(a)|(|G|/q)$. Dar atunci, $a^{|G|/q} = e$, ceea ce constituie o contradicție.

(3) a^t este generator pentru G dacă și numai dacă $\text{ord}_G(a^t) = |G|$. Însă această ultimă relație are loc dacă și numai dacă $(t, |G|) = 1$ (conform Teoremei 4.4.2(6)).

(4) Presupunem că G este ciclic și fie a un generator. Atunci,

$$G = \{a^0 = e, a^1, \dots, a^{|G|-1}\},$$

unde $a^i \neq a^j$, pentru orice $i \neq j$. Un element a^t este generator, unde $t \geq 1$, dacă și numai dacă $(t, |G|) = 1$. Cum există $\phi(|G|)$ numere strict pozitive mai mici decât $|G|$ și prime cu $|G|$, rezultă că t poate fi ales în $\phi(|G|)$ moduri. Deci G are $\phi(|G|)$ generatori. \square

4.5. Grupul \mathbb{Z}_m^*

Fie $m \geq 1$. Am văzut în Secțiunea 2.3 că un element $a \in \mathbb{Z}_m$ are un invers multiplicativ dacă și numai dacă $(a, m) = 1$. Atunci $\mathbb{Z}_m^* = \{a \in \mathbb{Z}_m | (a, m) = 1\}$ în raport cu înmulțirea claselor de resturi devine grup comutativ (numit grupul unităților inelului \mathbb{Z}_m – a se vedea și Secțiunea 5.1). Deoarece în \mathbb{Z}_m există $\phi(m)$ numere prime cu m (Secțiunea 2.4), ordinul grupului \mathbb{Z}_m^* este $\phi(m)$. Teorema 4.4.2(4) furnizează atunci

o nouă demonstrație a Teoremei lui Euler, conform căreia $a^{\phi(m)} \equiv 1 \pmod{m}$, pentru orice a ce este prim cu m .

Vom nota prin $\text{ord}_m(a)$ ordinul elementului $a \in \mathbb{Z}_m^*$ și ne vom referi la el ca fiind ordinul lui a modulo m ⁴. În cazul $m = 1$, $\mathbb{Z}_1^* = \{0\}$ și $\text{ord}_1(0) = 1$.

Particularizând rezultatele din secțiunea anterioară obținem.

Propoziția 4.5.1. Fie $m \geq 1$ și $a \in \mathbb{Z}_m^*$. Atunci au loc următoarele proprietăți:

- (1) $\text{ord}_m(a) = \min\{k \geq 1 | a^k \equiv 1 \pmod{m}\}$;
- (2) dacă $a^k \equiv 1 \pmod{m}$, atunci $\text{ord}_m(a)|k$. În particular, $\text{ord}_m(a)|\phi(m)$;
- (3) $\text{ord}_m(a) = \phi(m)$ dacă și numai dacă $a^{\phi(m)/q} \not\equiv 1 \pmod{m}$, pentru orice factor prim q al lui $\phi(m)$;
- (4) $a^k \equiv a^l \pmod{m}$ dacă și numai dacă $k \equiv l \pmod{\text{ord}_m(a)}$;
- (5) elementele $a^0 \pmod{m}, a^1 \pmod{m}, \dots, a^{\text{ord}_m(a)-1} \pmod{m}$ sunt distincte două câte două;
- (6) $\text{ord}_m(a^k) = \text{ord}_m(a)/(k, \text{ord}_m(a))$, pentru orice $k \geq 1$;
- (7) dacă $\text{ord}_m(a) = d_1d_2$, atunci $\text{ord}_m(a^{d_1}) = d_2$.

Corolarul 4.5.1. Fie $m \geq 1$ și $a, b \in \mathbb{Z}_m^*$. Dacă $\text{ord}_m(a)$ și $\text{ord}_m(b)$ sunt prime între ele, atunci $\text{ord}_m(ab \pmod{m}) = \text{ord}_m(a)\text{ord}_m(b)$.

Demonstrație. Direct de la Corolarul 4.4.1. \square

\mathbb{Z}_m cu adunarea formează grup ciclic, pentru orice $m \geq 1$. Nu același lucru se poate afirma despre \mathbb{Z}_m^* cu înmulțirea. Atunci când \mathbb{Z}_m^* este grup ciclic (în raport cu înmulțirea), generatorii lui mai sunt numiți rădăcini primitive modulo m ⁵. Este clar că $a \in \mathbb{Z}_m^*$ este rădăcină primitivă modulo m dacă și numai dacă $\text{ord}_m(a) = \phi(m)$. 0 este (singura) rădăcină primitivă modulo 1.

Propoziția 4.5.2. Fie $m \geq 1$ și $a \in \mathbb{Z}_m^*$. Atunci, au loc următoarele proprietăți:

- (1) a este rădăcină primitivă modulo m dacă și numai dacă $\text{ord}_m(a) = \phi(m)$;
- (2) a este rădăcină primitivă modulo m dacă și numai dacă are loc

$$(\forall q)(q \text{ factor prim } q \text{ al lui } \phi(m)) \Rightarrow a^{\phi(m)/q} \not\equiv 1 \pmod{m};$$

- (3) dacă a este rădăcină primitivă modulo m atunci, pentru orice $k \geq 1$, a^k este rădăcină primitivă modulo m dacă și numai dacă $(k, \phi(m)) = 1$;

⁴Notația $\text{ord}_m(a)$ a fost introdusă de Gauss în *Disquisitiones Arithmeticae* publicată în 1801 [66].

⁵Termenul de rădăcină primitivă a fost introdus de Euler în 1773. Demonstrația lui, conform căreia există rădăcini primitive modulo p , pentru orice număr prim p , s-a dovedit și era eronată. Gauss a furnizat mai multe demonstrații corecte acestui rezultat.

(4) dacă există rădăcini primitive modulo m , atunci numărul acestora este $\phi(\phi(m))$.

Demonstrație. De la Teorema 4.4.3. \square

Întrebarea fundamentală care se ridică acum este următoarea: pentru ce valori ale lui m există rădăcini primitive modulo m ? Răspunsul complet la această întrebare a fost dat de F. Gauss [66]. Vom prezenta mai jos acest rezultat fundamental, parcurgând mai multe etape în funcție de diverse valori ale lui m . Fiecare etapă, luată individual, constituie un rezultat important, de sine stătător, ce poate fi utilizat în diverse situații. În final, toate aceste rezultate intermedii vor fi corelate pentru a furniza rezultatul central.

Dacă $m \in \{1, 2, 4\}$, atunci este ușor de verificat că există rădăcini primitive modulo m .

Pentru cazul $m = p$, unde $p \geq 3$ este număr prim, avem următorul rezultat.

Lema 4.5.1. Există rădăcini primitive modulo p , pentru orice $p \geq 3$ număr prim.

Demonstrație. Fie $p \geq 3$ un număr prim. Vom arăta că există rădăcini primitive modulo p parcurgând următoarele două etape.

Etapa 1: Arătăm că dacă $e > 0$ și q este un număr prim cu proprietatea $q^e | p - 1$, atunci există $\alpha \in \mathbf{Z}_p^*$ de ordin q^e (modulo p).

În adevăr, ecuația

$$x^{(p-1)/q} \equiv 1 \pmod{p}$$

are cel mult $(p-1)/q$ soluții în \mathbf{Z}_p^* (Teorema 2.5.2). Deoarece $p \geq 3$ obținem

$$\frac{p-1}{q} \leq \frac{p-1}{2} \leq p-2,$$

care ne arată că există $a \in \mathbf{Z}_p^*$ ce nu este soluție a ecuației de mai sus. Deci există a astfel încât $a^{(p-1)/q} \not\equiv 1 \pmod{p}$. Vom arăta că $\alpha = a^{(p-1)/q^e} \pmod{p}$ este de ordin q^e (modulo p). Este clar că $\alpha^{q^e} \equiv 1 \pmod{p}$. Dacă ordinul lui α ar fi $d < q^e$, atunci $d | q^e$. Combinând toate acestea cu faptul că q este prim, obținem $d | q^{e-1}$, și deci

$$\begin{aligned} a^{(p-1)/q} &\equiv \alpha^{q^{e-1}} \pmod{p} \\ &\equiv 1 \pmod{p}, \end{aligned}$$

ceea ce constituie o contradicție.

Etapa 2: Arătăm că există rădăcini primitive modulo p .

Fie $p-1 = p_1^{e_1} \cdots p_k^{e_k}$ descompunerea în factori primi a lui $p-1$. Conform etapei 1, există α_i de ordin $p_i^{e_i}$, pentru orice $1 \leq i \leq k$. Aplicând repetat Corolarul 4.5.1 deducem că $\alpha = \alpha_1 \cdots \alpha_k \pmod{p}$ este de ordin $\phi(p) = p_1^{e_1} \cdots p_k^{e_k}$, ceea ce ne spune că α este rădăcină primitive modulo p . \square

Următoarea lemă face trecerea de la rădăcini primitive modulo p la rădăcini primitive modulo p^k , pentru orice $k > 1$.

Lema 4.5.2. Dacă $p \geq 3$ este un număr prim, atunci există rădăcini primitive modulo p ce sunt rădăcini primitive modulo p^k , pentru orice $k > 1$.

Demonstrație. Fie $p \geq 3$ un număr prim. Vom face demonstrația în două etape.

Etapa 1: Arătăm că există o rădăcină primitive modulo p , β , care satisface

$$(*) \quad (\forall k > 1)(\beta^{\phi(p^{k-1})} \not\equiv 1 \pmod{p^k}).$$

Fie α o rădăcină primitive modulo p . Atunci $(\alpha + p) \pmod{p}$ este, de asemenea, rădăcină primitive modulo p . Are loc:

$$\begin{aligned} ((\alpha + p) \pmod{p})^{p-1} &\equiv \alpha^{p-1} + p(p-1)\alpha^{p-2} \pmod{p^2} \\ &\equiv \alpha^{p-1} - p\alpha^{p-2} \pmod{p^2} \end{aligned}$$

Relația $((\alpha + p) \pmod{p})^{p-1} \equiv \alpha^{p-1} - p\alpha^{p-2} \pmod{p^2}$ ne spune că α sau $(\alpha + p) \pmod{p}$ trebuie să satisfacă relația $x^{p-1} \not\equiv 1 \pmod{p^2}$, deoarece altfel am obține $p|\alpha$, ceea ce ar fi o contradicție. Fie β rădăcină primitive care satisface această relație.

Vom arăta că β satisface $(*)$ prin inducție matematică după $k \geq 2$.

Pentru $k = 2$, $(*)$ a fost deja stabilită. Presupunem $(*)$ adevărată pentru $k \geq 2$. În baza Teoremei lui Euler, $\beta^{\phi(p^{k-1})} \equiv 1 \pmod{p^{k-1}}$. Ca urmare, există t astfel încât $\beta^{\phi(p^{k-1})} = 1 + tp^{k-1}$. Conform ipotezei inductive, p nu divide t . Atunci:

$$\begin{aligned} \beta^{\phi(p^k)} &= (1 + tp^{k-1})^p \\ &\equiv 1 + tp^k + (1/2)p(p-1)t^2p^{2k-2} \pmod{p^{k+1}} \\ &\equiv 1 + tp^k \pmod{p^{k+1}} \\ &\not\equiv 1 \pmod{p^{k+1}}, \end{aligned}$$

ce încheie demonstrația (ultima relație urmează de la faptul că dacă ar avea loc $1 + tp^k \equiv 1 \pmod{p^{k+1}}$, atunci p ar divide t , ceea ce ar constitui o contradicție).

Etapa 2: Arătăm că orice rădăcină primitive β ce satisface cerințele etapei 1 este rădăcină primitive modulo p^k , pentru orice $k > 1$.

Fie $k > 1$, β o rădăcină primitive modulo p ce satisface cerințele etapei 1, iar t ordinul ei modulo p^k . Atunci, $\beta^t \equiv 1 \pmod{p^k}$ și $t|\phi(p^k) = p^{k-1}(p-1)$. Relația $\beta^t \equiv 1 \pmod{p^k}$ conduce la $\beta^t \equiv 1 \pmod{p}$, de la care urmează $(p-1)|t$ (deoarece $\text{ord}_p(\beta) = p-1$). Combinând cu $t|p^{k-1}(p-1)$ obținem că există $0 \leq i \leq k-1$ astfel încât $t = p^i(p-1)$. Dacă $i = k-1$, atunci β este rădăcină primitive modulo p^k . Să arătăm că nu poate avea loc $i < k-1$. Dacă am presupune că $i < k-1$, atunci am obține

$$\beta^{\phi(p^{k-1})} = \beta^{p^{k-2}(p-1)} = (\beta^{p^i(p-1)})^{p^{k-2-i}} \equiv 1 \pmod{p^k},$$

ce ar contrazice relația $(*)$.

Ca urmare, β este rădăcină primitive modulo p^k . \square

Rădăcinile primitive modulo p^k conduc la rădăcini primitive modulo $2p^k$, pentru orice număr prim $p \geq 3$ și $k \geq 1$.

Lema 4.5.3. Fie $p \geq 3$ un număr prim și $k \geq 1$. Dacă α este rădăcină primitivă impară modulo p^k , atunci α este rădăcină primitivă modulo $2p^k$, iar dacă α este rădăcină primitivă pară modulo p^k , atunci $(\alpha + p^k) \text{ mod } p^k$ este rădăcină primitivă modulo $2p^k$.

Demonstrație. Fie α o rădăcină primitivă modulo p^k . Atunci $\phi(p^k)$ este cel mai mic exponent $x > 0$ ce satisface $\alpha^x \equiv 1 \text{ mod } p^k$.

Cum $\phi(p^k) = \phi(2p^k)$, obținem $\alpha^{\phi(2p^k)} \equiv 1 \text{ mod } p^k$. Vom analiza acum următoarele două cazuri.

Cazul 1: α este impar. Atunci $\alpha^{\phi(2p^k)} \equiv 1 \text{ mod } 2$, și deci $\alpha^{\phi(2p^k)} \equiv 1 \text{ mod } 2p^k$. Mai mult, $\phi(2p^k)$ este cel mai mic exponent strict pozitiv al lui α ce satisface această congruență, deoarece altfel s-ar contrazice faptul că α este rădăcină primitivă modulo p^k . Deci α este rădăcină primitivă modulo $2p^k$.

Cazul 2: α este par. Atunci $(\alpha + p^k) \text{ mod } p^k$ este impar. Deci

$$((\alpha + p^k) \text{ mod } p^k)^{\phi(2p^k)} \equiv 1 \text{ mod } 2.$$

Cum $\alpha + p^k \equiv \alpha \text{ mod } p^k$, deducem că are loc $((\alpha + p^k) \text{ mod } p^k)^{\phi(2p^k)} \equiv 1 \text{ mod } p^k$. Dar atunci $((\alpha + p^k) \text{ mod } p^k)^{\phi(2p^k)} \equiv 1 \text{ mod } 2p^k$, și nici o putere strict pozitivă mai mică decât $\phi(2p^k)$ nu poate satisface această congruență $((\alpha + p^k) \text{ mod } p^k)$ este, de asemenea, rădăcină primitivă modulo p^k). Ca urmare, $\alpha + p^k$ este rădăcină primitivă modulo $2p^k$. \square

Putem acum formula un rezultat fundamental asupra existenței rădăcinilor primitive, rezultat datorat lui Gauss [66].

Teorema 4.5.1. (Teorema lui Gauss)

Există rădăcini primitive modulo m dacă și numai dacă m este de forma $m = 1, 2, 4, p^k, 2p^k$, unde $p \geq 3$ este număr prim iar $k \geq 1$.

Demonstrație. Lemele 4.5.1, 4.5.2 și 4.5.3, cu observația că există rădăcini primitive modulo 1, 2 și 4, conduc la faptul că există rădăcini primitive modulo m , pentru orice m de forma $m = 1, 2, 4, p^k, 2p^k$, unde $p \geq 3$ este număr prim iar $k \geq 1$.

Reciproc, să presupunem că există rădăcini primitive modulo m , dar m nu este de nici una din formele din teoremă. Vom arăta în acest caz că are loc

$$(*) \quad a^{\phi(m)/2} \equiv 1 \text{ mod } m,$$

pentru orice $a \in \mathbf{Z}_m^*$, ceea ce va conduce la faptul că nici un element din \mathbf{Z}_m^* nu poate fi rădăcină primitivă modulo m .

Considerăm următoarele cazuri posibile asupra lui m .

Cazul 1: $m = 2^k$, unde $k \geq 3$. Atunci $\phi(m)/2 = 2^{k-2}$. Arătăm că are loc $(*)$ prin inducție matematică după $k \geq 3$. Pentru $k = 3$, $a^{\phi(m)/2} = a^2 \equiv 1 \text{ mod } 2^3$ deoarece

$a \in \{1, 3, 5, 7\} = \mathbf{Z}_m^*$. Presupunem $(*)$ adevărată pentru $k \geq 3$. Atunci, există t astfel încât $a^{2^{k-2}} = 1 + t2^k$. Obținem

$$a^{2^{k-1}} = (1 + t2^k)^2 = 1 + t2^{k+1} + t^22^{2k} \equiv 1 \text{ mod } 2^{k+1},$$

care încheie demonstrația în acest caz.

Cazul 2: $m = 2^k p_1^{e_1} \cdots p_n^{e_n}$, unde $k > 1$, $n > 0$, p_1, \dots, p_n sunt numere prime impare și distincte două câte două, și $e_1, \dots, e_n > 0$. Atunci

$$\phi(m)/2 = 2^{k-2}\phi(p_1^{e_1}) \cdots \phi(p_n^{e_n}).$$

Așa cum observăm, $\phi(m)/2$ este divizibil atât prin $\phi(2^k)$ cât și prin $\phi(p_i^{e_i})$, pentru orice i . Ca urmare, $a^{\phi(m)/2} \equiv 1 \text{ mod } 2^k$ și $a^{\phi(m)/2} \equiv 1 \text{ mod } p_i^{e_i}$, pentru orice i , de la care urmează $(*)$. \square

Direct de la Teorema 4.5.1 obținem:

Corolarul 4.5.2. Grupul multiplicativ \mathbf{Z}_m^* este ciclic dacă și numai dacă m este de forma $m = 1, 2, 4, p^k, 2p^k$, unde $p \geq 3$ este număr prim iar $k \geq 1$.

Următoarele rezultate furnizează informații suplimentare asupra rezolvării unor ecuații congruențiale în cazul în care \mathbf{Z}_m^* este ciclic.

Propoziția 4.5.3. Fie $m \geq 1$ astfel încât \mathbf{Z}_m^* este ciclic. Atunci ecuația

$$x^n \equiv 1 \text{ mod } m$$

are $(n, \phi(m))$ soluții în \mathbf{Z}_m^* .

Demonstrație. Fie α o rădăcină primitivă a grupului \mathbf{Z}_m^* . A determina $x \in \mathbf{Z}_m^*$ care să verifice ecuația din enunțul propoziției este echivalent cu a determina un număr $i \in \mathbf{Z}_{\phi(m)}$ astfel încât

$$\alpha^{in} \equiv 1 \text{ mod } m$$

sau, echivalent, a determina $i \in \mathbf{Z}_{\phi(m)}$ astfel încât

$$in \equiv 0 \text{ mod } \phi(m).$$

Teorema 2.5.1 ne arată că această ecuație are exact $(n, \phi(m))$ soluții în $\mathbf{Z}_{\phi(m)}$. Fiecare soluție i a acestei ecuații conduce la o soluție $\alpha^i \text{ mod } m$ a ecuației din enunțul propoziției. În plus, pentru $i \neq j$, soluțiile $\alpha^i \text{ mod } m$ și $\alpha^j \text{ mod } m$ sunt distincte. Deci ecuația dată are exact $(n, \phi(m))$ soluții în \mathbf{Z}_m^* . \square

Conform Propoziției 4.5.3, ecuația

$$x^n \equiv 1 \text{ mod } p^e$$

are exact $(n, p^e - p^{e-1})$ soluții în \mathbf{Z}_p^* , pentru orice număr prim p și $e \geq 1$.

Propoziția 4.5.4. Fie p un număr prim impar. Atunci ecuația

$$x^n \equiv -1 \pmod{p}$$

are soluții dacă și numai dacă $(n, p-1)|(p-1)/2$ și, în acest caz, are exact $(n, p-1)$ soluții în \mathbf{Z}_p^* .

Demonstrație. Fie α o rădăcină primă a modulu p . Cum $\alpha^{(p-1)/2} \equiv -1 \pmod{p}$, ecuația din enunțul propoziției poate fi rescrisă în mod echivalent la

$$x^n \equiv \alpha^{(p-1)/2} \pmod{p}.$$

Cum orice element $a \in \mathbf{Z}_p^*$ este de forma $a = \alpha^i \pmod{p}$, unde $i \in \mathbf{Z}_{p-1}$, deducem că a determină soluțiile ecuației de mai sus se reduce la a determina i astfel încât

$$\alpha^{in} \equiv \alpha^{(p-1)/2} \pmod{p}.$$

Aceasta este echivalent cu a determina soluțiile ecuației

$$in \equiv (p-1)/2 \pmod{(p-1)},$$

în necunoscuta i . Această ecuație are soluții dacă și numai dacă $(n, p-1)|(p-1)/2$.

În plus, dacă are soluții, atunci ea are exact $(n, p-1)$ soluții în \mathbf{Z}_{p-1} (Teorema 2.5.1). Demonstrația propoziției continuă ca în cazul propoziției anterioare. \square

Încheiem secțiunea prin câteva remarci asupra ecuației din Propoziția 4.5.4. Dacă scriem $n = 2^s t$ și $p-1 = 2^{s'} t'$, unde $s, s', t, t' \geq 1$ iar t și t' sunt impare, atunci obținem:

- dacă $s \geq s'$, atunci $(2^s t, 2^{s'} t') = 2^{s'}(t, t')$ care nu divide $(p-1)/2 = 2^{s'-1}t'$. Deci, în acest caz, ecuația nu are soluții;
- dacă $s < s'$, atunci $(2^s t, 2^{s'} t') = 2^s(t, t')|(p-1)/2 = 2^{s'-1}t'$. Deci, în acest caz, ecuația are exact $2^s(t, t')$ soluții în \mathbf{Z}_p^* .

4.6. Problema logaritmului discret

Fie G un grup ciclic finit și a un generator al său. Atunci

$$G = \{a^0 = e, a^1, \dots, a^{|G|-1}\},$$

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 numește *indexul* sau *logaritmul discret al lui b în grupul G relativ la a*. Dacă $G = \mathbf{Z}_m^*$, atunci k se mai numește și *indexul* sau *logaritmul discret al lui b modulo m relativ la a*.

Determinarea algoritmică a indexului k poartă denumirea de *problema logaritmului discret*. În informatică, această problemă se enunță astfel:

Problema logaritmului discret

Instanță: un grup ciclic finit G , un generator a al său și $b \in G$;

Întrebare: determinați $k < |G|$ astfel încât $b = a^k$.

Soluția cea mai la îndemână de rezolvare a acestei probleme este de a enumera elementele grupului G în forma

$$a^0, a^1, a^2, \dots, a^{|G|-1}$$

până se găsește a^k ce este b . Această soluție necesită, în cazul cel mai nefavorabil, parcurserea a $|G| - 1$ pași în care, exceptând primii doi pași, restul pașilor necesită o înmulțire cu a și o comparație a rezultatului cu b .

Dacă $|G|$ este foarte mare, de exemplu $|G| \geq 10^{100}$, atunci această soluție nu poate fi pusă în practică.

În [188], Daniel Shanks a propus un algoritm de complexitate $\mathcal{O}(\sqrt{|G|})$. Ideea acestuia se bazează pe următoarele. Fie k cu $0 \leq k < |G|$ și $n = \lceil \sqrt{|G|} \rceil$. Atunci k poate fi scris în forma $k = qn + r$, unde $0 \leq q, r < n$ (folosind teorema împărțirii cu rest și observând că $q < n$). Ca urmare, determinarea numărului k cu proprietatea $b = a^k$ se poate reduce la determinarea a două numere q și r ce satisfac $0 \leq q, r < n$ și $b = a^{qn+r}$. Dacă rescriem egalitatea $b = a^{qn+r}$ în forma $a^r = b(a^{-n})^q$, atunci problema noastră se reduce la a găsi două numere q și r ce satisfac $0 \leq q, r < n$ și $a^r = b(a^{-n})^q$. Determinarea acestor numere se poate face astfel:

- se calculează valorile a^0, a^1, \dots, a^{n-1} și se stochează într-o listă L ;
- se calculează succesiv $b, ba^{-n}, b(a^{-n})^2, \dots, b(a^{-n})^{n-1}$ și, pentru fiecare valoare calculată se verifică dacă aceasta se găsește în lista L . Apartenența unei astfel de valori la lista L determină automat r și q , întrerupând totodată și procesul de generare.

Algoritmul sugerat este următorul.

Algoritmul Baby-step Giant-step

input: un grup ciclic finit G , a un generator al său și $b \in G$;

output: un număr k astfel încât $0 \leq k < |G|$ și $b = a^k$;

begin

1. $n := \lceil \sqrt{|G|} \rceil$;
2. calculează o listă $L := \{(r, a^r) | 0 \leq r < n\}$ și sortează-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** $c := ca^{-n}$;
9. **end.**

Algoritmul de mai sus necesită $\mathcal{O}(n)$ spațiu pentru memorarea listei L . Calculul acesteia necesită $\mathcal{O}(n)$ înmulțiri. Complexitatea sortării ei este $\mathcal{O}(n \log n)$ (lista poate fi sortată o dată cu generarea fiecărei noi perechi ce urmează a fi adăugată). Pre-calculând lista L , pasul 3 necesită o înmulțire și o determinare de invers, iar pasul 5 necesită $\mathcal{O}(n)$ înmulțiri și $\mathcal{O}(n)$ căutări în tabelă.

Denumirea algoritmului providează de la faptul că determinarea elementelor listei se face prin înmulțiri cu a (*baby-steps*), iar reactualizarea lui c se face prin înmulțiri cu a^{-n} (*giant-steps*).

Încheiem secțiunea prin remarcă că nu se cunoaște nici o soluție polinomială (în raport cu $\log |G|$) de rezolvare a acestei probleme. Problema este considerată intractabilă pentru valori mari ale lui $|G|$ (adică algoritmii existenți nu pot fi utilizati în practică), ceea ce o face destul de potrivită pentru dezvoltarea de tehnici criptografice bazate pe ea. De exemplu, semnatura digitală ElGamal își bazează securitatea pe această problemă (cine dorește să atace cu succes semnatura ElGamal este pus în situația rezolvării unei instanțe a acestei probleme.).

4.7. Aplicații: criptografie cu chei publice

În această secțiune vom puncta câteva din aplicațiile majore ale teoriei grupurilor, și în special a grupurilor ciclice, în criptografie. Prezentarea noastră va urma [213]. Atragem atenția de la bun început că vom adopta conceptul de sistem de criptare în varianta Shannon [190], unde criptarea și decriptarea este definită funcțional. Criptografia modernă utilizează această abordare și o extinde prin criptare probabilistă [213]. Pentru scopul materialului nostru este suficient să urmăm abordarea Shannon.

4.7.1. Introducere în criptografie

Criptosistem. Metoda fundamentală de securizare a comunicației între două părți A și B față de o terță parte C constă în stabilirea, de comun acord de către A și B , a unui *sistem de criptare* (*cifrare*) a mesajelor. Un astfel de sistem trebuie să ofere posibilitatea criptării mesajelor ce urmează a fi transmise și, totodată, a decriptării “corecte” a mesajelor (criptate) recepționate. De exemplu, A și B pot conveni ca fiecare literă a mesajului să fie înlocuită ciclic cu litera aflată la distanța k de litera în cauză (considerând alfabetul ordonat în mod ușual). 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 condiția de a putea fi stabilită de A și B în deplină securitate).

Mesajele originale, ce urmează a fi criptate și transmise, se construiesc pe baza unei mulțimi finite de elemente “atomice”, mulțime numită *alfabet*. Elementele acestei mulțimi pot fi literele și cifrele caracteristice unei anumite limbi (engleză, japoneză, chineză etc.), secvențe formate din anumite litere și/sau cifre (gândite ca entități indivizibile), logograme sau diverse simboluri speciale reprezentând cuvinte sau fraze. Mesajele formate pe baza unui astfel de alfabet vor fi numite *texte inițiale* sau *texte sursă* sau *texte clare* sau *plaintexte*⁶. Ele sunt secvențe (înșiruiri) 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*⁷.

Definiția 4.7.1. 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 mulțime finită și nevidă numită *alfabetul plaintextelor*. Elementele ei sunt numite *simboluri/carcătere plaintext*;
- (ii) \mathcal{C} este o mulțime finită și nevidă numită *alfabetul criptotextelor*. Elementele ei sunt numite *simboluri/carcătere criptotext*;
- (iii) \mathcal{K} este o mulțime finită și nevidă numită *spațiul cheilor de criptare*. Elementele ei sunt numite *chei* (de criptare);
- (iv) \mathcal{E} și \mathcal{D} sunt două mulțimi de funcții,

$$\mathcal{E} = \{e_K : \mathcal{P} \rightarrow \mathcal{C} | K \in \mathcal{K}\}$$

și

$$\mathcal{D} = \{d_K : \mathcal{C} \rightarrow \mathcal{P} | K \in \mathcal{K}\},$$

astfel încât pentru orice $K \in \mathcal{K}$ și $x \in \mathcal{P}$ 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 mulțimii tuturor textelor construite pe baza alfabetului \mathcal{C} . Pentru fiecare cheie $K \in \mathcal{K}$, funcția e_K (d_K) se numește *funcția/regula de criptare* (*decriptare*). Observăm că funcția d_K este invers la stânga al funcției e_K , iar e_K este invers la dreapta al funcției d_K . Ca urmare, e_K este funcție injectivă, iar d_K , surjectivă. Injectivitatea regulilor de criptare conduce imediat la $|\mathcal{P}| \leq |\mathcal{C}|$.

⁶Terminologia de “plaintext” este de proveniență englezescă și preferăm să o utilizăm datorită ușurinței în exprimare.

⁷Terminologia de “criptotext” provine de la cuvântul englezesc “cryptotext”.

Moduri de operare. Criptarea unui plaintext $x_1 \dots x_n$, 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 și se criptează fiecare caracter x_i prin e_K . Adică criptotextul va fi

$$e_K(x_1) \dots e_K(x_n);$$

- *modul de operare cu cheie variabilă*. Pentru fiecare i se determină o cheie K_i și se criptează x_i prin e_{K_i} . Adică criptotextul va fi

$$e_{K_1}(x_1) \dots e_{K_n}(x_n).$$

Determinarea cheilor K_i se face de obicei printr-un așa numit *generator de chei*, pornindu-se de la o cheie inițială, un anumit șir de inițializare și, eventual, plaintextele x_j și criptotextele asociate lor, pentru orice $j < i$.

Criptosistemele cărora li se asociază un generator de chei și, deci, care se utilizează în modul de operare cu cheie variabilă, se mai numesc și *criptosisteme cu chei șir*.

Criptosistemele dezvoltate până în prezent pot fi împărțite în două clase fundamentale:

- *criptosisteme simetrice*, caracterizate prin aceea că regula decriptare se poate determina "ușor" cunoșcând regula de criptare, și invers;
- *criptosisteme asimetrice sau cu chei publice*, caracterizate prin aceea că cheia este împărțită în două subchei: una publică și una privată (secretă). Subcheia publică este utilizată de oricine dorește să transmită mesaje criptate deținătorului cheii. Subcheia secretă este cunoscută numai de deținătorul cheii, fiind utilizată pentru decriptarea mesajelor primite. Determinarea subcheii concrete prin cunoașterea cheii publice trebuie să fie o problemă intractabilă.

Adesea, în cadrul criptosistemelor simetrice, alfabetul plaintextelor este o mulțime de secvențe de lungime $m \geq 1$ formate din elemente ale unei alte mulțimi P . Adică $\mathcal{P} \subseteq P^m$. Dacă utilizatorul își formează plaintextele din elemente ale mulțimii P , atunci criptarea cu un altfel de criptosistem se realizează prin împărțirea 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 și *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.

Interacțiunea dintre utilizatorii legali și cei ilegali ai unui criptosistem poate fi reprezentată schematic ca în Figura 4.1, unde:

- A și B sunt *utilizatori legali* ce au convenit a comunica utilizând un criptosistem a priori stabilit între ei, și cu o anumită cheie de criptare K ;
- C este *utilizator ilegal* ce încearcă a captura criptotexte ce se transmit între A și 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 și B .

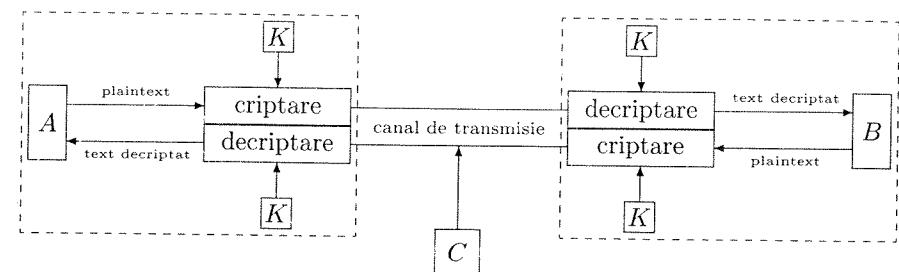


Figura 4.1: Interacțiunea dintre utilizatorii legali și ilegali ai unui criptosistem

Construcția criptosistemelor trebuie realizată având în vedere următoarele două deziderate fundamentale:

- (1) pentru orice cheie K , funcțiile e_K și d_K să poată fi calculate "eficient"⁸ (cunoșcând cheia K);
- (2) determinarea cheii de criptare pe baza unor criptotexte cunoscute (eventual și prin cunoașterea plaintextelor corespunzătoare) să fie "imposibilă" (în sens opus sensului de la (1) de calcul eficient).

Criptanaliza este știința ce se ocupă cu studiul tehniciilor (metodelor) prin care se poate reface un plaintext pornind de la unul sau mai multe criptotexte, fără a cunoaște, a priori, cheia de criptare. Fiecare tehnică în parte este numită *tehnica de criptanaliză* sau *tehnica 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 și practicarea tehniciilor de atac se numesc *criptanalisti*. Cuplul celor două științe, criptografie și criptanaliză, este întâlnit adesea sub terminologia de *criptologie*.

O presupunere fundamentală în criptanaliză, enunțată pentru prima dată de către Dutchman A. Kerckhoffs în secolul XIX (a se vedea [103]), constă în aceea că securitatea oferită de un criptosistem trebuie să se bazeze numai și numai pe cheia de criptare (nu și pe anumite detalii constructive ale acestora deoarece, mai devreme sau mai târziu, acestea pot deveni cunoscute criptanalistilor). Altfel spus, Kerckhoffs

⁸Prin *calculul eficient (ușor)* al unei funcții vom înțelege existența unui algoritm determinist de complexitate timp polinomială de calcul al respectivei funcții.

pornește de la ideea că, în procesul de criptanaliză, criptanalista au toate detaliile constructive ale criptosistemului, exceptând cheia utilizată la momentul respectiv. Chiar dacă în situații 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 și la atacuri pentru care această presupunere nu este satisfăcută.

Există, în general, 4 tipuri de atacuri:

1. *atac de criptotext*. Criptanalistul are la dispoziție un număr de criptotexte (obținute cu aceeași cheie), și 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 cunoaște cheia de criptare). Schematic, aceasta poate fi exprimată ca mai jos:

Date inițiale: $y_1 = e_K(x_1), \dots, y_i = e_K(x_i)$;

Cerințe: x_1, \dots, x_i sau K sau un algoritm de determinare a lui x_{i+1} ;

2. *atac de plaintext cunoscut*. Criptanalistul are la dispoziție un număr de criptotexte (obținute cu aceeași cheie), dar și 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 cunoaște cheia de criptare). Schematic, aceasta poate fi exprimată ca mai jos:

Date inițiale: $(x_1, y_1 = e_K(x_1)), \dots, (x_i, y_i = e_K(x_i))$;

Cerințe: K sau un algoritm de determinare a lui x_{i+1} ;

3. *atac de plaintext aleator*. 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 cunoaște cheia de criptare). Schematic, aceasta poate fi exprimată ca mai jos:

Date inițiale: $(x_1, y_1 = e_K(x_1)), \dots, (x_i, y_i = e_K(x_i))$, unde x_1, \dots, x_i sunt alese de criptanalist;

Cerințe: 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 aleator*. Aceasta este un caz particular al metodei de atac de plaintext aleator. Criptanalistul are posibilitatea de a cunoaște criptotextele asociate plaintextelor alese de el și, mai mult, are posibilitatea de a determina noi perechi (plaintext,criptotext) cu plaintextul aleș în funcție de concluziile obținute prin analiza perechilor (plaintext,criptotext) anterioare.

Există și alte metode de atac, mai puțin frecvente sau eficiente. Unul dintre acestea este *atacul de criptotext aleator* în care se presupune că criptanalistul are posibilitatea de a determina plaintextele corespunzătoare unor criptotexte alese de el, iar un altul

constă în enumerarea și 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 EKS⁹.

Corespondența literă–cifră. Mareea majoritate a criptosistemelor ce sunt utilizate astăzi în practică sunt construite utilizând aparatul algebric, uzând de o anumită corespondență bijectivă între setul de caractere necesar construirii plaintextelor și criptotextelor și o anumită submulțime de numere naturale (o astfel de corespondență ar putea fi, de exemplu, codul EBCDIC sau ASCII cu numerele private în sistemul zecimal). O astfel de “codificare” a setului de caractere utilizat nu trebuie gândită ca o metodă de creștere a securității criptosistemelor; ea nu este altceva decât un pas intermediu în aplicarea unor metode criptografice bazate în mod esențial pe teoria numerelor.

Fără a aduce nici un prejudiciu teoriei generale, se utilizează un set redus de caractere ca cel din Figura 4.2, împreună cu *corespondența 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 4.2: Corespondența literă–cifră

Astfel, se înlocuiesc ‘ă’ prin ‘a’, ‘â’ prin ‘a’, ‘î’ prin ‘i’, ‘ş’ prin ‘s’ și ‘ş’ prin ‘t’. De asemenea, nu se face distincție între literele mari și mici, spațiile sunt inserate între caractere doar pentru claritate, nu se folosesc semne de punctuație, caractere matematice sau de orice alt tip. Aceasta se bazează pe faptul că orice limbaj este dependent de context în mare măsură, sensul unei propoziții fiind clar chiar prin convențiile făcute mai sus.

Dacă $x = x_1 \dots x_n$ este un text de lungime $n \geq 1$ peste alfabetul considerat mai sus, atunci prin *deplasarea/shiftarea* lui x cu k poziții, unde $0 \leq k \leq 25$, înțelegem textul obținut din x prin înlocuirea fiecărei litere x_i cu litera de pe poziția $(k_i + k) \bmod 26$, unde k_i este poziția literei x_i , pentru orice i .

Criptosisteme cu chei publice. O caracteristică esențială 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 și cea de decriptare, fiecare din aceste reguli fiind algoritmic calculabilă în timp “eficient” (un astfel de criptosistem va fi prezentat în Secțiunea 5.6). 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).

⁹De la *Exhaustive Key Search*.

Opus ideii de cheie secretă se află conceptul de *cheie asimetrică*. Astfel de chei sunt împărțite în două subchei, una publică și una privată (secretă). Subcheia publică este publică, putând fi utilizată de oricine pentru criptare de mesaje. Subcheia secretă este cunoscută numai de deținătorul cheii, fiind utilizată pentru decriptare. Cunoașterea subcheii publice nu trebuie să permită determinarea ușoară a subcheii secrete. Criptosistemele bazate pe astfel de chei se numesc *criptosisteme asimetrice* sau *criptosisteme cu chei publice*. Ideea realizării unor astfel de criptosisteme a fost prezentată în 1976 de către W. Diffie și M.E. Hellman [46] la o Conferință Națională de Informatică în SUA (a se vedea și [47])¹⁰. Primul criptosistem de acest tip a fost propus de Merkle și Hellman [134, 135, 87]¹¹. Multe alte criptosisteme au fost apoi propuse, iar unele 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. Puține dintre ele s-au dovedit a fi atât sigure cât și practic utilizabile. Dintre acestea menționăm criptosistemul RSA, criptosistemul Rabin, criptosistemul ElGamal, criptosistemul Chor-Rivest (din clasa criptosistemelor bazate pe problema rucsacului), criptosistemul McEliece și criptosisteme bazate pe curbe eliptice. Primele trei pot fi utilizate și în construcția de semnături digitale.

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 obține $e_K(x) = y$. Perechile (x, y) astfel obținute pot atunci constitui baza unui atac de plaintext cunoscut. Ca urmare, cu cât spațiul plaintextelor este mai mic, cu atât mai mult criptosistemul este nesigur. De fapt, discuția de mai sus ne spune că criptosistemele cu chei publice nu pot asigura securitate necondiționată. Deci ceea ce putem cere de la astfel de criptosisteme este doar securitatea algoritmică (ceea ce nu este puțin dacă această securitate algoritmică este “bine aleasă” adică, bazată pe probleme a căror rezolvare este algoritmic “foarte dificilă” la momentul actual).

4.7.2. Criptosistemul RSA

Criptosistemul RSA a fost propus de Ronald Rivest, Adi Shamir și Leonard Adleman în 1977 [173], luându-și 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 ușor de implementat și utilizat. El se bazează pe “dificultatea” factorizării numerelor foarte mari (a se vedea discuția de la sfârșitul Secțiunii 2.8.2).

¹⁰Independent, această idee apare și la R. Merkle dar, datorită unor probleme editoriale, prima lucrare a acestuia apare abia în 1978 [133].

¹¹Și în acest caz, lucrarea lui Merkle și 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.

4.7.2.1. Descrierea criptosistemului

Criptosistemul RSA folosește aritmetică modulară în \mathbf{Z}_n , unde n este produsul a două numere prime distincte p și q .

Descrierea criptosistemului

- fie p și q două numere prime distincte și fie $n = pq$;
- $\mathcal{P} = \mathcal{C} = \mathbf{Z}_n$;
- $\mathcal{K} = \{(n, p, q, e, d) | e \in \mathbf{Z}_{\phi(n)}^* \wedge ed \equiv 1 \pmod{\phi(n)}\}$;
- pentru orice cheie $K = (n, p, q, e, d) \in \mathcal{K}$ și $x, y \in \mathbf{Z}_n$,

$$e_K(x) = x^e \pmod{n} \quad \text{și} \quad d_K(y) = y^d \pmod{n};$$

- pentru orice cheie $K = (n, p, q, e, d) \in \mathcal{K}$, (n, e) este cheia publică, iar (p, q, d) este cheia secretă.

Arătăm că pentru orice cheie $K = (n, p, q, e, d)$, d_K este invers la stânga a funcție e_K , adică $x^{ed} \equiv x \pmod{n}$, pentru orice $x \in \mathbf{Z}_n$. Avem de analizat două cazuri:

- (1) $x \in \mathbf{Z}_n^*$. Atunci, $x^{ed} \equiv x \pmod{n}$ deoarece $ed \equiv 1 \pmod{\phi(n)}$;
- (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 $x^{ed} \equiv x \pmod{n}$. Presupunem deci că $x \neq 0$. Atunci, deoarece x și n nu sunt prime între ele, $(x, n) \geq 2$. Cum $n = pq$ și $x < n$, 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 și, în baza teoremei lui Fermat, deducem că are loc

$$x^{q-1} \equiv 1 \pmod{q}$$

de la care urmează $x^{t\phi(n)} \equiv 1 \pmod{q}$. Combinând aceasta cu faptul că $p|x$ deducem

$$x^{t\phi(n)+1} \equiv x \pmod{pq}.$$

Adică $x^{ed} \equiv x \pmod{n}$.

Ca urmare, structura definită mai sus este un criptosistem.

Presupunem acum că două părți A și B doresc să schimbe mesaje criptându-le cu RSA cu parametrii p , q și n (p și q cunoscuți doar de A și 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) și reține secret d_A (cheia lui secretă). Similar, B alege (e_B, d_B) , face public e_B și reține secret d_B . Presupunem acum că A dorește a cripta și transmite un mesaj α către B. Acest mesaj se codifică numeric (folosind, de exemplu, corespondența deja adoptată) după care sirul numeric obținut 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 aceeași lungime ca și celelalte blocuri). Fie $x = x_1 \cdots x_m$ această descompunere. Se calculează apoi

$$y_i = x_i^e \text{ mod } n,$$

pentru orice $1 \leq i \leq m$. Secvența $y = y_1 \cdots y_m$ reprezintă codul numeric al criptotextului.

Decriptarea secvenței y cere determinarea subsecvențelor y_1, \dots, y_m (ca mai sus). Aceasta poate constitui o problemă serioasă pentru B, ca urmare a faptului că aceste subsecvențe pot avea lungimi diferite iar B nu va ști cum să le separe. Această problemă poate fi depășită dacă, înainte de a fi trimise, secvențele y_1, \dots, y_m sunt completate la stânga cu zero-uri astfel încât toate să aibă aceeași lungime ca și blocurile x_i . Fie y'_1, \dots, y'_m aceste noi blocuri. Atunci, decriptarea decurge calculând

$$x_i = (y'_i)^d \text{ mod } n,$$

pentru orice $1 \leq i \leq m$.

Exemplul 4.7.2.1. Fie $p = 101$ și $q = 113$ două numere prime. Atunci

$$n = 11413 \text{ și } \phi(n) = 11200.$$

Deoarece $\phi(n) = 2^{65}27$, urmează că e poate fi ales ca fiind orice număr mai mic sau egal cu $n - 1 = 11200$. Această limită este strictă decât $n - 1$ este divizibil prin 2, 5 sau 7. Fie $e = 3533$. Atunci, cu ajutorul algoritmului lui Euclid determinăm $d = e^{-1} \text{ mod } 11200 = 6597$.

Numeralele $n = 11413$ și $e = 3533$ sunt publice; p , q și d sunt secrete. Criptarea textului 9726 produce criptotextul 5761 deoarece

$$9726^{3533} \text{ mod } 11413 = 5761.$$

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 pași:

- alegera a două numere prime p și q (este necesar ca aceste numere să fie mari, de cel puțin 100 de cifre fiecare, pentru a asigura securitatea criptosistemului);
- alegera unui număr $e \in \mathbf{Z}_{\phi(n)}^*$ și determinarea inversului modulo $\phi(n)$ al acestuia, fie el d .

Este recomandabil ca numeralele p și q să aibă lungimi apropiate (dacă nu chiar egale). Alegera lui e constituie un alt punct cheie atât pentru asigurarea securității criptosistemului, cât și pentru eficiența implementării. Se constată că valori mici ale lui e asigură criptare rapidă și, totodată, dă posibilitatea unui factor mare de

decriptare d , ceea ce mărește securitatea criptosistemului (asupra acestui aspect vom reveni). Cele mai utilizate valori pentru e sunt 3, 17 și 65537 ($2^{16} + 1$) (repräsentarea binară a lui 65537 are doar doi biți 1, ceea ce ne spune că se vor face doar 16 ridicări la puterea a 2-a și o înmulțire).

Algoritmi ce implementeză regulile e_K și d_K sunt eficienți, de complexitate timp $\mathcal{O}((\log n)^3)$.

Generarea numerelor prime p și q , de 100 de cifre să spunem, ar ridica următoarea problemă care, aparent, ar fi căt se poate de serioasă: nu epuizăm, într-o perioadă de utilizare a unui astfel de criptosistem, toate numeralele prime de 100 de cifre? Răspunsul este căt se poate de simplu dacă facem apel la Teorema numerelor prime. În baza ei putem spune că există proximativ $4 \cdot 10^{97}$ numere prime cu 100 de cifre, număr ce depășește numărul de atomi din universul vizibil (a se vedea Secțiunea 2.1). Ca urmare, nu trebuie să ne fie teamă că am “epuiza” toate numeralele prime într-o perioadă, fie ea oricât de lungă, de utilizare a criptosistemului RSA.

Testarea primalității unui număr p se poate face prin algoritmi probabilisti de complexitate $\mathcal{O}((\log p)^3)$ (de exemplu, algoritmul Miller-Rabin). Dezavantajul 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 nu poate spune că numărul este prim, chiar dacă acesta nu este). Iterarea unui algoritm probabilist peste o aceeași intrare (același număr pentru care se testează primalitatea) de un număr de ori și obținerea 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 $1/4^{100}$. În 2002, Agrawal, Kayal și Saxena au arătat că problema primalității poate fi rezolvată prin algoritmi determiniști de complexitate timp polinomială (lucrarea acestora a fost publicată în 2004 [1]). Algoritmi determiniști de complexitate timp polinomială cunoscuți la momentul actual au complexitate destul de ridicată, $\mathcal{O}((\log p)^{10.5})$, ceea ce face ca toți algoritmi probabilisti să fie preferați 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 generația aleatoare a numerelor este următoarea: câte numere mari trebuie generate aleator pentru a depista cel puțin 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 este un număr de 512 biți, atunci probabilitatea de mai sus este aproximativ $1/177$. Adică, în medie, generația a 177 de numere mari va include și 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 mașină SPARC II a condus la depistarea unor numere prime de 256 de biți în 2.8 secunde, a unor numere prime de 512 de biți în 24 secunde, a unor numere prime de 768 de biți în 2 minute, și a unor numere prime de 1024 de biți în 5.1 minute [184].

4.7.2.2. Criptanaliză RSA

Trebuie să remarcăm că dacă se cunoaște descompunerea lui n în factori primi, $n = pq$, criptosistemul RSA este compromis (în ipoteza că această descompunere este cunoscută de o persoană neautorizată). În adevăr, cunoșcând p și q se poate determina ușor $\phi(n)$, și atunci cunoașterea 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 cunoșcând doar n , are aceleași consecințe. Calculul lui $\phi(n)$ în timp eficient face apel tot la factorizarea lui n . Ca urmare, putem spune că criptosistemul RSA își bazează securitatea pe problema factorizării pentru care, la momentul actual, nu se cunosc metode eficiente (dacă numerele prime p și q sunt aleator și suficient de mari).

Vom prezenta în continuare 3 modalități de a ataca criptosistemul RSA, fără a cunoaște factorizarea lui n . Fiecare din aceste atacuri speculează utilizare "greșită" a criptosistemului sau alegere nepotrivită a parametrilor.

Atacul lui Davida a fost propus în [40], el exploatând utilizare cu greșeli a criptosistemului.

Presupunem că A trimite un mesaj x lui B folosind cheia publică e_B . Mai presupunem că C intercepteză mesajul transmis, $y = x^{e_B} \text{ mod } n$, și multiplică y cu z , unde z este ales astfel încât există $z^{-1} \text{ mod } n$, și trimite $y' = yz^{e_B} \text{ mod } n$ lui B. Decriptând acest mesaj, B va găsi $x' = y'^{d_B} \text{ mod } n$ care, foarte probabil, este un plaintext fără înțeles. Ca urmare, el va renunța la acest mesaj. Dacă C va putea obține x' , atunci el va putea determina x prin

$$x = x'z^{-1} \text{ mod } n.$$

Atacul lui Lenstra a fost propus în [119], el exploatând, ca și atacul lui Davida, utilizare cu greșeli a criptosistemului.

Decriptarea unui mesaj criptat cu RSA poate fi realizată eficient dacă, dat $y = x^e \text{ mod } n$, se calculează

$$x_p = y^d \text{ mod } p^{p-1} \text{ mod } p$$

și

$$x_q = y^d \text{ mod } q^{q-1} \text{ mod } q.$$

De la acestea, x se obține cu ajutorul Teoremei chineze a resturilor ca fiind unică soluție modulo $n = pq$ a sistemului

$$\begin{cases} x \equiv x_p \text{ mod } p \\ x \equiv x_q \text{ mod } q \end{cases}$$

Să presupunem că x_p a fost calculat corect dar x_q a fost calculat greșit. Ca urmare, rezolvarea sistemului de mai sus va conduce la un x' diferit de x și, privit ca plaintext, fără înțeles. Dacă acest x' este obținut de C, atunci se poate determina în timp eficient p prin

$$p = ((x'^{e_B} - y) \text{ mod } n, n).$$

Ca urmare, criptosistemul este complet neutralizat de C.

Atac de exponent secret mic (atacul lui Wiener). Atacul pe care îl prezentăm aici este datorat lui M. Wiener [223]. El exploatează alegerea unui parametru e care produce o cheie secretă d "prea mică".

Teorema 4.7.2.1. Fie p și q numere prime astfel încât $q < p < 2q$, $n = pq$ și $e, d \in \mathbb{Z}_{\phi(n)}^*$ astfel încât $ed \equiv 1 \pmod{\phi(n)}$. Dacă $d < 1/3\sqrt[4]{n}$, atunci d se poate determina în timp polinomial determinant în raport cu $\log n$ cunoșcând doar n și e .

Demonstrație. Relația $e, d \in \mathbb{Z}_{\phi(n)}^*$ conduce la existența unui număr natural k astfel încât $ed - 1 = k\phi(n)$ sau, echivalent,

$$\left| \frac{e}{\phi(n)} - \frac{k}{d} \right| = \frac{1}{d\phi(n)}. \quad (1)$$

Mai mult, $ed - k\phi(n) = 1$ ne spune că $(k, d) = 1$, deci fractia k/d este ireductibilă.

Ca urmare, determinarea lui d poate fi redusă la determinarea unor fractii ireductibile k/d care să verifice (1). Vom arăta că orice fractie ireductibilă k/d ce verifică (1) va verifica și

$$\left| \frac{e}{n} - \frac{k}{d} \right| < \frac{1}{2d^2}. \quad (2)$$

În plus, determinarea fractiilor 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 relații:

$$\begin{aligned} n - \phi(n) &= p + q - 1 \\ &< 3q - 1 && (\text{de la } p < 2q) \\ &< 3q \\ &< 3\sqrt{n} && (\text{de la } q < p \text{ și } n = pq) \end{aligned}$$

și

$$\begin{aligned} k\phi(n) &= ed - 1 \\ &< ed \\ &< d\phi(n) && (\text{de la } e < \phi(n)) \\ &< 1/3\sqrt[4]{n}\phi(n), && (\text{conform ipotezei}) \end{aligned}$$

care conduce la $k < 1/3\sqrt[4]{n}$.

Acum obținem:

$$\begin{aligned} |e/n - k/d| &= |ed - kn|/(nd) \\ &= |1 + k\phi(n) - kn|/(nd) \\ &= |k(n - \phi(n)) - 1|/(nd) \\ &< (k(n - \phi(n)))/(nd) \\ &< (3k\sqrt{n})/(nd) \\ &= (3k)/(d\sqrt{n}) \\ &< (\sqrt[4]{n})/(d\sqrt{n}) \\ &= 1/(d\sqrt[4]{n}) \\ &< 1/(3d^2) && (\text{de la } \sqrt[4]{n} > 3d) \\ &< 1/(2d^2). \end{aligned}$$

Deci k/d verifică (2).

Fracțiile k/d ce verifică (2) sunt exact convergențele fracției continue finite simple asociate lui e/n [83]. Acestea se obțin pe baza cărurilor succesive ale împărțirii lui e la n prin algoritmul lui Euclid (a se vedea Secțiunea 2.2). Există $\log n$ astfel de convergențe ce pot fi determinate în timp polinomial în raport cu $\log n$. Verificarea faptului că una din aceste convergențe este exact fracția 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 $\log n$. \square

Cu valori de 512 biți pentru p și q , va rezulta n de 1024 biți. Ca urmare, pentru a contracara atacul lui Wiener, d trebuie să fie de cel puțin 256 biți.

Cele prezentate de noi în această secțiune nu au dorit altceva decât să scoată în evidență subtilitatea atacurilor ce se pot monta asupra criptosistemelor și, de ce nu, să atragă interesul cititorului spre acest domeniu extrem de util și interesant.

4.7.3. Semnături digitale

O altă aplicație importantă a problemelor algoritmice ce apar în cadrul grupului \mathbb{Z}_m^* o constituie semnatura digitală.

4.7.3.1. Introducere

Semnătura are scopul de a certifica originalitatea datelor care se transmit între diverse părți. Frecvent, semnăm cecuri, scrisori, contracte; originalitatea semnăturii noastre conferă textului în cauză caracterul de original. În general, o semnătură trebuie să satisfacă următoarele cerințe:

- să fie autentică (produsul original al semnatarului);
- să fie nefalsificabilă;
- să nu poată 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.

În realitate, nici una din cerințele de mai sus nu este complet satisfăcută. Există diverse metode de verificare a originalității unei semnături (analize grafologice etc.) dar care nu conferă garanții complete.

Necesitatea utilizării semnăturilor este cât se poate de clară pentru oricine dintre noi. Să presupunem că A și B sunt două părți ce doresc să comunice, schimbând între ele diverse mesaje (documente, acte, scrisori etc.). Există cel puțin două aspecte fundamentale care trebuie luate în considerație:

1. dacă B primește un mesaj x de la A , atunci B dorește să aibă garanția că x este, în adevăr, mesajul original trimis de A (acest mesaj nu a fost schimbat pe parcurs de o terță 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 dorește să aibă garanția că mesajul x este cel recepționat de B (mesajul nu a fost schimbat pe parcurs de o terță persoană C sau, chiar de către B).

Într-un astfel de context, utilizarea semnăturii (personale) constituie soluția problemei. A va semna suplimentar mesajul x prin determinarea, după o procedură de semnare secretă (personală), a entității $sig(x)$ (semnătură grafică, amprentă digitală etc.). Când B va receptiona cuplul $(x, sig(x))$, acesta va trebui să verifice, cu o procedură publică $ver(x, sig(x))$, autenticitatea semnăturii $sig(x)$ asupra mesajului x . Interpunerea lui C între A și B , fără a cunoaște procedura de semnare, trebuie să fie ineficientă.

Discuția purtată până acum poate fi bine descrisă prin intermediul Figurii 4.3.

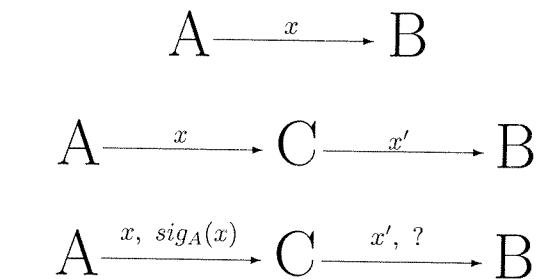


Figura 4.3: Schema de semnare cu interpunerea unei terțe părți

În 1976, Diffie și Hellman [47], odată cu lansarea ideii de cheie publică propune utilizarea semnăturilor și în cazul transmisiei datelor prin intermediul calculatorului (mesaje, programe etc.). Cerințele pe care trebuie să le satisfacă o semnătură, cât și precauțiile (măsurile de siguranță) pe care trebuie să și le ia părțile implicate în comunicare, discutate mai sus, trebuie să rămână valabile și în acest caz. Spre deosebire de semnătura uzuală a fiecărui, ce este în linii mari aceeași pe orice document, semnătura digitală variază de la document la document.

Definiția 4.7.3.1. 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 mulțime finită și nevidă ale cărei elemente sunt numite *mesaje*;

- (2) \mathcal{S} este o mulțime finită și nevidă ale cărei elemente sunt numite *semnături*;
 (3) \mathcal{K} este o mulțime finită și nevidă ale cărei elemente sunt numite *chei de semnare*;
 (4) $\mathcal{A}_s = \{sig_K : \mathcal{P} \rightarrow \mathcal{S} | K \in \mathcal{K}\}$ și $\mathcal{A}_v = \{ver_K : \mathcal{P} \times \mathcal{S} \rightarrow \{0, 1\} | K \in \mathcal{K}\}$ sunt mulțimi \mathcal{K} -indexate ale căror elemente sunt numite *reguli/algoritmi de semnare* și, respectiv, *reguli/algoritmi de verificare*, astfel încât

$$ver_K(x, y) = 1 \Leftrightarrow sig_K(x) = y,$$

pentru orice $x \in \mathcal{P}$, $y \in \mathcal{S}$ și $K \in \mathcal{K}$.

Utilizarea unei scheme de semnare de către două părți A și 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 obținând $sig_K(x)$, și transmite lui B cuplul $(x, sig_K(x))$;
- *verificare.* B primește un cuplu $(x, sig_K(x))$ și acceptă semnătura lui A dacă și numai dacă $ver_K(x, sig_K(x)) = 1$.

O cerință naturală asupra schemelor de semnare digitală constă în aceea că semnarea și verificarea trebuie să se facă “ușor” (în timp polinomial determinist). Algoritmii de semnare trebuie să fie secrete, iar cei de verificare, publici. Interceptarea unui mesaj și a unei semnături asociate nu trebuie să permită determinarea algoritmului de semnare și nici falsificarea semnăturii prin alte mijloace (semnarea corectă a altor mesaje fără a cunoaște algoritmul de semnare).

Semnăturile digitale nu pot asigura securitate necondiționată pentru simplul motiv că verificarea tuturor posibilităților (ce sunt în număr finit) conduce la determinarea algoritmului de semnare utilizat.

Semnăturile digitale pot fi utilizate în conjuncție cu metodele de criptare cu chei publice astfel. Să presupunem că A dorește să transmită lui B un mesaj x . A își alege un algoritm (propriu) de semnare sig_A , semnează mesajul, fie $y = sig_A(x)$, și criptează cuplul (x, y) folosind cheia publică e_B a lui B . Rezultatul, $z = e_B(x, y)$, este trimis lui B .

Evident, A poate cripta întâi pe x prin $e_B(x)$ și apoi semna rezultatul. 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 cunoaște mesajul original x , prin semnătura lui proprie $sig_C(e_B(x))$ și trimită lui B cuplul

$$(e_B(x), sig_C(e_B(x))).$$

În urma recepționării acestui cuplu, B va considera mesajul ca venind din partea lui C și nu a lui A (mesajul poartă semnătura lui C). O astfel de situație poartă denumirea

de *impersonificare*. Pentru a o evita, se recomandă întâi semnarea mesajului și apoi criptarea cuplului (mesaj, semnătură).

Adesea cheia K utilizată pentru semnare conține un parametru k , numit *parametru de securitate*, care, la schimbarea cheii, se schimbă doar el. În acest caz este de preferat de păstrat restul cheii drept cheie și de gândit parametrul de securitate ca fiind un parametru auxiliar. Altfel spus, este de preferat ca procedura de semnare să fie o funcție de două variabile, $sig_K(x, k)$. Evident, aceasta nu modifică cu absolut nimic strategia generală.

4.7.3.2. Semnătura ElGamal

Semnătura ElGamal a fost propusă în 1985 [51]. 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 $x \in \mathbb{Z}_p^*$, unde p este un număr prim, printr-o pereche $(\gamma, \delta) \in \mathbb{Z}_p^* \times \mathbb{Z}_{p-1}$ astfel încât x este combinație liniară de γ și δ cu doi parametri secrete a și k ,

$$x = (a\gamma + k\delta) \bmod (p - 1).$$

Această combinație este aleasă modulo $p - 1$ deoarece verificarea semnăturii se va face prin intermediul echivalenței

$$u \equiv v \bmod (p - 1) \Leftrightarrow \alpha^u \equiv \alpha^v \bmod p,$$

pentru orice rădăcină primitivă α a lui \mathbb{Z}_p^* .

Descrierea semnăturii:

- fie p un număr prim și α o rădăcină primitivă modulo p ;
- $\mathcal{P} = \mathbb{Z}_p^*$;
- $\mathcal{S} = \mathbb{Z}_p^* \times \mathbb{Z}_{p-1}$;
- $\mathcal{K} = \{(p, \alpha, a, \beta) | a \in \mathbb{Z}_{p-1}, \beta = \alpha^a \bmod p\}$;
- pentru orice $K = (p, \alpha, a, \beta)$ și $k \in \mathbb{Z}_{p-1}^*$, mesajul $x \in \mathbb{Z}_p^*$ este semnat prin

$$sig_K(x, k) = (\gamma, \delta),$$

unde

$$\gamma = \alpha^k \bmod p \quad \text{și} \quad \delta = (x - a\gamma)k^{-1} \bmod (p - 1)$$

$(k^{-1}$ este determinat modulo $p - 1$), iar verificarea semnăturii (γ, δ) pentru mesajul x se face prin

$$ver_K(x, (\gamma, \delta)) = 1 \Leftrightarrow \beta^\gamma \gamma^\delta \equiv \alpha^x \bmod p.$$

Numerele p , α și β sunt publice (unui grup de utilizatori), iar a este secret (particular fiecărui utilizator în parte). În plus, semnarea unui mesaj x presupune alegerea unui parametru de securitate k , ceea ce întărește securitatea schemei fără a afecta eficiența verificării (observăm că algoritmul de verificare nu depinde de k).

Schema descrisă mai sus este o schemă de semnare. În adevăr, are loc următorul săr de echivalențe:

$$\begin{aligned} \delta = (x - a\gamma)k^{-1} \bmod p - 1 &\Leftrightarrow k\delta \equiv (x - a\gamma) \bmod p - 1 \\ &\Leftrightarrow x \equiv (a\gamma + k\delta) \bmod p - 1 \\ &\Leftrightarrow \alpha^x \equiv \alpha^{a\gamma+k\delta} \bmod p \\ &\Leftrightarrow \alpha^x \equiv \beta^\gamma \gamma^\delta \bmod p. \end{aligned}$$

Exemplul 4.7.3.1. Fie $p = 467$, $\alpha = 2$ și $a = 127$. Atunci

$$\begin{aligned} \beta &= \alpha^a \bmod p \\ &= 2^{127} \bmod 467 \\ &= 132. \end{aligned}$$

Presupunem că se dorește a se semnă mesajul $x = 100$ folosind parametrul de securitate $k = 213$ ($k \in \mathbf{Z}_{466}^*$ și $k^{-1} = 431$). Numerele γ și δ vor fi date prin:

$$\gamma = 2^{213} \bmod 467 = 29$$

și

$$\delta = (100 - 127 \cdot 29) \cdot 431 \bmod 466 = 51.$$

Ca urmare, $sig_K(x, k) = (29, 51)$.

Verificarea se face calculând

$$132^{29} \cdot 29^{51} \bmod 467$$

și

$$2^{100} \bmod 467$$

și constatănd că ele sunt congruente modulo 467.

Algoritmii de semnare și verificare pentru semnătura ElGamal au complexitatea $\mathcal{O}((\log p)^3)$.

Semnătura ElGamal își bazează securitatea pe intractabilitatea problemei logaritmului discret. Ca urmare, p și α trebuie aleși astfel încât să fie îndeplinit acest deziderat (p trebuie să fie număr prim suficient de mare, în general de cel puțin 1024 biți, iar α să fie rădăcină primitivă modulo p generată aleator).

Chiar dacă p și α sunt aleși ca mai sus, există situații care compromit semnătura, așa cum se va vedea mai jos. Menționăm însă că nici una din situațiile descrise 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 cunoaște a . Presupunem că, printr-o anume metodă, am determinat γ . Atunci urmează să determinăm δ . Aceasta se reduce la rezolvarea ecuației

$$\gamma^\delta \equiv \alpha^x \beta^{-\gamma} \bmod p,$$

care constituie subiectul problemei logaritmului discret.

Dacă presupunem că, printr-o anume metodă, am determinat δ , determinarea lui γ se reduce la rezolvarea ecuației

$$\beta^\gamma \gamma^\delta \equiv \alpha^x \bmod p.$$

Pentru rezolvarea acestei ecuații, care nu "pare" a fi problema logaritmului discret, nu se cunoaște 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 ecuației

$$\beta^\gamma \gamma^\delta \equiv \alpha^x \bmod p,$$

care este o instanță a problemei logaritmului discret.

Determinarea simultană a semnăturii și a mesajului. Există posibilitatea determinării "simultane" a 3 numere γ , δ și x astfel încât (γ, δ) să fie semnătură pentru x . În adevăr, pentru orice i și j cu $0 \leq i, j \leq p - 2$ și $(j, p - 1) = 1$, numerele:

$$\begin{aligned} \gamma &= \alpha^i \beta^j \bmod p \\ \delta &= -\gamma j^{-1} \bmod (p - 1) \\ x &= -\gamma i j^{-1} \bmod (p - 1), \end{aligned}$$

unde j^{-1} este determinat modulo $p - 1$, verifică:

$$\begin{aligned} \beta^\gamma \gamma^\delta &\equiv \beta^{\alpha^i \beta^j} (\alpha^i \beta^j)^{-\alpha^i \beta^j j^{-1}} \bmod p \\ &\equiv \beta^{\alpha^i \beta^i} \alpha^{-ij^{-1}} \alpha^i \beta^j \beta^{-\alpha^i \beta^j} \bmod p \\ &\equiv \alpha^{-ij^{-1}} \alpha^i \beta^j \bmod p \\ &\equiv \alpha^{-\gamma i j^{-1}} \bmod p \\ &\equiv \alpha^x \bmod p, \end{aligned}$$

ceea ce arată că (γ, δ) este semnătură pentru x .

Falsificarea semnături cunoscând o semnătură. Dacă se poate intercepta un cuplu $(x, (\gamma, \delta))$, unde (γ, δ) este semnătură a lui x , atunci se pot semnă (valid) și alte mesaje. În adevăr, pentru orice h , i și j cu $0 \leq h, i, j \leq p - 2$ și $(h\gamma - j\delta, p - 1) = 1$, numerele:

$$\begin{aligned} \gamma' &= \gamma^h \alpha^i \beta^j \bmod p \\ \delta' &= \delta \gamma' (h\gamma - j\delta)^{-1} \bmod (p - 1) \\ x' &= \gamma' (hx + i\delta) (h\gamma - j\delta)^{-1} \bmod (p - 1), \end{aligned}$$

unde $(h\gamma - j\delta)^{-1}$ este determinat modulo $p - 1$, verifică

$$\beta^{\gamma'}(\gamma')^{\delta'} \equiv \alpha^{x'} \pmod{p},$$

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 cunoaște nici un atac serios la adresa acestei semnături, exceptând unele "neglijente" de protocol pe care le vom semnală în continuare.

Cunoașterea numărului secret k . Presupunem că s-a interceptat $(x, (\gamma, \delta))$, unde (γ, δ) este semnătură a lui x . Atunci cunoașterea lui k conduce la determinarea imediată a lui a prin relația

$$a = (x - k\gamma)\delta^{-1} \pmod{p - 1},$$

ceea ce compromite semnătura.

Utilizarea același k pentru a semna mesaje diferite. Dacă se utilizează același număr k pentru a semna două mesaje distincte x_1 și x_2 , atunci semnăturile sunt de forma (γ, δ_1) și (γ, δ_2) . Ca urmare,

$$\beta^{\gamma}\gamma^{\delta_1} \equiv \alpha^{x_1} \pmod{p}$$

și

$$\beta^{\gamma}\gamma^{\delta_2} \equiv \alpha^{x_2} \pmod{p},$$

ceea ce conduce la

$$\alpha^{x_1-x_2} \equiv \gamma^{\delta_1-\delta_2} \pmod{p}.$$

Deoarece $\gamma = \alpha^k \pmod{p}$, relația de mai sus conduce la

$$\alpha^{x_1-x_2} \equiv \alpha^{k(\delta_1-\delta_2)} \pmod{p},$$

care este echivalentă cu

$$k(\delta_1 - \delta_2) \equiv x_1 - x_2 \pmod{p - 1}.$$

Aceasta este o ecuație modulară în necunoscuta k ce admite soluție (k a fost utilizat pentru a semna x_1 și x_2). Conform Teoremei 2.5.1, soluțiile în \mathbf{Z}_{p-1} ale acestei ecuații sunt

$$(k_0 + i(p-1)/d) \pmod{p-1},$$

unde k_0 este o soluție arbitrară a ei, $d = (\delta_1 - \delta_2, p - 1)$ și $0 \leq i < d$.

Determinarea unei soluții k_0 se poate face cu algoritmul extins al lui Euclid, așa cum a fost discutat în Secțiunea 2.2. Valoarea reală a lui k se determină testând relația $\gamma \equiv \alpha^k \pmod{p}$.

4.7.3.3. Semnătura DSS

Semnătura DSS (Digital Signature Standard) a fost propusă în august 1991 de NIST ca metodă standard de schemă de semnare [52]. După 3 ani de dispute și critici asupra ei, în mai 1994 semnătura DSS a fost adoptată și publicată [161]. Ea este, în esență, o variație a schemei de semnare ElGamal, variație generată pe baza următoarei observații. 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 își păstreze intacte calitățile pentru ani și 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 biți sau, așa cum se sugerează, chiar 1024 de biți). Dacă p este ales de 512 biți, semnătura va avea 1024 de biți, 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 biți fără a compromite securitatea acesteia. Idea de bază constă în utilizarea unui număr prim de 512 biți (sau chiar 1024 de biți) și a unui factor prim q de 160 de biți 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 combinație liniară de γ și δ cu doi parametri secereti $a \in \mathbf{Z}_q$ și $k \in \mathbf{Z}_q^*$,

$$x = (-a\gamma + k\delta) \pmod{q}.$$

Această combinație este aleasă modulo q deoarece verificarea semnături se va face prin intermediul echivalenței

$$u \equiv v \pmod{q} \Leftrightarrow \alpha^u \equiv \alpha^v \pmod{p},$$

pentru orice element $\alpha \in \mathbf{Z}_p^*$ de ordin q . Se poate vedea astfel analogia cu schema ElGamal, înlocuind rădăcina primă 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 și α astfel încât p este prim, problema logaritmului discret în \mathbf{Z}_p^* este intractabilă, q este un factor prim al lui $p - 1$, iar $\alpha \in \mathbf{Z}_p^*$ este un element de ordin q (ca urmare, $\alpha \pmod{q}$ este rădăcina primă în \mathbf{Z}_q^*);
- $\mathcal{P} = \mathbf{Z}_p^*$;
- $\mathcal{S} = \mathbf{Z}_q \times \mathbf{Z}_q$;
- $\mathcal{K} = \{(p, q, \alpha, a, \beta) | a \in \mathbf{Z}_q \wedge \beta = \alpha^a \pmod{p}\}$;
- pentru orice cheie $K = (p, q, \alpha, a, \beta)$ și $k \in \mathbf{Z}_q^*$,

$$\text{sig}_K(x, k) = (\gamma, \delta),$$

unde

$$\gamma = (\alpha^k \bmod p) \bmod q \quad \text{și} \quad \delta = (x + a\gamma)k^{-1} \bmod q,$$

iar

$$ver_K(x, (\gamma, \delta)) = 1 \Leftrightarrow (\alpha^{e_1}\beta^{e_2} \bmod p) \bmod q = \gamma,$$

unde

$$e_1 = x\delta^{-1} \bmod q \quad \text{și} \quad e_2 = \gamma\delta^{-1} \bmod q,$$

pentru orice $x \in \mathbf{Z}_p^*$ (k^{-1} și δ^{-1} sunt calculate modulo q).

Numerele p, q, α și β sunt publice (unui grup de utilizatori), iar a este secret (particular fiecarui utilizator în parte).

Verificarea faptului că schema descrisă mai sus este, în adevăr, o schemă de semnare decurge astfel. Relația $k\delta \equiv (x + a\gamma) \bmod q$ este echivalentă, succesiv, cu:

$$\begin{aligned} k\delta \equiv (x + a\gamma) \bmod q &\Leftrightarrow k \equiv (x + a\gamma)\delta^{-1} \bmod q \\ &\Leftrightarrow \alpha^k \equiv \alpha^{(x+a\gamma)\delta^{-1}} \bmod p \\ &\Leftrightarrow \alpha^k \equiv \alpha^{x\delta^{-1}}(\alpha^a)\gamma\delta^{-1} \bmod p \\ &\Leftrightarrow \alpha^k \equiv \alpha^{x\delta^{-1}}\beta^{a\delta^{-1}} \bmod p \\ &\Leftrightarrow \alpha^k \equiv \alpha^{e_1}\beta^{e_2} \bmod p \end{aligned}$$

(pentru cea de a doua echivalență s-a utilizat faptul că α are ordinul q în \mathbf{Z}_p^*). De aici urmează $\gamma = (\alpha^{e_1}\beta^{e_2} \bmod p) \bmod q$.

Exemplul 4.7.3.2. Fie $q = 101$, $p = 78q + 1 = 7879$ și 3 un element primitiv în \mathbf{Z}_{7879} . Considerăm $\alpha = 3^{78} \bmod 7879 = 170$ și $a = 75$. Atunci

$$\beta = \alpha^a \bmod p = 4567.$$

Presupunem că se dorește a se semnă mesajul $x = 1234$ folosind parametrul de securitate $k = 50$ ($k \in \mathbf{Z}_{75}$ și $k^{-1} = 99$). Numerele γ și δ vor fi date prin:

$$\gamma = (170^{50} \bmod 7879) \bmod 101 = 94,$$

și

$$\delta = (1234 + 75 \cdot 94) \cdot 99 \bmod 101 = 97.$$

Că urmare, $sig_K(x, k) = (94, 97)$.

Algoritmii de semnare și verificare au complexitate cel mult $\mathcal{O}((\log p)^3)$. Se recomandă ca p și q să satisfacă

$$2^{l-1} < p < 2^l, \quad 2^{159} < q < 2^{160}$$

cu $512 \leq l \leq 1024$ multiplu de 64 [54].

Numărul α se poate alege pornind de la un element primitiv α_0 în \mathbf{Z}_p , prin

$$\alpha = \alpha_0^{\frac{p-1}{q}} \bmod p.$$

Prin modul de alegere al lui α , numerele β și γ vor fi, de asemenea, rădăcini de ordin q ale lui 1 modulo p . Necesitatea calculului lui δ^{-1} cere îndeplinirea relației $\delta \not\equiv 0 \bmod q$ (probabilitatea de a se obține un δ cu $\delta \equiv 0 \bmod q$ este de 2^{-160}).

Capitolul 5

Inele și corpuri

Din punct de vedere istoric, teoria inelelor își are rădăcinile în studiul polinoamelor și al numerelor. Conceptul de inel a fost introdus de Richard Dedekind, dar terminologia de *inel* a fost propusă de David Hilbert în [91]. Prima definiție axiomatică a inelelor a fost propusă de Fraenkel în 1914 [56].

În acest capitol vom prezenta câteva elemente de bază de teoria inelelor și corpuri, orientând materialul spre corpuri finite care au importanță deosebită în informatică în domenii ca teoria codurilor bloc și criptografie. Pentru detalii recomandăm [37, 126, 96, 141, 196].

5.1. Definiții. Exemple. Proprietăți de bază

Așa cum s-a spus în Secțiunea 1.4.4.1, un *inel* este o algebră $(R, +, -, 0, \cdot)$ ce satisfac:

- (1) $(R, +, -, 0)$ este grup comutativ;
- (2) (R, \cdot) este semigrup;
- (3) \cdot este distributivă la stânga și la dreapta față de $+$.

Operația nulară 0 este ușual numită *elementul zero* al inelului. Inelul este numit *comutativ* dacă operația \cdot este comutativă. Această proprietate se va transfera la toate structurile pe care le vom introduce în continuare și care au la bază conceptul de inel.

Atunci când nu va fi pericol de confuzie ne vom referi la inelul $(R, +, -, 0, \cdot)$ prin intermediul mulțimii R , așa cum am făcut și în cadrul semigrupurilor, monoizilor sau grupurilor. Adică, vom spune simplu că “ R este inel”.

Conform definiției, toate notațiile, precum și proprietățile corespunzătoare, introduse pentru grupuri aditive și semigrupuri moltiplicative (iar atunci când va fi cazul,

monoizi multiplicativi sau grupuri multiplicative), vor fi transferate și la inele. Astfel, vom utiliza multiplii întregi cât și puteri strict pozitive (pozitive sau întregi, atunci când este posibil) a elementelor inelului. Pe lângă proprietățile lor deja cunoscute putem stabili noi proprietăți specifice inelelor.

Propoziția 5.1.1. Fie $(R, +, -, 0, \cdot)$ un inel. Atunci au loc următoarele proprietăți:

- (1) $a0 = 0a = 0$, pentru orice $a \in R$;
- (2) $(-a)b = a(-b) = -(ab)$, pentru orice $a, b \in R$;
- (3) $(-a)(-b) = ab$, pentru orice $a, b \in R$;
- (4) $a(b - c) = ab - ac$ și $(b - c)a = ba - ca$, pentru orice $a, b, c \in R$;
- (5) $(\sum_{i=1}^n a_i)(\sum_{j=1}^m b_j) = \sum_{i=1}^n \sum_{j=1}^m a_i b_j$, pentru orice $n, m \geq 1$ și $a_i, b_j \in R$, $1 \leq i \leq n$ și $1 \leq j \leq m$.

Demonstrație. (1) Fie $a \in R$. Au loc relațiile:

$$a0 = a(0 + 0) = a0 + a0,$$

de unde urmează $a0 = 0$ prin utilizarea inversului aditiv al (opusului) lui $a0$. În mod similar, se obține $0a = 0$.

(2) Fie $a, b \in R$. Au loc relațiile:

$$0 = (a + (-a))b = ab + (-a)b,$$

de unde obținem $(-a)b = -(ab)$. În mod similar se arată că $a(-b) = -(ab)$.

(3) și (4) urmează cu ușurință de la (2), iar (5) se demonstrează prin inducție matematică utilizând proprietatea de distributivitate a operației \cdot față de $+$. □

Demonstrația următoarei propoziții rămâne în seama cititorului.

Propoziția 5.1.2. În orice inel R au loc următoarele proprietăți:

- (1) $(-m)a = -(ma)$;
- (2) $(m + n)a = ma + na$;
- (3) $m(a + b) = ma + mb$;
- (4) $(mn)a = m(na)$;
- (5) $m(ab) = (ma)b = a(mb)$;
- (6) $(ma)(nb) = (mn)(ab)$,

pentru orice $a, b \in R$ și $m, n \geq 1$.

Propoziția 5.1.3. Fie R un inel comutativ. Atunci, pentru orice $a, b \in R$ și $n \geq 1$, are loc:

$$(a + b)^n = \sum_{k=0}^n C_n^k a^{n-k} b^k$$

unde $C_n^k = n!/(k!(n - k)!)$, pentru orice $0 \leq k \leq n$.

Demonstrație. Prin inducție după $n \geq 1$ și prin utilizarea formulei

$$C_n^k + C_n^{k+1} = C_{n+1}^{k+1},$$

pentru orice $0 \leq k < n$. □

Un inel cu unitate (Exemplul 1.4.4.1(6)) este o algebră $(R, +, -, 0, \cdot, e)$ ce satisfac:

- $(R, +, -, 0)$ este grup comutativ;
- (R, \cdot, e) este monoid;
- \cdot este distributivă la stânga și la dreapta față de $+$.

Operația nulară e este ușual numită *unitatea inelului*, fiind unică cu proprietatea de a fi unitate relativ la operația notată multiplicativ.

Dacă $(R, +, -, 0, \cdot, e)$ este un inel cu unitate, atunci mulțimea

$$U(R) = \{a \in R | (\exists b \in R)(ab = ba = e)\}$$

formează grup comutativ în raport cu operația notată multiplicativ. Acest grup se numește *grupul unităților inelului R* (a se vedea Secțiunea 3.1 pentru grupul unităților unui monoid și Secțiunea 4.5 pentru grupul unităților inelului \mathbf{Z}_m).

Propoziția 5.1.4. Într-un inel cu unitate $(R, +, -, 0, \cdot, e)$ are loc $e = 0$ dacă și numai dacă $R = \{0\}$.

Demonstrație. Dacă $e = 0$ atunci, pentru orice $a \in R$ are loc

$$a = ae = a0 = 0,$$

ceea ce arată că $R = \{0\}$. Reciproc, este ușor de văzut că $R = \{0\}$ formează inel cu unitate, și atunci $e = 0$. □

Ca urmare a acestei propoziții, egalitatea $e = 0$ are loc numai în inele triviale, numite și *nule*. Inelele care nu sunt nule vor fi numite *nenule*. Vom exclude cu precădere inelul nul din considerațiile noastre ca fiind un caz degenerat care nu ridică probleme semnificative de analiză și studiu.

Vom prezenta în continuare câteva clase speciale de inele.

Definiția 5.1.1.

(1) Se numește *inel cu diviziune* orice algebră $(R, +, -, 0, \cdot, ', e)$ ce satisfac:

- $(R, +, -, 0)$ este grup comutativ;
- (R, \cdot, e) este monoid și $e \neq 0$;
- $'$ este o operație unară ce satisfac $aa' = a'a = e$ pentru orice $a \neq 0$;
- \cdot este distributivă la stânga și la dreapta față de $+$.

(2) Se numește *corp* orice inel comutativ cu diviziune.

În unele lucrări, inelele cu diviziune mai sunt numite și *corpuri necomutative*. Ca și în cazul grupurilor, se arată imediat că pentru orice inel cu diviziune $(R, +, -, 0, \cdot, ', e)$ și $a \in R$, elementul a' este unic. El va fi notat și prin a^{-1} (a se vedea notațiile introduse în cadrul grupurilor).

În Definiția 5.1.1(1) nu se cere inversabilitatea lui 0 deoarece $0a = a0 = 0$, pentru orice $a \in R$. Ca urmare, 0 este inversabil dacă și numai dacă inelul este nul. Cerând în definiție ca unitatea să fie diferită de elementul zero, acest caz este exclus automat.

Am fi tentați să spunem că dacă $(R, +, -, 0, \cdot, ', e)$ este inel cu diviziune, atunci $(R - \{0\}, \cdot, ', e)$ este grup. Această afirmație este adevărată dacă arătăm că, pentru orice $a, b \in R - \{0\}$, produsul ab este diferit de 0 (adică, este element al mulțimii $R - \{0\}$). Acest fapt va fi stabilit mai jos.

Definiția 5.1.2. Fie R un inel. Un element $a \in R - \{0\}$ este numit *divizor al lui zero* dacă există $b \in R - \{0\}$ astfel încât $ab = 0$ sau $ba = 0$.

Evident, inelele pot fi clasificate acum în inele cu divizori ai lui zero și inele fără divizori ai lui zero. Un inel este fără divizori ai lui zero dacă are loc

$$ab = 0 \Rightarrow a = 0 \text{ sau } b = 0,$$

pentru orice $a, b \in R$.

Inelul nul este fără divizori ai lui zero.

Propoziția 5.1.5. Dacă R este un inel și $c \in R - \{0\}$ nu este divizor al lui zero, atunci oricare din relațiile $ac = bc$ sau $ca = cb$ implică $a = b$, pentru orice $a, b \in R$.

Demonstrație. Relația $ac = bc$ conduce la $(a - b)c = 0$, de unde urmează $a = b$ deoarece c nu este divizor al lui zero. \square

Propoziția 5.1.6. Inelele cu diviziune nu au divizori ai lui zero.

Demonstrație. Fie R un inel cu diviziune și $a, b \in R$ astfel încât $a \neq 0$ și $ab = 0$. Aplicând a^{-1} la stânga acestei relații obținem $b = 0$. Deci R nu are divizori ai lui zero. \square

Ca urmare a acestei propoziții, dacă $(R, +, -, 0, \cdot, ', e)$ este inel cu diviziune, atunci $(R - \{0\}, \cdot, ', e)$ este grup.

Definiția 5.1.3. Un inel nenul cu unitate, fără divizori ai lui zero și comutativ se numește *domeniu de integritate*.

Inelele cu diviziune nu includ domeniile de integritate datorită comutativității (care nu este o cerință în cadrul inelelor cu diviziune).

Propoziția 5.1.7. Orice corp este domeniu de integritate.

Demonstrație. De la definiții și Propoziția 5.1.6. \square

În cazul finit, Propoziția 5.1.7 admite și o reciprocă.

Propoziția 5.1.8. Orice domeniu de integritate finit este corp.

Demonstrație. Fie $(R, +, -, 0, \cdot, ', e)$ un domeniu de integritate ale cărui elemente sunt a_1, \dots, a_n (toate aceste elemente sunt presupuse distințe două câte două). Fie $a \in R$ cu $a \neq 0$. Atunci, $aa_1, \dots, aa_n \in R$ și $aa_i \neq aa_j$, pentru orice $i \neq j$. În adevăr, dacă ar exista două numere i și j cu $i \neq j$ și $aa_i = aa_j$, atunci am obține $a(a_i - a_j) = 0$, ceea ce ar conduce la $a_i - a_j = 0$ deoarece $a \neq 0$. Aceasta contrazice însă presupunerea inițială.

Ca urmare, unul dintre elementele aa_i trebuie să fie e . Combinând cu comutativitatea, deducem că a admite invers și, deci, R este corp. \square

Corolarul 5.1.1. Fie $p \geq 2$. \mathbf{Z}_p este corp dacă și numai dacă p este prim.

Demonstrație. Să presupunem că p este prim. Pentru a arăta că \mathbf{Z}_p este corp este suficient de arătat că \mathbf{Z}_p este domeniu de integritate (el fiind inel comutativ cu unitate). Deci este suficient de arătat că nu are divizori ai lui zero.

Fie $a, b \in \mathbf{Z}_p$. Presupunem că are loc $ab = 0$, adică $ab \equiv 0 \pmod p$. Atunci, $p|ab$ și, cum p este prim, urmează că $p|a$ sau $p|b$. Ca urmare, ori $a = 0$ ori $b = 0$. Deci, \mathbf{Z}_p este fără divizori ai lui 0.

Reciproc, dacă presupunem că \mathbf{Z}_p este corp dar p este compus, fie de exemplu $p = rs$ cu $1 < r, s < p$, atunci

$$s = r^{-1}rs = r^{-1}(rs) = r^{-1}p = 0,$$

ceea ce constituie o contradicție (r^{-1} este inversul modulo p a lui r ; acesta există deoarece p este prim). \square

Exemplul 5.1.1.

- (1) Fie $(R, +, -, 0)$ un grup comutativ. Definim pe R operația binară \cdot prin $a \cdot b = 0$, pentru orice $a, b \in R$. Atunci $(R, +, -, 0, \cdot)$ este inel.
- (2) \mathbf{Z} , cu operațiile uzuale, formează domeniu de integritate, dar nu corp.
- (3) \mathbf{Q} , \mathbf{R} și \mathbf{C} , cu operațiile uzuale, formează corpuri.

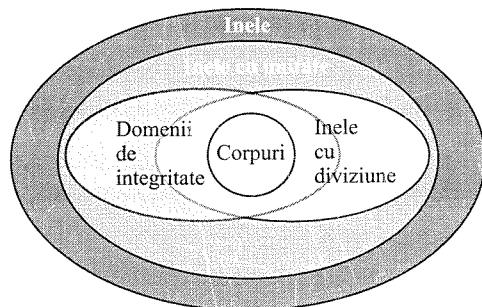


Figura 5.1: Relații între clase de inele

- (4) $n\mathbb{Z}$ este inel comutativ fără divizori ai lui zero. Acest inel are unitate doar în cazurile $n = -1$, $n = 0$ sau $n = 1$ (pentru $n = 0$ el este inel nul).
- (5) \mathbb{Z}_n este inel comutativ cu unitate. Pentru anumite valori ale lui n este posibil ca \mathbb{Z}_n să aibă divizori ai lui zero, cum ar fi cazul $n = 6$. De exemplu, $2 \cdot 3 = 0$ (în \mathbb{Z}_6), dar $2 \neq 0 \neq 3$.
- Dacă n este prim, atunci \mathbb{Z}_n este corp (Corolarul 5.1.1).
- (6) Multimea matricilor pătratice de tip $n \times n$ peste \mathbf{Q} (\mathbf{R} , \mathbf{C}), cu operațiile uzuale, formează inel cu unitate, necomutativ.

Diagrama din Figura 5.1 ne arată relațiile dintre clasele de inele introduse până acum. Incluziunile sunt stricte.

Există o strânsă legătură între inelele în care orice element este idempotent relativ la operația notată multiplicativ și algebrele Booleene.

Definiția 5.1.4. Un inel $(R, +, -, 0, \cdot)$ este numit *inel Boolean* dacă $x^2 = x$, pentru orice $x \in R$.

Exemplul 5.1.2. Pentru orice mulțime A , $(\mathcal{P}(A), \Delta, -, \emptyset, \cap, A)$, unde Δ este diferența simetrică iar $-$ este operația de complementariere a mulțimilor în raport cu A , este inel Boolean cu unitate.

Propoziția 5.1.9. Fie $(R, +, -, 0, \cdot)$ un inel Boolean. Atunci

- (1) $x + x = 0$, pentru orice $x \in R$;
- (2) $xy = yx$, pentru orice $x, y \in R$.

Demonstrație. (1) Conform definiției,

$$x + x = (x + x)^2 = x^2 + 2x^2 + x^2 = x + 2x + x,$$

pentru orice $x \in R$, de la care urmează $x + x = 0$.

(2) Conform definiției,

$$x + y = (x + y)^2 = x^2 + xy + yx + y^2 = x + xy + yx + y,$$

pentru orice $x, y \in R$, de la care urmează $xy + yx = 0$. Dar atunci, în baza lui (1), obținem $xy = -yx = (-y)x = yx$, ceea ce stabilește proprietatea cerută. \square

Teorema 5.1.1.

(1) Dacă $(R, +, -, 0, \cdot, 1)$ este un inel Boolean cu unitate, atunci $(R, \vee, \wedge, ', 0, 1)$, unde \vee, \wedge și $'$ sunt date prin:

- $x \vee y = x + y - xy$;
- $x \wedge y = xy$;
- $x' = 1 + x$,

pentru orice $x, y \in R$, este o algebră Booleană.

(2) Dacă $(R, \vee, \wedge, ', 0, 1)$ este o algebră Booleană, atunci $(R, +, -, 0, \cdot, 1)$, unde $+$, $-$ și \cdot sunt date prin:

- $x + y = (x \wedge y') \vee (x' \wedge y)$;
- $x \cdot y = x \wedge y$;
- $-x = x'$,

pentru orice $x, y \in R$, este inel Boolean cu unitate.

Demonstrație. (1) Se utilizează din plin faptul că $x^2 = x$, pentru orice x , și faptul că inelele Booleene sunt comutative (Propoziția 5.1.9(2)). Noi vom verifica, drept exemplu, doar una din proprietățile de distributivitate din definiția algebrelor Booleene. Astfel, vom arăta că este satisfăcută relația

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z),$$

pentru orice $x, y, z \in R$. Are loc

$$x \vee (y \wedge z) = x + yz - xyz$$

și

$$\begin{aligned} (x \vee y) \wedge (x \vee z) &= (x + y - xy)(x + z - xz) \\ &= x^2 + xz - x^2z + yx + yz - yxz - yxy - xyz + xyxz \\ &= x + y - xyz, \end{aligned}$$

ceea ce stabilește proprietatea cerută.

(2) Ca și în cazul proprietății precedente, vom verifica doar una din cerințele de inel Boolean cu unitate, și anume faptul că 0 este unitate aditivă a inelului. Are loc

$$x + 0 = (x \wedge 0') \vee (x' \wedge 0) = (x \wedge 1) \vee 0 = x \vee 0 = x,$$

pentru orice $x \in R$. \square

Observația 5.1.1. Pentru orice mulțime A , $(\mathcal{P}(A), \cup, \cap, ', \emptyset, A)$, unde $'$ este operația de complementariere a mulțimilor în raport cu A , este algebră Booleană (a se vedea și Exemplul 1.4.5.1(1)). Construcția din Teorema 5.1.1(2) asociază acestei algebrelle Booleene inelul Boolean cu unitate din Exemplul 5.1.2.

5.2. Homomorfisme, subinele și ideale

Reamintim că homomorfismele de inele sunt cazuri particulare de homomorfisme de algebrelle de același tip (Secțiunea 1.4.4.3). Înținând cont de particularitățile homomorfismelor de grupuri, putem spune că $h : R_1 \rightarrow R_2$ este *homomorfism de inele* dacă au loc următoarele proprietăți:

- $h(a+b) = h(a)+h(b)$, pentru orice $a, b \in R_1$ (h este homomorfism între grupurile aditive ale celor două inele);
- $h(ab) = h(a)h(b)$, pentru orice $a, b \in R_1$ (h este homomorfism între semigrupurile multiplicative ale celor două inele)

(operațiile binare ale celor două inele au fost notate la fel). Este ușor de văzut că proprietatea a doua poate fi cerută doar pentru $a, b \in R_1 - \{0\}$. În adevăr, dacă unul din cele două elemente este 0, de exemplu b , atunci

$$h(a0) = h(0) = 0 = h(a)0 = h(a)h(0).$$

Dacă R_1 și R_2 au și unități, e_1 și, respectiv, e_2 , atunci este necesar să fie satisfăcută și relația:

- $h(e_1) = e_2$

(pentru a asigura faptul că h este homomorfism între monoidii multiplicativi ai celor două inele). Dacă însă R_1 și R_2 sunt inele cu diviziune, atunci această cerință nu este necesară, deoarece în acest caz $R_1 - \{0\}$ și $R_2 - \{0\}$ formează grupuri multiplicativi, iar cerința

- $h(ab) = h(a)h(b)$, pentru orice $a, b \in R_1 - \{0\}$,

stabilăște un homomorfism între aceste grupuri.

Propoziția 5.2.1. Orice homomorfism între două inele cu diviziune este injectiv.

Demonstrație. Fie R_1 și R_2 inele cu diviziune și h un homomorfism de la R_1 la R_2 . Fie $a, b \in R_1$ astfel încât $h(a) = h(b)$, și fie $c = a - b$. Atunci

$$h(c) = h(a) - h(b) = 0.$$

Ca urmare, dacă presupunem că $c \neq 0$, atunci

$$h(e_1) = h(cc') = h(c)h(c') = 0,$$

ceea ce constituie o contradicție (e_1 este unitatea lui R_1 , iar c' este inversul lui c în R_1). \square

Apelând la Secțiunea 1.4.4.2, pentru ca o submulțime S a unui inel R să fie închisă în R sau să fie subinel al lui R este suficient să cerem:

- $0 \in S$ și $a - b \in S$, pentru orice $a, b \in S$ (ca urmare, S formează subgrup al grupului aditiv al lui R);
- $ab \in S$, pentru orice $a, b \in S$ (deci, S formează subsemigrup al semigrupului multiplicativ al lui R).

Este ușor de văzut că în cea de a doua cerință putem considera $a, b \neq 0$. Dacă R este inel cu unitate, atunci închiderea mulțimii S în R cere ca și unitatea lui R să fie în S , iar dacă R este inel cu diviziune, atunci cea de a doua cerință trebuie schimbată cu “ $e \in S$ și $ab^{-1} \in S$, pentru orice $a, b \in S$ diferite de 0”.

Intersecția oricărei familii nevide de subinele ale unui inel, este subinel în acel inel.

Definiția 5.2.1. Fie R un inel și $J \subseteq R$.

(1) J este numită *ideal stâng* în R dacă au loc proprietățile:

- J este subgrup al grupului aditiv al lui R ;
- $RJ \subseteq J$.

(2) J este numită *ideal drept* în R dacă au loc proprietățile:

- J este subgrup al grupului aditiv al lui R ;
- $JR \subseteq J$.

(3) J este numită *ideal* în R , și notăm $J \triangleleft R$, dacă J este atât ideal stâng, cât și ideal drept în R .

În orice inel R , $\{0\}$ și R sunt ideale (stângi, drepte). Idealul $\{0\}$ mai este numit și *idealul nul* sau *trivial* și este notat, în mod ușual, tot prin 0. Un ideal diferit de 0 și R este numit *ideal propriu* al inelului R . Idealele (stângi, drepte) generate de un singur element a sunt numite *ideale principale (stângi, drepte)*.

Observația 5.2.1. Fie R un inel.

- (1) Orice ideal (stâng, drept) al lui R este subinel în R . În adevăr, dacă J este ideal stâng, de exemplu, atunci relația $RJ \subseteq J$ conduce la $J^2 \subseteq J$ care, combinată cu faptul că J este subgrup al grupului aditiv al inelului, ne arată că J este subinel în R .
- (2) Dacă R este inel cu unitate, atunci nu este garantat faptul că orice ideal (stâng, drept) al acestuia conține unitatea inelului. Ca urmare, în acest caz nu este garantat faptul că orice ideal (stâng, drept) al inelului este subinel al acestuia. Dar dacă un ideal (stâng, drept) al inelului conține și unitatea inelului, atunci acesta coincide cu inelul.
- (3) Dacă R este comutativ, orice ideal stâng în R este și ideal drept în R , și reciproc.

Exemplul 5.2.1.

- (1) Dacă $h : R_1 \rightarrow R_2$ este homomorfism de inele, atunci $\{a \in R_1 | h(a) = 0\}$ este ideal în R_1 , iar $h(R_1)$ este subinel în R_2 .
- (2) \mathbf{Z} nu este ideal în \mathbf{Q} deoarece $1 \in \mathbf{Z}$, $1/2 \in \mathbf{Q}$, dar $(1/2) \cdot 1 = 1/2 \notin \mathbf{Z}$.
- (3) Fie R un inel comutativ. Atunci, pentru orice $a \in R$, $S = \{ra | r \in R\}$ este ideal în R .
- (4) Fie R un inel comutativ și $a \in R$. Este ușor de văzut că idealul generat de a este dat prin

$$\langle a \rangle = \{ra + na | r \in R, n \in \mathbf{Z}\},$$

iar dacă R are și unitate, atunci

$$\langle a \rangle = \{ra | r \in R\}.$$

Copurile nu conțin ideale proprii.

Propoziția 5.2.2. Fie R un inel comutativ cu unitate. Atunci R este corp dacă și numai dacă singurele lui ideale sunt 0 și R .

Demonstrație. Presupunem că R este corp și fie J un ideal al lui. Dacă $J \neq \{0\}$ atunci vom arăta că $J = R$. Fie $a \in J - \{0\}$. Cum R este corp, a admite invers în R , fie acesta a' .

Pentru orice $b \in R$ are loc

$$b = b \cdot e = b(a'a) = (ba')a \in J.$$

Ca urmare, $R \subseteq J$ și, deci, $R = J$.

Reciproc, fie $a \in R - \{0\}$. Mulțimea $J = \{ba | b \in R\}$ este ideal în R (Exemplul 5.2.1(3)). Conform ipotezei, $J = R$ (J fiind nevidă). Ca urmare, unitatea lui R este

element al idealului J , ceea ce arată că există $b \in R$ cu proprietatea $ba = e$. Cum R este comutativ, aceasta conduce la existența inversului lui a . Deci, R este corp. \square

Orice ideal J într-un inel R este subgrup normal în grupul aditiv $(R, +, -, 0)$ al lui R . Aceasta pentru că J este subgrup al acestui grup, iar comutativitatea grupului $(R, +, -, 0)$ conduce imediat la relația $J = a + J + (-a)$, pentru orice $a \in R$. Ca urmare, J induce o relație de echivalență pe R , notată \sim_J și dată prin

$$a \sim_J b \Leftrightarrow (\exists c \in J)(b = a + c),$$

pentru orice $a, b \in R$. Mai mult, această echivalență este chiar congruență. În adevăr, să presupunem că are loc $a \sim_J b$ și $c \sim_J d$. Atunci, există $x, y \in J$ astfel încât $b = a + x$ și $d = c + y$. Obținem:

- $b + d = (a + x) + (c + y) = (a + c) + (x + y)$, ceea ce implică $a + c \sim_J b + d$;
- $bd = ac + (ay + xc + xy)$, ceea ce implică $ac \sim_J bd$ (J fiind ideal, elementele ay , xc și xy aparțin acestuia);
- $-b = -a - x$, ceea ce arată că $-a \sim_J -b$.

Acestea ne arată că, în adevăr, \sim_J este congruență în inel.

Putem atunci discuta despre *inelul cât* R/\sim_J care se notează în mod ușual prin R/J . Clasele de echivalență sunt de forma $[a] = a + J$, pentru orice $a \in R$. Operațiile inelului cât sunt:

- $(a + J) + (b + J) = (a + b) + J$;
- $0 + J = J$ este elementul zero;
- $-(a + J) = (-a) + J$ este opusul lui $a + J$;
- $(a + J)(b + J) = (ab) + J$,

pentru orice $a, b \in R$.

Dacă R este inel comutativ sau are unitate, atunci aceleși proprietăți sunt valabile și pentru R/J .

Exemplul 5.2.2. Vom exemplifica prin construcția lui \mathbf{Z}_n ca inel cât. Fie $n \in \mathbf{Z}$. Idealul principal generat de n este

$$\langle n \rangle = \{kn | k \in \mathbf{Z}\}.$$

Clasele de echivalență induse de $\sim_{\langle n \rangle}$ sunt de forma

$$[a] = a + \langle n \rangle = \{a + kn | k \in \mathbf{Z}\},$$

pentru orice $a \in \mathbf{Z}$.

Observăm că $[a] = [r]$, unde $a = qn + r$, $q \in \mathbf{Z}$ și $0 \leq r < n$. Ca urmare,

$$\mathbf{Z}/\sim_{\langle n \rangle} = \{[a] | 0 \leq a < n\}.$$

Idealele lui \mathbf{Z} se găsesc printre subgrupurile grupului aditiv \mathbf{Z} , care sunt de forma $n\mathbf{Z}$, cu $n \geq 0$. Însă e ușor de observat că orice subgrup $n\mathbf{Z}$ este ideal în \mathbf{Z} , ceea ce conduce la faptul că idealele lui \mathbf{Z} sunt $n\mathbf{Z}$ cu $n \geq 0$, și toate sunt principale.

Propoziția 5.2.3. Fie R un inel și J un ideal în R . Atunci există o funcție bijectivă între idealele inelului R/J și idealele ce includ J ale inelului R . În plus, această funcție este strict monotonă în raport cu incluziunea.

Demonstrație. Considerăm funcția f ce asociază idealului \bar{J} al lui R/J idealul $\{a|[a] \in \bar{J}\}$ al lui R ce include J .

Injectivitatea funcției f rezultă cu ușurință în baza faptului că R/J este partiție a lui R . Ca urmare, dacă \bar{J}_1 și \bar{J}_2 sunt ideale distințe ale lui R/J , atunci $f(\bar{J}_1)$ și $f(\bar{J}_2)$ vor fi distințe.

Dacă J' este ideal al lui R ce include J , atunci $\bar{J} = \{[a]|a \in J'\}$ este ideal al lui R/J , ceea ce se verifică cu ușurință, și $f(\bar{J}) = J'$.

Stricta monotonie în raport cu incluziunea a funcției f se obține direct de la definiția acesteia. \square

Definiția 5.2.2. Un ideal J al unui inel R este numit *maximal* dacă $J \neq R$ și, pentru orice ideal J' al lui R , relația $J \subseteq J' \subseteq R$ conduce la $J' = J$ sau $J' = R$.

Observăm că dacă R nu are ideale proprii, atunci 0 este ideal maximal în R .

Corolarul 5.2.1. Fie R un inel comutativ cu unitate și J un ideal în R . Atunci, R/J este corp dacă și numai dacă J este ideal maximal în R .

Demonstrație. Să presupunem că R/J este corp. Dacă presupunem că J nu este ideal maximal în R , atunci există un ideal J' în R astfel încât $J \subset J' \subset R$. Însă aceasta ne spune că R/J are ideale proprii (Propoziția 5.2.3), ceea ce contrazice Propoziția 5.2.2. Reciproc, demonstrația decurge în mod similar. \square

În mod similar grupurilor, se pot demonstra următoarele teoreme de izomorfism pentru inele. Dar, ca și în cazul grupurilor, atragem atenția asupra faptului că nucleul unui homomorfism este definit, în lucrarea noastră, diferit de alte lucrări cu care cititorul ar putea lăua cunoștință. Aceasta nu constituie o problemă atâtă timp cât concepțile sunt manipulate cu atenție.

Teorema 5.2.1. (Prima teoremă de izomorfism)

- (1) Fie R_1 și R_2 inele, iar $h : R_1 \rightarrow R_2$ un epimorfism. Atunci $R_1/\ker(h)$ și R_2 sunt inele izomorfe.
- (2) Fie R un inel și $J \triangleleft R$. Atunci funcția $h : R \rightarrow R/J$ dată prin $h(a) = a + J$, pentru orice $a \in R$, este epimorfism. În plus, $\ker(h) = \sim_J$.

Teorema 5.2.2. (A două teoremă de izomorfism)

Fie R un inel, $S \leq R$ și $J \triangleleft R$. Atunci $S \cap J \triangleleft S$ și $J \triangleleft S + J$, iar inelele $S/(S \cap J)$ și $(S + J)/J$ sunt izomorfe.

Teorema 5.2.3. (A treia teoremă de izomorfism)

Fie R un inel, $J_1 \triangleleft R$ și $J_2 \triangleleft R$ astfel încât $J_1 \subseteq J_2$. Atunci $J_1 \triangleleft J_2$ și $J_2/J_1 \triangleleft R/J_1$, iar inelele $(R/J_1)/(J_2/J_1)$ și R/J_2 sunt izomorfe.

5.3. Caracteristica unui inel

Conceptul de caracteristică este central în teoria inelelor și corporilor, aşa cum se va vedea mai departe.

Definiția 5.3.1. Spunem că un inel R are *caracteristica* $n \geq 1$, dacă n este cel mai mic număr natural strict pozitiv ce satisfacă $na = 0$, pentru orice $a \in R$ (atunci când există un astfel de număr).

Dacă nu există $n \geq 1$ astfel încât $na = 0$ pentru orice $a \in R$, atunci vom spune că inelul R are *caracteristica zero*.

Vom nota prin $\text{char}(R)$ caracteristica inelului R .

Observația 5.3.1.

- (1) Fie $(R, +, -, 0, \cdot)$ un inel și $a \in R$. Presupunem că ordinul lui a în grupul aditiv $(R, +, -, 0)$ este $k \geq 1$. Adică, k este cel mai mic număr natural strict pozitiv ce satisfacă $ka = 0$. Dacă $\text{char}(R) = n \geq 1$, atunci $na = 0$ și, deci, $k|n$ (Teorema 4.4.2(3)).

Că urmare, caracteristica lui R , atunci când nu este zero, este cel mai mic multiplu comun al ordinelor elementelor inelului (relativ la operația notată aditiv).

- (2) Caracteristica unui inel nenul nu poate fi 1 deoarece $1a \neq 0$, pentru orice $a \neq 0$ din inel. Deci, singurul inel de caracteristică 1 este inelul nul.
- (3) Dacă un inel R are caracteristica 2, atunci are loc $2a = a + a = 0$, pentru orice $a \in R$, ceea ce conduce la $-a = a$.

Exemplul 5.3.1.

- (1) Pentru orice $m \geq 1$, \mathbf{Z}_m are caracteristica m deoarece m este cel mai mic număr natural nenul ce satisfacă $ma = 0$, pentru orice $a \in \mathbf{Z}_m$.
- (2) \mathbf{Z} este domeniu de integritate de caracteristică zero.

(3) **Q, R și C** sunt corpuri de caracteristică zero.

Teorema 5.3.1. Fie $(R, +, -, 0, \cdot, e)$ un inel cu unitate de caracteristică $n \geq 1$.

(1) n este cel mai mic număr strict pozitiv ce satisface $ne = 0$.

(2) Dacă R este nenul și nu are divizori ai lui zero, atunci n este prim.

Demonstrație. (1) Presupunem că $k \geq 1$ este cel mai mic număr natural cu proprietatea $ke = 0$. Atunci, pentru orice $a \in R$, are loc

$$ka = k(ea) = (ke)a = 0a = 0,$$

ceea ce ne arată că $k = n$ deoarece $\text{char}(R) = n$.

(2) În primul rând observăm că n nu poate fi 1 deoarece inelul este nenul. Deci, $n \geq 2$. Dacă n nu ar fi prim, atunci n s-ar scrie $n = rs$ cu $1 < r, s < n$. Are loc:

$$0 = ne = rse = (rs)(ee) = (re)(se),$$

ceea ce conduce la $re = 0$ sau $se = 0$ deoarece R este fără divizori ai lui zero. Atunci, de la (1), urmează că, caracteristica lui R este r sau s , ceea ce constituie o contradicție. \square

Corolarul 5.3.1. Dacă caracteristica unui domeniu de integritate nu este zero, atunci ea este un număr prim.

Demonstrație. Direct de la Teorema 5.3.1. \square

Corolarul 5.3.2. Orice corp finit are caracteristica un număr prim.

Demonstrație. Este suficient de arătat că nici un corp finit nu are caracteristica zero.

Fie R un corp finit. Considerăm secvența

$$e, 2e, 3e, \dots$$

Deoarece R este finit rezultă că există i și j distințe astfel încât $ie = je$. Presupunem $i < j$. Atunci $(j - i)e = 0$, ceea ce ne arată că, caracteristica lui R este nenulă (remarcăm că $j - i > 0$). \square

Utilizând rezultatele de mai sus putem redemonstra că \mathbf{Z}_p este corp dacă și numai dacă p este prim.

Observația 5.3.2. Dacă caracteristica unui inel cu unitate și fără divizori ai lui zero este $n \geq 1$, atunci n este număr prim. Aceasta ne spune că toate elementele diferite de 0 ale inelului au același ordin aditiv, care este n .

Fie $(R, +, -, 0, \cdot, e)$ un corp. Intersecția tuturor subcorpilor lui R este subcorp, numit *subcorpul prim* al lui R , notat P_R .

Să presupunem că R are caracteristica p . Atunci $p \geq 2$ este număr prim. P_R conține 0 și e și, ca urmare, conține

$$0, 1e = e, 2e, \dots, (p-1)e.$$

Mai mult, mulțimea $\{0, 1e, 2e, \dots, (p-1)e\}$ formează subcorp al lui R (ceea ce este ușor de văzut) și, ca urmare, ea este chiar P_R . Putem spune astfel că R include o “copie” a lui \mathbf{Z}_p .

Să presupunem acum că R are caracteristica zero. Atunci, P_R conține ne , pentru orice $n \in \mathbf{Z}$, și $(ne)(me)'$, pentru orice $n, m \in \mathbf{Z}$ cu $m \neq 0$. Mai mult, P_R este exact mulțimea elementelor de forma $(ne)(me)'$, cu $n, m \in \mathbf{Z}$ și $m \neq 0$. Ca urmare, R include o “copie” a lui \mathbf{Q} .

Am obținut astfel următorul rezultat foarte important.

Teorema 5.3.2. Fie $(R, +, -, 0, \cdot, e)$ un corp.

- (1) Dacă R are caracteristica un număr p , atunci există un unic izomorfism între P_R și corpul \mathbf{Z}_p .
- (2) Dacă R are caracteristica zero, atunci există un unic izomorfism între P_R și corpul \mathbf{Q} .

Demonstrație. Vom demonstra (1), (2) obținându-se în manieră similară.

Este ușor de văzut că $P_R = \{ne | 0 \leq n < p\}$. Funcția $h : \mathbf{Z}_p \rightarrow P_R$ dată prin $h(n) = ne$, pentru orice $n \in \mathbf{Z}_p$, este injectivă (Propoziția 5.2.1), surjectivă, și satisface proprietățile de homomorfism:

$$h(n+m) = (n+m)e = ne + me = h(n) + h(m),$$

și

$$h(nm) = nme = nmee = (ne)(me) = h(n)h(m),$$

pentru orice $n, m \in \mathbf{Z}_p$. Deci, h este izomorfism între \mathbf{Z}_p și P_R .

Dacă h' este un alt izomorfism de la \mathbf{Z}_p la P_R , atunci $h'(0) = 0 = h(0)$ și $h'(1) = 1e = h(1)$. Atunci, pentru orice $n \in \mathbf{Z}$ cu $n > 1$,

$$h'(n) = h'\underbrace{(1 + \dots + 1)}_{n \text{ ori}} = \underbrace{h'(1) + \dots + h'(1)}_{n \text{ ori}} = \underbrace{h(1) + \dots + h(1)}_{n \text{ ori}} = h(n),$$

ceea ce stabilește unicitatea izomorfismului h . \square

Teorema 5.3.3. Fie R un inel comutativ de caracteristică număr prim p . Atunci:

- (1) $(a+b)^{p^n} = a^{p^n} + b^{p^n}$, pentru orice $a, b \in R$ și $n \in \mathbf{N}$;
- (2) $(a-b)^{p^n} = a^{p^n} - b^{p^n}$, pentru orice $a, b \in R$ și $n \in \mathbf{N}$.

Demonstrație. (1) Deoarece $C_p^i \equiv 0 \pmod{p}$, pentru orice $0 < i < p$, obținem

$$(a+b)^p = a^p + C_p^1 a^{p-1} b + \dots + C_p^{p-1} a b^{p-1} + b^p = a^p + b^p.$$

Vom face inducție după $n \geq 0$. Pentru $n = 0$ și $n = 1$ formula este trivial satisfăcută. Pentru $n > 1$ are loc

$$(a+b)^{p^n} = ((a+p)^{p^{n-1}})^p = (a^{p^{n-1}} + b^{p^{n-1}})^p = a^{p^n} + b^{p^n}$$

(ultimele două egalități sunt conform ipotezei inducțive).

(2) Are loc

$$a^{p^n} = ((a-b)+b)^{p^n} = (a-b)^{p^n} + b^{p^n},$$

de unde urmează relația din teoremă. \square

Vom încheia secțiunea anticipând câteva concepte care vor fi reluate într-un cadru mai larg în capitolul dedicat spațiilor vectoriale.

Fie K un subcorp al unui corp R . Un element de forma

$$a = k_1 a_1 + \dots + k_n a_n,$$

unde $k_i \in K$ și $a_i \in R$, pentru orice $1 \leq i \leq n$, va fi numită *combinație liniară* a elementelor a_1, \dots, a_n cu coeficienți din K .

Elementele a_1, \dots, a_n sunt numite *liniar independente peste K* dacă pentru orice combinație liniară $k_1 a_1 + \dots + k_n a_n$ a lor are loc

$$k_1 a_1 + \dots + k_n a_n = 0 \Rightarrow (\forall 1 \leq i \leq n)(k_i = 0).$$

Altfel spus, nici un element a_i nu poate fi scris ca o combinație liniară a celorlalte elemente $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n$ în care cel puțin un coeficient din K să fie nenul.

Este important de remarcat că un element $a \in R$ nu poate fi scris în două moduri distințe ca o combinație liniară de a_1, \dots, a_n dacă a_1, \dots, a_n sunt liniar independente. În adevăr, dacă $a \in R$ să ar scrie

$$a = \sum_{i=1}^n k_i a_i = \sum_{i=1}^n l_i a_i,$$

atunci am obține

$$\sum_{i=1}^n (k_i - l_i) a_i = 0$$

care ar conduce la $k_i = l_i$, pentru orice i (în baza liniar independenței elementelor a_1, \dots, a_n).

Dacă a_1, \dots, a_n sunt liniar independente peste K și orice element din R se scrie ca o combinație liniară a acestor elemente, atunci vom spune că mulțimea $B = \{a_1, \dots, a_n\}$ formează o bază (finită) peste K pentru R . Nu este greu de arătat că dacă B și B' sunt

două baze (finite) pentru R (ambele peste K), atunci $|B| = |B'|$ (demonstrația poate fi făcută ad hoc urmând aceeași idee ca în demonstrația Lemei 6.2.1 și Corolarului 6.2.1).

Dacă există o bază B pentru R peste K , atunci numărul $|B|$ poartă denumirea de *dimensiunea corpului R peste K* și se notează prin $[R : K]$. Dacă R este finit, atunci K este finit și $|R| = |K|^{[R : K]} = |K|^{|B|}$ (orice element din R se scrie unic ca o combinație liniară a elementelor bazei cu coeficienți din K , iar fiecare coeficient poate fi ales în $|K|$ moduri).

Drept consecință a acestei discuții obținem:

Teorema 5.3.4. Fie R un corp finit de caracteristică p . Atunci, există $n \geq 1$ astfel încât ordinul lui R este p^n .

Demonstrație. Subcorpul prim al lui R are exact p elemente. Atunci, $|R| = p^n$, unde $n = [R : P_R]$. \square

Fie K un subcorp al unui corp L care este subcorp al corpului R . Presupunem că $[L : K] = m$ și $[R : L] = n$. Fie $B_1 = \{a_1, \dots, a_m\}$ o bază în L peste K și $B_2 = \{b_1, \dots, b_n\}$ o bază în R peste L . Orice element $a \in R$ se poate scrie în mod unic în forma

$$a = \sum_{i=1}^n \beta_i b_i,$$

unde $\beta_i \in L$. Cum fiecare β_i poate fi scris în forma $\beta_i = \sum_{j=1}^m \alpha_{ij} a_j$, obținem:

$$a = \sum_{i,j} \alpha_{ij} a_j b_i.$$

În plus, este imediat de verificat că mulțimea $B = \{a_j b_i \mid 1 \leq i \leq m, 1 \leq j \leq n\}$ este liniar independentă. Ca urmare, am obținut următorul rezultat:

Teorema 5.3.5. Fie K un subcorp al corpului L care este subcorp al corpului R . Dacă $[L : K] = m$ și $[R : L] = n$, atunci $[R : K] = mn$.

5.4. Inele de polinoame

Polinoamele peste un inel R pot fi introduse, în principal, în două moduri echivalente: ca secvențe infinite de elemente peste R , sau ca expresii formale sintactice construite cu elemente din R și elemente distințe numite variabile. În prezentarea noastră vom urma cea de a doua direcție. Menționăm încă de la început că nu vom intra în detaliu asupra chestiunilor elementare de teoria polinoamelor (adunare, înmulțire, împărțire,

cel mai mare divizor comun etc.), chestiuni pentru care cititorul este îndrumat către manualele de liceu.

5.4.1. Polinoame. Proprietăți de bază

Fie R un inel arbitrar. Un *polinom* peste R este o expresie de forma

$$f(x) = a_0 + a_1x + \cdots + a_nx^n,$$

unde $n \geq 0$, $a_i \in R$ pentru orice $0 \leq i \leq n$, iar x este un element ce nu este în R . Elementele a_i sunt numite *coeficienții polinomului*, iar x , *necunoscuta sau variabilă* acestuia¹. Simplificat, polinomul $f(x)$ se mai notează prin $f(x) = \sum_{i=0}^n a_i x^i$. Mai mult, atunci când necunoscuta x se subînțelege din context ne vom referi la $f(x)$ doar prin f . Vom nota prin $R[x]$ mulțimea tuturor polinoamelor în necunoscuta x și cu coeficienți în R .

Facem convenția ca în expresia polinomului $f(x)$ să nu scriem termii $a_i x^i$ pentru care $a_i = 0$. Cu aceasta, polinomul $f(x)$ poate fi scris și în forma

$$f(x) = a_0 + a_1x + \cdots + a_nx^n + 0x^{n+1} + \cdots + 0x^{n+h},$$

pentru orice $h \geq 0$. Această convenție este importantă prin aceea că ori de câte ori vom avea de comparat două polinoame putem presupune că ele au același număr de termeni. Astfel, vom spune că polinoamele $f(x) = \sum_{i=0}^n a_i x^i$ și $g(x) = \sum_{i=0}^m b_i x^i$ sunt *egale*, și notăm $f(x) = g(x)$, dacă $a_i = b_i$, pentru orice i .

Definim *suma* a două polinoame $f(x) = \sum_{i=0}^n a_i x^i$ și $g(x) = \sum_{i=0}^m b_i x^i$ ca fiind polinomul notat $f(x) + g(x)$ și dat prin

$$f(x) + g(x) = \sum_{i=0}^n (a_i + b_i)x^i.$$

Produsul a două polinoame $f(x) = \sum_{i=0}^n a_i x^i$ și $g(x) = \sum_{i=0}^m b_i x^i$, notat $f(x)g(x)$, este polinomul

$$f(x)g(x) = \sum_{k=0}^{n+m} c_k x^k,$$

unde

$$c_k = \sum_{0 \leq i \leq n, 0 \leq j \leq m, i+j=k} a_i b_j,$$

pentru orice $0 \leq k \leq n+m$.

$R[x]$, cu adunarea și înmulțirea polinoamelor, formează inel. Elementul zero este *polinomul zero* sau *nul* al cărui coeficienți sunt toți 0, iar *opusul* unui polinom $f(x) =$

¹Nu trebuie confundată expresia formală $a_0 + a_1x + \cdots + a_nx^n$ ce definește $f(x)$ ca funcție polomială asociată ce va fi discutată ulterior (dar pe care cititorul probabil că o cunoaște).

$\sum_{i=0}^n a_i x^i$ este polinomul $-f(x) = \sum_{i=0}^n (-a_i)x^i$. Dacă R are unitate, atunci $R[x]$ are *polinom unitate*. Polinoamele nul și unitate, cel de-al doilea atunci când există, vor fi notate prin 0, respectiv, 1.

Definiția 5.4.1.1. Spunem că un polinom nenul $f(x) = \sum_{i=0}^n a_i x^i \in R[x]$ are *gradul* n dacă $a_n \neq 0$.

Prin convenție, *gradul polinomului nul* este $-\infty$, unde $-\infty$ verifică proprietățile:

- $-\infty < n$, și
- $-\infty + n = -\infty - n = -\infty - \infty = -\infty$,

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

Notăm gradul polinomului f prin $\text{grad}(f)$. Polinoamele de grad 0 sau $-\infty$ sunt numite și *polinoame constante*². Dacă identificăm elementele $a \in R$ cu polinoamele constante $a \in R[x]$, atunci R poate fi văzut ca un subinel al inelului $R[x]$.

Dacă R are unitate și un polinom f de grad n are coeficientul a_n egal cu unitatea lui R , atunci vom spune că f este un *polinom monic*.

Următoarele proprietăți sunt imediat de verificat.

Propoziția 5.4.1.1. Fie R un inel și $f, g \in R[x]$. Atunci:

- (1) $\text{grad}(f + g) \leq \max\{\text{grad}(f), \text{grad}(g)\}$;
- (2) $\text{grad}(fg) \leq \text{grad}(f) + \text{grad}(g)$. Dacă R este domeniu de integritate, atunci inegalitatea are loc prin egalitate.

Propoziția 5.4.1.2. Fie R un inel. Atunci au loc următoarele proprietăți:

- (1) $R[x]$ este inel comutativ dacă și numai dacă R este inel comutativ;
- (2) $R[x]$ este inel cu unitate dacă și numai dacă R este inel cu unitate;
- (3) $R[x]$ este domeniu de integritate dacă și numai dacă R este domeniu de integritate.

Observația 5.4.1. Dacă R este corp, grupul unităților lui $R[x]$ este format din exact toate polinoamele constante nenule. Ca urmare, $R[x]$ nu este corp.

Vom fi interesați în principal de polinoame cu coeficienți într-un corp și, ca urmare, în ceea ce urmează vom considera că F desemnează un corp arbitrar³. Unitatea corpului F va fi întotdeauna notată prin 1.

²Separarea polinomului nul de celelalte polinoame constante și atribuirea acestuia un grad mai mic decât 0 este făcută din rațiuni tehnice.

³Unele din proprietățile ce vor urma pot avea loc cerând ca F să satisfacă condiții mai slabe decât cele de corp.

Definiția 5.4.1.2. Spunem că polinomul $f \in F[x]$ divide polinomul $g \in F[x]$, și notăm $f|g$, dacă există $h \in F[x]$ astfel încât $g = fh$.

Dacă $f|g$, atunci vom mai spune că f este divizor al lui g sau că g este multiplu al lui f .

Ca și în cazul lui \mathbf{Z} , putem defini

$$f \equiv g \pmod{h}$$

dacă și numai dacă $h|(f - g)$, pentru orice polinoame f, g și h . Dacă $f \equiv g \pmod{h}$ atunci vom spune că f și g sunt congruente modulo h .

Teorema 5.4.1.1. (Teorema împărțirii cu rest pentru polinoame)

Fie g un polinom nenul în $F[x]$. Atunci, pentru orice $f \in F[x]$, există două polinoame $q, r \in F[x]$ astfel încât $f = qg + r$ și $\text{grad}(r) < \text{grad}(g)$. În plus, q și r sunt unice cu aceste proprietăți.

Demonstrație. Prin inducție după gradul polinomului f . \square

Observația 5.4.1.1.

- (1) Fie F un corp. Pentru orice polinom nenul $f \in F[x]$, idealul $J = \langle f \rangle$ este dat prin

$$J = \langle f \rangle = \{\alpha f | \alpha \in F\}.$$

Atunci $g \sim_J h$ dacă și numai dacă $g = h + \alpha f$, unde $\alpha \in F$, ceea ce este echivalent cu $f|g - h$. Ca urmare, $g \sim_J h$ dacă și numai dacă $g \equiv h \pmod{f}$. Deci clasele de echivalență induse de J în $F[x]$ sunt formate din polinoame congruente modulo f .

Orice clasă de echivalență $g + \langle f \rangle$ conține un polinom r cu $\text{grad}(r) < \text{grad}(f)$ (obținut prin împărțirea lui g la f). În plus, r este unic cu această proprietate. În adevăr, dacă ar exista $r' \in g + \langle f \rangle$ cu $\text{grad}(r') < \text{grad}(f)$, atunci $r - r'$ ar fi divizibil prin f (deoarece r și r' sunt în aceeași clasă de echivalență indusă de f), iar relația $\text{grad}(r - r') < \text{grad}(f)$ ar conduce la $r = r'$. Ca urmare, clasele de echivalență induse de $\langle f \rangle$ sunt exact clasele ai căror reprezentanți de clasă sunt polinoame de grad mai mic decât f cu coeficienți din F .

- (2) Cititorul este invitat să compare construcția lui $F[x]/\langle f \rangle$ (așa cum este prezentată la (1)) cu construcția lui \mathbf{Z}_n (Exemplul 5.2.2).

- (3) Considerând $F = \mathbf{Z}_p$ și $f \in F[x]$ de grad n , există exact p^n polinoame de grad mai mic ca f și, deci, $\mathbf{Z}_p/\langle f \rangle$ are exact p^n elemente. De exemplu, considerând $f(x) = x^2 + x + 1 \in \mathbf{Z}_2[x]$, atunci $\mathbf{Z}_2[x]/\langle f \rangle$ are 2^2 elemente. Acestea sunt $[0], [1], [x]$ și $[x + 1]$. Operațiile cu aceste elemente se obțin aplicând operațiile corpului $\mathbf{Z}_2[x]$ asupra reprezentanților de clasă și apoi aplicând reducerea modulo $f(x)$. Astfel,

$$[x] \cdot [x] = [x^2 \pmod{f(x)}] = [x + 1].$$

Pentru adunare este suficient să se aplique adunarea în \mathbf{Z}_2 coeficienților polinoamelor. De exemplul

$$[x] + [x + 1] = [(1 + 1 \pmod{2})x + (1 + 0 \pmod{2})] = [1].$$

Este însă interesant de remarcat că $F[x]$ are numai ideale principale.

Teorema 5.4.1.2. Orice ideal nenul J în $F[x]$ este ideal principal generat de un unic polinom monic f . În plus, f este de grad minim în J .

Demonstrație. Fie J un ideal nenul în $F[x]$ și g un polinom nenul de grad minim în J . Fie a coeficientul termenului de grad maxim al lui g . Atunci $f(x) = a^{-1}g(x)$ este un polinom monic în J . Vom arăta că f satisface teorema.

Fie $h(x)$ un polinom din J . Conform teoremei împărțirii cu rest pentru polinoame, există $q, r \in F[x]$ astfel încât $h = fq + r$ și $\text{grad}(r) < \text{grad}(f) = \text{grad}(g)$. Deoarece J este ideal, $h - fq \in J$ și, ca urmare, modul de alegere al polinomului f conduce la $r = 0$. Ca urmare, $h = fq$, ceea ce ne arată că $J = \langle f \rangle$.

Unicitatea lui f se obține astfel. Dacă f' este un alt polinom monic din J cu $J = \langle f' \rangle$, atunci $f = f'h_1$ și $f' = fh_2$, unde h_1 și h_2 sunt polinoame din $F[x]$. Aceasta conduce la $f = fh_2h_1$, de unde urmează $h_2h_1 = 1$. Ca urmare, h_1 și h_2 sunt polinoame constante. Cum f și f' sunt polinoame monice, deducem că $h_1 = 1$ și, deci, $f = f'$. \square

Definiția 5.4.1.3. Spunem că un polinom $f \in F[x]$ este ireductibil sau prim peste/ \mathbb{F} dacă $\text{grad}(f) > 0$ iar relația $f = gh$ cu $g, h \in F[x]$ conduce la faptul că g sau h este polinom constant.

Teorema 5.4.1.3. Fie $f \in F[x]$. Dacă f este ireductibil peste $F[x]$, atunci $F[x]/\langle f \rangle$ este corp.

Demonstrație. Fie f un ireductibil peste $F[x]$. Considerăm idealul $\langle f \rangle$ în $F[x]$ și arătăm că acesta este maximal. Atunci, în baza Corolarului 5.2.1 va rezulta că $F[x]/\langle f \rangle$ este corp.

Fie J un ideal în $F[x]$ astfel încât $\langle f \rangle \subseteq J$. Cum $F[x]$ are numai ideale principale (Teorema 5.4.1.2), există un polinom g astfel încât $J = \langle g \rangle$. Atunci, inclusiv $\langle f \rangle \subseteq J$ conduce la $f \in \langle g \rangle$ care este echivalentă cu faptul că există $\alpha \in F$ astfel încât $f = \alpha g$. Cum f este ireductibil, α sau g trebuie să fie polinom constant. Dar atunci $J = F[x]$ (dacă g este polinom constant) sau $J = \langle f \rangle$ (dacă α este polinom constant). Ca urmare, $\langle f \rangle$ este ideal maximal. \square

Fie F un corp, $f(x) = a_0 + a_1x + \dots + a_nx^n$ un polinom în $F[x]$ și $a \in F$. Înlocuind (substituind) x în expresia lui $f(x)$ cu a și efectuând calculele în corpul F , se obține un element din F , și anume

$$a_0 + a_1a + \dots + a_na^n.$$

Acest element este notat prin $f(a)$ (această notație nu poate crea confuzii atât timp cât x este specificat clar prin intermediul lui $F[x]$ și el nu este element din F). Este ușor de văzut că funcția φ_a care asociază unui polinom $f(x) \in F[x]$ elementul $f(a) \in F$ este un homomorfism de inele comutative cu unitate.

Principiul substituibilității sub incidență căruia vom lucra spune că substituind x într-o identitate de polinoame din $F[x]$ rezultă o identitate în F .

Definiția 5.4.1.4. Fie F un corp, $a \in F$ și $f(x) \in F[x]$. $a \in F$ este numit *rădăcină* a polinomului $f(x)$ dacă $f(a) = 0$.

Propoziția 5.4.1.3. Fie F un corp. Un element $a \in F$ este rădăcină a polinomului $f(x) \in F[x]$ dacă și numai dacă $x - a$ divide $f(x)$.

Demonstrație. În baza teoremei împărțirii cu rest pentru polinoame obținem

$$f(x) = q(x)(x - a) + c,$$

unde $c \in F$. Deci $f(a) = c$. Atunci urmează imediat că a este rădăcină a polinomului f dacă și numai dacă $x - a$ divide f . \square

Definiția 5.4.1.5. Fie F un corp. Un element $a \in F$ este numit *rădăcină de multiplicitate* $k \geq 1$ a polinomului $f(x) \in F[x]$ dacă $(x - a)^k | f$.

Dacă a este rădăcină de multiplicitate $k = 1$ a polinomului f , atunci vom mai spune că a este *rădăcină simplă a lui* f , iar dacă $k > 1$, atunci vom mai spune că a este *rădăcină multiplă a lui* f .

În studiul rădăcinilor multiple ale unui polinom, conceptul de derivată este unul din cele mai importante.

Definiția 5.4.1.6. Fie F un corp și $f(x) = a_0 + a_1x + \dots + a_nx^n \in F[x]$ un polinom. Derivata polinomului f , notată f' , este polinomul $f'(x) = a_1 + 2a_2x + \dots + na_nx^{n-1}$ (în cazul $n = 0$, derivata lui f este 0).

Un simplu exercițiu conduce la:

Propoziția 5.4.1.4. Fie F un corp. Un element $a \in F$ este rădăcină multiplă a polinomului $f(x) \in F[x]$ dacă și numai dacă a este rădăcină a lui f și f' .

5.4.2. Extensii, elemente algebrice, descompuneri

Dacă K este un subcorp al unui corp F , atunci vom mai spune că F este o *extensie* a lui K . Vom fi interesanți în cele ce urmează de anumite tipuri de extensii ale unui corp.

Definiția 5.4.2.1. O extensie F a unui corp K este numită *finită* dacă există o bază finită pentru F peste K .

Dacă F este o extensie finită a corpului K , atunci orice element din F se scrie în mod unic ca o combinație liniară a elementelor unei baze B peste K , cu coeficienți în K .

Definiția 5.4.2.2. Fie K un subcorp al unui corp F și $M \subseteq F$. Cel mai mic subcorp al lui F ce include K și M , notat prin $K(M)$, este numit *extensia lui K prin M*.

Evident, $K(M)$ există întotdeauna, fiind intersecția tuturor subcorpuriilor lui F ce includ K și M . Atunci când M este finită, $M = \{a_1, \dots, a_n\}$, $K(M)$ va mai fi notat prin $K(a_1, \dots, a_n)$, iar când $M = \{a\}$, $K(a)$ va mai fi numit *extensie simplă a lui K prin a*.

Definiția 5.4.2.3. Fie K un subcorp al unui corp F și $a \in F$.

- (1) Spunem că a este *algebric peste K* dacă există $f \in K[x]$ astfel încât $f(a) = 0$.
- (2) Spunem că F este *algebric peste K* sau că este o *extensie algebrică a lui K* dacă toate elementele lui sunt algebrice peste K .

Elementele care nu sunt algebrice se numesc *transcendente*. De exemplu, π este transcendental (demonstrația acestui fapt depășește cadrul lucrării noastre).

Am introdus astfel concepțile de extensie, extensie finită, extensie prin M și extensie algebrică a unui corp. Evident, orice extensie algebrică a unui corp K este extensie a lui K . Reciproc, dacă cerem ca o extensie să fie finită, atunci ea este și algebrică.

Propoziția 5.4.2.1. Orice extensie finită a unui corp K este extensie algebrică peste K .

Demonstrație. Fie F o extensie finită a unui corp K , $m = [F : K]$ și $a \in F$. Elementele $1, a, \dots, a^m$ sunt liniar dependente peste K (deoarece numărul acestora depășește dimensiunea lui F peste K). Ca urmare, există $k_i \in K$, $0 \leq i \leq m$, nu toate 0, astfel încât $k_0 + k_1a + \dots + k_ma^m = 0$. Aceasta ne spune că a este algebric peste K deoarece el este rădăcină a polinomului $f(x) = k_0 + k_1x + \dots + k_mx^m$. Deci F este extensie algebrică peste K . \square

Ne vom concentra în cele ce urmează asupra extensiilor algebrice simple.

Observația 5.4.2.1. Fie K un subcorp al unui corp F și $a \in F$ un element algebric peste K . Considerăm multimea

$$J_a = \{f \in K[x] | f(a) = 0\}.$$

Este ușor de văzut că aceasta formează ideal al lui $K[x]$. Ea este nevidă deoarece a este algebraic peste K . Ca urmare, Teorema 5.4.1.2 conduce la existența unui unic polinom monic $f \in J_a$ astfel încât $J_a = \langle f \rangle$. Mai mult, f este de grad minim în J .

Definiția 5.4.2.4. Fie K un subcorp al unui corp F și $a \in F$ un element algebraic peste K . Unicul polinom monic f ce generează idealul J_a este numit *polinomul minimal al lui a peste K* , iar gradul acestuia este numit *gradul lui a peste K* .

Propoziția 5.4.2.2. Fie K un subcorp al unui corp F , $a \in F$ un element algebraic peste K și f polinomul minimal al lui a . Atunci f este ireductibil în $K[x]$.

Demonstrație. Dacă am presupune că polinomul f se poate scrie în forma $f = g_1g_2$, unde $g_1, g_2 \in K[x]$ și $1 \leq \text{grad}(g_1), \text{grad}(g_2) < \text{grad}(f)$, atunci am obținut $f(a) = g_1(a)g_2(a) = 0$, ceea ce ar implica că g_1 sau g_2 este în J_a , contrazicând astfel minimalitatea gradului lui f . \square

Ca urmare, dacă K este un subcorp al unui corp F iar $a \in F$ este un element algebraic peste K , atunci idealul J_a este generat de un unic polinom monic, ireductibil și de grad minim în J_a .

Exemplul 5.4.2.1. Elementul $a = \sqrt{2} + \sqrt{3} \in \mathbf{R}$ este algebraic peste \mathbf{Q} și are gradul 4. În adevăr, polinomul $f(x) = x^4 - 10x^2 + 1 \in \mathbf{Q}[x]$ are pe a ca rădăcină (în \mathbf{R}). El este și ireductibil în $\mathbf{Q}[x]$, ceea ce se poate arăta ușor prin contradicție. Ca urmare, a are gradul 4 peste $\mathbf{Q}[x]$, $f(x)$ fiind polinomul minimal al acestuia.

Fie F un corp, K un subcorp al lui F și $a \in F$ un element algebraic peste K . Vom face în continuare câteva remarcări asupra corpului $K(a)$.

$K(a)$ este cel mai mic subcorp al lui F ce include K și a . Conform teoriei închiderii, $K(a)$ este format din toate combinațiile de elemente din K și puteri ale lui a , combinații care sunt de forma

$$k_0 + k_1a + \cdots + k_ma^m,$$

cu $m \geq 0$ și $k_0, \dots, k_m \in K$. Așa cum observăm, aceste combinații sunt obținute prin evaluarea tuturor polinoamelor din $K[x]$ în a . Evident, pot exista polinoame distincte care evaluate în a să conducă la același element din $K(a)$. Dar, prin congruență $\ker(\varphi)$, unde $\varphi : K[x] \rightarrow K(a)$ este dată prin $\varphi(h) = h(a)$, pentru orice $h \in K[x]$, polinoamele care sunt evaluate identic în a sunt colectate într-o aceeași clasă de echivalență. Deci $K[x]/\ker(\varphi)$ devine izomorf cu $K(a)$.

Să presupunem acum că $f(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1} + x^n$ este polinomul minimal al lui a . Aceasta este un polinom monic și de grad minim relativ la proprietatea $f(a) = 0$. Așa cum am spus, orice element din $K(a)$ este de forma

$$b = k_0 + k_1a + \cdots + k_ma^m,$$

obținut prin evaluarea unui polinom $g(x) = k_0 + k_1x + \cdots + k_mx^m$ în a . Utilizând teorema împărțirii cu rest, g poate fi scris în forma $g = fq + r$, unde $\text{grad}(r) < \text{grad}(f)$.

În plus, $b = r(a)$. Ca urmare, b poate fi definit prin evaluarea în a a unui polinom de grad mai mic decât $f(x)$. Ca urmare, orice element din $K(a)$ poate fi scris ca o combinație liniară cu coeficienți din K a elementelor $1, a, \dots, a^{n-1}$. Mai mult, aceste elemente sunt liniar independente (altfel, s-ar contrazice minimalitatea gradului polinomului f) și, deci, ele formează o bază pentru $K(a)$ peste K .

Colectăm acum toate observațiile în următoarea teoremă importantă.

Teorema 5.4.2.1. Fie K un subcorp al unui corp F , $a \in F$ un element algebraic peste K și f polinomul minimal al lui a . Atunci:

- (1) $K(a)$ este izomorf cu $K[x]/\ker(\varphi)$, unde $\varphi : K[x] \rightarrow K(a)$ este dată prin $\varphi(h) = h(a)$, pentru orice $h \in K[x]$;
- (2) $K(a)$ este izomorf cu $K[x]/\langle f \rangle$;
- (3) $[K(a) : K] = \text{grad}(f)$, iar mulțimea $\{1, a, \dots, a^{\text{grad}(f)-1}\}$ este bază pentru $K(a)$ peste K ;
- (4) orice $b \in K(a)$ este algebraic peste K iar gradul lui b este divizor al gradului lui a .

Demonstrație. (1) Fie $\varphi : K[x] \rightarrow K(a)$ data prin $\varphi(h) = h(a)$, pentru orice $h \in K[x]$. Este ușor de văzut că φ este homomorfism de inele și

$$K \subseteq \varphi(K[x]) \subseteq K(a).$$

Prima teoremă de izomorfism pentru inele conduce la faptul că $K[x]/\ker(\varphi)$ este izomorf cu $\varphi(K[x])$. Cum $K[x]/\ker(\varphi)$ este corp (Teorema 5.4.1.3), urmărează că $\varphi(K[x])$ este corp, iar $a \in \varphi(K[x])$ și definiția lui $K(a)$ conduce la $\varphi(K[x]) = K(a)$, ceea ce încheie demonstrația pentru (1).

(2) Considerăm idealul $J_a = \{h \in K[x] | h(a) = 0\} = \langle f \rangle$ și vom arăta că $\ker(\varphi) = \sim_{J_a}$ ceea ce, prin prisma lui (1), va demonstra (2).

Fie $(g, h) \in \sim_{J_a}$. Atunci există $\alpha \in J_a$ astfel încât $g = h + \alpha$. Are loc

$$g(a) = h(a) + \alpha(a) = h(a),$$

ceea ce arată că $(g, h) \in \ker(\varphi)$.

Fie acum $(g, h) \in \ker(\varphi)$. Deci $g(a) = h(a)$. Considerăm $\alpha = g - h$. Deoarece $\alpha(a) = 0$, deducem că $\alpha \in J_a$. În plus, $g = h + \alpha$, ceea ce arată că $(g, h) \in \sim_{J_a}$.

Am demonstrat astfel că $\ker(\varphi) = \sim_{J_a}$.

(3) Fie $n = \text{grad}(f)$. Pentru orice $b \in K(a)$ există $g \in K[x]$ astfel încât $b = g(a)$. Conform teoremei împărțirii cu rest, există q și r astfel încât $g = fq + r$ și $\text{grad}(r) < \text{grad}(f) = n$. Atunci

$$b = g(a) = f(a)q(a) + r(a) = r(a),$$

ceea ce arată că b poate fi scris ca o combinație liniară a elementelor $1, a, \dots, a^{n-1}$.

Fie $a_0 + a_1a + \cdots + a_{n-1}a^{n-1} = 0$ o combinație liniară 0 a acestor elemente, unde $a_0, \dots, a_{n-1} \in K$. Atunci a este rădăcină pentru $h(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1}$, iar definiția lui f ne spune că $h = 0$. Ca urmare, $a_i = 0$ pentru orice i . Deci $1, a, \dots, a^{n-1}$ sunt liniar independente, demonstrând astfel (3).

(4) $K(a)$ este extensie finită a lui K (conform cu (3)) și, ca urmare, este extensie algebrică (Propoziția 5.4.2.1). Fie $b \in K(a)$. Atunci $K(b)$ este subcorp al lui $K(a)$, iar Teorema 5.3.5 conduce la

$$[K(a) : K] = [K(a) : K(b)][K(b) : K],$$

ceea ce arată că gradul lui b divide gradul lui a (în baza definiției gradului unui element algebric și a proprietății (3)). \square

Să analizăm încă o dată construcția din Teorema 5.4.2.1. Să presupunem acum că avem un corp K și un polinom $f \in K[x]$ de grad $n \geq 2$ ireductibil peste K . Corpul $K[x]/\langle f \rangle$ este izomorf cu o extensie simplă a lui K cu o anumită rădăcină a lui f (care, deocamdată, nu ne interesează). Funcția ψ de la K la multimea claselor de echivalență induse de polinoamele constante, dată prin $\psi(a) = [a]$, este un izomorfism între K și subcorpul K' al lui $K[x]/\langle f \rangle$ format numai din astfel de clase de echivalență (reamintim că singurele elemente inversabile în $K[x]$ sunt polinoamele constante nenule). Astfel, putem identifica K' prin K și privi pe $K[x]/\langle f \rangle$ ca o extensie a lui K . Dacă facem aceasta, atunci elementele lui $K[x]/\langle f \rangle$ sunt clase de echivalență induse de polinoame $h \in K[x]$ de grad strict mai mic decât n ce au forma

$$[h] = [a_0 + a_1x + \cdots + a_mx^m] = a_0 + a_1[x] + \cdots + a_m[x]^m,$$

unde $h(x) = a_0 + a_1x + \cdots + a_mx^m \in K[x]$ și $m < n$ (clasa de echivalență $[a_i]$ a fost identificată prin a_i , pentru orice i). În plus, $0[f] = [0]$. Ca urmare, orice element din $K[x]/\langle f \rangle$ poate fi scris ca o expresie polinomială în $[x]$ cu coeficienți din K .

Să considerăm acum un element θ ce nu este în K , și fie L multimea tuturor polinoamelor peste K în necunoscuta θ , de grad strict mai mic decât n . Această multime, cu operațiile de adunare și înmulțire a polinoamelor modulo polinomul $f(\theta)$, devine corp izomorf cu $K[x]/\langle f \rangle$. În plus, θ este rădăcină a lui $f(x)$ în L . Deci, L este o extensie simplă a lui K printr-o rădăcină a polinomului f .

Am obținut astfel

Corolarul 5.4.2.1. Fie K un corp și $f \in K[x]$ un polinom ireductibil peste $K[x]$. Atunci există o extensie algebrică simplă a lui K printr-o rădăcină a lui f .

Exemplul 5.4.2.2. Vom exemplifica construcția de mai sus considerând $K = \mathbf{Z}_3$ și polinomul ireductibil peste \mathbf{Z}_3 , $f(x) = x^2 + x + 2$.

Fie θ un element ce nu este în \mathbf{Z}_3 . Atunci multimea

$$L = \{0, 1, 2, \theta, \theta + 1, \theta + 2, 2\theta, 2\theta + 1, 2\theta + 2\},$$

cu operațiile de adunare și înmulțire a polinoamelor modulo $\theta^2 + \theta + 2$, este o extensie algebrică simplă a lui \mathbf{Z}_3 . În această extensie, θ este rădăcină a lui $f(x)$ deoarece

$$f(\theta) \equiv 0 \pmod{\theta^2 + \theta + 2}.$$

Adunarea elementelor $\theta + 1$ și $2\theta + 1$ în L produce elementul 2, iar înmulțirea acestora, θ .

Următoarea propoziție este trivial de verificat.

Propoziția 5.4.2.3. Fie K un corp și f un polinom ireductibil peste K . Dacă L_1 și L_2 sunt două extensii algebrice simple prin căte o rădăcină a lui f , a , respectiv. b , atunci există un izomorfism φ de la L_1 la L_2 cu proprietățile $\varphi(a) = b$ și $\varphi(c) = c$, pentru orice $c \in K$.

Definiția 5.4.2.5. Fie F un corp și K un subcorp al lui. Un polinom $f \in K[x]$ de grad strict pozitiv spunem că se descompune peste/în F dacă poate fi scris în forma

$$f(x) = a(x - a_1) \cdots (x - a_n),$$

unde $n = \text{grad}(f)$ și $a, a_1, \dots, a_n \in F$.

Dacă un polinom $f \in K[x]$ se descompune în F și $F = K(a_1, \dots, a_n)$, atunci vom mai spune că F este un *corp de descompunere pentru f*.

Teorema 5.4.2.2. (Existența și unicitatea corporilor de descompunere)

Pentru orice corp K și polinom $f \in K[x]$ de grad strict pozitiv, există un corp de descompunere pentru f . În plus, acesta este unic până la un izomorfism ce păstrează fixe elementele din K și aplică rădăcini ale lui f în rădăcini ale lui f .

Demonstrație. Se aplică repetat Teorema 5.4.2.1 și Propoziția 5.4.2.3. \square

5.5. Corpuri finite

În Secțiunea 5.3 am văzut că orice corp finit are caracteristica un număr prim și, dacă acest număr prim este p , atunci el are p^n elemente, unde n este dimensiunea corpului peste subcorpul prim al lui.

Întrebarea naturală care se pune este următoarea: pentru ce numere naturale q există corpori cu q elemente? Este clar că q trebuie să fie de forma p^n , unde p este un număr prim și $n \geq 1$. Dar pentru orice număr prim p și $n \geq 1$ există un corp cu p^n elemente? Răspunsul este pozitiv și conținut în cele ce urmează.

Teorema 5.5.1. (Teorema de existență și unicitate a corpurilor finite)

Pentru orice număr prim p și orice $n \geq 1$ există un corp cu p^n elemente. În plus, orice corp cu p^n elemente este izomorf cu corpul de descompunere al polinomului $x^{p^n} - x \in \mathbf{Z}_p[x]$.

Demonstrație. Fie F corpul de descompunere al polinomului $f(x) = x^{p^n} - x$ din $\mathbf{Z}_p[x]$ (Teorema 5.4.2.2). Deoarece $f'(x) = p^n x^{p^n-1} - 1 = p - 1 \in \mathbf{Z}_p$, $f(x)$ nu are rădăcini multiple și, deci, are exact p^n rădăcini în F .

Vom arăta că multimea rădăcinilor acestui polinom, notată prin S , coincide exact cu F . Observăm că S conține 0 și 1. Apoi, dacă a și b sunt în S , atunci

$$(a - b)^{p^n} = a^{p^n} - b^{p^n} = a - b,$$

ceea ce arată că $a - b \in S$. De asemenea,

$$(ab^{-1})^{p^n} = a^{p^n} b^{-p^n} = ab^{-1},$$

arătând că $ab^{-1} \in S$. Ca urmare, S este subcorp al lui F ce conține toate rădăcinile lui F . Definiția lui F conduce atunci la $F = S$ și, deci, $|F| = p^n$.

Fie F un corp cu p^n elemente. Atunci, F trebuie să aibă caracteristica p (Teorema 5.3.4). Ca urmare, putem presupune că F conține pe \mathbf{Z}_p ca subcorp. Fie $f(x) = x^{p^n} - x \in \mathbf{Z}_p[x]$.

Grupul multiplicativ al lui F are ordinul $p^n - 1$ (0 nu este element al acestui grup). Ca urmare, $a^{p^n-1} = 1$, pentru orice $a \in F - \{0\}$. De aici obținem imediat că orice element $a \in F$ este rădăcină a lui $f(x)$. Cum F are exact p^n elemente, F trebuie să fie corp de descompunere pentru $f(x)$. \square

Teoremele 5.3.4 și 5.5.1 dau răspuns complet la întrebarea asupra numărului de elemente ale corpurilor finite. Mai mult, Teorema 5.5.1 ne permite să vorbim despre corpul cu p^n elemente, acesta fiind unic până la un izomorfism. Acest corp se mai notează prin $GF(p^n)$ sau \mathbf{F}_{p^n} (sau F_{p^n}) și se mai numește *corpul Galois cu p^n elemente*.

Corolarul 5.5.1. Fie F_{p^n} un corp finit. Atunci orice subcorp al lui F_{p^n} are p^m elemente, unde $m|n$. Reciproc, dacă $m|n$, atunci există exact un subcorp al lui F_{p^n} cu p^m elemente.

Demonstrație. Fie K un subcorp al lui F_{p^n} . Este clar că acesta are p^m elemente, unde $m \geq 1$ (K include subcorful prim al lui F_{p^n}). Atunci F_{p^n} va trebui să aibă $(p^m)^k$ elemente, unde $k \geq 1$. De aici urmează că $m|n$.

Reciproc, dacă $m|n$, atunci $(p^m - 1)|(p^n - 1)$ și, deci, $(x^{p^m-1} - 1)|(x^{p^n-1} - 1)$ în $F_p[x]$. Ca urmare, $(x^{p^m} - x)|(x^{p^n} - x)$ în $F_p[x]$, ceea ce ne spune că orice rădăcină a lui $x^{p^n} - x$ este rădăcină și a polinomului $x^{p^m} - x$. Dar atunci F_{p^n} trebuie să conțină un subcorp de descompunere pentru $x^{p^m} - x$ și, aşa cum s-a arătat în Teorema 5.5.1, acesta trebuie să aibă ordinul p^m . Dacă ar exista două astfel de subcorpuri de descompunere pentru $x^{p^m} - x$ în F_{p^n} , atunci acestea ar conține mai mult de p^m

rădăcini ale polinomului $x^{p^m} - x$, ceea ce evident ar fi o contradicție. Deci există un unic subcorp al lui F_{p^n} cu p^m elemente. \square

5.6. Aplicații: criptosistemul Rijndael

Secțiunea 4.7 a introdus cititorul în domeniul criptografiei. Așa cum s-a văzut, criptosistemele se clasifică în simetrice și asimetrice (sau cu chei publice), iar Secțiunea 4.7.2 a prezentat criptosistemul asimetric RSA.

În această secțiune vom descrie un criptosistem simetric cunoscut sub numele de *criptosistemul Rijndael*. Deoarece acesta a fost acceptat ca nou standard de criptare simetrică de către SUA, el mai poate fi găsit și sub denumirea de *criptosistemul AES* (Advanced Encryption Standard). Rijndael înlocuiește faimosul criptosistem DES (Data Encryption Standard) despre care s-a arătat, în 1998, că poate fi atacat cu succes prin căutare exhaustivă în spațiul de chei [50].

Criptosistemul Rijndael a fost propus de Joan Daemen și Vincent Rijmen [55, 39]. În octombrie 2002 el a fost ales ca nou standard de criptare, în urma unei competiții lansate de American National Institute for Standards and Technology (NIST). Criptosistemul se bazează pe aritmetică în $GF(2^8)$ cu polinomul ireductibil

$$f(x) = x^8 + x^4 + x^3 + x + 1 \in \mathbf{Z}_2[x].$$

Cum elementele acestui corp sunt polinoame de grad cel mult 7 cu coeficienți în \mathbf{Z}_2 , acestea pot fi reprezentate ca secvențe binare de exact 8 biți. Astfel, secvența binară 01010111 poate fi utilizată pentru a reprezenta polinomul $x^6 + x^4 + x^2 + x + 1$. Evident, în locul secvențelor binare se pot utiliza vectori de lungime 8 peste \mathbf{Z}_2 sau notația hexazecimală pentru a avea o scriere și mai condensată a elementelor corpului. Astfel, $(57)_h$ reprezintă același polinom ca și secvența binară 01010111 (“ $(\cdot)_h$ ” specifică notația hexazecimală).

În corpul $GF(2^8)$, adunarea polinoamelor (elementelor corpului) se face pe componente modulo 2. Astfel,

$$(x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) = x^7 + x^6 + x^4 + x^2.$$

Înmulțirea se realizează modulo $f(x)$. De exemplu,

$$(x^6 + x^4 + x^2 + x + 1)(x^7 + x + 1) = x^7 + x^6 + 1.$$

Dacă adunarea polinoamelor poate fi ușor simulată prin operații la nivel de octet (XOR pe componente), cu înmulțirea lucrurile sunt ceva mai delicate. Se poate constata că înmulțirea a două polinoame se reduce la înmulțirea unui polinom cu x de un număr de ori și la operații de adunare. De exemplu, înmulțirea dintre $x^3 + 1$ și $x^2 + x$ se reduce la înmulțirea lui $x^3 + 1$ cu x , obținându-se $x^4 + x$, înmulțirea

lui $x^4 + x$ cu x , obținându-se $x^5 + x^2$, și adunarea polinoamelor $x^4 + x$ și $x^5 + x^2$. Înmulțirea unui polinom cu x poate fi ușor simulață la nivel de octet:

- octetul ce reprezintă polinomul g ce trebuie înmulțit cu x este deplasat la stânga cu o poziție, completându-se ultima lui poziție cu 0;
- dacă primul bit în octetul ce reprezintă g este 1, atunci din octetul obținut la pasul anterior trebuie scăzut 00011011 ce reprezintă ultimii 8 biți ai lui $f(x)$ (polinomul ireductibil al corpului considerat). Aceasta înseamnă aplicarea unei operații XOR octetului obținut la pasul anterior cu 00011011.

De exemplu,

$$(01010111) \cdot (00000010) = 10101110$$

și

$$(10101110) \cdot (00000010) = 01000111.$$

Vom trece acum la prezentarea criptosistemului Rijndael. Acesta procesează blocuri de date de $4 \times m \times 8$ biți folosind o cheie de $4 \times k \times 8$ biți, unde $m, k \in \{4, 6, 8\}$. În general, criptarea decurge astfel:

- întâi, blocul de date ce urmează a fi criptat este împărțit în grupe de câte 8 biți (octeți) obținându-se astfel o secvență de octeți

$$b_0 b_1 \cdots b_{4m-1}$$

Această secvență este apoi organizată ca o matrice de tip $4 \times m$

$$\begin{pmatrix} b_0 & b_4 & \cdots & b_{4m-4} \\ b_1 & b_5 & \cdots & b_{4m-3} \\ b_2 & b_6 & \cdots & b_{4m-2} \\ b_3 & b_7 & \cdots & b_{4m-1} \end{pmatrix}$$

- matricea astfel obținută este considerată ca fiind un caracter plaintext. Acest caracter este criptat folosind o serie de transformări bazate în principal pe aritmetică în $GF(2^8)$. Rezultatul acestor transformări este o matrice de tip $4 \times m$ ce constituie caracterul criptotext asociat.

Formal, criptosistemul Rijndael poate fi descris astfel:

1. $\mathcal{P} = \mathcal{C} = \mathcal{M}_{4 \times m}(\mathbf{Z}_2^8)$, unde $m \in \{4, 6, 8\}$;
2. $\mathcal{K} = \mathcal{M}_{4 \times k}(\mathbf{Z}_2^8)$, unde $k \in \{4, 6, 8\}$;
3. pentru fiecare $K \in \mathcal{K}$,

$$e_K = T_{K_n}^f \circ T_{K_{n-1}} \circ \cdots \circ T_{K_1} \circ T_{K_0}^i$$

și

$$d_K = T_{K_0}^{-f} \circ T_{K_1}^{-1} \circ \cdots \circ T_{K_{n-1}}^{-1} \circ T_{K_n}^i$$

unde:

- 3.1. n reprezintă numărul de *runde* ce urmează a fi executate. Acest număr de runde depinde de cheia utilizată și de lungimea blocului, așa cum este arătat în Figura 5.2;

n	$m = 4$	$m = 6$	$m = 8$
$k = 4$	10	12	14
$k = 6$	12	12	14
$k = 8$	14	14	14

Figura 5.2: Numărul de runde în Rijndael

- 3.2. T_Z^i , T_Z și T_Z^f sunt transformări date prin:

- $T_Z^i = A_Z$,
- $T_Z = A_Z \circ Mc \circ Sh \circ S$,
- $T_Z^f = A_Z \circ Sh \circ S$,

pentru orice $Z \in \mathcal{M}_{4 \times m}(\mathbf{Z}_2^8)$, unde:

- 3.2.1. A_Z , numită transformarea *AddRoundKey*, nu face altceva decât de a extinde operația XOR la nivel de matrici. Adică,

$$A_Z(X)(i, j) = X(i, j) \oplus Z(i, j),$$

pentru orice $X \in \mathcal{M}_{4 \times m}(\mathbf{Z}_2^8)$, $0 \leq i \leq 3$ și $0 \leq j \leq m - 1$. Vom scrie simplificat $A_Z(X) = X \oplus Z$;

- 3.2.2. Sh , numită transformarea *ShiftRows*, permutează ciclic liniile matricei de intrare cu un număr de poziții. Prima linie este permuatată cu $C_0 = 0$ poziții (adică, nu se efectuează nici o transformare a ei), a doua linie cu $C_1 = 1$ poziții, a treia linie cu $C_2 = 2$ poziții și ultima cu $C_3 = 3$ poziții. Această transformare este reprezentată în Figura 5.3.

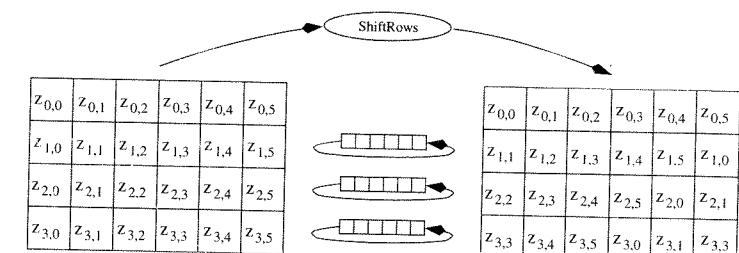


Figura 5.3: Transformarea ShiftRows

Din punct de vedere formal,

$$Sh(X)(i, j) = X(i, (j + C_i) \bmod m),$$

pentru orice $X \in \mathcal{M}_{4 \times m}(\mathbf{Z}_2^8)$, $0 \leq i \leq 3$ și $0 \leq j \leq m - 1$;

3.2.3. S , numită transformarea *SubBytes*, este o substituție neliniară ce operează independent pe fiecare octet al matricei de intrare. Ea folosește o tabelă de substituție, numită *S-box*, care este inversabilă. Această tabelă poate fi calculată prin

$$S(X)(i, j)^t = M_1 \cdot X(i, j)' \oplus C,$$

unde

$$M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \quad C = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix},$$

și

$$X(i, j)' = \begin{cases} (0, 0, 0, 0, 0, 0, 0, 0)^t, & \text{if } X(i, j) = (00)_h \\ (X(i, j)^{-1})^t, & \text{altfel} \end{cases}$$

(inversa este calculată în corpul $GF(2^8)$, iar exponentul "t" denotă transpusa matricei asupra căreia este aplicat).

Transformarea *SubBytes* este reprezentată grafic în Figura 5.4;

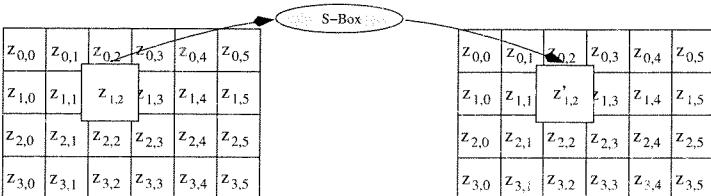


Figura 5.4: Transformarea *SubBytes* aplicată lui $z_{1,2}$

3.2.4. Mc , numită transformarea *MixColumns*, consideră fiecare coloană a matricei de intrare ca un polinom peste $GF(2^8)$, care este de grad cel mult 3, și înmulțește modulo $x^4 + 1$ fiecare astfel de polinom cu polinomul

$$a(x) = (03)_h x^3 + (01)_h x^2 + (01)_h x + (02)_h.$$

Această transformare poate fi scrisă ca înmulțire de matrice în corpul $GF(2^8)[x]$, prin

$$Mc(X) = M_2 \bullet X,$$

unde

$$M_2 = \begin{pmatrix} (02)_h & (03)_h & (01)_h & (01)_h \\ (01)_h & (02)_h & (03)_h & (01)_h \\ (01)_h & (01)_h & (02)_h & (03)_h \\ (03)_h & (01)_h & (01)_h & (02)_h \end{pmatrix}$$

Matricea M_2 este inversabilă și inversa ei este

$$M_2^{-1} = \begin{pmatrix} (0e)_h & (0b)_h & (0d)_h & (09)_h \\ (09)_h & (0e)_h & (0b)_h & (0d)_h \\ (0d)_h & (09)_h & (0e)_h & (0b)_h \\ (0b)_h & (0d)_h & (09)_h & (0e)_h \end{pmatrix}$$

Transformarea *MixColumns* este reprezentată grafic în Figura 5.5;

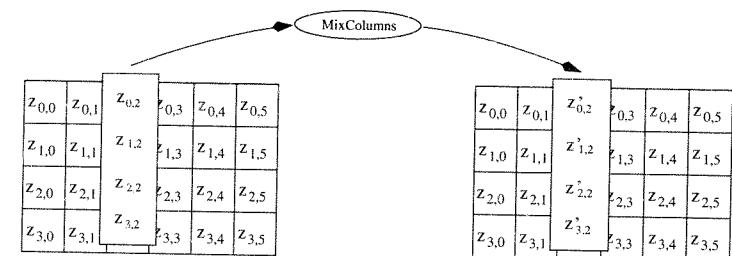


Figura 5.5: Transformarea *MixColumns* aplicată coloanei a treia

3.3. T_Z^{-1} și T_Z^{-f} sunt transformări date prin:

- $T_Z^{-1} = A_{Mc^{-1}(Z)} \circ Mc^{-1} \circ Sh^{-1} \circ S^{-1}$
- $T_Z^{-f} = A_Z \circ Sh^{-1} \circ S^{-1}$,

pentru orice $Z \in \mathcal{M}_{4 \times m}(\mathbf{Z}_2^8)$ (așa cum se va vedea mai departe, transformările S , Sh și Mc sunt inversabile);

3.4. K_0, \dots, K_n sunt numite *chei de rundă*. Ele sunt obținute de la K aplicând următoarea transformare:

- întâi se generează o secvență de $m(n+1)$ vectori coloană (secvență numită *expansiunea cheii*);
- apoi cheile de rundă sunt luate din expansiunea cheii astfel: prima cheie de rundă constă din primii m vectori, a doua din următorii m vectori etc. (cum există $n+1$ runde și în fiecare rundă se prelucră un bloc de m coloane, iar printre operații intervine și adunarea modulo 2 pe componente, sunt necesare $m(n+1)$ vectori coloană pentru a forma cele $n+1$ chei de rundă).

Expanziunea cheii este o secvență de vectori

$$W_0, W_1, \dots, W_{m(n+1)-1}$$

definiți astfel:

- pentru $0 \leq i \leq k-1$, $W_i = K(-, i)$;
- pentru $k \leq i \leq m(n+1)-1$, $W_i = W_{i-k} \oplus T(W_{i-1})$, unde:

$$* T(W) = \begin{cases} SB(RB(W)) \oplus Rcon(i/k), & \text{dacă } i \bmod k = 0 \\ SB(W), & \text{dacă } k > 6 \text{ și} \\ & i \bmod k = 4 \\ W, & \text{altfel;} \end{cases}$$
- * $RB((z_0, z_1, z_2, z_3)^t) = (z_1, z_2, z_3, z_0)^t$;
- * $SB((z_0, z_1, z_2, z_3)^t) = (S(z_0), S(z_1), S(z_2), S(z_3))^t$;
- * $Rcon(i)$ este un vector ce conține valorile

$$(RC(i), (00)_h, (00)_h, (00)_h),$$

unde

- $RC(1) = (01)_h$;
- $RC(i) = x \bullet RC(i-1)$.

Acum cheia de rundă K_i este matricea

$$K_i = (W_{im}, W_{im+1}, \dots, W_{(i+1)m-1}).$$

Vom demonstra că, în adevăr, descrierea de mai sus este un criptosistem. De fapt, va trebui să arătăm că are loc $d_K(e_K(X)) = X$, pentru orice simbol plaintext X și cheie K .

Propoziția 5.6.1. Pentru orice $Z \in \mathcal{M}_{4 \times m}(\mathbf{Z}_2^8)$ și $m \in \{4, 6, 8\}$ are loc:

- (1) M_1 este inversabilă și, ca urmare, transformarea S este inversabilă;
- (2) M_2 este inversabilă și, ca urmare, transformarea Mc este inversabilă;
- (3) $A_Z \circ A_Z = Id$;
- (4) $Sh \circ S = S \circ Sh$;
- (5) $Mc^{-1} \circ A_Z = A_{Mc^{-1}(Z)} \circ Mc^{-1}$;
- (6) $T_Z^i \circ T_Z^f = Sh \circ S$;
- (7) $T_Z^{-1} \circ Sh \circ S \circ T_Z = Sh \circ S$;
- (8) $T_Z^{-f} \circ Sh \circ S \circ T_Z^i = Id$.

Demonstrație. Aceasta urmează direct de la definițiile acestor transformări. \square

Corolarul 5.6.1. Pentru orice $X \in \mathcal{M}_{4 \times m}(\mathbf{Z}_2^8)$, $K \in \mathcal{M}_{4 \times k}(\mathbf{Z}_2^8)$ și $m, k \in \{4, 6, 8\}$ are loc $d_K(e_K(X)) = X$.

Demonstrație. Direct de la Propoziția 5.6.1. \square

Există multe proprietăți matematice interesante ale acestui criptosistem. La momentul actual se fac studii intense asupra posibilităților teoretice și practice de atac asupra lui, iar pagina web de la NIST dedicată acestuia furnizează informații de actualitate. Foarte pe scurt, putem spune că securitatea acestui criptosistem se bazează pe faptul că operații simple cu biți sunt translatăte într-un corp finit, iar inversarea acestora în respectivul corp, fără a cunoaște informații suplimentare asupra cheii, este dificil de realizat.

Capitolul 6

Spații vectoriale

Algebra liniară este ramură a matematicii care se ocupă cu studiul *vectorilor*, *spațiilor vectoriale* (numite, din rațiuni istorice, și *spații liniare*), *transformărilor liniare* și *sistemelor de ecuații liniare*. Spațiile vectoriale, și de fapt întreaga algebră liniară, au aplicații majore în multe ramuri ale matematicii dar și în științele sociale. În informatică, algebra liniară este fundamentală într-o multitudine de domenii, cum ar fi teoria codurilor detectoare și corectoare de erori, criptografie, geometrie computațională etc.

Din punct de vedere istoric, bazele algebrei liniare au fost puse de William Rowan Hamilton prin descoperirea quaternionilor în 1843 [81] (el este și cel care a introdus terminologia de *vector*), urmat imediat de Hermann Grassmann care în 1844 a publicat cartea *Die lineare Ausdehnungslehre dargestellt und durch Anwendungen auf die übrigen Zweige der Mathematik, wie auch auf die Statik, Mechanik, die Lehre vom Magnetismus und die Krystallonomie*. Grassmann a considerat în cartea sa combinații liniare, combinații liniare unice (ce duc la conceptul de bază a unui spațiu vectorial) dar, fără a folosi un formalism adecvat care, la momentul respectiv, nu era disponibil. Definiția formală a spațiilor vectoriale apare în matematică în jurul anilor 1920 [197, 219].

Pentru o bună introducere în teoria spațiilor vectoriale indicăm [79, 116, 37].

6.1. Definiții. Exemple. Proprietăți de bază

Spațiile vectoriale sunt similare algebrelor definite în Secțiunea 1.4.4, deosebirea constând în aceea că se consideră două mulțimi peste care se definesc anumite operații.

Definiția 6.1.1. Fie $(F, +, -, 0, \circ, ', e)$ un corp. Un *spațiu vectorial* peste F este un sistem $(V, \oplus, \ominus, \mathbf{0}, \cdot)$ format dintr-un grup comutativ $(V, \oplus, \ominus, \mathbf{0})$ și o funcție $\cdot : F \times V \rightarrow V$ ce satisface:

- (1) $\alpha \cdot (x \oplus y) = \alpha \cdot x \oplus \alpha \cdot y$, pentru orice $\alpha \in F$ și $x, y \in V$;
- (2) $(\alpha + \beta) \cdot x = \alpha \cdot x \oplus \beta \cdot x$, pentru orice $\alpha, \beta \in F$ și $x \in V$;
- (3) $(\alpha \circ \beta) \cdot x = \alpha \cdot (\beta \cdot x)$, pentru orice $\alpha, \beta \in F$ și $x \in V$;
- (4) $e \cdot x = x$, pentru orice $x \in V$.

Numim elementele lui V *vectori*, elementele lui F *scări*, iar F , *corpul de scări al spațiului vectorial* V . Operația \cdot este numită *înmulțirea cu scări*.

Așa cum se vede din definiție, trebuie făcută distincție între “adunarea” în F și cea în V (*adunarea vectorilor*), între opusul unui element în F și opusul unui vector în V , între elementul zero al corpului F și *vectorul zero* sau *nul 0* al spațiului V . Pentru a face aceste distincții, în Definiția 6.1.1 s-au utilizat notații diferite pentru operațiile celor două structuri algebrice. Aceasta conduce însă la o manevrare anevoieasă a formulelor și calculelor, motiv pentru care operațiile celor două structuri se notează identic, distincția rezultând din context. Astfel, operațiile corpului F vor fi noteate în varianta $(F, +, -, 0, \cdot, ', 1)$, iar cele ale spațiului vectorial prin $(V, +, -, 0, \cdot)$. Mai mult, semnul operației \cdot (de fapt ea notează două operații, înmulțirea în F și înmulțirea cu scări) este omis. Cu acestea, cerințele din Definiția 6.1.1 se rescriu astfel:

- (1) $\alpha(x + y) = \alpha x + \alpha y$, pentru orice $\alpha \in F$ și $x, y \in V$;
- (2) $(\alpha + \beta)x = \alpha x + \beta x$, pentru orice $\alpha, \beta \in F$ și $x \in V$;
- (3) $(\alpha\beta)x = \alpha(\beta x)$, pentru orice $\alpha, \beta \in F$ și $x \in V$;
- (4) $1x = x$, pentru orice $x \in V$.

Scăderea vectorilor se definește în mod ușual prin $x - y = x + (-y)$, pentru orice $x, y \in V$.

Spațiul vectorial format doar dintr-un singur element, 0, poartă denumirea de *spațiu vectorial trivial*. În general, vom evita acest spațiu din considerațiile noastre deoarece el nu prezintă interes din punct de vedere al studiului.

Observația 6.1.1. Fie V un spațiu vectorial peste un corp F . Fiecare scalar $\alpha \in F$ definește o funcție $f_\alpha : V \rightarrow V$ dată prin

$$f_\alpha(x) = \alpha x,$$

pentru orice $x \in V$.

Atunci, Definiția 6.1.1(1) ne spune că f_α este endomorfism al grupului comutativ $(V, +, -, 0)$.

Exemplul 6.1.1.

- (1) Fie F un corp și $n \geq 1$ un număr natural. Considerăm mulțimea F^n a tuturor vectorilor n -ari peste F , adică mulțimea tuturor elementelor de forma (a_1, \dots, a_n) cu $a_1, \dots, a_n \in F$.

Această mulțime poate fi structurată ca spațiu vectorial peste F considerând adunarea vectorilor

$$(a_1, \dots, a_n) + (b_1, \dots, b_n) = (a_1 + b_1, \dots, a_n + b_n)$$

și înmulțirea cu scări

$$b(a_1, \dots, a_n) = (ba_1, \dots, ba_n),$$

pentru orice $(a_1, \dots, a_n), (b_1, \dots, b_n) \in F^n$ și $b \in F$.

În cazul $n = 1$ identificăm F^1 cu F , și atunci corpul F poate fi privit ca un spațiu vectorial peste el însuși.

- (2) Fie F un corp și $m, n \geq 1$ numere naturale. Notăm prin ${}^m F^n$ sau $M_{m \times n}(F)$ mulțimea tuturor matricelor de tip $m \times n$ cu elemente din corpul F . Această mulțime poate fi organizată ca spațiu vectorial peste F cu adunarea și înmulțirea ușuală a acestora cu scări. Matricea ale cărei componente sunt toate zero, numită și *matricea nulă* sau *matricea zero*, este vectorul zero al spațiului vectorial.
- (3) \mathbf{Q}^n , \mathbf{R}^n , \mathbf{C}^n sunt spații vectoriale obținute ca la (1), pe care cititorul cu siguranță le-a întâlnit în studiile sale matematice (cel puțin pentru $n = 2$ sau $n = 3$).
- (4) \mathbf{C} poate fi privit ca spațiu vectorial peste \mathbf{R} , și atât \mathbf{C} , cât și \mathbf{R} pot fi privite ca spații vectoriale peste corpul \mathbf{Q} .
- (5) Mulțimea tuturor funcțiilor de la \mathbf{R} la \mathbf{R} , cu operația de adunare a funcțiilor $f + g$ și înmulțirea cu scări αf ($(\alpha f)(x) = \alpha f(x)$, pentru orice x), formează spațiu vectorial peste \mathbf{R} .

Următoarele proprietăți urmează cu ușurință de la definiții.

Propoziția 6.1.1. Fie V un spațiu vectorial peste un corp F . Atunci, pentru orice $x, y \in V$ și $\alpha, \beta \in F$, au loc următoarele proprietăți:

- (1) $0x = 0$ (în stânga egalității, 0 este elementul zero al corpului F , iar în dreapta, 0 este vectorul zero al lui V);
- (2) $(-1)x = -x$ ($-x$ fiind opusul lui x în V);
- (3) $(-\alpha)x = \alpha(-x) = -\alpha x$;
- (4) $\alpha 0 = 0$ (0 fiind vectorul zero al lui V);
- (5) dacă $\alpha x = 0$, atunci $\alpha = 0$ sau $x = 0$;
- (6) dacă $\alpha x = \alpha y$, atunci $\alpha = 0$ sau $x = y$;

(7) dacă $\alpha x = \beta x$, atunci $\alpha = \beta$ sau $x = 0$.

Demonstrație. (1) Au loc relațiile:

$$0x = (0 + 0)x = 0x + 0x,$$

de unde urmează $0x = 0$.

(2) Au loc relațiile:

$$0 = 0x = (1 + (-1))x = 1x + (-1)x = x + (-1)x,$$

de unde urmează $(-1)x = -x$.

(3) Utilizând (2) și definiția spațiilor vectoriale, obținem:

$$\alpha(-x) = \alpha((-1)x) = (\alpha(-1))x = ((-1)\alpha)x = (-1)(\alpha x) = -\alpha x$$

și $(-\alpha)x = ((-1)\alpha)x = -\alpha x$.

(4) Au loc relațiile:

$$\alpha x = \alpha(0 + x) = \alpha 0 + \alpha x,$$

de unde urmează $\alpha 0 = 0$.

(5) Pentru $\alpha = 0$ proprietatea este trivial satisfăcută. Pentru $\alpha \neq 0$ are loc:

$$x = 1x = (\alpha^{-1}\alpha)x = \alpha^{-1}(\alpha x) = 0.$$

(6) și (7) sunt consecințe imediate ale proprietăților precedente. \square

Fie $m \geq 1$ un număr natural. În corpul F , notația $m \cdot 1$ reprezintă 1 adunat cu el însuși de m ori. Dat un vector x , $(m \cdot 1)x$ este un element al spațiului vectorial V , dat prin

$$(m \cdot 1)x = \underbrace{(1 + \dots + 1)}_{m \text{ ori}} x = \underbrace{1x + \dots + 1x}_{m \text{ ori}} = \underbrace{x + \dots + x}_{m \text{ ori}}$$

(ceea ce se poate arăta ușor prin inducție matematică).

Ca urmare, putem folosi liber notația mx pentru $(m \cdot 1)x$. În mod similar putem obține și multiplii negativi ai unui vector x .

Dacă F are caracteristica p , atunci $p \cdot 1 = 0$ și, deci, $px = 0$, pentru orice $x \in V$.

Definiția 6.1.2. Fie V un spațiu vectorial peste un corp F și $U \subseteq V$. Spunem că U este *închisă* în V dacă au loc proprietățile:

(1) U este închisă în grupul $(V, +, -, 0)$;

(2) dacă $x \in U$ și $\alpha \in F$, atunci $\alpha x \in U$.

Evident, Definiția 6.1.2(1) poate fi înlocuită cu oricare din proprietățile ce stabilesc închiderea unei submulțimi într-un grup. De exemplu, ea poate fi înlocuită prin următoarele două proprietăți:

- $0 \in U$;
- $x - y \in U$, pentru orice $x, y \in U$.

Dacă luăm în considerație faptul că $(-1)x = -x$, atunci proprietatea (2) din Definiția 6.1.2 ne permite să înlocuim proprietatea (1) din aceeași definiție prin

- $0 \in U$;
- $x + y \in U$, pentru orice $x, y \in U$.

Definiția 6.1.3. Fie V și U spații vectoriale peste un corp F . Spunem că U este *subspațiu vectorial* al lui V , și notăm $U \leq V$, dacă $U \subseteq V$ și restricțiile operațiilor spațiului V la U coincid cu operațiile spațiului U .

Dacă $U \leq V$, atunci U este submulțime închisă în V . Reciproc, orice submulțime închisă în V poate fi structurată ca un subspațiu vectorial al lui V prin restricționarea operațiilor spațiului V la U .

Următoarea propoziție urmează imediat de la definiții.

Propoziția 6.1.2. Intersecția oricărei familii nevide de subspații vectoriale ale unui spațiu vectorial este subspațiu vectorial al acestuia

Exemplul 6.1.2.

- (1) Dacă V este un spațiu vectorial peste un corp F , atunci $\{0\}$ și V sunt subspații ale spațiului V .
- (2) Fie F un corp și $n \geq 1$. Dacă considerăm $U \subseteq F^n$ ca fiind submulțimea tuturor vectorilor a căror ultimă coordonată este 0, atunci U este închisă în F^n . Ca urmare, U definește un subspațiu vectorial al lui F^n . Dacă $n \geq 2$, atunci acest subspațiu poate fi "identificat" cu F^{n-1} .

Fie V un spațiu vectorial peste un corp F , $x_1, \dots, x_k \in V$ și $\alpha_1, \dots, \alpha_k \in F$, unde $k \geq 1$. O sumă de forma

$$\alpha_1 x_1 + \dots + \alpha_k x_k$$

va fi numită *combinăție liniară* a vectorilor x_1, \dots, x_k . Uneori, ea va fi scrisă simplificat în forma $\sum_{i=1}^k \alpha_i x_i$ sau chiar $\sum \alpha_i x_i$, atunci când numărul k nu este important sau poate fi dedus din context.

Mulțimea tuturor combinațiilor liniare ale vectorilor x_1, \dots, x_k formează un subspațiu vectorial al spațiului V , ce poartă denumirea de *subspațiu (vectorial) generat* de x_1, \dots, x_k . El se mai notează prin $\langle x_1, \dots, x_k \rangle_V$ sau $\langle x_1, \dots, x_k \rangle$, atunci când V este clar din context. Ca urmare,

$$\langle x_1, \dots, x_k \rangle = \{\alpha_1 x_1 + \dots + \alpha_k x_k \mid \alpha_1, \dots, \alpha_k \in F\}.$$

Este clar că $0 \in \langle x_1, \dots, x_k \rangle$ și $\langle x_1, \dots, x_k \rangle$ este intersecția tuturor subspațiilor vectoriale ale lui V ce conțin x_1, \dots, x_k .

Dacă $x = \sum \alpha_i x_i$, atunci spunem că x este o *combinație liniară a vectorilor* x_1, \dots, x_k sau că x este *liniar dependent de* x_1, \dots, x_k .

Definiția 6.1.4. Fie V un spațiu vectorial peste un corp F . Vectorii x_1, \dots, x_k din V sunt *numiți liniar dependenți* dacă există $\alpha_1, \dots, \alpha_k \in F$, nu toți 0, astfel încât $\sum \alpha_i x_i = 0$.

Dacă x_1, \dots, x_k nu sunt liniar dependenți, atunci ei se numesc *liniar independenți*. Formal, vectorii x_1, \dots, x_k sunt liniar independenți dacă pentru orice $\alpha_1, \dots, \alpha_k \in F$, relația $\sum \alpha_i x_i = 0$ conduce la $\alpha_1 = \dots = \alpha_k = 0$.

Observația 6.1.2. Fie V un spațiu vectorial peste un corp F .

- (1) Un vector $x \in V$ este liniar independent dacă și numai dacă $x \neq 0$.
- (2) Dacă $x_1, \dots, x_k \in V$ sunt vectori liniar independenți, atunci nu există i astfel încât $x_i = 0$. Mai mult, nu există i și j cu $i \neq j$ și $x_i = x_j$.
- (3) Atragem atenția asupra faptului că noțiunile de combinație liniară de vectori, vectori liniar dependenți sau independenți s-au introdus luând în considerare un număr finit de vectori.

Propoziția 6.1.3. Fie V un spațiu vectorial peste un corp F . Vectorii x_1, \dots, x_k din V sunt liniar dependenți dacă și numai dacă există i , $1 \leq i \leq k$, astfel încât x_i este liniar dependent de restul vectorilor.

Demonstrație. Să presupunem că vectorii x_1, \dots, x_k sunt liniar dependenți. Atunci există $\alpha_1, \dots, \alpha_k$, nu toți 0, astfel încât $\sum \alpha_i x_i = 0$.

Fie i cu proprietatea $\alpha_i \neq 0$. Atunci

$$x_i = \sum_{j \neq i} (-\alpha_i^{-1} \alpha_j) x_j,$$

ceea ce arată că x_i este combinație liniară a celorlalți vectori.

Reciproc, dacă $x_i = \sum_{j \neq i} \alpha_j x_j$, atunci $(-1)x_i + \sum_{j \neq i} \alpha_j x_j = 0$, ceea ce arată că x_1, \dots, x_k sunt liniar dependenți. \square

6.2. Bază și dimensiune

Specificarea unui spațiu vectorial într-un mod finitar, adică printr-un număr finit de vectori ai acestuia care să genereze întreg spațiul, este de importanță foarte mare. Dacă acești vectori sunt și liniar independenți, atunci ei formează o bază.

Definiția 6.2.1. Fie V un spațiu vectorial netrivial peste un corp F . Se numește *bază* a spațiului V orice submulțime finită $B \subseteq V$ de vectori liniar independenți ce generează spațiul V (adică orice vector din V se scrie ca o combinație liniară de vectori din B).

Observația 6.2.1.

(1) Spațiul vectorial trivial $\{0\}$ este general de vectorul 0, dar 0 nu este liniar independent. Acesta este motivul pentru care Definiția 6.2.1 evită spațiul trivial. Dacă ea ar fi luat în considerare și acest caz, atunci spațiul trivial nu ar fi admis nici o bază. Ca urmare, ori am fi acceptat că spațiul trivial nu admite nici o bază, ori ar fi trebuit să modificăm definiția pentru a include și pe acesta. Ambele cazuri ar fi condus la analize de genul "spațiu trivial/spațiu netrivial", care nu ar fi avut nici un sens și nu ar fi furnizat nimic în plus.

Ca urmare, conform Definiției 6.2.1, vom discuta despre baze numai pentru spații netriviale.

- (2) Dacă x_1, \dots, x_k este o secvență de vectori ce formează o bază pentru un spațiu vectorial V , atunci orice doi vectori din această secvență sunt distincți (Observația 6.1.2(2)). Ca urmare, putem spune că mulțimea $\{x_1, \dots, x_k\}$, care are exact k vectori, este bază a spațiului vectorial V . Aceasta este de fapt punctul de vedere adoptat în Definiția 6.2.1.
- (3) Definiția 6.2.1 consideră numai *baze finite* (formate dintr-un număr finit de vectori). Aceasta este în concordanță cu Observația 6.1.2(3). Există abordări pentru cazul spațiilor vectoriale cu *baze infinite*, dar acestea nu vor fi considerate în acest material.

Orice bază a unui spațiu vectorial este nevidă, conform Definiției 6.2.1. Mai mult, nici o bază nu conține vectorul 0.

Exemplul 6.2.1.

- (1) Fie F un corp și $n \geq 1$ un număr natural. Spațiul vectorial F^n (introdus în Exemplul 6.1.1(1)) poate fi generat de vectorii

$$\begin{aligned} e_1 &= (1, 0, 0, \dots, 0, 0) \\ e_2 &= (0, 1, 0, \dots, 0, 0) \\ &\dots \\ e_n &= (0, 0, 0, \dots, 0, 1). \end{aligned}$$

- (2) Fie F un corp și $m, n \geq 1$ numere naturale. Spațiul vectorial ${}^m F^n$ (introdus în Exemplul 6.1.1(2)) poate fi generat de vectorii (matricele) E_{ij} date prin

$$E_{ij}(u, v) = \begin{cases} 1, & \text{dacă } u = i \text{ și } v = j \\ 0, & \text{altfel,} \end{cases}$$

pentru orice $i, u \in \{1, \dots, m\}$ și $v, j \in \{1, \dots, n\}$.

Teorema 6.2.1. Fie V un spațiu vectorial netrivial peste un corp F . Mulțimea $B = \{x_1, \dots, x_k\} \subseteq V$ este bază a spațiului vectorial V dacă și numai dacă orice vector $x \in V$ se scrie în mod unic ca o combinație liniară de vectori ai lui B .

Demonstrație. Fie $B = \{x_1, \dots, x_k\}$ o bază a spațiului vectorial V . Dacă un vector $x \in V$ ar avea cel puțin două scrieri distincte ca o combinație liniară de vectori ai bazei,

$$x = \sum_{i=1}^k \alpha_i x_i = \sum_{i=1}^k \beta_i x_i,$$

atunci s-ar obține

$$\sum_{i=1}^k (\alpha_i - \beta_i) x_i = 0,$$

ceea ce ar contrazice liniar independenta vectorilor bazei (menționăm că există i astfel încât $\alpha_i - \beta_i \neq 0$).

Pentru reciproca teoremei avem de arătat doar că B este formată din vectori liniar independenti. Fie $\sum \alpha_i x_i = 0$ o combinație liniară ce conduce la vectorul 0. Unicitatea scrierii acestui vector ca o combinație liniară de vectori ai lui B conduce la $\alpha_i = 0$, pentru orice i . Deci, B este formată din vectori liniar independenti. \square

Scrierea unică a vectorilor $x \in V$ în formă de combinații liniare de vectori ai unei baze $B = \{x_1, \dots, x_k\}$ permite introducerea conceptului de *coordonată*. Astfel, dacă

$$x = \alpha_1 x_1 + \dots + \alpha_k x_k,$$

atunci vom spune că α_i este *coordonata a i -a sau corespunzătoare lui x_i* a vectorului x . k -uplul $(\alpha_1, \dots, \alpha_k)$ se numește *coordonata lui x în spațiul vectorial V în raport cu baza B* . Așa cum se vede, am considerat implicit ordonarea vectorilor bazei în raport cu indicii acestora.

Propoziția 6.2.1. Fie V un spațiu vectorial netrivial peste un corp F . Orice submulțime finită $A \subseteq V$ ce generează V include o bază pentru V .

Demonstrație. Dacă vectorii mulțimii A sunt liniar independenti, atunci ei formează o bază. Altfel, există $x \in A$ ce este liniar dependent de ceilalți vectori din A . Eliminăm acest vector din mulțimea A și vectorii rămași generează în continuare V .

Se repetă acest procedeu până când se obține o mulțime liniar independentă de vectori care va fi bază a spațiului V . Faptul că V este netrivial garantează că în urma acestui proces va rămaîne cel puțin un vector nenul. \square

Definiția 6.2.2. Fie V un spațiu vectorial peste un corp F . O mulțime de vectori $B = \{x_1, \dots, x_k\} \subseteq V$ este numită *mulțime maximală de vectori liniar independenti* în V , dacă B este o mulțime de vectori liniar independenti și $B \cup \{x\}$ este mulțime de vectori liniar dependenți, pentru orice $x \in V - B$.

Evident, $B = \{x_1, \dots, x_k\} \subseteq V$ este mulțime maximală de vectori liniar independenti în V dacă pentru orice $x \in V - B$ există $\alpha_1, \dots, \alpha_k \in F$, nu toți 0, astfel încât $x = \sum \alpha_i x_i$. Direct de la această remarcă obținem:

Propoziția 6.2.2. Fie V un spațiu vectorial netrivial peste un corp F . Dacă B este mulțime maximală de vectori liniar independenti în V , atunci B este bază pentru V .

Următorul rezultat va fi de importanță foarte mare.

Lema 6.2.1. (Lema de schimb a lui Steinitz)

Fie V un spațiu vectorial netrivial peste un corp F . Dacă V este generat de $A = \{x_1, \dots, x_k\}$ și $B = \{y_1, \dots, y_m\} \subseteq V$ este o mulțime de vectori liniar independenti, atunci $m \leq k$. În plus, anumiți vectori din A pot fi înlocuiți cu vectori din B astfel încât mulțimea nou obținută să aibă tot k vectori, să genereze V și să includă B .

Demonstrație. Arătăm că dacă un vector $y_i \in B$ nu este în A , atunci există $x_j \in A - B$ ce poate fi înlocuit cu y_i . Mulțimea astfel obținută va genera în continuare V , va avea k vectori și va conține y_i .

Construcția este următoarea. Fie $y_i \in B - A$. Vectorul y_i se scrie ca o combinație de elemente din A , $y_i = \sum \alpha_j x_j$. În plus, există j astfel încât $x_j \in A - B$ și $\alpha_j \neq 0$ pentru că, altfel, y_i ar fi 0 sau s-ar scrie ca o combinație de elemente din $B - \{y_i\}$, ceea ce ar contrazice faptul că B este liniar independentă. Relația

$$y_i = \alpha_j x_j + \sum_{t \neq j} \alpha_t x_t$$

ne arată că y_i și x_j se pot exprima unul din altul. Construim atunci

$$A' = (A - \{x_j\}) \cup \{y_i\}$$

și constatăm că are loc $\langle A' \rangle = V$.

Repetând procedeul de mai sus cu toți vectorii din B ce nu sunt în A , obținem o mulțime ce satisfac cerințele lemei. \square

Corolarul 6.2.1. Dacă A și B sunt mulțimi finite liniar independente ce generează un spațiu vectorial V , atunci $|A| = |B|$.

Demonstrație. Direct de la Lema de schimb a lui Steinitz aplicată de două ori: o dată cu A ca mulțime generatoare a spațiului V , și altă dată cu B ca mulțime generatoare a spațiului V . \square

Exemplul 6.2.2. Fie F un corp. Notăm prin F^N mulțimea tuturor vectorilor infiniti cu coordonate din F , adică funcții de la N la F . Această mulțime poate fi structurată ca spațiu vectorial ca și F^n . Dar, spre deosebire de F^n , acest spațiu vectorial nu admite baze finite. În adevăr, să presupunem că el ar admite o bază finită B . Fie

$k = |B|$. Observăm că vectorii $f_i : \mathbf{N} \rightarrow F$ dați prin

$$f_i(n) = \begin{cases} 1, & \text{dacă } n = i \\ 0, & \text{altfel,} \end{cases}$$

pentru orice $1 \leq i \leq k+1$ și $n \in \mathbf{N}$, sunt liniar independenți. Cum

$$|\{f_i \mid 1 \leq i \leq k+1\}| = k+1 > k = |B|,$$

deducem că B nu poate fi o bază a acestui spațiu vectorial.

Corolarul 6.2.1 ne permite să introducем următorul concept important.

Definiția 6.2.3. Fie V un spațiu vectorial peste un corp F .

- (1) Spunem că V este *spațiu vectorial finit dimensional* dacă există o bază (finită) pentru V . Numărul vectorilor din această bază se numește *dimensiunea lui V* și se notează prin $\dim(V)$.
- (2) Spunem că V este *spațiu vectorial infinit dimensional* dacă V nu este finit dimensional.

Spațiul vectorial trivial este considerat finit dimensional.

Dacă B este o bază finită a spațiului vectorial V , atunci vom mai spune că V este $|B|$ -dimensional.

Exemplul 6.2.3.

- (1) Spațiul vectorial F^n din Exemplul 6.1.1(1) are dimensiunea n .
- (2) Spațiul vectorial mF^n din Exemplul 6.1.1(2) are dimensiunea mn .
- (3) Spațiul vectorial $F^{\mathbf{N}}$ din Exemplul 6.2.2 este infinit dimensional.

Corolarul 6.2.2. Fie V un spațiu vectorial k -dimensional, $k \geq 1$. Atunci, orice mulțime de k vectori liniar independenți din V formează o bază pentru V .

Demonstrație. Fie x_1, \dots, x_k vectori liniar independenți ce generează V (astfel de vectorii există deoarece V are dimensiunea k).

Dacă y_1, \dots, y_k ar fi alți k vectori liniar independenți, atunci, aplicând Lema de schimb a lui Steinitz, obținem că y_1, \dots, y_k generează V . Deci ei formează o bază pentru V . \square

Corolarul 6.2.3. Fie V un spațiu vectorial k -dimensional, $k \geq 1$. Atunci orice mulțime x_1, \dots, x_m de vectori liniar independenți cu proprietatea $m < k$, poate fi extinsă la o bază în V .

Demonstrație. Direct de la Lema de schimb a lui Steinitz aplicată unei baze a spațiului V și mulțimii $\{x_1, \dots, x_m\}$. \square

Corolarul 6.2.4. Fie V un spațiu vectorial finit dimensional și $U \leq V$. Dacă $\dim(U) = \dim(V)$, atunci $U = V$.

Demonstrație. Fie $k = \dim(U) = \dim(V)$. Spațiul vectorial U conține k vectori liniar independenți ce generează U . Acești vectori sunt și în V , și generează V . Ca urmare, $U = V$. \square

6.3. Funcții liniare

Din rațiuni istorice, homomorfismele de spații vectoriale sunt ușual numite *funcții (aplicații) liniare*.

Definiția 6.3.1. Fie U și V spații vectoriale peste un corp F . O funcție $f : U \rightarrow V$ este numită *liniară* dacă:

- (1) $f(x+y) = f(x) + f(y)$, pentru orice $x, y \in U$;
- (2) $f(\alpha x) = \alpha f(x)$, pentru orice $x \in U$ și $\alpha \in F$.

Cele două cerințe din Definiția 6.3.1 pot fi cumulate în una singură,

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y),$$

pentru orice $\alpha, \beta \in F$ și $x, y \in U$.

Dacă f este o aplicație liniară, atunci este clar că are loc:

$$f(\sum \alpha_i x_i) = \sum \alpha_i f(x_i),$$

pentru orice $\alpha_i \in F$ și $x_i \in U$ (cea ce poate fi ușor obținut prin inducție matematică).

Concepțele de *epimorfism*, *monomorfism*, *izomorfism*, *endomorfism* și *automorfism* de spații vectoriale se introduc în mod standard: epimorfismele de spații vectoriale sunt funcții liniare surjective, monomorfismele sunt funcții liniare injective, izomorfismele sunt funcții liniare bijective, endomorfismele sunt funcții liniare de la un spațiu vectorial la el însuși, iar automorfismele sunt endomorfisme bijective.

Atunci când două spații vectoriale U și V sunt izomorfe vom nota $U \cong V$.

Propoziția 6.3.1. Orice spațiu vectorial n -dimensional peste un corp F este izomorf cu F^n .

Demonstrație. Fie V un spațiu n -dimensional peste un corp F și $\{x_1, \dots, x_n\}$ o bază a lui.

Considerăm funcția $f : F^n \rightarrow V$ dată prin

$$f(\alpha_1, \dots, \alpha_n) = \sum \alpha_i x_i,$$

pentru orice $\alpha_1, \dots, \alpha_n \in F$.

Este trivial de verificat că această funcție este liniară. În plus, surjectivitatea ei urmează de la faptul că $\{x_1, \dots, x_n\}$ generează V , iar injectivitatea de la faptul că $\{x_1, \dots, x_n\}$ este mulțime de vectori liniari independenti. \square

Conform Propoziției 6.3.1, în cazul spațiilor vectoriale finit dimensionale peste un corp F , ne putem restrânge studiul, via un izomorfism, la spații vectoriale de forma F^n .

Corolarul 6.3.1. Orice două spații vectoriale peste același corp F și de aceeași dimensiune, sunt izomorfe.

Demonstrație. Dacă U și V sunt spații vectoriale n -dimensionale peste un corp F , atunci ambele sunt izomorfe cu F^n și, deci, izomorfe între ele. \square

Fie U și V două spații vectoriale peste un corp F . Mulțimea funcțiilor liniare de la U la V , notată $\text{Hom}_F(U, V)$, poate fi structurată ca spațiu vectorial peste F considerând adunarea vectorilor $f + g$ și înmulțirea cu scalari αf date prin:

- $(f + g)(x) = f(x) + g(x),$
- $(\alpha f)(x) = \alpha f(x),$

pentru orice $f, g \in \text{Hom}_F(U, V)$, $x \in U$ și $\alpha \in F$.

Să presupunem acum că U este de dimensiune m și V de dimensiune n . Vom arăta că $\text{Hom}_F(U, V)$ este de dimensiune mn .

Fie $B_1 = \{x_1, \dots, x_m\}$ o bază pentru U și $B_2 = \{y_1, \dots, y_n\}$ o bază pentru V . Vom fixa ordinea vectorilor în aceste baze ca fiind ordinea dată de indecșii vectorilor acestora.

Orice funcție liniară f de la U la V este complet specificată de imaginea vectorilor bazei B_1 prin ea. Însă, pentru orice i , $f(x_i)$ este o combinație liniară de vectori ai bazei B_2 ,

$$f(x_i) = \sum_{j=1}^n \alpha_{ij} y_j.$$

Ca urmare, f este complet și unic specificată de matricea

$$M_f = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \alpha_{m1} & \alpha_{m2} & \cdots & \alpha_{mn} \end{pmatrix}$$

(în raport cu ordinea vectorilor bazelor, fixată ca mai sus).

Știm că mulțimea ${}^m F^n$ a tuturor matricilor de tip $m \times n$ peste F formează spațiu vectorial de dimensiune mn . Ca urmare, ceea ce ne mai rămâne de arătat este că

funcția $h : \text{Hom}_F(U, V) \rightarrow {}^m F^n$ dată prin $h(f) = M_f$, este izomorfism de spații vectoriale.

Unicitatea specificării funcțiilor liniare de la U la V prin matrice, să cum s-a arătat mai sus, conduce imediat la faptul că h este bine definită și injectivă. Pentru orice matrice $M = (a_{ij}) \in {}^m F^n$, considerăm funcția f_M dată prin

$$f_M(x) = \sum_{i,j} \alpha_i a_{ij} y_j,$$

pentru orice $x \in U$, unde $x = \sum_{i=1}^m \alpha_i x_i$ este scrierea unică a lui x ca o combinație liniară de vectori ai bazei B_1 . f_M este funcție liniară (ceea ce se verifică cu ușurință) și $h(f_M) = M_{f_M} = M$. Deci, h este surjectivă.

Un simplu exercițiu ne arată că h este funcție liniară, care, combinată cu proprietățile de mai sus, conduce la $\text{Hom}_F(U, V) \cong {}^m F^n$.

S-a obținut astfel următorul rezultat important.

Teorema 6.3.1. Fie U și V două spații vectoriale peste un corp F , de dimensiune m și, respectiv, n . Atunci $\text{Hom}_F(U, V) \cong {}^m F^n$ și, deci, $\dim(\text{Hom}_F(U, V)) = mn$.

6.4. Sume directe și spații vectoriale cât

Fie V un spațiu vectorial peste un corp F și $U_1, \dots, U_k \leq V$, unde $k \geq 2$. Definim $U_1 + \dots + U_k$ prin

$$U_1 + \dots + U_k = \{x_1 + \dots + x_k | (\forall 1 \leq i \leq k)(x_i \in U_i)\}.$$

Este ușor de văzut că $U_1 + \dots + U_k \leq V$.

Dacă $U_1 + \dots + U_k = V$, atunci vom spune că V este suma subspațiilor vectoriale U_1, \dots, U_k .

Definiția 6.4.1. Fie V un spațiu vectorial peste un corp F și $U_1, \dots, U_k \leq V$. Spunem că V este suma directă a subspațiilor U_1, \dots, U_k , și notăm

$$V = U_1 \oplus \dots \oplus U_k,$$

dacă orice element $x \in V$ se scrie unic în forma

$$x = x_1 + \dots + x_k,$$

unde $x_i \in U_i$, pentru orice $1 \leq i \leq k$.

Unicitatea scrierii elementelor $x \in V$ ca în Definiția 6.4.1 poate fi exprimată și prin

$$\sum x_i = \sum y_i \Rightarrow (\forall 1 \leq i \leq k)(x_i = y_i),$$

sau prin

$$\sum x_i = 0 \Rightarrow (\forall 1 \leq i \leq k)(x_i = 0),$$

pentru orice $x_i, y_i \in U_i$, $1 \leq i \leq k$.

Teorema 6.4.1. Fie V un spațiu vectorial peste un corp F și U_1, \dots, U_k subspații ale lui V , unde $k \geq 2$. Dacă V este finit dimensional și $V = U_1 \oplus \dots \oplus U_k$, atunci $\dim(V) = \dim(U_1) + \dots + \dim(U_k)$.

Demonstrație. Fie B_i o bază pentru U_i , $1 \leq i \leq k$. Dacă arătăm că $B_i \cap B_j = \emptyset$, pentru orice $i \neq j$, și $B = \bigcup_{i=1}^k B_i$ este bază pentru V , atunci vom obține afirmația din teoremă.

Fie i și j cu $i < j$. Dacă am presupune că există $x \in B_i \cap B_j$, atunci elementul $2x \in V$ ar avea cel puțin două scrieri distincte în $U_1 \oplus \dots \oplus U_k$,

$$\begin{aligned} 2x &= 0 + \dots + 0 + \underbrace{2x}_{i} + 0 + \dots + 0 + \underbrace{0}_{j} + 0 + \dots + 0 \\ &= 0 + \dots + 0 + \underbrace{x}_{i} + 0 + \dots + 0 + \underbrace{x}_{j} + 0 + \dots + 0, \end{aligned}$$

ceea ce ar contrazice ipoteza. Ca urmare, $B_i \cap B_j = \emptyset$, pentru orice $i \neq j$.

Este clar că orice vector din V este generat de B . Ca urmare a unicării scrierii vectorilor din V ca sumă de vectori din U_i , $1 \leq i \leq k$, obținem că orice vector din V se scrie unic ca o sumă de combinații liniare, fiecare combinație liniară fiind peste exact o bază B_i . Deci orice vector din V se scrie unic ca o combinație liniară de vectori din B , ceea ce arată că B este bază pentru V . \square

Este interesant de știut când un spațiu vectorial se poate scrie ca sumă directă de subspații vectoriale. Pentru cazul $k = 2$ există un criteriu destul de simplu.

Teorema 6.4.2. Fie V un spațiu vectorial peste un corp F și U_1, U_2 subspații ale lui V . Atunci $V = U_1 \oplus U_2$ dacă și numai dacă $U_1 + U_2 = V$ și $U_1 \cap U_2 = \{0\}$.

Demonstrație. Dacă $V = U_1 \oplus U_2$, atunci este clar că $U_1 + U_2 = V$ și $U_1 \cap U_2 = \{0\}$.

Reciproc, va trebui să arătăm că orice sumă $x + y$ cu $x \in U_1$ și $y \in U_2$ este unică. Sau, echivalent, dacă presupunem $x + y = 0$, atunci $x = 0$ și $y = 0$. Relația $x + y = 0$ ne spune că y este invers aditiv al lui x . Însă $y \in U_2$ și $U_1 \cap U_2 = \{0\}$. Ca urmare, unicitatea inversului ne conduce la $x = 0$ și $y = 0$. \square

Atunci când un spațiu vectorial poate fi scris în forma $V = U_1 \oplus U_2$, vom spune că U_2 este un complement al lui U_1 în V .

Corolarul 6.4.1. Fie V un spațiu vectorial finit dimensional. Atunci orice subspațiu U al lui V admite un complement în V .

Demonstrație. Presupunem că V este un spațiu vectorial n -dimensional, $A = \{x_1, \dots, x_n\}$ este o bază pentru V , iar $U \leq V$. Presupunem că U este diferit de spațiul trivial (caz în care complementul ar fi V) și de spațiul V (caz în care complementul ar fi spațiul trivial).

Orice $n + 1$ vectori din U sunt liniar dependenți și, deci, există o bază pentru U . Fie $B = \{y_1, \dots, y_m\}$ o astfel de bază.

Aplicând Lema de schimb a lui Steinitz, putem găsi o nouă bază pentru V , fie aceasta C , care să includă B . Fie atunci $D = C - B$. D este nevidă și generează un subspațiu U' al lui V . Arătăm că U' este complement al lui U . În adevăr, $V = U + U'$ și $U \cap U' = \{0\}$. Prima relație decurge de la modul de construcție a bazei D , iar a doua de la faptul că D și B sunt disjuncte iar C este multime de vectori liniar independenți. \square

Teorema 6.4.3. Fie V un spațiu finit dimensional și $U_1, U_2 \leq V$. Atunci

$$\dim(U_1 + U_2) = \dim(U_1) + \dim(U_2) - \dim(U_1 \cap U_2).$$

Demonstrație. Fie $U = U_1 \cap U_2$. U este subspațiu vectorial atât în U_1 , cât și în U_2 . Fie W_1 complementul lui U în U_1 , și W_2 complementul lui U în U_2 .

Dacă arătăm că are loc

$$U_1 + U_2 = W_1 \oplus W_2 \oplus U,$$

atunci obținem

$$\begin{aligned} \dim(U_1 + U_2) &= \dim(W_1) + \dim(W_2) + \dim(U) \\ &= \dim(U_1) + \dim(U_2) - \dim(U), \end{aligned}$$

ceea ce va încheia demonstrația.

Este clar că orice element din $U_1 + U_2$ se scrie ca sumă de elemente din W_1, W_2 și U . Ne rămâne de demonstrație unicitatea scrierii.

Fie $y_1 \in W_1, y_2 \in W_2$ și $z \in U$ astfel încât $y_1 + y_2 + z = 0$. De aici urmează $y_1 = -(y_2 + z)$, care, combinată cu $y_1 \in U_1$ și $-(y_2 + z) \in U_2$, conduce la $y_1 \in U$. Deci, $y_1 \in W_1 \cap U = \{0\}$, ceea ce arată că $y_1 = 0$. Un raționament similar conduce la $y_2 = 0$, și apoi la $z = 0$. \square

Orice subspațiu U al unui spațiu vectorial V este subgrup al grupului comutativ al spațiului gazdă. Ca urmare, U este subgrup normal și, deci, induce o congruență \sim_U în grupul $(V, +, -, 0)$. Această congruență este compatibilă și cu înmulțirea cu scalari. În adevăr, dacă $x \sim_U y$, atunci există $z \in U$ astfel încât $y = x + z$. Pentru orice $\alpha \in F$, $\alpha y = \alpha x + \alpha z$, ceea ce arată că $\alpha x \sim_U \alpha y$ (atrăgem atenția că $\alpha z \in U$ deoarece $U \leq V$).

Deci \sim_U este congruență în spațiul vectorial V . Ca urmare, putem discuta despre *spațiul cât* V/\sim_U , care se notează în mod ușual prin V/U . Elementele acestui spațiu vectorial sunt clase de echivalență induse de \sim_U , ce au forma $x+U$, cu $x \in V$. Funcția $f : V \rightarrow V/U$ dată prin $f(x) = x+U$, pentru orice $x \in V$, este epimorfism.

6.5. Produs scalar. Ortogonalitate. Spațiu dual

Definiția 6.5.1. Fie V un spațiu vectorial peste un corp F . Se numește *produs scalar* pe V orice funcție $\langle \cdot, \cdot \rangle : V^2 \rightarrow F$ ce satisfac:

- (1) $\langle x, y \rangle = \langle y, x \rangle$, pentru orice $x, y \in V$;
- (2) $\langle x, y+z \rangle = \langle x, y \rangle + \langle x, z \rangle$, pentru orice $x, y, z \in V$;
- (3) $\langle \alpha x, y \rangle = \alpha \langle x, y \rangle$ și $\langle x, \alpha y \rangle = \alpha \langle x, y \rangle$, pentru orice $x, y \in V$ și $\alpha \in F$.

Definiția 6.5.2. Fie V un spațiu vectorial peste un corp F . Un produs scalar $\langle \cdot, \cdot \rangle$ pe V este numit *nedegenerat* dacă are loc:

$$(\forall x \in V)((\forall y \in V)(\langle x, y \rangle = 0) \Rightarrow x = 0).$$

Exemplul 6.5.1. Fie spațiu vectorial $V = F^n$, unde F este un corp și $n \geq 1$. Dacă considerăm funcția ce asociază oricărora doi vectori $x = (x_1, \dots, x_n)$ și $y = (y_1, \dots, y_n)$ scalarul $\sum x_i y_i$, atunci aceasta este produs scalar nedegenerat pe V .

Definiția 6.5.3. Fie V un spațiu vectorial peste un corp F , împreună cu un produs scalar $\langle \cdot, \cdot \rangle$ pe V . Spunem că doi vectori $x, y \in V$ sunt *ortogonali* sau *perpendiculari*, și notăm $x \perp y$, dacă $\langle x, y \rangle = 0$.

Observația 6.5.1. Fie V un spațiu vectorial peste un corp F , împreună cu un produs scalar $\langle \cdot, \cdot \rangle$ pe V . Dacă $x \in V$ este ortogonal pe vectorii $y_1, \dots, y_k \in V$, atunci el este ortogonal pe orice combinație liniară a lor.

Dat un spațiu vectorial V și S o submulțime nevidă a acestuia, vom nota prin S^\perp mulțimea tuturor vectorilor din V ce sunt ortogonali pe toți vectorii din S . Adică

$$S^\perp = \{y \in V | (\forall x \in S)(x \perp y)\}.$$

Este trivial de verificat, utilizând definiția produsului scalar, că S^\perp este subspațiu vectorial al spațiului vectorial V . El se numește *spațiu ortogonal* asociat lui (indus de) S .

Definiția 6.5.4. Fie V un spațiu vectorial finit dimensional peste un corp F , împreună cu un produs scalar $\langle \cdot, \cdot \rangle$ pe V . Se numește *bază ortogonală* a spațiului V orice bază cu proprietatea că orice doi vectori distincți ai ei sunt ortogonali.

Teorema 6.5.1. Orice spațiu vectorial finit dimensional admite o bază ortogonală.

Demonstrație. Vom demonstra teorema prin inducție matematică după dimensiunea spațiului vectorial:

- pentru spații vectoriale de dimensiune 1, teorema este trivial satisfăcută;
- presupunem teorema adevărată pentru spații vectoriale de dimensiune cel mult $n-1$, unde $n > 1$, și fie V un spațiu vectorial de dimensiune n . Vom considera două cazuri:

1. $\langle x, x \rangle = 0$, pentru orice $x \in V$. În acest caz, orice doi vectori din V sunt ortogonali deoarece relația

$$\langle x+y, x+y \rangle = \langle x, x \rangle + \langle y, y \rangle + 2\langle x, y \rangle$$

conduce la $\langle x, y \rangle = 0$. Ca urmare, orice bază a spațiului V este bază ortogonală;

2. există $x \in V$ cu $\langle x, x \rangle \neq 0$. Fie U spațiu ortogonal indus de x . Vom arăta că are loc $V = U \oplus U^\perp$, și pentru aceasta vom apela la Teorema 6.4.2. Fie $y \in V$. Considerăm scalarul $\alpha = \langle y, x \rangle \langle x, x \rangle^{-1}$ (care este bine definit conform ipotezei) și vectorul $z = y - \alpha x$. Un calcul simplu ne arată că $z \in U$, $\alpha x \in U^\perp$ și $y = z + \alpha x$. Deci $V = U + U^\perp$. Deoarece $\dim(U) = 1$ și $x \notin U$ (conform ipotezei) dar $x \in U^\perp$, $U \cap U^\perp$ nu poate fi decât subspațiuul $\{0\}$. Deci Teorema 6.4.2 conduce la $V = U \oplus U^\perp$.

Ca urmare a relației $V = U \oplus U^\perp$, $\dim(U^\perp) = n-1$. Ipoteza inducțivă ne spune că U^\perp admite o bază ortogonală, și fie $\{x_2, \dots, x_n\}$ o astfel de bază. Atunci este trivial de văzut că $\{x, x_2, \dots, x_n\}$ este o bază pentru V (a se vedea demonstrația Teoremei 6.4.1) și, în plus, ea este ortogonală conform construcției.

Conform principiului inducției matematice, teorema este demonstrată. \square

Procedeul de construcție a bazei ortogonale din Teorema 6.5.1 se numește *procedeul de ortogonalizare Gram-Schmidt*.

Orice corp F poate fi privit ca spațiu vectorial peste el însuși, așa cum s-a văzut în Exemplul 6.1.1(1), de dimensiune 1. Ca urmare, dat un spațiu vectorial V peste F , putem considera mulțimea tuturor funcțiilor liniare de la V la F care formează spațiu vectorial peste F (Secțiunea 6.3). Acest spațiu vectorial se mai notează prin V^* și se numește *spațiu vectorial dual* spațiului V .

Direct de la Teorema 6.3.1 obținem:

Corolarul 6.5.1. Pentru orice spațiu vectorial finit dimensional V peste un corp F are loc $\dim(V^*) = \dim(V)$.

Fie V un spațiu n -dimensional peste un corp F , și fie $B = \{x_1, \dots, x_n\}$ o bază a acestuia. Pentru orice $1 \leq i \leq n$, considerăm funcția liniară $\varphi_i : V \rightarrow F$ dată prin

$$\varphi_i(x_j) = \begin{cases} 1, & \text{dacă } j = i \\ 0, & \text{dacă } j \neq i \end{cases}$$

pentru orice $x_j \in B$. $\varphi_i(v)$ furnizează coordonata corespunzătoare lui x_i (în raport cu baza B) din unica scriere a lui v ca o combinație liniară a vectorilor bazei.

Este ușor de văzut că $\{\varphi_1, \dots, \varphi_n\}$ formează o bază pentru V^* . Această bază poartă denumirea de *duala bazei* B , și va fi notată prin B^\perp .

Stabilim acum următorul rezultat important.

Teorema 6.5.2. Fie V un spațiu n -dimensional peste un corp F și W un subspațiu netrivial al lui V . Atunci $\dim(W) + \dim(W^\perp) = n$.

Demonstrație. Fie W un subspațiu netrivial al lui V și $B = \{x_1, \dots, x_m\}$ o bază a lui. Extindem B la o bază $B' = \{x_1, \dots, x_m, x_{m+1}, \dots, x_n\}$ a spațiului V , și fie $B'^\perp = \{\varphi_1, \dots, \varphi_{m+1}, \dots, \varphi_n\}$ duala bazei B' .

Vom arăta că $\{\varphi_{m+1}, \dots, \varphi_n\}$ este bază a subspațiului vectorial W' al lui V^* , unde

$$W' = \{\varphi \in V^* \mid \varphi(W) = \{0\}\}$$

(se poate verifica imediat că W' este, în adevăr, subspațiu vectorial al lui V^*). De la definiția funcției φ_i obținem $\varphi_i(W) = \{0\}$, pentru orice $m+1 \leq i \leq n$, ceea ce arată că $\{\varphi_{m+1}, \dots, \varphi_n\} \subseteq W'$. Orice element $\varphi \in W'$ poate fi scris în forma

$$\varphi = a_1\varphi_1 + \dots + a_n\varphi_n.$$

Cum $\varphi(W) = \{0\}$ și $B \subseteq W$, obținem $a_i = \varphi(x_i) = 0$, pentru orice $1 \leq i \leq m$. Deci,

$$\varphi = a_{m+1}\varphi_{m+1} + \dots + a_n\varphi_n,$$

ceea ce arată că B'^\perp este bază pentru W' . Ca urmare, $\dim(W') = n - \dim(W)$.

Vom arăta acum că W' și W^\perp sunt izomorfe, ceea ce va încheia demonstrația. Aceasta rezultă însă cu ușurință dacă vom considera funcția $\psi : W^\perp \rightarrow W'$ dată prin $\psi(v) = f_v$, unde $f_v : V \rightarrow F$ este dată prin $f_v(w) = \langle v, w \rangle$, pentru orice $v \in W^\perp$ și $w \in V$. Nu vom argumenta decât buna definire a funcției ψ . Fie $v \in W^\perp$. Atunci $\langle v, w \rangle = 0$, pentru orice $w \in W$, ceea ce arată că $f_v(W) = \{0\}$ și, deci, $f_v \in W'$. \square

6.6. Aplicații: coduri detectoare și corectoare de erori

În Secțiunea 3.5 s-a introdus conceptul de cod de lungime variabilă și s-au prezentat unele rezultate de caracterizare și aplicații ale acestor codurilor în teoria compresiei datelor.

Codurile bloc sunt coduri pentru care toate cuvintele cod au aceeași lungime. Codul ASCII pe 8 biți este un astfel de cod. Aceste tipuri de coduri au aplicații majore în codificarea informației ce urmează a fi transmisă pe canale ce o pot altera. Atmosfera este un canal care poate altera informația (prin factori perturbatori ca vând, ploaie etc.). Interesul major în astfel de situații este de a detecta biții alterați și, eventual, de a-i corecta. Codurile detectoare și corectoare de erori, care sunt coduri bloc, au acest scop.

Rolul acestei secțiuni este de a introduce cititorul într-un mod facil în teoria codurilor detectoare și corectoare de erori. Vom începe cu exemple care să motiveze temeinic necesitatea acestor tipuri de coduri, și vom avansa spre clasa codurilor liniare care sunt, de fapt, subspații vectoriale peste un corp finit. Prin aceasta vom arăta importanța și necesitatea înțelegerei conceptelor de spațiu vectorial și corp finit. Atragem atenția că această secțiune este departe de a constitui un studiu complet al codurilor detectoare și corectoare de erori, dar constituie o bună introducere în teoria acestora. Pentru aprofundarea acestei teorii recomandăm [141, 126, 90, 140].

6.6.1. Introducere

6.6.1.1. Transmiterea informației prin canale cu zgomot

Din punct de vedere al transmisiei informației, teoria codurilor implică existența unui *emițător* (*codor, codificator*), a unui *receptor* (*decoder, decodificator*) și a unui *canal de transmisie* (*comunicație*) a mesajelor de la emițător la receptor. Emițătorul, receptorul și canalul de transmisie pot avea naturi diverse:

- stație satelit, stație pământ, atmosferă;
- aparat de emisie, aparat de recepție, cablu telefonic.

Este căt se poate de natural a presupune imperfecțiunea canalului de transmisie, ce altereză într-o anumită măsură informația transmisă prin el. Factorii care conduc la alterarea informației ce se transmite prin astfel de canale vor fi numiți, în mod generic, *zgomot*. Deci orice canal de transmisie a informației este supus unor factori de perturbare externi, deci zgomotului. Schematic, transmiterea informației poate fi reprezentată ca în diagrama din Figura 6.1.

În practică, controlul asupra zgomotului se face prin alegerea unui canal adecvat de transmisie și/sau utilizarea unor filtre de combatere a zgomotelor. Acestea sunt însă probleme ingineresti și ele nu constituie obiectul prezentării noastre. Premiza de la care vom pleca noi este aceea că în practică nu întâlnim canale perfecte și, ca urmare, problema noastră constă în stabilirea unor coduri pentru care să avem îndeplinite următoarele deziderate:

- (1) codificare rapidă a informației;

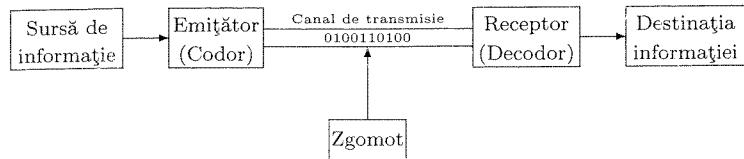


Figura 6.1: Transmiterea informației prin canale cu zgomot

- (2) transmisie ușoară a mesajelor codificate;
- (3) decodificare rapidă a mesajelor;
- (4) corectarea erorilor generate de canalul de transmisie;
- (5) transfer maxim de informație în unitatea de timp.

Obiectivul major din cele cinci menționate mai sus este al patrulea. El poate să nu fie, și în general nu este, compatibil cu celelalte. Adică, un cod care permite corectarea erorilor într-o măsură mare poate să nu conducă la codificare sau decodificare optimă, sau poate să conducă la transfer greoi de informație pe canal.

Pentru a facilita și consolida înțelegerea, în toate subsecțiunile Secțiunii 6.6.1 vom considera coduri peste corpul $F_2 = \{0, 1\}$ (în Secțiunea 6.6.2 vom trece la coduri peste un corp F_q arbitrar). Dacă $x \in \{0, 1\}$, atunci prin \bar{x} vom nota bitul din $\{0, 1\} - \{x\}$ (*complementul modulo 2 al bitului x*).

Reamintim că un *cod binar* este o mulțime nevidă C de cuvinte peste $F_2 = \{0, 1\}$ cu proprietatea de decodificare unică (Definiția 3.5.1.2(1)). *Codurile bloc* sunt mulțimi nevide C de cuvinte ce satisfac

$$(\forall c_1, c_2 \in C)(|c_1| = |c_2|).$$

Evident, astfel de mulțimi de cuvinte au proprietatea de decodificare unică și, deci, sunt coduri în sensul Definiției 3.5.1.1. Numărul natural n cu proprietatea $|c| = n$, pentru orice $c \in C$, se numește *lungimea codului C*.

Întrucât toată Secțiunea 6.6 se referă numai la coduri bloc, vom simplifica terminologia la cea de cod, înțelegând că este vorba despre coduri bloc.

Canalele de transmisie utilizate pentru transmiterea de mesaje codificate binar vor fi numite *canale binare*. Acestea pot fi clasificate în:

- *canale fără zgomot*, ceea ce înseamnă că bitul $x \in \{0, 1\}$ transmis este bitul x recepționat (informația nu este alterată);
- *canale cu zgomot*, ceea ce înseamnă că bitul $x \in \{0, 1\}$ transmis este bitul x recepționat cu probabilitatea $p_x \in (0, 1)$. Altfel spus, nu există certitudinea că x transmis este x recepționat. De asemenea, observăm că 0 și 1 pot avea probabilități diferite de păstrare a intactității lor. Ca urmare, putem face o nouă clasificare a canalelor:

- *simetrice*, unde cei doi biți au aceeași probabilitate de păstrare a intactității la transmisie;
- *asimetrice*, care nu sunt simetrice.

Canalele fără zgomot mai sunt numite și canale *perfecte*, dar, aşa cum am spus, astfel de canale nu prea apar în practică de zi cu zi¹. Noi vom considera în continuare canale binare simetrice, abreviat *canale BSC*, și vom face suplimentar următoarele presupuneri:

- o secvență de m biți transmisă printr-un astfel de canal este recepționată tot ca o secvență de m biți (canalul nu modifică lungimea secvenței);
- ordinea de transmisie a bițiilor coincide cu ordinea de recepționare a lor, astfel că nu există nici o problemă în decodificarea mesajului, exceptând cazul în care anumiți biți au fost alterați de canal;
- alterarea unui bit nu conduce la alterarea următorilor (a vecinilor săi). Altfel spus, nu există erori în *cascadă*.

Probabilitatea de păstrare a intactității bițiilor printr-un canal BSC nu depinde de bit în sine; această probabilitate este numită *fiabilitatea* sau *acuratețea* canalului. Deci, fiabilitatea unui canal BSC este un număr real $p \in (0, 1)$ care reprezintă probabilitatea ca bitul $x \in \{0, 1\}$ transmis să fie bitul x recepționat. Ca urmare, $1 - p$ va fi probabilitatea ca bitul x transmis să fie bitul \bar{x} recepționat. Diagrama din Figura 6.2 arată schematic modul în care operează un canal BSC în transmisia informației.

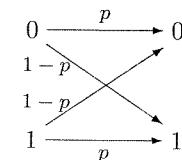


Figura 6.2: Canal BSC

Exprimarea fiabilității unui canal în termeni de probabilitate ne permite să spunem că un canal este mai fiabil decât un alt canal. Canalele cu fiabilitatea $p = 1$ sau $p = 0$ vor fi excluse din studiile noastre (pentru cele cu $p = 0$, bitul x transmis este exact bitul \bar{x} recepționat). De asemenea, orice canal cu fiabilitatea $0 < p \leq 1/2$ poate fi ușor convertit într-un canal cu fiabilitatea $1/2 < p < 1$ (se înlocuiește 0 cu 1 și 1 cu 0). Deci, canalele considerate de noi vor fi canale BSC cu fiabilitatea $1/2 < p < 1$.

¹Teoria codurilor detectoare și corectoare de erori ar fi inutilă dacă toate canalele ar fi perfecte.

6.6.1.2. Detectia și corecția erorilor

Prezentăm într-o manieră intuitivă ce înseamnă a detecta și corecta o eroare de transmisie. Considerăm codul

$$C_1 = \{000000, 010101, 101010, 111111\}$$

și presupunem că printr-un canal BSC s-a transmis un cuvânt cod și s-a recepționat cuvântul $w = 110101$. Receptorul detectează o eroare în transmisie (receptorul cunoaște codul utilizat în transmisie fără a ști însă care este cuvântul cod transmis). Corecția lui w poate fi realizată în 4 moduri diferite (C_1 are 4 elemente) dar, dintr-un anumit punct de vedere, este natural să corectăm w la cel mai apropiat cuvânt cod din C_1 .

Să considerăm acum codul

$$C_2 = \{00, 01, 10, 11\}.$$

Este ușor de constatat că orice transmisie s-ar realiza printr-un canal BSC, nu se detectează nici o eroare. Aceasta nu înseamnă că nu există alterări ale mesajelor pe parcurs, ci că ele nu pot fi detectate. Evident, un cod ca C_2 nu este de preferat pentru că nu avem nici un control asupra transmisiei informației. Modificăm C_2 adăugând la fiecare cuvânt cod al lui câte un bit, la sfârșit, astfel încât numărul de biți 1 în fiecare cuvânt cod să fie par. Obținem astfel codul

$$C_3 = \{000, 011, 101, 110\}.$$

Bitul adăugat se numește *bit de verificare a parității* sau *bit de paritate*. Să presupunem acum că la transmisia unui cuvânt cod s-a recepționat cuvântul $v = 010$. Evident, s-a detectat o eroare. Candidații pentru cuvântul cod transmis pot fi oricare din cuvintele 110, 000 sau 011. Evident, și 101 poate fi un candidat dar, deocamdată, pare mai natural să corectăm v la unul din primele 3 cuvinte menționate.

Ce câștigăm și ce pierdem prin utilizarea lui C_3 în locul lui C_2 ? Ca să răspundem la această întrebare introducem următorul concept.

Definiția 6.6.1.1. Numim *rata informației* codului C de lungime n numărul

$$ri(C) = \frac{\log_2 |C|}{n}.$$

Deoarece $1 \leq |C| \leq 2^n$, este clar că rata informației codului C este un număr real între 0 și 1. Ea este 1 dacă fiecare cuvânt de lungime n este cuvânt cod, și 0 dacă $|C| = 1$.

Dar ce măsoară rata informației? $|C|$ reprezintă numărul de cuvinte cod din C și, totodată, numărul de caractere ale alfabetului peste care se definesc mesajele ce urmează a fi codificate. Numărul n reprezintă numărul de biți utilizati de codul C pentru a codifica un caracter (simbol) arbitrar al alfabetului, iar $\lfloor \log_2 |C| \rfloor + 1$ reprezintă numărul maxim de biți ce s-ar putea utiliza pentru a codifica un caracter arbitrar

al alfabetului. Ca urmare, rata informației reprezintă cantitatea de informație reală într-un cuvânt cod (ce proporție dintr-un cuvânt cod reprezintă informație codificată, restul fiind introdusă din rațiuni tehnice).

Revenind acum la întrebarea de mai sus, putem spune următoarele. Rata informației codului C_2 este 1, deci un cuvânt cod conține în totalitate informație reală. Rata informației codului C_3 este $2/3$, ceea ce înseamnă că doar două treimi din orice cuvânt cod al lui este informație reală (de fapt, doi biți sunt utilizati pentru a codifica informația transmisă, iar al treilea este adăugat din rațiuni tehnice). Ca urmare, C_3 este mai puțin eficient decât C_2 , dar el poate detecta orice eroare singulară în timp ce C_2 nu poate detecta nici una.

Este, în adevăr, important să adăugăm informație suplimentară cuvintelor cod pentru a detecta erori? Vom răspunde la această întrebare considerând următorul exemplu.

Presupunem că avem un cod C de lungime 11 ce conține toate cele 2^{11} cuvinte. Așa cum am văzut, transmisia cu un astfel de cod nu conduce la detectare de erori. Să presupunem că fiabilitatea canalului este $p = 1 - 10^{-8}$ (un bit din 10^8 biți transmis este, în medie, alterat) și rata de transmisie a biți este de 10^7 biți/secundă. Atunci, probabilitatea ca un cuvânt cod să fie transmis doar cu o eroare este $11p^{10}(1-p)$ (probabilitatea ca 10 biți din cei 11 să nu fie alterați este p^{10} , iar probabilitatea ca un bit din cei 11 să fie alterat este $1-p$; dar, acest bit alterat poate fi oricare din cei 11 biți). Un calcul simplu ne arată că $11p^{10}(1-p)$ este aproximativ $11/10^8$. Deci

$$\frac{11}{10^8} \cdot \frac{10^7}{11} = 0.1 \text{ cuvinte cod/secundă}$$

vor fi transmise cu exact o eroare care nu va putea fi detectată. Aceasta înseamnă un cuvânt cod eronat la 10 secunde, 6 la un minut, 360 într-o oră, 8640 într-o zi. Deci nu este prea bine.

Să presupunem acum că adăugăm codului C bitul de paritate și obținem codul C' . Cele $2^{11} = 2048$ cuvinte ale lui C' vor avea acum lungimea 12. Codul C' detectează erorile singulare, dar nu va putea detecta alterările a cel puțin doi biți în același cuvânt cod. Probabilitatea ca cel puțin 2 biți să fie alterați, în același cuvânt cod transmis, se obține scăzând din 1 probabilitatea ca nici un bit să fie alterat (p^{12}) și probabilitatea ca exact un bit să fie alterat ($12p^{11}(1-p)$). Deci, această probabilitate este

$$1 - p^{12} - 12p^{11}(1-p) \approx C_{12}^2 p^{10}(1-p)^2 \approx \frac{66}{10^{16}}.$$

Dar aceasta înseamnă că aproximativ

$$\frac{66}{10^{16}} \cdot \frac{10^7}{12} \approx \frac{5.5}{10^9} \text{ cuvinte cod/secundă}$$

vor fi transmise cu erori ce nu vor putea fi detectate. Adică, un cuvânt cod la aproximativ 2000 de zile, ceea ce pare a fi mult mai bine.

Ce se întâmplă însă cu rata informației codului C' ? Evident că aceasta va scădea ceea ce ne va spune că în unitatea de timp se transmit mai puține informații.

6.6.1.3. Determinarea cuvântului cod transmis

Fie C un cod de lungime n , $w \in \{0, 1\}^n$ și $v \in C$. Fie d numărul de poziții pe care diferă cuvintele w și v . Atunci probabilitatea ca v să fi fost transmis atunci când w a fost recepționat este

$$\phi_p(v, w) = p^{n-d}(1-p)^d,$$

unde p este fiabilitatea canalului (adică probabilitatea ca $(n-d)$ biți să fi fost transmiși intacți, și d biți, alterați).

În practică cunoaștem w dar nu-l știm pe v . De cele mai multe ori v se alege satisfăcînd

$$\phi_p(v, w) = \max\{\phi_p(u, w) | u \in C\}.$$

Este clar că un astfel de v poate să nu fie unic. Alegerea lui v în această variantă este în strînsă legătură cu numărul de poziții pe care diferă v și w .

Teorema 6.6.1.1. Fie C un cod de lungime n , $v_1, v_2 \in C$ și $w \in \{0, 1\}^n$, iar d_1 și d_2 numărul de poziții pe care diferă v_1 și w și, respectiv, v_2 și w . Atunci

$$\phi_p(v_1, w) \leq \phi_p(v_2, w) \Leftrightarrow d_1 \geq d_2$$

(se presupune că fiabilitatea p a canalului satisfacă $1/2 < p < 1$).

Demonstrație. Au loc relațiile:

$$\begin{aligned} \phi_p(v_1, w) \leq \phi_p(v_2, w) &\Leftrightarrow p^{n-d_1}(1-p)^{d_1} \leq p^{n-d_2}(1-p)^{d_2} \\ &\Leftrightarrow \left(\frac{p}{1-p}\right)^{d_2-d_1} \leq 1 \\ &\Leftrightarrow d_2 \leq d_1 \end{aligned}$$

(ultima relație urmează de la $p/(1-p) > 1$). \square

Această teoremă nu ne spune altceva decât că corectarea erorilor în maniera adoptată de noi ca fiind naturală se bazează pe determinarea unui cuvânt cod care să difere de w pe cât mai puține poziții.

Problema care se pune acum este de a găsi un formalism adecvat pentru a exprima și studia concepțele de eroare și cel mai apropiat cuvânt cod față de un cuvânt dat. Pentru aceasta vom apela la spații vectoriale. Considerând corpul finit $F_2 = \{0, 1\}$, mulțimea F_2^n a tuturor cuvintelor de lungime n peste F_2 formează un spațiu vectorial peste F_2 (Exemplul 6.1.1(1)). Vom nota elementele lui F_2^n în manieră vectorială așa cum am făcut în Exemplul 6.1.1(1), sau prin cuvintele de lungime n peste F_2 . Ambele variante vor fi utilizate și nu vor crea ambiguități. Adunarea vectorilor și înmulțirea cu scalari sunt date prin:

- $x_1 \cdots x_n + y_1 \cdots y_n = (x_1 + y_1) \cdots (x_n + y_n);$
- $\alpha(x_1 \cdots x_n) = (\alpha \cdot x_1) \cdots (\alpha \cdot x_n),$

unde $\alpha, x_i, y_i \in F_2$, $x_i + y_i$ este adunarea modulo 2, iar $\alpha \cdot x_i$ este dată prin

$$0 \cdot 0 = 0 \cdot 1 = 1 \cdot 0 = 0 \text{ și } 1 \cdot 1 = 1.$$

Semnul operației \cdot va fi omis în cele mai multe cazuri. Vectorul zero va fi notat simplificat prin 0 în loc de $\mathbf{0}$ sau 0^n sau $(0, \dots, 0)$ (din context se va deduce clar atunci când este vorba de vectorul zero sau elementul zero al corpului F_2).

Definiția 6.6.1.2. Fie v un cuvînt peste $\{0, 1\}$. Pondere Hamming a lui v , notată $Hw(v)$, este dată de numărul de apariții a simbolului 1 în v .

Dacă s-a transmis cuvântul cod v și s-a recepționat w , atunci $v + w$ reprezintă cuvântul ce are 1 pe acele poziții pe care diferă v și w . Adică, $v + w$ semnalizează prin biții 1 locul unde a apărut o eroare de transmisie. De exemplu, dacă $v = 110110$ și $w = 010010$, atunci $v + w = 100100$. Deci, au apărut două erori de transmisie pe pozițiile 1 și 4. $v + w$ va fi numită *eroare de transmisie* sau, mai simplu, *eroare*. Ca urmare, dat un cod de lungime n , erorile posibile de transmisie sunt elementele lui F_2^n , diferite de 0.

Definiția 6.6.1.3. Fie v și w două cuvinte de aceeași lungime peste $\{0, 1\}$. Distanța Hamming dintre v și w , notată $Hd(v, w)$, este definită prin

$$Hd(v, w) = Hw(v + w).$$

Adică, distanța Hamming dintre v și w este ponderea Hamming a erorii $v + w$ sau, altfel spus, numărul de poziții pe care diferă cele două cuvinte. Astfel, formula ce dă probabilitatea ca la transmiterea lui v să se recepționeze w poate fi acum rescrisă la

$$\phi_p(v, w) = p^{n-Hw(u)}(1-p)^{Hw(u)},$$

unde $u = v + w$.

Propoziția 6.6.1.1. Au loc următoarele proprietăți:

- (1) $0 \leq Hw(v) \leq n$;
- (2) $Hw(v) = 0$ dacă și numai dacă $v = 0$;
- (3) $0 \leq Hd(v, w) \leq n$;
- (4) $Hd(v, w) = 0$ dacă și numai dacă $v = w$;
- (5) $Hd(v, w) = Hd(w, v)$;
- (6) $Hw(v + w) \leq Hw(v) + Hw(w)$;
- (7) $Hd(v, w) \leq Hd(v, u) + Hd(u, w)$;
- (8) $Hw(av) = aHw(v)$;

(9) $Hd(av, aw) = aHd(v, w)$,
pentru orice $u, v, w \in \{0, 1\}^n$ și $a \in \{0, 1\}$, unde $n \geq 1$.

Demonstrație. Vom demonstra doar (7), ca exemplu. Au loc relațiile:

$$\begin{aligned} Hd(v, w) &= Hw(v + w) \\ &= Hw(v + u + u + w) \\ &\leq Hw(v + u) + Hw(u + w) \\ &= Hd(v, u) + Hd(u, w), \end{aligned}$$

pentru orice $u, v, w \in \{0, 1\}^n$ (inegalitatea urmează de la (6)). \square

Putem trata acum cele două probleme principale ale practicii codurilor: codificarea, cu alegerea unui cod, și decodificarea, cu detectarea și corectarea erorilor.

Codificare. Să presupunem că avem un alfabet A peste care se construiesc mesajele ce urmează a fi transmise. Determinăm un număr natural k astfel încât $|A|^k \leq 2^k$. k biți vor fi suficienți pentru codificarea tuturor caracterelor din M . În continuare, va trebui să stabilim numărul de biți pe care îi vom adăuga fiecărui cuvânt de lungime k astfel încât să obținem un criteriu optim de detecție și corecție a erorilor. Evident, acesta nu este un pas simplu și construcția diverselor tipuri de coduri pornește de aici.

Decodificare. Să presupunem că se receptionează un cuvânt w . Decodificarea acestuia se poate face după una din următoarele două strategii:

Strategia completă, abreviată CMLD, decodifică w prin cel mai apropiat cuvânt cod (în sensul distanței Hamming), dacă acesta este unic; altfel, se alege arbitrar unul din ele;

Strategia incompletă, abreviat IMLD, decodifică w prin cel mai apropiat cuvânt cod, dacă acesta este unic; altfel, se cere o retransmisie. De asemenea se poate cere o retransmisie dacă cuvântul recepționat este “prea departe” de cuvintele cod.

Încercăm în continuare să exprimăm *fiabilitatea* principiului CMLD. De fapt, vom calcula probabilitatea pentru care strategia CMLD decide asupra cuvântului cod v ca fiind corect transmis. Această probabilitate o vom nota prin $\theta_p(C, v)$ (evident, ea depinde de fiabilitatea p a canalului și de codul C).

Fie $L(v)$ mulțimea tuturor cuvintelor din $\{0, 1\}^n$, n fiind lungimea codului C , care sunt la fel de apropiate de v , și cel mai apropiate. Remarcăm că $L(v)$ reprezintă exact mulțimea acelor cuvinte care, dacă ar fi receptionate, strategia CMLD ar decide asupra cuvântului cod v . Este clar atunci că are loc

$$\theta_p(C, v) = \sum_{w \in L(v)} \phi_p(v, w).$$

θ_p poate fi utilizată pentru a compara două coduri. Trebuie să notăm că în calculul lui θ_p este ignorată posibilitatea unei retransmisii.

6.6.1.4. Coduri detectoare și corectoare de erori

Codurile cu un singur cuvânt cod, numite și *coduri triviale*, sunt total neimportante în practică, deoarece ele pot codifica doar mesaje formate din repetări ale aceluiași simbol. Ca urmare, vom presupune, în tot ceea ce urmează de acum mai departe că orice cod are cel puțin două cuvinte cod.

Definiția 6.6.1.4. Fie C un cod de lungime n .

- (1) Spunem că C detectează eroarea $u \in \{0, 1\}^n - \{0^n\}$ dacă $v + u \notin C$, pentru orice $v \in C$.
- (2) Spunem că C este cod *t-detector de erori* dacă C detectează orice eroare a cărei pondere Hamming este cel mult t , dar există erori cu ponderea Hamming $t + 1$ ce nu pot fi detectate de C .

C detectează eroarea u dacă oricum am transmite un cuvânt cod v ce s-ar receptiona cu eroarea u , deci s-ar receptiona $w = v + u$, cuvântul w nu este în C , căci altfel nu am putem spune că a apărut eroarea u .

Este clar că putem efectiv verifica dacă un cod C detectează o anumită eroare u . După cum vom vedea, nici un cod nu poate detecta orice eroare.

Definiția 6.6.1.5. Fie C un cod. *Distanța codului* C , notată $d(C)$, este definită ca fiind

$$d(C) = \min\{Hd(v, w) | v, w \in C, v \neq w\}.$$

Distanța oricărui cod este cel puțin 1 (reamintim că, aşa cum am considerat mai sus, orice cod are cel puțin două cuvinte cod).

Teorema 6.6.1.2. Fie C un cod de lungime n și distanță d . Atunci au loc următoarele proprietăți:

- (1) C detectează orice eroare $u \in \{0, 1\}^n - \{0^n\}$ cu $Hw(u) \leq d - 1$;
- (2) există cel puțin o eroare $u \in \{0, 1\}^n - \{0^n\}$ cu $Hw(u) = d$ ce nu poate fi detectată de C .

Demonstrație. (1) Fie u o eroare astfel încât $0 < Hw(u) < d$. Atunci, pentru orice $v \in C$, are loc

$$Hd(v, v + u) = Hw(v + v + u) = Hw(u) < d,$$

ceea ce ne arată că $v + u \notin C$ deoarece C are distanță d . Deci codul C detectează eroarea u .

(2) Conform definiției distanței unui cod, există v și w cuvinte cod astfel încât $Hd(v, w) = d$. Considerăm eroarea $u = v + w$ ce are ponderea Hamming d și, utilizând egalitatea $w = v + u$, obținem că C nu detectează eroarea u a cărei pondere Hamming este d . \square

Teorema 6.6.1.2 ne spune că orice cod de dimensiune d este $(d - 1)$ -detector de erori.

Dacă cuvântul cod $v \in C$ a fost transmis pe un canal BSC și s-a recepționat w cu eroarea u atunci, în ipoteza în care w este cel mai aproape de v , strategia CMLD va concluziona că v a fost cel transmis. Dacă aceasta se va întâmpla ori de câte ori se înregistrează eroarea u , atunci vom spune că C corectează eroarea u .

Definiția 6.6.1.6. Fie C un cod de lungime n .

- (1) Spunem că C corectează eroarea $u \in \{0, 1\}^n - \{0^n\}$ dacă, pentru orice $v \in C$ și $w \in C - \{v\}$, are loc $Hd(v + u, v) < Hd(v + u, w)$.
- (2) Spunem că C este un cod t -corector de erori dacă C corectează toate erorile cu ponderea Hamming cel mult t , dar există erori cu ponderea Hamming $t + 1$ ce nu pot fi corectate de C .

Teorema 6.6.1.3. Fie C un cod de lungime n și distanță d . Atunci au loc următoarele proprietăți:

- (1) C corectează orice eroare $u \in \{0, 1\}^n - \{0^n\}$ cu $Hw(u) \leq \lfloor (d - 1)/2 \rfloor$;
- (2) există cel puțin o eroare $u \in \{0, 1\}^n - \{0^n\}$ cu $Hw(u) = \lfloor (d - 1)/2 \rfloor + 1$ ce nu poate fi corectată de C .

Demonstrație. (1) Fie u o eroare cu ponderea Hamming cel mult $\lfloor (d - 1)/2 \rfloor$. Atunci, pentru orice $v \in C$, are loc

$$Hd(v, v + u) = Hw(v + v + u) = Hw(u) \leq \lfloor (d - 1)/2 \rfloor < d/2,$$

ceea ce arată că $Hd(v, v + u) < Hd(w, v + u)$, pentru orice $w \in C - \{v\}$.

(2) Fie v și w cuvinte cod cu proprietatea $Hd(v, w) = d$. Considerăm cuvântul $v + w$. Acesta are exact d biți 1 (restul fiind 0). Modificăm $v + w$ lăsând exact $\lfloor (d - 1)/2 \rfloor + 1$ biți 1, restul transformându-i în 0. Fie u un cuvânt astfel obținut. El va avea ponderea Hamming $\lfloor (d - 1)/2 \rfloor + 1$. Arătăm că u nu poate fi corectat de C . În adevăr, pentru orice $v \in C$ și $w \in C - \{v\}$, are loc:

- dacă d este par, atunci

$$Hd(v, v + u) = Hw(u) = Hd(w, v + u);$$

- dacă d este impar, atunci

$$Hd(v, v + u) = Hw(u) > Hd(w, v + u).$$

Deci C nu poate corecta eroarea u . \square

6.6.1.5. Problema fundamentală a teoriei codurilor bloc

Concluzia pe care o desprindem din secțiunile anterioare este că am vrea un cod C care să satisfacă următoarele proprietăți:

- C să aibă o lungime n mică pentru a accelera transmisia informației;
- $|C|$ să fie destul de mare pentru a acoperi un alfabet cât mai larg;
- distanța d a codului să fie mare pentru a putea detecta și corecta cât mai multe erori.

Acești 3 parametri sunt în conflict: nu putem avea n mic și, în același timp, $|C|$ mare. *Problema fundamentală în teoria codurilor bloc* este de a optimiza unul din acești parametri pornind de la valori date ale celorlalți. În mod ușual, n și d se fixează, și atunci se caută cea mai mare valoare posibilă pentru $|C|$. Această valoare se notează prin $A_2(n, d)$ (indicele 2 este datorat faptului că lucrăm peste F_2). Atunci când vom lucra peste F_q vom schimba 2 în q .

Un cod cu parametrii n , $|C| = m$ și d va fi numit un *cod de tip* (n, m, d) . Este clar că dacă fixăm n și $d = 1$, atunci $m = 2^n$ este cea mai mare valoare posibilă pentru m . Similar, dacă fixăm $d = n$, atunci singura valoare posibilă pentru m este $m = 2$.

Teorema 6.6.1.4. Fie $n \geq 1$ și $1 \leq d \leq n$ un număr impar. Atunci există coduri de tip (n, m, d) dacă și numai dacă există coduri de tip $(n + 1, m, d + 1)$.

Demonstrație. Fie $n \geq 1$ și $1 \leq d \leq n$ un număr impar.

Presupunem că C este un cod de tip (n, m, d) . Construim un cod C' de lungime $n + 1$ extinzând cuvintele cod ale lui C prin adăugarea bitului de paritate, adică:

$$\begin{aligned} C' = & \{x_1 \cdots x_n 0 | x_1 \cdots x_n \in C \wedge Hw(x_1 \cdots x_n) \text{ este par}\} \\ & \cup \{x_1 \cdots x_n 1 | x_1 \cdots x_n \in C \wedge Hw(x_1 \cdots x_n) \text{ este impar}\}. \end{aligned}$$

Este clar că $|C'| = |C|$ și distanța codului crește cu cel mult o unitate față de cea a codului C . Cum distanța codului C' este număr par (conform construcției), iar d este impar, urmează că $d(C') = d + 1$, ceea ce demonstrează implicația în sens direct a teoremei.

Reciproc, fie C un cod de tip $(n + 1, m, d + 1)$, iar x și y cuvinte cod astfel încât $Hd(x, y) = d + 1$. Fixăm o poziție k , $1 \leq k \leq n + 1$, cu proprietatea că pe această poziție k cele două cuvinte cod au biți diferenți. Eliminăm bitul de pe poziția k din toate cuvintele cod ale lui C , obținând astfel un nou cod C' ce va avea lungimea n , distanța d și același număr de cuvinte cod ca și C . \square

Corolarul 6.6.1.1. Fie $n \geq 1$ și $1 \leq d \leq n$. Dacă d este impar, atunci $A_2(n, d) = A_2(n + 1, d + 1)$, iar dacă d este par, atunci $A_2(n, d) = A_2(n - 1, d - 1)$.

Demonstrație. Evident, este suficient să demonstrăm doar prima parte a corolarului. Presupunem că d este impar și fie C un cod de tip $(n, A_2(n, d), d)$. Dacă

ar exista un cod C' de tip $(n+1, A_2(n+1, d+1), d+1)$ pentru care ar avea loc $A_2(n+1, d+1) > A_2(n, d)$, atunci, conform Teoremei 6.6.1.4, ar exista un cod C'' de tip $(n, A_2(n+1, d+1), d)$, ceea ce ar constitui o contradicție (cea mai mare valoare posibilă pentru m , atunci când n și d sunt dați, este $A_2(n, d)$). \square

Determinarea valorii $A_2(n, d)$ nu este un lucru simplu. Tabelul de mai jos, preluat din [131] și completat cu rezultate noi din [195], prezintă câteva din valorile cunoscute pentru $A_2(n, d)$ (acolo unde sunt prezentate două limite, cum ar fi de exemplu 72–79, valoarea nu este cunoscută, dar se știe că ea se găsește între limitele respective).

n	$d = 3$	$d = 5$	$d = 7$
5	4	2	—
6	8	8	—
7	16	2	2
8	20	4	2
9	40	6	2
10	72 – 79	12	2
11	144 – 158	24	4
12	256	32	4
13	512	64	8
14	1024	128	16
15	2048	256	32
16	2560 – 3276	256 – 340	36 – 37

Exemplul 6.6.1.1. Conform tabelului de mai sus, $A_2(5, 3) = 4$. Următorul cod este de tip $(5, 4, 3)$ (el este specificat ca o matrice, linile acesteia fiind cuvinte cod)

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{matrix}$$

Aplicând acestui cod construcția din demonstrația Teoremei 6.6.1.4, obținem codul de tip $(6, 4, 4)$ de mai jos

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{matrix}$$

Să reluăm problema decodificării prezentată în secțiunea anterioară. Aceasta se bazează pe determinarea celui mai apropiat cuvânt cod față de cuvântul recepționat. Ideal este să existe întotdeauna exact un cel mai apropiat cuvânt cod. Vom investiga în continuare această posibilitate.

Definiția 6.6.1.7. Fie $x \in F_2^n$ și $r \geq 0$ un număr natural. Sfera de centru x și rază r este definită ca fiind mulțimea

$$S(x, r) = \{y \in F_2^n \mid Hd(x, y) \leq r\}.$$

Propoziția 6.6.1.2. Fie $x \in F_2^n$ și $0 \leq r \leq n$. Atunci,

$$|S(x, r)| = \sum_{i=0}^r C_n^i,$$

unde $C_n^i = n!/(i!(n-i)!)$, pentru orice n și $i \leq n$.

Demonstrație. Distanța dintre x și un cuvânt arbitrar $y \in F_2^n$ poate fi $0, 1, \dots, r$. Pentru fiecare i , $0 \leq i \leq r$, există exact C_n^i cuvinte y cu $Hd(x, y) = i$, deoarece există exact C_n^i moduri distincte de a modifica exact i biți ai cuvântului x . De aici urmează relația din propoziție. \square

Teorema 6.6.1.5. (Inegalitatea lui Hamming)

Orice cod de tipul $(n, m, 2t + 1)$ satisfacă relația

$$m \sum_{i=0}^t C_n^i \leq 2^n.$$

Demonstrație. Fie C un cod de tip $(n, m, 2t + 1)$. Pentru orice $x, y \in C$, dacă $x \neq y$, atunci $S(x, t) \cap S(y, t) = \emptyset$. Ca urmare, un vector din F_2^n se poate afla în cel mult o sferă de rază t centrată în unul din cuvintele cod ale lui C . Deci suma tuturor vectorilor din aceste sfere nu depășește numărul vectorilor din F_2^n . În baza Propoziției 6.6.1.2, aceasta înseamnă

$$m \sum_{i=0}^t C_n^i \leq 2^n,$$

ceea ce încheie demonstrația. \square

Exemplul 6.6.1.2. Orice cod de tip $(5, m, 3)$ satisfacă $m \cdot 6 \leq 32$.

Faptul că un număr natural m satisfacă inegalitatea lui Hamming relativ la alte două numere n și t (ca în Teorema 6.6.1.5), nu înseamnă că există un cod de tip $(n, m, 2t + 1)$.

Definiția 6.6.1.8. Un cod C de tip $(n, m, 2t + 1)$ este numit *perfect* dacă el satisfacă inegalitatea lui Hamming prin egalitate.

Evident, codurile perfecte sunt de preferat deoarece, orice cuvânt s-ar receptiona acesta se va găsi în exact o sferă centrată în unul din cuvintele cod. Ca urmare, cuvântul va putea fi decodificat prin “central” sferei.

Problema determinării de coduri perfecte s-a dovedit a fi una din cele mai dificile probleme din teoria codurilor bloc.

6.6.2. Coduri liniare

Tot ceea ce am prezentat în Secțiunea 6.6.1 poate fi extins cu ușurință la coduri peste un corp arbitrar F_q . Vom puncta câteva din elementele la care trebuie acordată atenție atunci când se face o astfel de extensie.

Un cod de lungime n peste F_q va fi o submulțime nevidă $C \subseteq F_q^n$. Vom presupune că orice cod are cel puțin două cuvinte cod (așa cum am făcut și în Secțiunea 6.6.1.4). Elementele lui F_q^n vor fi numite vectori sau *cuvinte de lungime n peste F_q* .

Ponderea Hamming a unui cuvânt $u \in F_q^n$, notată $Hw(u)$, este numărul de poziții din u diferite de 0. *Distanța Hamming* dintre u și v , notată $Hd(u, v)$, este ponderea Hamming a cuvântului $u - v$ ($-v$ este opusul lui v în F_q^n). În F_2^n , $u - v = u + v$, și acesta este un aspect la care cititorul trebuie să acorde atenție. *Distanța codului C*, notată $d(C)$, se definește și ea în mod ușual ca fiind

$$d(C) = \min\{Hd(u, v) | u, v \in C, u \neq v\}.$$

Conform convenției adoptate, $d(C) \geq 1$. Egalitatea din Propoziția 6.6.1.2 devine

$$|S(x, r)| = \sum_{i=0}^r C_n^i (q-1)^i,$$

iar inegalitatea lui Hamming,

$$m \sum_{i=0}^t C_n^i (q-1)^i \leq q^n.$$

Vom trece acum la prezentarea unei clase foarte importante de coduri, clasa *codurilor liniare*.

6.6.2.1. Definiții. Exemple. Proprietăți de bază

Definiția 6.6.2.1. Se numește *cod liniar de tip $[n, k]$ peste F_q* , unde $n \geq 1$ și $1 \leq k \leq n$, orice subspațiu vectorial $C \subseteq F_q^n$ de dimensiune k .

Dacă C este un cod liniar de tip $[n, k]$ peste F_q de dimensiune d , atunci vom mai spune că C este *cod de tip $[n, k, d]$* .

Observația 6.6.2.1. Fie C este un cod liniar de tip $[n, k, d]$ peste F_q .

- (1) Atragem atenția asupra notațiilor “[n, k, d]” și “(n, m, d)”. În prima, k reprezintă dimensiunea subspațiului vectorial C și nu numărul de vectori în C așa cum desemnează m în cea de a doua notație. Legătura între ele este dată de faptul că C are q^k vectori.
- (2) Faptul că C este subspațiu vectorial al spațiului vectorial F_q^n conduce la:

- $u + v \in C$, pentru orice $u, v \in C$;
- $-u \in C$, pentru orice $u \in C$ ($-u$ este opusul lui u în F_q^n);
- $0 \in C$;
- $\alpha u \in C$, pentru orice $u \in C$ și $\alpha \in F_q$.

- (3) Codul C poate fi specificat printr-o bază a acestuia, ce are k vectori liniari independenti. Dacă $B = \{x_1, \dots, x_k\}$ este o astfel de bază, atunci vectorii bazei pot fi distribuiți într-o matrice astfel încât fiecare din vectori formează o linie a matricei. Fără a restrâng generalitatea putem presupune că linia i a matricei este dată de vectorul x_i , pentru orice $1 \leq i \leq k$. Matricea astfel obținută poartă denumirea de *matrice generatoare* a codului C . Ea este o matrice de tip $k \times n$ peste F_q .

O proprietate importantă a codurilor liniare constă în aceea că distanța lor se poate determina inspectând cuvintele cod diferențe de 0.

Propoziția 6.6.2.1. Fie C un cod liniar de tip $[n, k]$ peste F_q . Atunci

$$d(C) = \min\{Hw(c) | c \in C - \{0\}\}.$$

Demonstrație. Conform definiției, $d(C) = \min\{Hd(u, v) | u, v \in C, u \neq v\}$. Fie $Hw(C) = \min\{Hw(c) | c \in C - \{0\}\}$. Atunci:

$$\begin{aligned} d(C) &= \min\{Hd(u, v) | u, v \in C, u \neq v\} \\ &= \min\{Hw(u - v) | u, v \in C, u \neq v\} \\ &\geq Hw(C) \end{aligned}$$

(inegalitatea urmează de la faptul că $u - v \in C$, pentru orice $u, v \in C$).

Pe de altă parte,

$$\begin{aligned} Hw(C) &= \min\{Hw(c) | c \in C - \{0\}\} \\ &= \min\{Hw(c - 0) | c \in C - \{0\}\} \\ &= \min\{Hd(c, 0) | c \in C - \{0\}\} \\ &\geq d(C). \end{aligned}$$

Ca urmare, $d(C) = Hw(C)$. □

Putem acum prezenta câteva avantaje și posibile dezavantaje ale codurilor liniare.

1. Printre *avantajele* codurilor liniare menționăm:

- (a) distanța lor poate fi determinată mult mai eficient decât a codurilor neliniare. Astfel, pentru un cod liniar cu m cuvinte cod vor fi necesare cel mult $m - 1$ calcule de ponderi Hamming, în timp ce pentru coduri neliniare cu m cuvinte sunt necesare $m(m - 1)/2$ calcule de ponderi Hamming;
- (b) codurile liniare pot fi specificate prin matrice generatoare, ceea ce constituie un mod destul de economic;
- (c) pentru coduri liniare există proceduri eficiente de codificare și decodificare (această afirmație va fi justificată în secțiunile următoare);
2. Ca posibile *dezavantaje* ale codurilor liniare menționăm:

- (a) deoarece codurile liniare sunt subspații ale unor spații vectoriale de tip F_q^n , q trebuie să fie o putere a unui număr prim. Aceasta ar putea crea anumite inconveniențe relativ la alegerea codului. De exemplu, dacă dorim un cod peste un alfabet cu 10 caractere atunci nu putem folosi în mod direct teoria codurilor liniare. Putem obține astfel de coduri din coduri liniare dacă, de exemplu, considerăm coduri peste F_{11} și restricționăm apoi cuvintele cod pentru a obține coduri peste un alfabet cu 10 caractere. Un exemplu de aplicare a acestei tehnici îl constituie codul ISBN care va fi prezentat mai jos;
- (b) restricția de liniaritate ar putea părea o restricție destul de tare. Însă practica dovedește că multe din codurile considerate eficiente din anumite puncte de vedere sunt coduri liniare.

Exemplul 6.6.2.1. Codul ISBN (International Standard Book Number) este un cod de lungime 10 peste un alfabet cu 11 caractere, $0, 1, \dots, 9, X$. Un cuvânt cod al acestui cod se scrie în forma (caracterul “–” este doar o linie de despărțire și nu operatorul “diferență”)

$$x_1 - x_2 x_3 - x_4 x_5 x_6 x_7 x_8 x_9 - x_{10},$$

unde:

- $x_1, \dots, x_9 \in \{0, \dots, 9\}$;
- x_1 indică limba utilizată;
- $x_2 x_3$ indică editura;
- $x_4 x_5 x_6 x_7 x_8 x_9$ indică numărul cărții în cadrul editurii;
- x_{10} este “cifra” de stabilire a parității, ce satisfacă:

$$\sum_{i=1}^{10} i x_i \equiv 0 \pmod{11}.$$

Atunci când $x_{10} = 10$, ea este înlocuită prin X .

Un astfel de cod detectează orice eroare singulară sau produsă prin inversarea a două cifre. În adevăr, fie $x = x_1 \cdots x_{10}$ un cuvânt cod (am omis liniile de despărțire în scrierea acestuia deoarece acestea nu au importanță în studiul codului). Dacă cifra x_i este schimbată, de exemplu în $x_i + a$ cu $a \neq 0$, atunci

$$\left(\sum_{j=1, j \neq i}^{10} j x_j \right) + i(x_i + a) \equiv ia \pmod{11},$$

iar $ia \not\equiv 0 \pmod{11}$ deoarece i și a sunt diferite de 0.

Dacă schimbăm x_i cu x_j , unde $i \neq j$ și $x_i \neq x_j$, atunci

$$\left(\sum_{k=1, k \neq i,j}^{10} k x_k \right) + ix_j + j x_i \equiv (j-i)(x_i - x_j) \pmod{11},$$

iar $(j-i)(x_i - x_j) \not\equiv 0 \pmod{11}$ deoarece $i \neq j$ și $x_i \neq x_j$.

Acest cod poate fi obținut din codul liniar

$$C = \{x_1 \cdots x_{10} \in F_{11}^{10} \mid \sum_{i=1}^{10} i x_i = 0\}$$

peste F_{11}^{10} prin eliminarea cuvintelor cod care conțin cel puțin un $x_i = 10$, $1 \leq i \leq 9$.

Vom specifica un cod liniar C de tip $[n, k]$ peste F_q printr-o matrice generatoare G de tip $k \times n$ peste F_q . C are q^k cuvinte cod, fiecare cuvânt cod având lungimea n . Aceste cuvinte cod sunt combinații liniare de vectori ai bazei, deci combinații liniare ale liniilor matricei generatoare. Ca urmare, un cuvânt cod este unic determinat de scalarii care furnizează combinația corespunzătoare lui, deci de un cuvânt $u \in F_q^k$. Pentru un astfel de cuvânt, cuvântul cod va fi uG .

Ca urmare, codificarea cu C va decurge astfel:

- dat un mesaj ca secvență de simboluri din F_q , împărțim mesajul în grupe de câte k simboluri, de la stânga la dreapta (putem presupune, fără a restrâng generalitatea, că lungimea mesajului este multiplu de k ; altfel, îl putem completa la dreapta cu un simbol fixat pentru a fi îndeplinită această cerință);
- fiecare grupă de câte k simboluri este privită ca un vector linie k -dimensional. Dacă u este un astfel de vector, atunci el va fi codificat prin cuvântul cod uG .

Exemplul 6.6.2.2. Fie codul liniar de tip $[7, 4]$ peste F_2 dat prin matricea generatoare

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

(se poate verifica cu ușurință că liniile acestei matrice sunt liniar independente).

Atunci vectorul $u_1 = (0, 0, 0, 0)$ va fi codificat prin

$$u_1G = (0, 0, 0, 0, 0, 0, 0),$$

vectorul $u_2 = (1, 0, 0, 0)$ prin

$$u_2G = (1, 0, 0, 0, 1, 0, 1),$$

iar vectorul $u_3 = (1, 1, 1, 0)$ prin

$$u_3G = (1, 1, 1, 0, 1, 0, 0).$$

Ca urmare, mesajul 000010001110 va fi codificat prin

$$000000010001011110100.$$

Așa cum se observă, matricea din Exemplul 6.6.2.2 are o proprietate particulară care face codificarea foarte eficientă: prima parte a ei este I_k (matricea unitate de ordin k). Putem găsi pentru orice cod C o matrice generatoare cu o astfel de proprietate? Răspunsul este afirmativ. Pornim cu o matrice generatoare G și o transformăm în mod “echivalent” până o aducem la forma $(I_k|A)$, numită *formă standard*. Operațiile de care avem nevoie sunt următoarele:

- (T1) interschimbarea a două linii ale matricei;
- (T2) înmulțirea unei linii cu un scalar și adăugarea rezultatului la o altă linie;
- (T3) interschimbarea a două coloane a matricei;
- (T4) înmulțirea unei coloane cu un scalar diferit de 0.

Dacă aplicăm aceste transformări de un număr finit de ori unei matrice generatoare G a unui cod liniar C , vom obține o nouă matrice G' ce are liniile liniar independente și, deci, ea va fi matrice generatoare a unui cod liniar C' . Vom spune că C și C' sunt *coduri liniare echivalente*. Este ușor de văzut că echivalența codurilor păstrează tipul acestora (inclusiv și distanța). Dacă G' se obține numai prin transformări de tipul (T1) și (T2), atunci ea generează același cod C . Singura diferență este că asocierea cuvintelor cod poate fi diferită (adică pentru $u \in F_q^k$, uG și uG' pot fi diferite).

Este cunoscut că prin transformări (T1)-(T4) aplicate unei matrice G de tip $k \times n$ ce are liniile liniar independente se poate obține o nouă matrice de forma $G' = (I_k|A)$ ce are liniile liniar independente.

Exemplul 6.6.2.3. Fie G matricea 4×7 peste F_2

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Adăugăm prima linie la linia a doua și a treia și obținem:

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Adăugăm a doua linie la prima, a treia și a patra linie și obținem:

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Interschimbăm liniile 3 și 4 și obținem:

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Ultimii pași constau în adunarea liniei 3 la linia 2, urmată apoi de adăugarea liniei 4 la liniile 2 și 1:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Matricea astfel obținută este în formă standard.

6.6.2.2. Decodificare Slepian și sindrom

O primă metodă de decodificare a mesajelor codificate prin coduri liniare, pe care o discutăm în această secțiune, a fost propusă de Slepian în 1960 [194]. Ideea este foarte simplă și se bazează pe faptul că orice subspațiu vectorial al unui spațiu vectorial este subgrup normal al grupului aditiv al spațiului gazdă. Ca urmare, el va induce o congruență care partiziionează spațiul gazdă.

Fie C un cod de tip $[n, k]$ peste F_q . Atunci congruența \sim_C indușă de C partiziunează F_q^n în $q^n/q^k = q^{n-k}$ clase de echivalență:

$$a_0 + C, a_1 + C, \dots, a_s + C,$$

unde $a_0, a_1, \dots, a_s \in F_q^n$, $a_0 = 0$ și $s = q^{n-k} - 1$.

Pentru orice clasă de echivalență $a_i + C$, putem presupune că a_i este de pondere Hamming minimă în $a_i + C$. În adevăr, dacă există $a \in a_i + C$ de pondere Hamming mai mică decât a_i , atunci putem interschimba a_i cu a deoarece are loc $a_i + C = a + C$. Pentru a vedea aceasta facem apel la proprietățile codurilor liniare:

- fie $a = a_i + c$, unde $c \in C$;
- pentru orice $c' \in C$, $-c + c'$ este în C și, atunci,

$$a + c' = a_i + c - c + c' = a_i + c' \in a_i + C.$$

Deci, $a + C \subseteq a_i + C$;

- pentru orice $c' \in C$,

$$a_i + c' = a_i + c - c + c' = a + (-c + c') \in a + C,$$

ceea ce arată că are loc $a_i + C \subseteq a + C$.

Ca urmare, putem presupune că a_i are pondere minimă în $a_i + C$. Acest element, numit *lider* al clasei de echivalență, va fi considerat ca fiind eroarea de transmisie. Deci $a_i + C$ reprezintă mulțimea tuturor cuvintelor cod la care adăugăm eroarea de transmisie a_i . Ca urmare, pentru un cuvânt y recepționat, va exista un unic i , $0 \leq i \leq s$, cu $y \in a_i + C$. Atunci, y va fi decodificat prin $y - a_i$. Aceasta corespunde procedurii de decodificare prin cel mai apropiat cuvânt cod deoarece determinarea acestui cel mai apropiat cuvânt cod corespunde acceptării celei mai mici erori (ca pondere Hamming).

Pentru a avea o procedură elegantă de decodificare, clasele de echivalență $a_i + C$ se organizează în forma unui tabel, numit *tabel Slepian*, astfel:

1. se enumeră pe o linie elementele codului C , într-o ordine arbitrară dar fixată, începând cu 0;
2. dacă s-au construit i linii $0+C, \dots, a_{i-1}+C$, unde $1 \leq i \leq s$, atunci se determină un cuvânt din F_q^n de pondere minimă ce nu este în liniile anterioare, fie acesta a_i , și se construiește o nouă linie $a_i + C$ obținută prin adăugarea succesivă a elementelor codului C la a_i , în ordinea aleasă la pasul inițial. Procedura se încheie când linia $i = s$ a fost procesată.

Exemplul 6.6.2.4. Fie C codul de tip $[4, 2]$ peste F_2 dat prin matricea generatoare

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

Ca urmare, $C = \{0000, 1011, 0101, 1110\}$. Atunci, un tabel Slepian poate fi de forma:

lider			
0000	1011	0101	1110
1000	0011	1101	0110
0100	1111	0001	1010
0010	1001	0111	1100

Dacă se recepționează $y = 1110$, atunci acesta se găsește în prima linie a tabelului și, deci, eroarea este considerată 0000 (altfel spus, el este cuvânt cod). Dacă se recepționează $y = 1111$, atunci acesta este în linia a treia a tabelului și, deci, eroarea va fi considerată ca fiind 0100. Deci el este decodificat prin $y + 0100 = 1011$.

Dacă q și n sunt mari, metoda de decodificare Slepian poate fi greoaie din două motive:

1. căutarea vectorului în tabelul Slepian poate fi mare consumatoare de timp;
2. memorarea tabelului Slepian poate constitui o problemă.

În cadrul tabelului Slepian suntem interesați de coloana liderilor (care este, de fapt, coloana erorilor). O idee foarte bună ar consta în găsirea unei metode prin care, un vector recepționat y să poate fi încadrat în linia corespunzătoare a tabelului Slepian fără a face căutarea acestuia în tot tabelul. Aceasta s-ar putea face dacă pentru fiecare linie a tabelului am putea stabili un parametru unic care să caracterizeze toate elementele din acea linie. Atunci, de îndată ce s-ar receptiona y , s-ar determina parametrul acestuia și y ar fi încadrat la linia corespunzătoare lui. Deci s-ar ști eroarea de transmisie (după metoda cuvântului cod cel mai apropiat).

O metodă de a stabili un parametru ca cel discutat mai sus este următoarea. Un vector $a = (a_1, \dots, a_n) \in F_q^n$ este numit *vector de verificare a parității* dacă are loc

$$a_1x_1 + \dots + a_nx_n = 0,$$

pentru orice $x = (x_1, \dots, x_n) \in C$.

Exemplul 6.6.2.5.

- (1) În cazul codurilor peste F_2 obținute prin adăugarea bitului de paritate, vectorul $(1, \dots, 1)$ este vector de verificarea a parității deoarece fiecare cuvânt cod $x = (x_1, \dots, x_n) \in C$ trebuie să verifice $\sum_{i=1}^n x_i = 0$ (în F_2).
- (2) Un cazul codului ISBN, vectorul $(1, 2, \dots, 10)$ este vector de verificare a parității.

Ca urmare, vectorii de verificare a parității sunt exact vectorii ortogonali pe C . Mulțimea acestora este spațiul ortogonal C^\perp . Conform rezultatelor din Secțiunea 6.5, $\dim(C^\perp) = n - k$. Ca urmare, C^\perp este un cod liniar de tip $[n, n - k]$ peste F_q . El se numește *codul dual* asociat codului C . Orice matrice generatoare a codului dual se numește *matrice de verificare a parității*. Dacă H este o astfel de matrice, atunci ea verifică $Hx^t = 0$, pentru orice $x \in C$, unde x^t este transpusul lui x .

Dacă $y \in F_q^n$, Hy^t se numește *sindromul vectorului* y .

Propoziția 6.6.2.2. Fie C un cod de tip $[n, k]$ peste F_q și H o matrice de verificare a parității codului C . Atunci, pentru orice $x, y \in F_q^n$, $Hx^t = Hy^t$ dacă și numai dacă $x \sim_C y$.

Demonstrație. Au loc relațiile:

$$\begin{aligned} x \sim_c y &\Leftrightarrow x + C = y + C \\ &\Leftrightarrow x - y \in C \\ &\Leftrightarrow H(x - y)^t = 0 \\ &\Leftrightarrow Hx^t = Hy^t, \end{aligned}$$

ce stabilesc rezultatul din propoziție. \square

Ca urmare a acestei propoziții, elementele de pe o linie a tabelului Slepian au același unic sindrom, diferit de al celorlalte linii. Aceasta poate fi calculat luând în considerare doar liderul liniei.

Următorul exemplu ne arată cum decurge decodificarea bazată pe sindrom.

Exemplul 6.6.2.6. Fie C codul din Exemplul 6.6.2.4. Matricea H de mai jos este matrice de verificare a parității acestui cod

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

Atunci, un tabel Slepian cu sindrom poate fi de forma:

lider	sindrom
0000	00
1000	11
0100	01
0010	10

Dacă se recepționează $y = 1110$, atunci acesta are sindromul $Hy^t = 00$, ceea ce ne arată că el se găsește în prima linie a tabelului și, deci, eroarea este considerată 0000. În acest caz y este cuvânt cod.

Dacă se recepționează $y = 1111$, atunci acesta are sindromul $Hy^t = 01$, ceea ce ne arată că y este în linia a treia a tabelului, și astfel eroarea va fi considerată ca fiind 0100. Deci el este decodificat prin $y + 0100 = 1011$.

Un singur lucru ne mai rămâne de arătat în cadrul acestei metode de decodificare, și anume, cum putem genera în mod eficient o matrice de verificare a parității.

Propoziția 6.6.2.3. Fie C un cod de tip $[n, k]$ peste F_q și $G = (I_k | A)$ o matrice generatoare pentru C , în formă standard. Atunci $H = (-A^t | I_{n-k})$ este matrice de verificare a parității codului C .

Demonstrație. Se verifică imediat că liniile matricei H sunt liniar independente (folosind faptul că liniile matricei G sunt liniat independente) și că fiecare linie a

matricei H este ortogonală pe G . Cum C^\perp are dimensiunea $n - k$, liniile matricei H formează o bază a acestui subspațiu vectorial. \square

Ca urmare, având o matrice generatoare pentru C , fie aceasta G , putem transforma G în formă standard $G' = (I_k | A)$ (ea va fi matrice generatoare a unui cod C' echivalent lui G), după care se poate obține imediat o matrice de verificare a parității pentru G' . Codificarea și decodificarea se va face cu C' , dar aceasta nu are nici o importanță deoarece codurile echivalente au aceeași putere de detectare și corectare de erori. În plus, utilizarea un cod pentru care matricea generatoare este în formă standard poate conduce la eficiență în procesul de codificare și decodificare.

Ca o ultimă remarcă, codurile liniare sunt unic determinate de o matrice de verificare a parității. Aceasta se bazează pe faptul că $(C^\perp)^\perp = C$. Dacă H este o matrice de verificare a parității pentru codul C , $H = (B | I_{n-k})$, atunci $G = (I_k | -B^t)$ este o matrice generatoare pentru C . Aceasta face ca de multe ori, în practică, codurile liniare să fie specificate printr-o matrice de verificare a parității.

Capitolul 7

Teoria mulțimilor parțial ordonate

7.1. Completitudine

Noțiunea de *completitudine* joacă un rol deosebit de important în teoria mulțimilor parțial ordonate și a aplicațiilor acesteia. În general, o mpo este numită completă dacă orice submulțime a ei ce satisface o proprietate dată admite infimum și/sau supremum. Prima noțiune de completitudine a fost introdusă de Birkhoff în 1933 [9] prin considerarea conceptului de latice completă (mpo pentru care orice submulțime nevidă admite infimum și supremum). Ulterior, apar și alte tipuri de completitudine; acestea, împreună cu câteva proprietăți de bază ale lor, vor fi discutate în capitolul de față.

7.1.1. Completitudine prin submulțimi. Latici complete

Așa cum deja s-a menționat, prima noțiune de completitudine în teoria mulțimilor parțial ordonate a fost introdusă de Birkhoff în 1933 [9]¹.

Definiția 7.1.1.1. Fie $M = (A, \leq)$ o mpo.

- (1) Spunem că M este *inf-completă prin submulțimi* dacă orice submulțime nevidă $B \subseteq A$ admite infimum.
- (2) Spunem că M este *sup-completă prin submulțimi* dacă orice submulțime nevidă $B \subseteq A$ admite supremum.

¹Birkhoff a considerat noțiunea de latice definită ca structură algebrică echivalentă cu ceea ce numim astăzi latice completă ([9], pag. 442).

- (3) Spunem că M este *completă prin submulțimi* dacă orice submulțime nevidă $B \subseteq A$ admite infimum și supremum.

Mulțimile parțial ordonate inf-complete (sup-complete, complete) prin submulțimi sunt exact inf-semilaticele (sup-semilaticele, laticele) complete. Din acest motiv putem folosi aceste terminologii în mod echivalent. De exemplu, ne putem referi la o mpo ca fiind inf-completă prin submulțimi sau inf-semilatice completă, în mod echivalent.

Câteva proprietăți simple ale laticelor complete au fost deja prezentate în Secțiunea 1.4.3.1. Vom continua în această secțiune cu două rezultate de caracterizare ale laticelor complete, rezultate prin care vom vedea că în anumite situații putem înlocui cerința de existență a supremumului și/sau infimumului unei submulțimi arbitrară prin existența supremumului și/sau infimumului unui lanț.

Vom spune că un lanț este *bine ordonat* dacă orice submulțime nevidă a lui are cel mai mic element.

Teorema 7.1.1.1. Fie $M = (A, \leq)$ o mpo nevidă. Atunci următoarele afirmații sunt echivalente:

- (1) M este latice completă;
- (2) M este sup-semilatice și pentru orice lanț $L \subseteq A$ există $\sup(L)$;
- (3) M este sup-semilatice și pentru orice lanț bine ordonat $L \subseteq A$ există $\sup(L)$.

Demonstrație. Este clar că (1) implică (2) și (2) implică (3).

Să presupunem că are loc afirmația de la (3). Vom arăta că pentru orice submulțime $B \subseteq A$ există $\inf(B)$, ceea ce va stabili completitudinea lui M conform Teoremei 1.4.3.1. Deci va avea loc (1).

Fie $B \subseteq A$. Dacă $B = \emptyset$, atunci B este lanț bine ordonat, iar ipoteza asigură existența elementului $\sup(B)$ care este \perp_M .

Să presupunem $B \neq \emptyset$. Mulțimea B^- a minoranților lui B este nevidă deoarece conține cel puțin \perp_M . Considerăm mulțimea \mathcal{C} a tuturor lanțurilor bine ordonate în $(B^-, \leq|_{B^-})$. Este clar că această mulțime este nevidă. Pe ea definim relația \preceq prin

$$L_1 \preceq L_2 \Leftrightarrow L_1 = L_2 \text{ sau } (\exists b \in L_2)(L_1 = \{a \in L_2 \mid a < b\}),$$

pentru orice $L_1, L_2 \in \mathcal{C}$. Această relație structurează \mathcal{C} ca o mpo. În plus, orice lanț \mathcal{L} în (\mathcal{C}, \preceq) are cel puțin un majorant în \mathcal{C} (este suficient de considerat $\bigcup \mathcal{L}$, care este lanț bine ordonat în $(B^-, \leq|_{B^-})$, deci element al mulțimii \mathcal{C} ; acesta este chiar $\sup(\mathcal{L})$). Atunci, Lema lui Zorn asigură existența unui element maximal L în (\mathcal{C}, \preceq) .

Conform ipotezei, există $\sup(L)$. Vom arăta că $\sup(L)$ este infimumul mulțimii B , astfel:

- a) conform Lemei 1.4.2.1, $\sup(L)$ este minorant al mulțimii B ;

- b) dacă a este minorant al mulțimii B , conform aceleiasi Leme 1.4.2.1 și a faptului că M este sup-semilatice, $\sup(\{a, \sup(L)\})$ este minorant al mulțimii B . Atunci, maximalitatea lanțului $L \subseteq B^-$ conduce la

$$\sup(L) = \sup(\{a, \sup(L)\})$$

și, deci, $a \leq \sup(L)$.

Așa arătat astfel că $\sup(L)$ este cel mai mare minorant al mulțimii B . Deci $\inf(B)$ există. Teorema 1.4.3.1 ne spune atunci că M este latice completă, ceea ce stabilește (1) și încheie demonstrația teoremei. \square

Evident, Teorema 7.1.1.1 poate fi dualizată. Atragem atenția asupra importanței cerinței conform căreia M este sup-semilatice, cerință ce este esențială în a demonstra că (3) implică (1). De asemenea, cerința ca M să fie nevidă este importantă deoarece, altfel, (1) ar fi satisfăcută (laticea vidă este completă), dar (2) sau (3) nu (pentru lanțul vid nu ar exista supremum).

Observația 7.1.1.1. Cerința “ M este sup-semilatice” în Teorema 7.1.1.1 merită o discuție mai detaliată. Fie B o submulțime nevidă a unei mpo $M = (A, \leq)$. Presupunem, de asemenea, că $B^- \neq \emptyset$. Știm că dacă $\inf(B)$ există atunci acesta este dat de $\inf(B) = \sup(B^-)$. Teorema 7.1.1.1 ne spune că $\sup(B^-)$ poate fi substituit prin supremul unui lanț maximal bine ordonat în B^- (a se vedea demonstrația teoremei). Dar, ca această substituire să funcționeze, este necesar ca M să fie sup-semilatice, ceea ce în acest caz înseamnă că pentru orice doi minoranți ai mulțimii B există supremul acestora și el este tot minorant al mulțimii B^- (Lema 1.4.2.1). Deci putem substitui $\sup(B^-)$ prin supremul unui lanț maximal bine ordonat în B^- dacă B^- este închisă la supremumul oricărora două elemente (aceasta fiind satisfăcută dacă M este sup-semilatice).

Dar este esențial ca B^- să fie închisă la supremumul oricărora două elemente? Nu poate fi relaxată această cerință? Dacă analizăm cu atenție demonstrația Teoremei 7.1.1.1 constatăm că putem cere doar ca pentru orice doi minoranți ai mulțimii B să existe un majorant al acestora care să fie minorant al mulțimii B . Altfel spus, B^- trebuie să fie *mulțime dirijată*.

Ca urmare, cerința “ M este sup-semilatice” în Teorema 7.1.1.1 poate fi înlocuită cu “pentru orice submulțime nevidă $B \subseteq A$, mulțimea B^- este dirijată”.

Teorema 7.1.1.2. Fie $M = (A, \leq)$ o mpo. Atunci următoarele afirmații sunt echivalente:

- (1) M este latice completă;
- (2) M este latice și pentru orice lanț nevidă $L \subseteq A$ există $\sup(L)$ și $\inf(L)$;
- (3) M este latice și orice lanț maximal $L \subseteq A$ este latice completă.

Demonstrație. Este clar că cele trei afirmații din teoremă sunt echivalente în cazul mpo vide. Să presupunem că M este o mpo nevidă. Este clar că (1) implică (2). Vom arăta că (1) implică și (3).

Fie L un lanț maximal în M . L este nevid deoarece conține cel mai mic și cel mai mare element al laticei complete M . Fie $B \subseteq L$. Dacă $B = \emptyset$, atunci $\inf(B) = \top_M$ și $\sup(B) = \perp_M$ și ambele elemente sunt în L . Să presupunem că B este nevidă. Conform ipotezei, există $\sup_M(B)$. În plus, acest element trebuie să fie în L deoarece el este cel mai mic majorant al mulțimii B și, deci, mai mic decât orice majorant al mulțimii B ce se găsește în L . Neapartenența acestuia la L ar contrazice maximalitatea lanțului.

În mod similar se arată că există $\inf(B)$. Deci, L este latice completă.

Să arătăm că (3) implică (2). Fie $L \subseteq A$ un lanț nevid în M . Există atunci un lanț maximal L' ce include L . Conform ipotezei, există $\sup_{L'}(L)$. Vom arăta că $\sup_{L'}(L) = \sup(L)$.

Fie a un majorant (în M) al lanțului L . Conform Lemiei 1.4.2.1 și faptului că M este latice, $\inf(\{a, \sup_{L'}(L)\})$ este majorant al lanțului L . Inegalitatea

$$\inf(\{a, \sup_{L'}(L)\}) \leq \sup_{L'}(L)$$

nu poate fi strictă, deoarece ori s-ar contrazice maximalitatea lanțului L' (în cazul $\inf(\{a, \sup_{L'}(L)\}) \notin L'$), ori s-ar contrazice faptul că $\sup_{L'}(L)$ este supremumul lanțului L în L' (în cazul $\inf(\{a, \sup_{L'}(L)\}) \in L'$).

Deci,

$$\inf(\{a, \sup_{L'}(L)\}) = \sup_{L'}(L) \leq a,$$

cea ce ne arată că $\sup_{L'}(L)$ este supremumul lanțului L în M .

În mod similar se arată că există $\inf(L)$ în M .

Să arătăm acum că (2) implică (1). Fie $B \subseteq A$ o submulțime nevidă a mulțimii A . Vom arăta că există un lanț nevid L astfel încât $L^+ = B^+$, ceea ce va conduce, conform ipotezei, la faptul că există $\sup(B)$.

Fie $B = \{b_\alpha | \alpha < \gamma\}$ o ordonare bună a mulțimii B , unde $\gamma = |B|$. Definim $L = \{a_\alpha | \alpha < \gamma\}$ prin recursie transfiniță astfel:

- $a_0 = b_0$;
- $a_\alpha = \sup(\{a_{\alpha'}, b_\alpha\})$, dacă α este ordinal succesor și α' este predecesorul imediat al lui α ;
- $a_\alpha = \sup(\{\sup(\{a_\beta | \beta < \alpha\}, b_\alpha)\})$, dacă α este ordinal limită,

pentru orice $0 < \alpha < \gamma$. Conform ipotezelor, a_α există, pentru orice α .

Este clar că L este lanț și orice majorant al lanțului L este majorant al mulțimii B . Arătăm prin inducție transfiniță că orice majorant al mulțimii B este și majorant al lanțului L .

Fie c un majorant al mulțimii B . c este majorant pentru b_0 și, deci și pentru a_0 . Fie $\alpha < \gamma$. Presupunem că c este majorant pentru a_β , pentru orice $0 \leq \beta < \alpha < \gamma$. Avem două cazuri de analizat:

- α este ordinal succesor. Fie $\alpha = \alpha' + 1$. Elementul c fiind majorant pentru $a_{\alpha'}$ și b_α , va fi majorant și al celui mai mic majorant al acestor două elemente, deci al lui a_α ;
- α este ordinal limită. Atunci c va fi majorant al elementului $\sup(\{a_\beta | \beta < \alpha\})$ deoarece este majorant al fiecărui element a_β cu $\beta < \alpha$. Cum c este majorant și al elementului b_α , deducem că el este majorant al celui mai mic majorant al elementelor $\sup(\{a_\beta | \beta < \alpha\})$ și b_α , deci al lui a_α .

Ca urmare, $L^+ = B^+$, iar existența supremumului lanțului L asigură existența supremumului mulțimii nevide B .

În mod similar se arată că există infimumul mulțimii B . \square

Atât Teorema 7.1.1.1, cât și Teorema 7.1.1.2 fac apel, indirect, la Axioma alegerii.

7.1.2. Completitudine prin mulțimi dirijate

Completitudinea prin submulțimi dirijate (filtrate) se introduce ca și în cazul laticelor complete dar cu deosebirea că submulțimile în cauză sunt dirijate (filtrate).

Definiția 7.1.2.1. Spunem că o mpo $M = (A, \leq)$ este *completă prin submulțimi dirijate (filtrate)* dacă M are cel mai mic (cel mai mare) element și orice submulțime dirijată (filtrată) admite supremum (infimum).

Evident, (\emptyset, \emptyset) este mpo completă atât prin submulțimi dirijate, cât și filtrate.

În Secțiunea 7.1.1 s-a văzut că laticele complete pot fi caracterizate, în anumite situații, prin existența supremumului/infimumului lanțurilor. Mai exact, s-a arătat că, în anumite situații, existența supremumului/infimumului unei submulțimi B arbitrară poate fi redusă la existența supremumului/infimumului unui anumit (tip de) lanț. În cazul mulțimilor dirijate, aceste rezultate pot fi întărite considerabil arătând că, în general, existența supremumului mulțimilor dirijate poate fi redusă la existența supremumului lanțurilor. Pentru aceasta, vom demonstra întâi următoarea teoremă de descompunere a mulțimilor dirijate [137].

Teorema 7.1.2.1. Pentru orice mulțime dirijată infinită (A, \leq) există o secvență transfiniță de submulțimi dirijate $(A_\alpha | \alpha < \gamma)$, unde $\gamma = |A|$, astfel încât:

- (1) A_α este finită, pentru orice ordinal finit $\alpha < |A|$;
- (2) $|A_\alpha| = |\alpha|$, pentru orice ordinal infinit $\alpha < |A|$;
- (3) $A_\alpha \subseteq A_\beta$, pentru orice $\alpha < \beta < |A|$;
- (4) $A = \bigcup_{\alpha < \gamma} A_\alpha$.

Demonstrație. Fie $A = \{a_\alpha | \alpha < \gamma\}$ o ordonare bună a mulțimii A . Vom nota prin u_Y un majorant arbitrar al unei submulțimi finite și nevidă $Y \subseteq A$ (existența unui astfel de majorant este asigurată de faptul că (A, \leq) este dirijată).

Definim prin recursie:

- $A_0 = \{a_0\}$;
- $A_{i+1} = A_i \cup \{b_{i+1}, u_{A_i \cup \{b_{i+1}\}}\}$, unde b_{i+1} este cel mai mic element al mulțimii $A - A_i$ (existența lui b_{i+1} fiind asigurată de buna ordonare a mulțimii A), pentru orice $i \leq \omega$;
- $A_\omega = \bigcup_{i<\omega} A_i$.

Dacă $|A| = \omega$, atunci este ușor de văzut că secvența de mai sus satisfac cerințele teoremei. Altfel, definim A_α , pentru $\omega < \alpha < \gamma$, prin recursie transfinิตă astfel:

- dacă α este ordinal limită, atunci $A_\alpha = \bigcup_{\beta < \alpha} A_\beta$;
- dacă α este ordinal succesor, $\alpha = \beta + 1$, atunci $A_\alpha = \bigcup_{i<\omega} A_{\alpha,i}$, unde:
 - $A_{\alpha,0} = A_\beta \cup \{b_\beta\}$, unde b_β este cel mai mic element al mulțimii $A - A_\beta$;
 - $A_{\alpha,i+1} = A_{\alpha,i} \cup \{u_Y | Y \subseteq A_{\alpha,i} \text{ și } Y \text{ finită}\}$.

Să arătăm că, în adevăr, secvența astfel definită satisfac teorema. Aceasta decurge printr-o simplă inducție transfinิตă. Vom lua în considerare numai cazul ordinalilor strict mari decât ω deoarece pentru restul ordinalilor mulțimile corespunzătoare satisfac proprietățile din teoremă.

Fie α un ordinal limită. Presupunem că toate submulțimile A_β cu $\beta < \alpha$ sunt dirijate și satisfac (2) și (3). A_α este dirijată deoarece orice submulțime finită și nevidă a ei va fi în una din mulțimile A_β cu $\beta < \alpha$ și A_β este dirijată. În plus, proprietățile (2) și (3) vor fi trivial satisfăcute și de A_α .

Să presupunem că α este ordinal succesor, $\alpha = \beta + 1$ și A_β satisfac (2) și (3). Orice submulțime finită și nevidă a mulțimii A_α se găsește în una din submulțimile $A_{\alpha,i}$ și, conform definiției mulțimii $A_{\alpha,i+1}$, aceasta va avea un majorant în $A_{\alpha,i+1}$. Ca urmare, A_α este dirijată. Proprietatea (3) este trivial satisfăcută. Pentru a arăta că are loc (2) folosim faptul că, pentru orice mulțime infinită X , cardinalul mulțimii tuturor submulțimilor finite ale ei egalează cardinalul mulțimii X . În baza acestei proprietăți urmează că are loc $|A_{\alpha,i}| = |A_\beta|$, pentru orice i . Atunci

$$|A_\alpha| \leq |\omega| \times |A_\beta| = |A_\beta| = |\beta| = |\alpha|,$$

care, combinată cu proprietatea

$$|\beta| = |A_\beta| \leq |A_\alpha|,$$

conduce la $|A_\alpha| = |\beta| = |\alpha|$.

(4) urmează direct de la construcția secvenței $(A_\alpha | \alpha < \gamma)$. \square

Vom rafina Definiția 7.1.2.1, în cazul mulțimilor dirijate, astfel:

Definiția 7.1.2.2. Spunem că o mpo M este γ -completă prin submulțimi dirijate, unde $\gamma > 0$ este un număr cardinal, dacă ea are cel mai mic element și orice submulțime dirijată de cardinal cel mult γ admite supremum.

Corolarul 7.1.2.1. Fie $M = (A, \leq)$ o mpo și $\gamma > 0$ un cardinal. M este γ -completă prin submulțimi dirijate dacă și numai dacă orice lanț de cardinal cel mult γ admite supremum.

Demonstrație. Orice lanț nevid de cardinal cel mult $\gamma > 0$ este submulțime dirijată de cardinal cel mult γ . Ca urmare, γ -completitudinea prin submulțimi dirijate implică existența supremumului oricărui lanț de cardinal cel mult γ .

Reciproc, presupunem că orice lanț de cardinal cel mult γ admite supremum. Presupunem, prin contradicție, că există o submulțime dirijată B de cardinal cel mult γ ce nu admite supremum. În plus, putem presupune că orice submulțime dirijată de cardinal strict mai mic decât $|B|$ admite supremum.

B nu poate fi finită. Fie $B = \bigcup_{\alpha < |B|} B_\alpha$ o descompunere a mulțimii B ca în Teorema 7.1.2.1. Considerăm atunci mulțimea $L = \{\sup(B_\alpha) | \alpha < |B|\}$. Conform presupunerii făcute, există $\sup(B_\alpha)$ pentru orice $\alpha < |B|$ deoarece fiecare submulțime B_α este dirijată și $|B_\alpha| < |B|$. Mulțimea L este lanț (conform proprietăților descompunerii lui B), supremul acestuia există și, în plus, $\sup(L) = \sup(B)$, ceea ce constituie o contradicție. \square

De la Corolarul 7.1.2.1 obținem:

Corolarul 7.1.2.2. O mpo este completă prin submulțimi dirijate dacă și numai dacă orice lanț al ei admite supremum.

O combinație a completitudinilor prin submulțimi dirijate și filtrate a fost considerată de McShane în 1952 [145, 146] și numită *completitudine în sens Dedekind*. Studiul acestui tip de completitudine a fost realizat în profunzime de Wolk [226]. Noi ne vom limita la a face doar câteva simple observații asupra acestui tip de completitudine.

Definiția 7.1.2.3. Fie $M = (A, \leq)$ o mpo. Spunem că M este *completă în sens Dedekind* (abreviat, *D-completă*) dacă au loc următoarele proprietăți:

- (1) M are cel mai mic și cel mai mare element;
- (2) orice submulțime dirijată $B \subseteq A$ admite supremum;
- (3) orice submulțime filtrată $B \subseteq A$ admite infimum.

În cazul laticelor, completitudinea uzuală și D-completitudinea coincid.

Propoziția 7.1.2.1. Fie $M = (A, \leq)$ o mpo. Atunci M este latice completă dacă și numai dacă M este latice D-completă.

Demonstrație. Este clar că dacă M este latică completă, atunci ea este latică D-completă.

Reciproc, fie $B \subseteq A$ o submulțime nevidă. Lema 1.4.2.1 și faptul că M este latică asigură că B^+ este filtrată, care va admite astfel infimum conform D-continuității. Cum $\sup(B) = \inf(B^+)$, deducem că există $\sup(B)$.

În mod similar se arată că orice submulțime nevidă $B \subseteq A$ admite infimum. Deci, M este latică completă. \square

Observația 7.1.2.1. Este interesant de observat că, dată o mpo M , Propoziția 7.1.2.1 și Teorema 7.1.1.2(1)(2) conduc la echivalența următoarelor afirmații:

- (1) M este latică completă;
- (2) M este latică și pentru orice lanț nevid $L \subseteq A$ există $\sup(L)$ și $\inf(L)$;
- (3) M este latică nevidă, orice submulțime dirijată admite supremum și orice submulțime filtrată admite infimum.

7.1.3. Completitudine prin lanțuri

Așa cum s-a văzut în secțiunile anterioare, multe caracterizări ale laticelor complete sau ale mpilor complete prin submulțimi dirijate fac apel la lanțuri. Pe de altă parte, multe studii în informatică teoretică asupra “recursivității” fac apel la supremumul unor tipuri de lanțuri, atunci când acesta există [23, 120, 175, 216]. Este natural atunci de a introduce un concept de completitudine prin lanțuri. Meritul unui studiu profund al unei astfel de completitudini aparține lui Markowski [136, 137, 138].

Definiția 7.1.3.1. Spunem că o mpo $M = (A, \leq)$ este *completă prin lanțuri* dacă orice lanț în M admite supremum.

Completitudinea prin lanțuri este echivalentă cu completitudinea prin submulțimi dirijate (Corolar 7.1.2.2). Deci, din acest punct de vedere, acest nou concept de completitudine nu aduce nimic nou. Vom prefera însă, cel puțin din punct de vedere tehnic, să utilizăm lanțuri în locul submulțimilor dirijate. Cum completitudinea prin lanțuri este, fără doar și poate, cel mai intens utilizat concept de completitudine, vom simplifica terminologia de *mpo completă prin lanțuri* la cea de *mpo completă* și vom trece la un studiu intensiv al acesteia (celealte concepte de completitudine vor fi utilizate cu terminologia deja adoptată).

Observația 7.1.3.1.

- (1) Definiția 7.1.3.1 poate fi reformulată echivalent prin: M este mpo completă prin lanțuri dacă au loc următoarele proprietăți:

– M are cel mai mic element;

– orice lanț nevid $L \subseteq A$ admite supremum.

Această reformulare este bazată pe faptul că existența supremumului lanțului vid este echivalentă cu existența celui mai mic element \perp_A . De multe ori vom prefera să folosim această variantă, deoarece în demonstrații apare frecvent necesitatea clasificării lanțurilor în vide și nevide.

- (2) Unii autori cer în Definiția 7.1.3.1 doar existența supremumului lanțurilor nvide, iar ceea ce am numit noi mpo completă ei numesc *mpo completă pointată*. Atunci când nu vom cere existența celui mai mic element ne vom referi la astfel de mpilor ca fiind *mpo slab complete* (deci în astfel de mpilor este asigurat supremumul oricărui lanț nevid; cel mai mic element poate să existe sau nu).

Exemplul 7.1.3.1.

- (1) Orice mpo care are cel mai mic element și pentru care orice lanț este finit, este completă. În particular, ordinalii finiți și mpilor plate sunt mpilor complete.
- (2) $(\mathcal{P}(A), \subseteq)$, unde A este o mulțime arbitrară, este mpo completă.
- (3) Orice latică completă este mpo D-completă, iar orice mpo D-completă este mpo completă. Figura 7.1 prezintă grafic relația dintre aceste clase de mpilor.

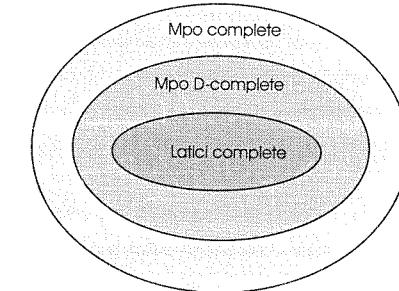


Figura 7.1: Relații între clase de mpilor

- (4) Mulțimea \mathbb{N} a numerelor naturale împreună cu ordinea uzuală nu este mpo completă deoarece lanțul \mathbb{N} nu admite supremum.
- (5) Fie A și B două mulțimi și $(A \rightsquigarrow B)$ mulțimea tuturor funcțiilor parțiale de la A la B . Considerăm pe $(A \rightsquigarrow B)$ relația binară \leq dată prin

$$f \leq g \Leftrightarrow \text{Dom}(f) \subseteq \text{Dom}(g) \wedge (\forall x \in \text{Dom}(f))(f(x) = g(x)),$$

pentru orice $f, g \in (A \rightsquigarrow B)$. Atunci $((A \rightsquigarrow B), \leq)$ este mpo completă. În adevăr, dacă L este un lanț în $(A \rightsquigarrow B)$, funcția $\bigcup L$ (existența acesteia este asigurată de faptul că L este mulțime de funcții compatibile două câte două (Secțiunea 1.1.3.5)) este supremumul acestui lanț.

Lema 7.1.3.1. Fie M și M' două mpo izomorfe. Atunci M este completă dacă și numai dacă M' este completă.

Demonstrație. Fie $M = (A, \leq)$ și $M' = (A', \leq')$ două mpo și $f : A \rightarrow A'$ un izomorfism între ele. Presupunem că M este completă. Atunci

- $f(\perp_M)$ este cel mai mic element al mpo M' ;
- pentru orice lanț L' în M' , $f^{-1}(L')$ este lanț în M . Atunci, $\sup_M(f^{-1}(L'))$ există și $\sup_{M'}(L') = f(\sup_M(f^{-1}(L')))$.

Deci, M' este completă.

Un raționament similar, realizat prin prisma funcției f^{-1} , ne arată că dacă M' este completă, atunci M este completă. \square

Fie A o mulțime și (B, \leq) o mpo. Pe mulțimea $(A \rightarrow B)$ a tuturor funcțiilor de la A la B definim relația binară $\leq_{(A \rightarrow B)}$ prin:

$$f \leq_{(A \rightarrow B)} g \Leftrightarrow (\forall a \in A)(f(a) \leq g(a)),$$

pentru orice $f, g : A \rightarrow B$. Este ușor de verificat că $\leq_{(A \rightarrow B)}$ este ordine parțială pe mulțimea $(A \rightarrow B)$.

Pentru orice $S \subseteq (A \rightarrow B)$ și $a \in A$ definim $S(a) = \{f(a) | f \in S\}$. Dacă $S = \emptyset$, atunci $S(a) = \emptyset$. Următoarea lemă, ce are un caracter tehnic, va fi intens utilizată în multe din demonstrațiile ce urmează.

Lema 7.1.3.2. Fie A o mulțime și (B, \leq) o mpo. Atunci, pentru orice submulțime nevidă $S \subseteq (A \rightarrow B)$ are loc

$$\exists \sup(S) \Leftrightarrow (\forall a \in A)(\exists \sup(S(a))).$$

În plus, dacă există $\sup(S)$ atunci are loc

$$(\forall a \in A)((\sup(S))(a) = \sup(S(a)))$$

(supremumul mulțimii S este considerat în raport cu $\leq_{(A \rightarrow B)}$, iar cel al mulțimii $S(a)$ în raport cu \leq).

Demonstrație. Presupunem că există $\sup(S)$ și fie acesta $f : A \rightarrow B$. Este clar că pentru orice $a \in A$, $f(a)$ este majorant pentru $S(a)$. Dacă presupunem, prin contradicție, că există $a \in A$ astfel încât $f(a)$ nu este cel mai mic majorant pentru $S(a)$, atunci există un element b ce este majorant pentru $S(a)$ și $b < f(a)$. Considerăm funcția $f' : A \rightarrow B$ dată prin

$$f'(x) = \begin{cases} f(x), & x \neq a \\ b, & x = a, \end{cases}$$

pentru orice $x \in A$. Atunci este ușor de verificat că f' este majorant pentru S și $f' <_{(A \rightarrow B)} f$, ceea ce contrazice faptul că f este cel mai mic majorant al mulțimii S . Deci, există $\sup(S(a))$ și 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 funcția $f : A \rightarrow B$ dată prin $f(a) = \sup(S(a))$, pentru orice $a \in A$. Este ușor de văzut că f este cel mai mic majorant pentru S . Ca urmare, există supremumul mulțimii S și, în plus, $(\sup(S))(a) = \sup(S(a))$, pentru orice $a \in A$. \square

Teorema 7.1.3.1. Fie A o mulțime nevidă și (B, \leq) o mpo. Dacă (B, \leq) este mpo completă, atunci $((A \rightarrow B), \leq_{(A \rightarrow B)})$ este mpo completă.

Demonstrație. Cel mai mic element al mpo $((A \rightarrow B), \leq_{(A \rightarrow B)})$ este funcția $\perp_{(A \rightarrow B)} : A \rightarrow B$ dată prin $\perp_{(A \rightarrow B)}(a) = \perp_B$, pentru orice $a \in A$.

Fie $L \subseteq (A \rightarrow B)$ un lanț nevid. Atunci, pentru orice $a \in A$, $L(a)$ este lanț nevid în B . Deoarece (B, \leq) este mpo completă, există $\sup(L(a))$ pentru orice $a \in A$. Atunci, Lema 7.1.3.2 asigură existența supremului lanțului L . Deci $((A \rightarrow B), \leq_{(A \rightarrow B)})$ este mpo completă. \square

Exemplul 7.1.3.2. Mulțimea parțial ordonată $((\mathbf{N}_\perp \rightarrow \mathbf{N}_\perp), \leq_{(\mathbf{N}_\perp \rightarrow \mathbf{N}_\perp)})$ este completă și, ca urmare, orice lanț va admite supremum. Să considerăm lanțul de funcții $L = \{f_i | i \geq 0\} \subseteq (\mathbf{N}_\perp \rightarrow \mathbf{N}_\perp)$ dat ca în diagrama de mai jos:

L	\perp	0	1	2	3	\dots
f_0	\perp	\perp	\perp	\perp	\perp	\dots
f_1	\perp	1	\perp	\perp	\perp	\dots
f_2	\perp	1	1!	\perp	\perp	\dots
f_3	\perp	1	1!	2!	\perp	\dots
\dots						

Acest lanț admite supremum, ce poate fi determinat utilizând Lema 7.1.3.2 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 lanțul L constituie o aproximare a funcției factorial: f_0 aproximează funcția factorial pentru \perp , f_1 aproximează funcția factorial pentru \perp și 0 etc. Aceste aproximări sunt înțelese în sensul

$$f_0 \leq_{\mathbf{N}_\perp \rightarrow \mathbf{N}_\perp} f_1 \leq_{\mathbf{N}_\perp \rightarrow \mathbf{N}_\perp} f_2 \leq_{\mathbf{N}_\perp \rightarrow \mathbf{N}_\perp} \dots \leq_{\mathbf{N}_\perp \rightarrow \mathbf{N}_\perp} \sup(L).$$

Exemplul 7.1.3.3. Exemplul 7.1.3.2 poate fi generalizat în mod natural. Orice funcție $f : \mathbf{N}_\perp \rightarrow \mathbf{N}_\perp$ poate fi aproximată printr-un lanț L de funcții definite ca

în diagrama de mai jos:

L	\perp	0	1	2	\dots
f_0	$f(\perp)$	\perp	\perp	\perp	\dots
f_1	$f(\perp)$	$f(0)$	\perp	\perp	\dots
f_2	$f(\perp)$	$f(0)$	$f(1)$	\perp	\dots
\dots	\dots	\dots	\dots	\dots	\dots

Operații cu mpo complete. Vom studia acum modul în care se păstrează proprietatea de completitudine prin trecere la intersecție, reuniune, sume și produse ordonate (așa cum au fost definite în Secțiunea 1.4.2.4).

Intersecția unei familii $((A_i, \leq_i) | i \in I)$ de mpo complete poate să nu fie mpo completă din două motive:

1. prin intersecție nu se păstrează cel mai mic element. De exemplu, să considerăm $\mathbf{Z}_1 = \mathbf{Z}_- \cup \{\perp_1\}$ și $\mathbf{Z}_2 = \mathbf{Z}_- \cup \{\perp_2\}$, unde \mathbf{Z}_- este mulțimea numerelor întregi negative, iar \perp_1 și \perp_2 sunt două elemente diferite între ele și diferite de orice număr întreg. Aceste două mulțimi cu ordinea naturală pe \mathbf{Z}_- extinsă prin considerarea elementelor \perp_1 și respectiv \perp_2 ca fiind cele mai mici elemente, conduc la mpo complete. Intersecția lor este \mathbf{Z}_- ce nu mai are cel mai mic element;
2. prin intersecție nu se mai asigură supremumul unor lanțuri. Intersecția dualelor mulțimilor parțial 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 existența celui mai mic element. Însă, $\perp \oplus \bigcup_{i \in I} (A_i, \leq_i)$ este mpo completă (considerând \perp un nou element). Dacă (A_i, \leq_i) nu este o familie disjunctă (dar elementele ei sunt mpo complete), $\perp \oplus \biguplus_{i \in I} (A_i, \leq_i)$ este mpo completă.

Vom analiza acum cazul sumei ordonate.

Propoziția 7.1.3.1. Fie $\mathcal{I} = (I, \leq)$ o mpo completă și $((A_i, \leq_i) | i \in I)$ o familie de mpo complete disjuncte. Atunci suma ordonată $\sum_{i \in I}^o (A_i, \leq_i)$ este mpo completă.

Demonstrație. Fie $(A, \leq') = \sum_{i \in I}^o (A_i, \leq_i)$. 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 mpo \mathcal{I} (existența acestuia este asigurată de faptul că \mathcal{I} este completă).

Fie $L \subseteq A$ un lanț și $K = \{i \in I | \exists a \in L : a \in A_i\}$. Mulțimea K este lanț în I . Deoarece \mathcal{I} este completă, va exista $k = sup(K)$. Considerăm acum sublanțul $L' = L \cap A_k$ al lanțului L . Deoarece (A_k, \leq_k) este completă, există $sup_{A_k}(L')$ și este ușor de arătat că $sup_A(L) = sup_{A_k}(L')$. \square

Există cazuri în care disjunctivitatea familiei $((A_i, \leq_i) | i \in I)$ nu este asigurată. Suma ordonată disjunctă (Secțiunea 1.4.2.4) 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 situații se pot face astfel. Fie $((A_i, \leq_i) | i \in I)$ o familie de mpo. Atunci *suma separată* a acestei familii, notată $\sum_{i \in I}^s (A_i, \leq_i)$, 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 parțială

$x \leq' y \Leftrightarrow$ are loc una din următoarele două proprietăți:

1. există $i \in I$ și $a, b \in A_i$ astfel încât $x = (a, i)$, $y = (b, i)$ și $a \leq_i b$, sau
2. $x = \perp$ și $y \in A$,

pentru orice $x, y \in A$.

Suma separată este oarecum similară sumei disjuncte, diferența 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 $\{(A_i, \leq_i) | i \in I\}$ o familie de mpo. Atunci, *suma de fuziune* a acestei familii, notată $\sum_{i \in I}^f (A_i, \leq_i)$, 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 parțială

$x \leq' y \Leftrightarrow$ are loc una din următoarele două proprietăți:

1. există $i \in I$ și $a, b \in A_i - \{\perp_{A_i}\}$ astfel încât $x = (a, i)$, $y = (b, i)$ și $a \leq_i b$, sau
2. $x = \perp$ și $y \in A$,

pentru orice $x, y \in A$.

Suma de fuziune poate fi considerată un caz particular de sumă separată (mpo în cauză nu are cel mai mic element). Ca urmare, sumă de fuziune de mpo complete este mpo completă.

În 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 (parțială sau totală) pe mulțimea indecșilor familiei. Primul nostru rezultat nu este altceva decât o simplă generalizare a Lemei 7.1.3.2.

Lema 7.1.3.3. Fie $((A_i, \leq_i) | i \in I)$ o familie nevidă de mpo. Atunci, pentru orice submulțime nevidă $S \subseteq \prod_{i \in I} A_i$ are loc

$$\exists \text{sup}(S) \Leftrightarrow (\forall i \in I)(\exists \text{sup}(S(i))).$$

În plus, dacă există $\text{sup}(S)$, atunci are loc

$$(\forall i \in I)((\text{sup}(S))(i) = \text{sup}(S(i))).$$

Demonstrație. Similară demonstrației Lemei 7.1.3.2. \square

Dacă în Lema 7.1.3.3 alegem $I = A$, $A_i = B$ și $\leq_i = \leq$ pentru orice $i \in I$, atunci obținem rezultatul din Lema 7.1.3.2.

Propoziția 7.1.3.2. Pentru orice familie nevidă $((A_i, \leq_i) | i \in I)$ de mpo complete, $\prod_{i \in I} (A_i, \leq_i)$ este mpo completă.

Demonstrație. Cel mai mic element al mpo $\prod_{i \in I} (A_i, \leq_i)$ este funcția $\perp_{\prod_{i \in I} A_i}$ dată prin $\perp_{\prod_{i \in I} A_i}(i) = \perp_{A_i}$, pentru orice $i \in I$.

Restul demonstrației decurge similar demonstrației Teoremei 7.1.3.1. \square

Corolarul 7.1.3.1. Fie $n \geq 1$ un număr natural și (A_i, \leq_i) mpo complete, unde $1 \leq i \leq n$. Atunci, $\times_{i=1}^n (A_i, \leq_i)$ este mpo completă.

Demonstrație. Este ușor de verificat că $\times_{i=1}^n (A_i, \leq_i)$ și $\prod_{i \in \{1, \dots, n\}} (A_i, \leq_i)$ sunt izomorfe. Atunci Propoziția 7.1.3.2 și Lema 7.1.3.1 conduc direct la faptul că mpo $\times_{i=1}^n (A_i, \leq_i)$ este completă. \square

Exemplul 7.1.3.4. Produs direct sau cartezian de mpo plate este mpo completă.

Submulțimi parțial ordonate complete. Vom discuta acum conceptul de submulțimi parțial ordonată completă care, menționăm încă de la început, nu trebuie confundat cu cel de sub-mpo care este și completă. Vom începe cu o observație care va justifica introducerea conceptului.

Observația 7.1.3.2. Fie $M = (A, \leq)$ o mpo completă și $M' = (A', \leq')$ o sub-mpo a lui M .

- (1) Dacă M' este completă, atunci, pentru orice lanț nevid L în M' , există $\text{sup}_M(L)$ și $\text{sup}_M(L) \leq \text{sup}_{M'}(L)$. Această inegalitate poate fi strictă. Drept exemplu, să considerăm $A = \mathcal{P}(\mathbb{N})$ și $A' = \{B \in A | B$ finită} $\cup \{\mathbb{N}\}$. Atunci, în raport cu incluziunea, A și A' sunt mulțimi parțial ordonate complete. În plus, A' este submulțime parțial ordonată a lui A . Fie lanțul

$$L = \{\{0\}, \{0, 2\}, \{0, 2, 4\}, \dots\} \subseteq A'.$$

Observăm că $\text{sup}_{M'}(L) = \mathbb{N}$, $\text{sup}_M(L) = \{n \in \mathbb{N} | n$ este par} și $\text{sup}_M(L) \subset \text{sup}_{M'}(L)$.

- (2) Dacă M' are cel mai mic element și $\text{sup}_M(L) \in M'$, pentru orice lanț nevid L în M' , atunci M' este completă. În acest caz are loc $\text{sup}_M(L) = \text{sup}_{M'}(L)$, pentru orice lanț nevid L în M' .

Definiția 7.1.3.2. Fie $M = (A, \leq)$ o mpo completă și $M' = (A', \leq')$ o sub-mpo a lui M . Spunem că M' este submulțime parțial ordonată completă (abreviat, sub-mpo completă) a lui M dacă pentru orice lanț L în M' are loc $\text{sup}_M(L) \in A'$.

Observația 7.1.3.3.

- (1) Definiția 7.1.3.2 este în concordanță cu Observația 7.1.3.1(1) în sensul că am putea înlocui cerința din această definiție prin:
 - (1') $\perp_M \in A'$;
 - (2') $(\forall L \subseteq A')(L$ lanț nevid în $M' \Rightarrow \text{sup}_M(L) \in A')$.

Aceasta se bazează pe faptul că $\text{sup}_M(\emptyset) = \perp_M$ care, conform cerinței (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 concordanță 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 și supremumul lanțurilor.

- (2) Cerința din Definiția 7.1.3.2 poate fi modificată echivalent la: pentru orice lanț $L \subseteq A'$ există $\text{sup}_{M'}(L)$ și are loc $\text{sup}_{M'}(L) = \text{sup}_M(L)$.
- (3) Orice sub-mpo completă a unei mpo complete este mpo completă.

Propoziția 7.1.3.3. Intersecția unei familii nevide de sub-mpo complete ale unei mpo complete este sub-mpo completă.

Demonstrație. 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 și în $\bigcap \mathcal{M}$. Similar, orice sub-mpo din \mathcal{M} păstrează supremumul lanțurilor calculat în M . Ca urmare, supremumul oricărui lanț nevid din \mathcal{M} , calculat în M , va fi în \mathcal{M} . \square

Reamintim că mulțimea tuturor funcțiilor 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 $\leq_{(A \rightarrow_m A')}$ restricția relației $\leq_{(A \rightarrow A')}$ la mulțimea $(A \rightarrow_m A')$. Cu această relație, $(A \rightarrow_m A')$ devine sub-mpo a mulțimii parțial ordonate $(A \rightarrow A')$. Vom arăta că $(A \rightarrow_m A')$ este chiar sub-mpo completă.

Teorema 7.1.3.2. Fie $M = (A, \leq)$ și $M' = (A', \leq')$ două mpo. Dacă M' este completă, atunci $(A \rightarrow_m A')$ este sub-mpo completă a mpo complete $(A \rightarrow A')$.

Demonstrație. Funcția $\perp_{(A \rightarrow A')}$ este monotonă, ceea ce ne spune că este element al mpo $(A \rightarrow_m A')$.

Fie $L \subseteq (A \rightarrow_m A')$ un lanț nevid de funcții monotone. Acest lanț admite supremum în $(A \rightarrow A')$. Vom arăta că acest supremum este funcție 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 7.1.3.2, este suficient de arătat că are loc

$$sup_{(A \rightarrow A')}(L(a)) \leq' sup_{(A \rightarrow A')}(L(b)),$$

iar conform Propoziției 1.4.2.6 este suficient de arătat că $L(b)$ este cofinal în $L(a)$. În primul rând remarcăm că $L(a)$ și $L(b)$ sunt lanțuri nevide în M' deoarece L este lanț nevid de funcții. Pe de altă parte, orice element din $L(a)$ este de forma $f(a)$ cu $f \in L$. Cum f este funcție monotonă și $a \leq b$, deducem că are loc $f(a) \leq' f(b)$, iar $f(b)$ este element al lanțului $L(b)$. 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 $(A \rightarrow A')$. \square

Observația 7.1.3.4. Teorema 7.1.3.2 poate fi reformulată echivalent astfel.

Fie $M = (A, \leq)$ și $M' = (A', \leq')$ două mpo. Dacă $M' = (A', \leq')$ este completă, atunci supremul oricărui lanț nevid de funcții monotone de la M la M' este funcție monotonă.

7.2. Teoria de punct fix a mulțimilor parțial ordonate

7.2.1. Funcții continue

În analiza matematică, o funcție f este continuă dacă este compatibilă cu limita sirurilor numerice în sensul că, pentru orice sir convergent $(a_n)_{n \geq 0}$, are loc

$$f(\lim_{n \rightarrow \infty} a_n) = \lim_{n \rightarrow \infty} f(a_n).$$

În cazul mulțimilor parțial ordonate, limita este supremumul sau infimumul unei submulțimi (submulțimile pot fi arbitrar, dirijate, lanțuri etc.), și atunci este natural să extindem conceptul de continuitate de mai sus la funcții definite pe mpo prin a cere compatibilitatea acestora cu supremumul sau infimumul. De exemplu, putem cere

$$f(sup(S)) = sup(f(S)),$$

pentru orice submulțime S ce satisfacă o anumită proprietate dată a priori, cum ar fi cea de lanț. Ca o astfel de definiție să funcționeze, domeniul pe care este definită funcția în cauză trebuie să aibă proprietatea că pentru orice submulțime 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 submulțimi, prin mulțimi dirijate, prin lanțuri etc.).

Conceptul de funcție continuă definită pe mpo complete (prin lanțuri) este unul dintre cele mai studiate concepte de continuitate. Aceasta va fi cel pe care îl vom prezenta și noi în cele ce urmează.

Definiția 7.2.1.1. Fie $M = (A, \leq)$ și $M' = (A', \leq')$ mpo complete și $f : A \rightarrow A'$ o funcție. Spunem că f este *continuă* dacă pentru orice lanț nevid L în M există $sup(f(L))$ și $f(sup(L)) = sup(f(L))$.

Observația 7.2.1.1.

- (1) Cerința “există $sup(f(L))$ ” din Definiția 7.2.1.1 este necesară deoarece în mpo complete este asigurat supremul lanțurilor, dar definiția nu garantează că $f(L)$ este lanț.
- (2) În Definiția 7.2.1.1 se cere că lanțul L să fie nevid. Dacă s-ar da posibilitatea ca relația din definiție să fie satisfăcută și de lanțul vid, atunci s-ar obține

$$f(\perp_M) = f(sup(\emptyset)) = sup(f(\emptyset)) = sup(\emptyset) = \perp_{M'},$$

deci s-ar păstra și cel mai mic element al mulțimilor parțial ordonate. În cazul în care o funcție f satisfacă și această condiție suplimentară ea este numită *funcție continuă strictă* sau *funcție continuă în sens strict*.

Funcțiile monotone păstrează lanțurile și, ca urmare, este de așteptat să existe o legătură destul de strânsă între monotonie și continuitate.

Teorema 7.2.1.1. (Continuitate și monotonie)

Fie $M = (A, \leq)$ și $M' = (A', \leq')$ mpo complete și $f : A \rightarrow A'$ o funcție. Atunci f este continuă dacă și numai dacă:

- (1) f este monotonă;
- (2) pentru orice lanț nevid $L \subseteq A$, $f(sup(L)) \leq' sup(f(L))$.

Demonstrație. Să presupunem că f este continuă. Fie $a, b \in A$ cu $a \leq b$. Considerăm lanțul $L = \{a, b\}$ al cărui supremum este b . Continuitatea funcției 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 definiția continuității.

Reciproc, presupunem că sunt îndeplinite condițiile (1) și (2) din enunțul teoremei. Fie $L \subseteq A$ un lanț nevid. Monotonia funcției f conduce la faptul că $f(L)$ este lanț (în M'), iar completitudinea mulțimii parțial ordonate M' conduce la existența supremumului acestui lanț. În plus, $f(sup(L))$ este majorant al lanțului $f(L)$ deoarece $sup(L)$ este majorant al lanțului L și f este monotonă. Cum $sup(f(L))$ este cel mai mic majorant al lanțului $f(L)$, obținem $sup(f(L)) \leq' f(sup(L))$, care combinată cu inegalitatea de la (2) conduce la $f(sup(L)) = sup(f(L))$. Deci f este continuă. \square

Observația 7.2.1.2. Punctul (2) al Teoremei 7.2.1.1 poate fi reformulat echivalent prin:

(2') pentru orice lanț nevid $L \subseteq A$, $f(sup(L)) = sup(f(L))$.

Corolarul 7.2.1.1. Fie $M = (A, \leq)$ și $M' = (A', \leq')$ mpo complete și $f : A \rightarrow A'$ o funcție. Dacă M are numai lanțuri finite, atunci f este continuă dacă și numai dacă este monotonă.

Demonstrație. Ceea ce avem de arătat este că monotonia funcției f implică continuitatea acesteia, în ipoteza în care M are numai lanțuri finite.

Fie $L \subseteq A$ un lanț nevid. Conform ipotezei, L este finit și, deci, are cel mai mare element, fie acesta a . Atunci monotonia funcției f conduce la

$$sup(f(L)) \leq' f(a) = f(sup(L)).$$

Deci conform Teoremei 7.2.1.1, f este continuă. \square

Observația 7.2.1.3. În general, nu orice funcție monotonă este continuă. Să considerăm, spre exemplu, funcția $\varphi : (\mathbf{N}_\perp \rightarrow \mathbf{N}_\perp) \rightarrow \mathbf{N}_\perp$ dată prin

$$\varphi(f) = \begin{cases} 1, & (\forall n \in \mathbf{N})(f(n) \neq \perp) \\ \perp, & \text{altfel,} \end{cases}$$

pentru orice $f \in (\mathbf{N}_\perp \rightarrow \mathbf{N}_\perp)$. Dacă $\varphi(f) = 1$ atunci vom spune că f este *totală*; altfel, spunem că f este *parțială*.

Este ușor de văzut că φ este funcție monotonă. În adevăr, pentru orice două funcții $f, g \in (\mathbf{N}_\perp \rightarrow \mathbf{N}_\perp)$ astfel încât $f \leq_{(\mathbf{N}_\perp \rightarrow \mathbf{N}_\perp)} g$, au loc relațiile:

- dacă f este parțială, atunci $\varphi(f) = \perp \leq \varphi(g)$ (\leq este ordinea parțială pe \mathbf{N}_\perp);
- dacă f este totală, atunci g este totală și, deci, $\varphi(f) = \varphi(g)$.

Fie acum lanțul de funcții L din diagrama de mai jos.

L	\perp	0	1	2	3	\dots
f_0	\perp	\perp	\perp	\perp	\perp	\dots
f_1	\perp	0	\perp	\perp	\perp	\dots
f_2	\perp	0	0	\perp	\perp	\dots
\dots	\dots					
$sup(L)$	\perp	0	0	0	0	\dots

Supremumul acestui lanț este funcția ce ia valoarea 0 pentru orice număr natural, și \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.

Că urmare, “ne-continuitatea” funcției φ rezidă din faptul că există lanțuri de funcții parțiale al căror supremum este funcție totală. Dacă unei funcții parțiale îi este asociată o anumită valoare (prin φ), iar unei funcții totale o altă valoare, atunci φ nu poate păstra supremumul unor astfel de lanțuri.

Încheiem observația prin a remarcă faptul că, dacă înlocuim “ \perp ” în definiția funcției φ prin 0, atunci φ nu este nici monotonă și nici continuă.

Exemplul 7.2.1.1.

- (1) Dacă $M_i = (A_i, \leq_i)$ sunt mpo plate, $1 \leq i \leq n$, și $M = (A, \leq)$ este o mpo completă, atunci orice funcție monotonă $f : A_1 \times \dots \times A_n \rightarrow A$ este continuă. În adevăr, mpo plate sunt complete și produs cartezian de mpo complete este mpo completă. În plus, $\times_{i=1}^n (A_i, \leq_i)$ are numai lanțuri finite.
- (2) Orice funcție constantă (definită pe o mpo completă și cu valori într-o mpo completă) este continuă.
- (3) Funcția identitate (definită pe o mpo completă și cu valori în aceeași mpo) este continuă.
- (4) Funcțiile proiecție $pr_i : A_1 \times \dots \times A_n \rightarrow A_i$, unde $1 \leq i \leq n$ și (A_j, \leq_j) sunt mpo complete, pentru orice $1 \leq j \leq n$, sunt funcții continue.

Teorema 7.2.1.2. Compunere de funcții continue este funcție continuă.

Demonstrație. Fie $M_1 = (A_1, \leq_1)$, $M_2 = (A_2, \leq_2)$ și $M_3 = (A_3, \leq_3)$ mpo complete, iar $f : A_1 \rightarrow A_2$ și $g : A_2 \rightarrow A_3$ funcții continue.

Fie $L \subseteq A_1$ un lanț nevid. Deoarece f este continuă, $f(L) \subseteq A_2$ este lanț nevid, există $sup(f(L))$ și $f(sup(f(L))) = sup(f(L))$.

$f(L)$ fiind lanț și g fiind continuă, există $sup(g(f(L)))$ și

$$g(sup(f(L))) = sup(g(f(L))).$$

Conbinând această relație cu cea de mai sus obținem:

$$g(f(sup(f(L)))) = g(sup(f(L))) = sup(g(f(L))),$$

ceea ce arată că $g \circ f$ este funcție continuă. \square

Continuitatea funcțiilor de tipul $f : A \rightarrow A_1 \times \cdots \times A_n$. Studiul continuității acestui tip de funcții face apel direct la funcțiile proiecție care sunt continue.

Teorema 7.2.1.3. Fie $M = (A, \leq)$ și $M_i = (A_i, \leq_i)$ mpo complete, unde $1 \leq i \leq n$ și $n \geq 2$, și $f : A \rightarrow A_1 \times \cdots \times A_n$ o funcție. Atunci f este continuă dacă și numai dacă $pr_i \circ f$ sunt funcții continue, pentru orice $1 \leq i \leq n$.

Demonstrație. Să presupunem că f este continuă. Atunci, cum funcția proiecție pr_i este continuă și compunere de funcții continue este funcție continuă, rezultă că $pr_i \circ f$ este funcție continuă, pentru orice i .

Reciproc, presupunem că pentru orice i , $pr_i \circ f$ este funcție 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 definiției relației de ordine parțială \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 lanț nevid. Pentru a arăta că

$$f(\sup(L)) \leq' \sup(f(L))$$

avem de demonstrat că

$$pr_i(f(\sup(L))) \leq_i pr_i(\sup(f(L))),$$

pentru orice i . Are loc:

$$\begin{aligned} pr_i(f(\sup(L))) &= \sup(pr_i(f(L))) && (pr_i \circ f \text{ este continuă}) \\ &= pr_i(\sup(f(L))), && (pr_i \text{ este continuă și } f(L) \text{ este lanț}) \end{aligned}$$

pentru orice i . Deci f este continuă. \square

Corolarul 7.2.1.2. Fie $M = (A, \leq)$ și $M_i = (A_i, \leq_i)$ mpo complete, unde $1 \leq i \leq n$ și $n \geq 2$, și $f_i : A_i \rightarrow A'_i$ funcții continue. Atunci funcția $f : A \rightarrow A_1 \times \cdots \times A_n$ data prin $f(a) = (f_1(a), \dots, f_n(a))$, pentru orice $a \in A$, este continuă.

Demonstrație. De la Teorema 7.2.1.3 și relația $pr_i \circ f = f_i$, pentru orice i . \square

În mod ușual, funcția din Corolarul 7.2.1.2 se notează prin (f_1, \dots, f_n) .

Corolarul 7.2.1.3. Fie $M = (A, \leq)$ și $M_i = (A_i, \leq_i)$ mpo complete și $f_i : A_i \rightarrow A'_i$ funcții continue, unde $1 \leq i \leq n$ și $n \geq 2$. Atunci funcția f definită pe $A_1 \times \cdots \times A_n$ și cu valori în $A'_1 \times \cdots \times A'_n$ data prin $f(a_1, \dots, a_n) = (f_1(a_1), \dots, f_n(a_n))$, pentru orice $(a_1, \dots, a_n) \in A_1 \times \cdots \times A_n$, este continuă.

Demonstrație. Putem scrie:

$$f(a_1, \dots, a_n) = ((f_1 \circ pr_1)(a_1, \dots, a_n), \dots, (f_n \circ pr_n)(a_1, \dots, a_n)),$$

pentru orice $(a_1, \dots, a_n) \in A_1 \times \cdots \times A_n$. \square

Atunci corolarul urmează de la Teorema 7.2.1.3 și Corolarul 7.2.1.2. \square

Corolarul 7.2.1.4. Fie $M = (A, \leq)$, $M_i = (A_i, \leq_i)$ și $M' = (A', \leq')$ mpo complete, unde $1 \leq i \leq n$ și $n \geq 2$, iar $f_i : A \rightarrow A_i$ și $f : A_1 \times \cdots \times A_n \rightarrow A'$ funcții continue. Atunci funcția $f \circ (f_1, \dots, f_n)$ este continuă.

Demonstrație. De la Corolarul 7.2.1.2 și Teorema 7.2.1.2. \square

Continuitatea funcțiilor de tipul $f : A_1 \times \cdots \times A_n \rightarrow A$. Studiul continuității acestui tip de funcții face apel la conceptul de *funcție Curry*².

Definiția 7.2.1.2. Fie $f : A_1 \times \cdots \times A_n \rightarrow A$ o funcție, unde $n \geq 2$. *Funcția Curry asociată funcției* f este funcția $f^c : A_1 \times \cdots \times A_{n-1} \rightarrow (A_n \rightarrow A)$ dată prin

$$f^c(a_1, \dots, a_{n-1})(a_n) = f(a_1, \dots, a_n),$$

pentru orice $(a_1, \dots, a_{n-1}) \in A_1 \times \cdots \times A_{n-1}$ și $a_n \in A_n$.

Vom conveni ca funcțiile $f^c(a_1, \dots, a_{n-1}) : A_n \rightarrow A$ definite ca în Definiția 7.2.1.2, pentru orice $(a_1, \dots, a_{n-1}) \in A_1 \times \cdots \times A_{n-1}$, să fie numite tot *funcții Curry asociate funcției* f .

Teorema 7.2.1.4. Fie $M = (A, \leq)$ și $M_i = (A_i, \leq_i)$ mpo, unde $1 \leq i \leq n$ și $n \geq 2$, și $f : A_1 \times \cdots \times A_n \rightarrow A$ o funcție. Atunci f este monotonă dacă și numai dacă funcțiile Curry asociate funcției f sunt monotone.

Demonstrație. Vom face demonstrația pentru cazul $n = 2$, raționamentul putând fi extins cu ușurință la cazul $n \geq 2$.

Presupunem că f este funcție monotonă. Fie $a_1 \in A_1$. Vom arăta că $f^c(a_1)$ este funcție monotonă. Fie $a_2, a'_2 \in A_2$ cu $a_2 \leq_2 a'_2$. Atunci:

$$\begin{aligned} f^c(a_1)(a_2) &= f(a_1, a_2) \\ &\leq f(a_1, a'_2) && ((a_1, a_2) \leq' (a_1, a'_2) \text{ și } f \text{ monotonă}) \\ &= f^c(a_1)(a'_2), \end{aligned}$$

ceea ce ne arată că $f^c(a_1)$ este monotonă (\leq' este ordinea parțială pe $A_1 \times A_2$).

Vom arăta acum că f^c este funcție monotonă. Fie $a_1, a'_1 \in A_1$ cu $a_1 \leq_1 a'_1$. Trebuie să arătăm că $f^c(a_1) \leq_{(A_2 \rightarrow A)} f^c(a'_1)$. Fie $a_2 \in A_2$. Atunci:

$$\begin{aligned} f^c(a_1)(a_2) &= f(a_1, a_2) \\ &\leq f(a'_1, a_2) && ((a_1, a_2) \leq' (a'_1, a_2) \text{ și } f \text{ monotonă}) \\ &= f^c(a'_1)(a_2), \end{aligned}$$

ceea ce ne arată că f^c este monotonă.

²Denumirea acestor funcții provine de la numele logicianului american Haskell B. Curry (1900-1982). Așa cum menționează logicianul rus Moses Schönfinkel în [185], aceste tipuri de funcții 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 funcții.

Reciproc, presupunem că funcțiile Curry asociate funcției 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:

$$\begin{aligned} f(a_1, a_2) &= f^c(a_1)(a_2) \\ &\leq f^c(a_1)(a'_2) \quad (f^c(a_1) \text{ monotonă}) \\ &\leq f^c(a'_1)(a'_2) \quad (f^c \text{ monotonă}) \\ &= f(a'_1, a'_2), \end{aligned}$$

ceea ce ne arată că f este monotonă. \square

Teorema 7.2.1.5. Fie $M = (A, \leq)$ și $(M_i = (A_i, \leq_i))$ mpo complete, unde $1 \leq i \leq n$ și $n \geq 2$, și $f : A_1 \times \cdots \times A_n \rightarrow A$ o funcție. Atunci f este continuă dacă și numai dacă funcțiile Curry asociate funcției f sunt continue.

Demonstrație. Ca și în cazul Teoremei 7.2.1.4, vom face demonstrația doar pentru $n = 2$.

Presupunem că f este funcție continuă. Atunci f este monotonă și, conform Teoremei 7.2.1.4, funcțiile Curry asociate sunt monotone.

Fie $a_1 \in A_1$ și $L \subseteq A_2$ un lanț nevid. Vom arăta că este satisfăcută relația

$$f^c(a_1)(\sup(L)) = \sup(f^c(a_1)(L)).$$

Aceea loc:

$$\begin{aligned} f^c(a_1)(\sup(L)) &= f(a_1, \sup(L)) \\ &= f(\sup(\{a_1\}), \sup(L)) \\ &= f(\sup(\{a_1\}) \times L) \quad (\{a_1\} \times L \text{ este lanț}) \\ &= \sup(f(\{a_1\} \times L)) \quad (f \text{ continuă}) \\ &= \sup(f^c(a_1)(L)), \end{aligned}$$

ceea ce arată că $f^c(a_1)$ este funcție continuă.

Fie acum $L \subseteq A_1$ un lanț nevid. Vom arăta că $f^c(\sup(L)) = \sup(f^c(L))$. Fie $a_2 \in A_2$. Atunci:

$$\begin{aligned} f^c(\sup(L))(a_2) &= f(\sup(L), a_2) \\ &= f(\sup(L), \sup(\{a_2\})) \\ &= f(\sup(L \times \{a_2\})) \quad (L \times \{a_2\} \text{ lanț}) \\ &= \sup(f(L \times \{a_2\})) \quad (f \text{ continuă}) \\ &= \sup(f^c(L)(a_2)) \\ &= \sup(f^c(L))(a_2), \quad (\text{Lema 7.1.3.2}) \end{aligned}$$

ceea ce arată că f^c este funcție continuă.

Reciproc, presupunem că funcțiile Curry asociate funcției f sunt continue. Deci ele sunt și monotone, ceea ce conduce la faptul că f este monotonă.

Fie $L \subseteq A_1 \times A_2$ un lanț nevid. Vom arăta că are loc $f(\sup(L)) \leq' \sup(f(L))$. Considerăm $L_1 = \{a | (\exists b)((a, b) \in L)\}$ și $L_2 = \{b | (\exists a)((a, b) \in L)\}$. Este clar că L_1 și L_2 sunt lanțuri nevide în M_1 și, respectiv, M_2 , și $\sup(L) = (\sup(L_1), \sup(L_2))$. Atunci:

$$\begin{aligned} f(\sup(L)) &= f(\sup(L_1), \sup(L_2)) \\ &= f^c(\sup(L_1))(\sup(L_2)) \\ &= \sup(f^c(\sup(L_1))(L_2)) \\ &= \sup(\{f^c(\sup(L_1))(a_2) | a_2 \in L_2\}) \\ &= \sup(\{\sup(f^c(L_1)(a_2)) | a_2 \in A_2\}) \\ &= \sup(\{\sup(\{f^c(a_1)(a_2) | a_1 \in L_1\}) | a_2 \in L_2\}) \\ &= \sup(\{\sup(\{f(a_1, a_2) | a_1 \in L_1\}) | a_2 \in L_2\}) \\ &= \sup(\{f(a_1, a_2) | a_1 \in L_1, a_2 \in L_2\}) \end{aligned}$$

(pentru ultima egalitate se verifică că sunt îndeplinite ipotezele Propoziției 1.4.2.5).

Vom arăta că $f(L)$ este cofinală în $f(L_1 \times L_2)$, ceea ce va stabili continuitatea funcției f .

Fie $(a_1, a_2) \in L_1 \times L_2$. Atunci există a'_1 și a'_2 astfel încât $(a_1, a'_2), (a'_1, a_2) \in L$. Fie \leq' ordinea parțială 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 $f(L_1 \times L_2)$ și demonstrația este astfel încheiată. \square

Prin $[M_1 \rightarrow M_2]$, sau $[A_1 \rightarrow A_2]$, atunci când M_1 și M_2 sunt subînțelese din context, vom nota mulțimea tuturor funcțiilor continue de la mpo completă $M_1 = (A_1, \leq_1)$ la mpo completă $M_2 = (A_2, \leq_2)$.

Corolarul 7.2.1.5. Fie $M_1 = (A_1, \leq_1)$ și $M_2 = (A_2, \leq_2)$ mpo complete. Atunci funcția $\psi : [A_1 \rightarrow A_2] \times A_1 \rightarrow A_2$ dată prin $\psi(f, a) = f(a)$, pentru orice $a \in A_1$ și $f \in [A_1 \rightarrow A_2]$, este continuă.

Demonstrație. Funcția ψ^c este funcția identitate, iar pentru $f \in [A_1 \rightarrow A_2]$, $\psi^c(f)$ este funcția f . Ca urmare, funcțiile Curry asociate funcției ψ sunt continue, ceea ce conduce la faptul că ψ este continuă. \square

Teorema 7.2.1.6. Fie $M_1 = (A_1, \leq_1)$ și $M_2 = (A_2, \leq_2)$ mpo complete. Atunci $[A_1 \rightarrow A_2]$ este sub-mpo completă a mpo completei $(A_1 \rightarrow_m A_2)$.

Demonstrație. Cel mai mic element al mulțimii $(A_1 \rightarrow_m A_2)$ este și cel mai mic element al mulțimii $[A_1 \rightarrow A_2]$.

Fie $L \subseteq [A_1 \rightarrow A_2]$ un lanț nevid. Deoarece orice funcție continuă este monotonă, lanțul L admite supremum în $(A_1 \rightarrow_m A_2)$. Vom arăta că acest supremum este

funcție continuă (el fiind funcție monotonă).

Fie $K \subseteq A_1$ un lanț nevid. Atunci:

$$\begin{aligned} (\sup_{(A_1 \rightarrow_m A_2)}(L))(\sup(K)) &= \sup(L(\sup(K))) \\ &= \sup(\{f(\sup(K))|f \in L\}) \\ &= \sup(\{\sup(f(K))|f \in L\}) \\ &= \sup(\{\sup(\{f(a)|a \in K\})|f \in L\}) \\ &= \sup(\{f(a)|a \in K, f \in L\}) \\ &= \sup((\sup_{(A_1 \rightarrow_m A_2)}(L))(K)), \end{aligned}$$

ceea ce arată că $\sup_{(A_1 \rightarrow_m A_2)}(L)$ este funcție continuă (aceste egalități urmează de la Lemă 7.1.3.2, de la faptul că f este continuă și de la Propoziția 1.4.2.5). \square

Observația 7.2.1.4. Teorema 7.2.1.6 poate fi reformulată echivalent astfel: “Fie $M = (A, \leq)$ și $M' = (A', \leq')$ două mpo. Dacă M și M' sunt complete, atunci supremumul oricărui lanț nevid de funcții continue de la A la A' este funcție continuă”.

7.2.2. Puncte fixe

Stabilirea existenței punctelor fixe ale unei funcții, cât și determinarea acestora, este de importanță uriașă în matematică și informatică. De exemplu, semantica denotațională a structurilor repetitive se bazează pe determinarea celui mai mic punct fix al unei funcții continue (detalii asupra aplicațiilor teoriei punctelor fixe vor fi date în Secțiunea 7.3.1).

Definiția 7.2.2.1. Fie A o mulțime nevidă și $f : A \rightarrow A$ o funcție. Se numește *punct fix* al funcției f orice element $a \in A$ cu proprietatea $f(a) = a$.

În această secțiune vom studia existența punctelor fixe pentru funcții monotone și continue definite pe mpo complete în unul din sensurile deja studiate. Înainte de aceasta facem observația că o funcție poate să nu aibă nici un punct fix, poate avea un număr finit de puncte fixe sau chiar o infinitate de puncte fixe. În plus, dacă funcția este definită pe o mpo, atunci putem discuta despre puncte fixe minime 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 7.2.2.1.³ Fie A o mulțime și $f : \mathcal{P}(A) \rightarrow \mathcal{P}(A)$ o funcție monotonă. Să considerăm o submulțime $X \subseteq A$ cu proprietatea $X \subseteq f(X)$ (evident, mulțimea vidă

³Rezultatul ce face subiectul acestui exemplu a fost descoperit de Knaster și Tarski în 1927 (conform celor menționate de Tarski în [203]). În [105] sunt prezentate un număr de aplicații ale acestui rezultat.

satisfac această proprietate). Dacă aplicăm f acestei submulțimi obținem $f(X)$ ce este inclusă în $f^2(X)$ în baza monotoniei funcției f . Repetând procedeul obținem

$$f^2(X) \subseteq f^3(X).$$

Intuitiv, continuând acest procedeu oricât de mult, mulțimea “limită” Y care s-ar obține nu s-ar mai modifica prin aplicarea funcției f , adică ea ar satisface $f(Y) = Y$. Această mulțime limită nu este alta decât

$$Y = \sup(\{X, f(X), f^2(X), \dots\})$$

(și ea există deoarece este supremumul unui lanț într-o mpo completă, în acest caz supremumul fiind reuniunea lanțului). Este ușor de văzut că, în adevăr, $f(Y) = Y$ deoarece:

$$\begin{aligned} Y &= \sup(\{X, f(X), f^2(X), \dots\}) \\ &= \sup(\{f(X), f^2(X), f^3(X), \dots\}) \\ &= \bigcup_{i \geq 1} f^i(X) \\ &= f(\bigcup_{i \geq 0} f^i(X)) \\ &= f(\sup(\{X, f(X), f^2(X), \dots\})) \\ &= f(Y). \end{aligned}$$

Ca urmare, pornind de la o submulțime X a mulțimii A ce satisfac $X \subseteq f(X)$, am putut pune în evidență atât un punct fix, cât și modul de determinare al acestuia.

Dar dacă dorim să determinăm cel mai mic punct fix al funcției f ? Intuitiv, pentru determinarea acestuia, ar trebui să pornim cu \emptyset .

În adevăr, $Y_0 = \sup(\{\emptyset, f(\emptyset), f^2(\emptyset), \dots\})$ este punct fix al funcției f și, dacă Z este un alt punct fix al acesteia, atunci relațiile:

- $\emptyset \subseteq Z$;
- $f(\emptyset) \subseteq f(Z) = Z$, pe baza monotoniei funcției f și a faptului că $f(Z) = Z$;
- $f(f(\emptyset)) \subseteq f(Z) = Z$ etc.

conduc la

$$Y_0 = \sup(\{\emptyset, f(\emptyset), f^2(\emptyset), \dots\}) \subseteq Z.$$

Deci Y_0 este cel mai mic punct fix al funcției f .

Exemplul 7.2.2.2. Să analizăm acum ce elemente de bază s-au folosit în obținerea rezultatelor din exemplul anterior:

- în primul rând, s-a folosit faptul că există supremumul lanțurilor. De fapt, $(\mathcal{P}(A), \subseteq)$ este o mpo completă;

- în al doilea rând, s-a utilizat faptul că $\sup(L) = \sup(L - \{X\})$, unde X este primul element al lanțului L . Această proprietate este însă satisfăcută în orice mpo de orice lanț ce are cel puțin 2 elemente (eliminarea celui mai mic element al lanțului, atunci când există, nu modifică supremumul lanțului);
- cea de a treia proprietate utilizată este $\sup(f(L)) = f(\sup(L))$, pentru orice lanț L . Această proprietate nu ne spune altceva decât că f este, de fapt, o funcție 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 $(\mathcal{P}(A), \subseteq)$. Existența celui mai mic element este însă garantată în orice mpo completă.

Ca urmare, rezultatul din Exemplul 7.2.2.1 poate fi generalizat la arăta că orice funcție continuă definită pe o mpo completă are un cel mai mic punct fix. Să intrăm puțin în detaliu.

Fie $M = (A, \leq)$ o mpo completă și $f : A \rightarrow A$ o funcție continuă. Prin inducție după $i \geq 0$ se arată cu ușurință 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 lanț în M . Completitudinea mpo M asigură existența supremumului acestui lanț. 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 inducție după $i \geq 0$ se obține $f^i(\perp_A) \leq a$, pentru orice $i \geq 0$, ceea ce arată că $\sup(L) \leq a$. Deci, $\sup(L)$ este cel mai mic punct fix al funcției f .

Rezultatul din Exemplul 7.2.2.2 poate fi întărit considerabil. Începem printr-un rezultat frecvent întâlnit în analiza matematică [20].

Teorema 7.2.2.1. Fie $M = (A, \leq)$ o mpo slab completă și nevidă, iar $f : A \rightarrow A$ o funcție ce satisfacă $a \leq f(a)$, pentru orice $a \in A$. Atunci f are cel puțin un punct fix.

Demonstrație. Conform Principiului de maximalitate al lui Hausdorff, M are cel puțin un lanț maximal. Fie L un astfel de lanț. Conform ipotezei, există $\sup(L)$. Proprietatea pe care o satisfacă funcția f ne spune că $\sup(L)$ este punct fix al acesteia.

Demonstrația de mai sus face apel la Principiul de maximalitate al lui Hausdorff și, implicit, la Axioma alegerii. Se poate da o demonstrație directă, mergându-se pe ideea de mai sus, și anume de a demonstra existența unui lanț maximal, având ca ipoteză faptul că M este slab completă. Să intrăm puțin în detaliu.

Fie $a \in A$ un element arbitrar (M este nevidă). Dacă a este maximal, atunci a este punct fix al funcției f . Altfel, considerăm mulțimea B a tuturor submulțimilor B ale lui A ce satisfac:

- conțin a ca prim element;
- $f(B) \subseteq B$;
- $\sup_A(L) \in B$, pentru orice lanț nevid L în B .

Fie $L = \bigcup \mathcal{B}$. L este nevidă deoarece conține cel puțin a , $f(L) \subseteq L$ în baza celei de a doua proprietăți, și supremumul oricărui lanț nevid din L este tot în L (supremumul fiind calculat în M). Dacă arătăm că L este lanț, atunci demonstrația teoremei este încheiată conform celor spuse mai sus.

Fie $c \in L$. Vom spune că c este *element extrem* dacă are loc

$$(\forall x \in L)(x < c \Rightarrow f(x) \leq c).$$

Considerăm acum mulțimea L_c a elementelor din lanțul L ce se găsesc în afara intervalului $(c, f(c))$. Adică

$$L_c = \{x \in L | x \leq c \vee f(c) \leq x\}.$$

Vom arăta următoarele:

1. $L_c = L$, pentru orice element extrem $c \in L$. Pentru aceasta este suficient de arătat că L_c este în \mathcal{B} . Conform ipotezei, $a \in L_c$.

Fie $x \in L_c$. Dacă $x < c$, atunci $f(x) \leq c$ și, deci, $f(x) \in L_c$. Dacă $x = c$, atunci $f(x) = f(c)$ și, deci, $f(x) \in L_c$. Deci $f(L_c) \subseteq L_c$.

Fie $C \subseteq L_c$ un lanț nevid. Conform ipotezei, există $\sup_A(C)$. Dacă toate elementele lanțului C sunt mai mici sau egale cu c , atunci $\sup_A(C) \leq c$, ceea ce ne arată că $\sup_A(C) \in L_c$. Dacă există $x \in C$ astfel încât $f(c) \leq x$, atunci $f(c) \leq x \leq \sup_A(C)$, ceea ce arată, și în acest caz, că $\sup_A(C)$ este în L_c .

Ca urmare, $L_c \in \mathcal{B}$, ceea ce ne arată că $L_c = L$;

2. orice element $c \in L$ este extrem. Fie $E \subseteq L$ mulțimea tuturor elementelor extreme ale lui L . Pentru a arăta că $E = L$ este suficient de arătat că $E \in \mathcal{B}$.

Conform ipotezei asupra elementului a , $a \in E$. Fie $c \in E$. Trebuie să arătăm că $f(c) \in E$. Fie $x \in L$ cu $x < f(c)$. În baza punctului anterior, $L = L_c$ și, deci, $x < c$ sau $x = c$ sau $f(c) \leq x$. Ultima posibilitate nu poate avea loc deoarece $x < f(c)$. Dacă are loc prima, atunci $f(x) \leq c \leq f(c)$, iar dacă are loc a doua, $f(x) = f(c)$. Deci $f(E) \subseteq E$.

Fie $C \subseteq E$ un lanț nevid. Pentru a obține $\sup_A(C) \in E$ va trebui să arătăm că $f(x) \leq \sup_A(C)$, pentru orice x din L cu $x < \sup_A(C)$. Fie deci $x \in L$ cu $x < \sup_A(C)$.

Dacă pentru fiecare $c \in E$ are loc $f(c) \leq x$, atunci $c \leq f(c) \leq x$ ceea ce ne arată că $\sup_A(C) \leq c$ și, deci, $\sup_A(C) \in E$. Altfel, deoarece $L = L_c$ pentru orice $c \in E$, trebuie să existe $c \in E$ cu $x \leq c$. Dacă $x < c$, atunci $f(x) \leq c \leq \sup_A(C)$,

iar dacă $x = c$, atunci $f(x) = f(c) \in E$ ceea ce ne arată că $f(x) \leq \sup_A(C)$. Deci, $\sup_A(C) \in E$.

Ca urmare, $E \in \mathcal{B}$, ceea ce ne arată că L este format numai din elemente extreme.

În baza acestor două rezultate putem arăta cu ușurință că L este lanț. În adevăr, dacă $x, y \in L$, atunci x este element extrem și $L = L_x$. Atunci, $y \leq x$ sau $f(x) \leq y$, ceea ce ne arată că x și y sunt comparabile. \square

Aplicarea Teoremei 7.2.2.1 se va face, în principiu, astfel. Datează o mpo $M = (A, \leq)$ și o funcție f , vom considera submulțimea $B = \{a \in A | a \leq f(a)\}$. Dacă ipotezele suplimentare asigură că:

- B este slab completă (cu ordinea parțială indușă de M);
- $f(B) \subseteq B$ (adică, $f|_B$ este funcție de la B la B),

atunci $f|_B$ admite cel puțin un punct fix în B , ceea ce ne spune că f va admite cel puțin un punct fix.

Următorul corolar urmează din plin această linie.

Corolarul 7.2.2.1. Fie $M = (A, \leq)$ o mpo completă și $f : A \rightarrow A$ o funcție monotonă. Atunci f are cel puțin un punct fix.

Demonstrație. Fie $B = \{a \in A | a \leq f(a)\}$. B este nevidă deoarece conține \perp și, în raport cu ordinea indușă, este mpo completă. În adevăr, dacă L este un lanț nevid în B , atunci există $\sup_A(L) \in B$ va trebui să arătăm că are loc $\sup_A(L) \leq f(\sup_A(L))$. Aceasta urmează cu ușurință de la

$$a \leq f(a) \leq f(\sup_A(L))$$

pentru orice $a \in L$, conform definiției mulțimii B , monotoniei funcției f și a faptului că $a \leq \sup_A(L)$. În plus, $f(B) \subseteq B$, ca urmare a monotoniei funcției f .

Deci putem aplica Teorema 7.2.2.1 funcției $f|_B : B \rightarrow B$ și, conform acesteia, $f|_B$ va admite cel puțin un punct fix. Orice punct fix al funcției $f|_B$ este punct fix al funcției f . \square

Cum mpo D-complete sau laticile complete sunt cazuri particulare de mpo complete, deducem că orice funcție monotonă definită pe o astfel de mpo cu valori în aceeași mpo admite cel puțin un punct fix.

Datează o mpo $M = (A, \leq)$ și o funcție $f : A \rightarrow A$, vom nota prin $fix_M(f)$ mulțimea tuturor punctelor fixe ale funcției f și prin $\mu_M(f)$ cel mai mic punct fix al funcției f (atunci când acesta există). Aceste notări vor fi simplificate la $fix_A(f)$, $fix(f)$, $\mu_A(f)$ sau $\mu(f)$, ori de câte ori mulțimea parțial ordonată M este subînteleasă din context.

Teorema 7.2.2.2. Fie $M = (A, \leq)$ o mpo completă și $f : A \rightarrow A$ o funcție monotonă. Atunci, au loc următoarele proprietăți:

- (1) există cel mai mic punct fix al funcției f ;
- (2) $\mu(f) \leq a$, pentru orice a ce satisfacă $f(a) \leq a$;
- (3) $fix(f)$, cu ordinea indușă, este mpo completă.

Demonstrație. (1) Conform Corolarului 7.2.2.1, $fix(f)$ este nevidă. Fie S mulțimea

$$S = \{a \in A | a \leq f(a) \wedge (\forall x \in fix(f))(a \leq x)\}.$$

Vom arăta că S este mpo completă (în raport cu ordinea indușă de M). În primul rând, $\perp \in S$. Fie acum $L \subseteq S$ un lanț nevid. Ca în demonstrația Corolarului 7.2.2.1 se arată că $\sup_A(L) \leq f(\sup_A(L))$. Cum orice element din $fix(f)$ este majorant pentru S , deducem că are loc $\sup_A(L) \leq x$, pentru orice $x \in fix(f)$. Combinând cu inegalitatea de mai sus obținem că $\sup_A(L)$ este un element din S , deci S este completă.

Datorită monotoniei funcției f și a faptului că $f(x) = x$, pentru orice $x \in fix(f)$, obținem că are loc $f(S) \subseteq S$. Ca urmare, aplicând Corolarul 7.2.2.1 funcției $f|_S$, deducem că $f|_S$ are cel puțin un punct fix în S . Acesta este și punct fix al funcției f și, fiind în S , va fi mai mic decât orice alt punct fix al funcției f . Ca urmare, el este cel mai mic punct fix al funcției f .

(2) Fie a astfel încât $f(a) \leq a$. Segmentul $[\perp, a] \subseteq A$ este mpo completă (cu ordinea indușă de M) ce satisfacă $f([\perp, a]) \subseteq [\perp, a]$ (conform ipotezei și a faptului că f este monotonă). Aplicând Corolarul 7.2.2.1 funcției $f|_{[\perp, a]}$ obținem că f are cel puțin un punct fix în $[\perp, a]$. Ca urmare, $\mu(f) \leq a$.

(3) Conform punctului (1), pentru a arăta că $fix(f)$ este mpo completă ne rămâne de arătat că orice lanț nevid $L \subseteq fix(f)$ admite supremum în $fix(f)$.

Dacă L este un astfel de lanț, considerăm mulțimea $\sup_A(L) \uparrow \subseteq A$ care este ușor de văzut că este completă. În plus, $f(\sup_A(L) \uparrow) \subseteq \sup_A(L) \uparrow$. Ca urmare, în baza punctului (1) aplicat restricției funcției f la $\sup_A(L) \uparrow$, deducem că f are cel mai mic punct fix în $\sup_A(L) \uparrow$. Acest cel mai mic punct fix este $\sup_{fix(f)}(L)$. Ca urmare, L admite supremum în $fix(f)$. \square

În 1955, Alfred Tarski a arătat că mulțimea punctelor fixe ale unei funcții monotone definite pe o latice completă (cu valori în aceeași latice) formează o latice completă [203]⁴. Acest rezultat poate fi dedus și din teorema anterioară.

Corolarul 7.2.2.2. Fie $M = (A, \leq)$ o latice completă și $f : A \rightarrow A$ o funcție monotonă. Atunci $fix(f)$ cu ordinea indușă formează o latice completă.

⁴Așa cum menționează Tarski în [203], acest rezultat, apărut în Pacific Journal of Mathematics în 1955, ar fi fost obținut în 1939 și comunicat prin lecții publice în perioada 1939–1942.

Demonstrație. Cum orice latice completă este și mpo completă, Teorema 7.2.2.2 conduce la faptul că $fix(f)$ este mpo completă. Conform Teoremei 7.1.1.1, ceea ce ne rămâne de arătat este că $fix(f)$ este sup-semilatice, adică există $sup_{fix(f)}(\{a, b\})$, pentru orice $a, b \in fix(f)$.

Fie $c = sup_A(\{a, b\})$. Multimea c^\uparrow este strict completă și $f(c^\uparrow) \subseteq c^\uparrow$ (ceea ce se arată ca în demonstrația punctului (3) al Teoremei 7.2.2.2). Aplicând Teorema 7.2.2.2(1) acestei mulțimi și funcției f restricționată la c^\uparrow , deducem că există cel mai mic punct fix al acestei funcții în c^\uparrow , iar acesta este chiar $sup_{fix(f)}(\{a, b\})$. \square

Pentru completitudinea laticilor, faptul că orice funcție monotonă are cel puțin un punct fix este definitoriu. Acest lucru a fost remarcat de Anne Davis în 1955 [41]⁵.

Teorema 7.2.2.3. Fie $M = (A, \leq)$ o latice. Dacă orice funcție monotonă de la M la M are cel puțin un punct fix, atunci M este completă.

Demonstrație. Presupunem că orice funcție monotonă de la M la M are cel puțin un punct fix, dar M nu este completă. Atunci, în baza dualei Teoremei 7.1.1.1 va exista un lanț bine ordonat L ce nu admite infimum și, ca în demonstrația aceleiași teoreme deducem că există un lanț maximal bine ordonat L' în L^- .

Prima observație constă în aceea că nu poate exista $x \in A$ astfel încât $c \leq x \leq b$ pentru orice $b \in L$ și $c \in L'$. Aceasta pentru că, altfel, un element x ce ar verifica această proprietate ar fi minorant al lanțului L . Cum L nu admite infimum, există $y \not\leq x$ minorant pentru L . Dar atunci este ușor de văzut că $sup(\{x, y\})$ este minorant pentru L ce satisface $sup(\{x, y\}) > x$. Mai mult, $L' \cup \{sup(\{x, y\})\}$ ar fi lanț în L^- , ceea ce ar contrazice maximalitatea lui L' . Deci nu poate exista $x \in A$ astfel încât $c \leq x \leq b$ pentru orice $b \in L$ și $c \in L'$.

Pentru orice $a \in A$ definim mulțimile

$$L(a) = \{b \in L \mid a \not\leq b\}$$

și

$$L'(a) = \{c \in L' \mid a \not\geq c\}.$$

În baza observației de mai sus, ori $L(a) \neq \emptyset$, ori $L'(a) \neq \emptyset$, pentru orice $a \in A$. Considerăm funcția $f : A \rightarrow A$ dată prin

$$f(a) = \begin{cases} max(L(a)), & \text{dacă } L(a) \neq \emptyset \\ min(L'(a)), & \text{altfel,} \end{cases}$$

pentru orice $a \in A$. Este clar că f este bine definită și, în plus, nu are nici un punct fix deoarece ori $a \not\leq f(a)$, ori $a \not\geq f(a)$, pentru orice $a \in A$.

Ne mai rămâne de arătat că f este monotonă. Fie $x, y \in A$ cu $x \leq y$. Considerăm următoarele cazuri:

⁵Este interesant de observat că lucrarea lui Anne Davis urmează imediat lucrării lui Alfred Tarski în același număr al revistei Pacific Journal of Mathematics.

- dacă $L(x) = \emptyset$, dar $L'(y) \neq \emptyset$, atunci $f(x) \in L'$ și $f(y) \in L$, ceea ce ne arată că $f(x) \leq f(y)$;
- dacă $L(x) = \emptyset = L'(y)$, atunci $f(x) \leq f(y)$ deoarece $L'(y) \subseteq L'(x)$;
- dacă $L(x) \neq \emptyset$, atunci $L'(y) \neq \emptyset$ și $f(x) \leq f(y)$ deoarece $L(x) \subseteq L(y)$.

Deci, funcția f este monotonă și nu are nici un punct fix, ceea ce constituie o contradicție. \square

Corolarul 7.2.2.3. O latice M este completă dacă și numai dacă orice funcție monotonă de la M la M are cel puțin un punct fix.

Demonstrație. De la Corolarul 7.2.2.2 și Teorema 7.2.2.3 \square

Dacă existența punctelor fixe ale funcțiilor monotone este definitorie pentru completitudinea laticilor, existența celui mai mic punct fix al funcțiilor monotone se dovedește definitorie pentru mpo complete [137].

Teorema 7.2.2.4. Fie $M = (A, \leq)$ o mpo. Atunci următoarele afirmații sunt echivalente:

- (1) M este completă.
- (2) Orice funcție monotonă $f : A \rightarrow A$ admite cel mai mic punct fix.
- (3) Orice funcție $f : A \rightarrow A$ ce păstrează infimum admite cel mai mic punct fix.

Demonstrație. Este clar că (1) implică atât (2) cât și (3), iar (2) implică (3). Ca urmare, ne rămâne de arătat că (3) implică (1).

Fie $L \subseteq A$ un lanț (posibil vid). Va trebui să arătăm că există supremumul acestui lanț. Ideea de demonstrație este următoarea. Vom considera mulțimea tuturor majoranților acestui lanț și vom defini o funcție ce păstrează infimumul astfel încât mulțimea punctelor fixe ale ei să fie exact mulțimea majoranților lanțului. Proprietatea (3) va conduce atunci la existența celui mai mic punct fix al acestei funcții, care, de fapt, este cel mai mic majorant al lanțului. Deci există supremumul lanțului.

Să formalizăm ideea de mai sus. Orice lanț include o submulțime cofinală bine ordonată. Cum supremumul lanțului există dacă și numai dacă există supremumul acelei submulțimi cofinale, putem presupune că L este lanț bine ordonat (altfel, raționalamentul de mai jos se poate realiza asupra unei submulțimi cofinale bine ordonate în lanț).

Fie L^+ mulțimea tuturor majoranților lanțului L și f funcția dată prin

$$f(x) = \begin{cases} x, & \text{dacă } x \in L^+ \\ min\{y \in L \mid y \not\leq x\}, & \text{dacă } x \notin L^+, \end{cases}$$

pentru orice $x \in A$. f este bine definită deoarece am presupus că L este lanț bine ordonat.

Orice element din L^+ este punct fix al funcției f și nici un alt element nu poate fi punct fix al ei. Ca urmare, $L^+ = \text{fix}(f)$. Dacă arătăm că f păstrează infimumul, atunci L^+ va trebui să fie nevidă deoarece în acest caz va exista $\mu(f) \in L^+$. În plus, $\text{sup}(L) = \mu(f)$.

Ca urmare, ceea ce ne rămâne de arătat este că f păstrează infimumul. Fie $X \subseteq A$ nevidă astfel încât există $\text{inf}(X)$. Avem de analizat două cazuri:

- dacă $X \subseteq L^+$, atunci $f(X) = X$ (conform definiției funcției f), ceea ce conduce la $\text{inf}(f(X)) = \text{inf}(X) = f(\text{inf}(X))$ (ultima egalitate urmează de la faptul că $\text{inf}(X) \in L^+$);
- dacă $X \not\subseteq L^+$, atunci considerăm $Y = X - L^+$. $\text{inf}(X)$ nu este în L^+ deoarece, altfel, X ar fi mulțime de majoranți pentru L^+ și, deci, X ar fi submulțime a lui L^+ . Deci, $\text{inf}(X) \notin L^+$. Mai mult,

$$\text{inf}(f(X)) = \text{inf}(f(Y)) = y_0,$$

unde y_0 este cel mai mic element din L astfel încât $y_0 \in f(Y)$. Acum, $y_0 \leq \text{inf}(X)$, deoarece altfel am avea $y_0 \leq x$ pentru orice $x \in X$ ceea ce ar arăta că $y_0 \notin f(Y)$.

Pentru orice $y \in L$, dacă $y < y_0$, atunci $y \leq x$ și, deci, $y \leq \text{inf}(X)$. Astfel, $f(\text{inf}(X)) = y_0 = \text{inf}(f(X))$.

Deci f păstrează infimumul. \square

7.2.3. Inducție de punct fix

Fie $M = (A, \leq)$ o mpo completă. Notăm prin μ_M , sau μ dacă M se subînțelege din context, funcția $\mu : [A \rightarrow A] \rightarrow A$ dată prin

$$\mu(f) = \text{cel mai mic punct fix al funcției } f,$$

pentru orice funcție continuă $f : A \rightarrow A$. Funcția μ_M se mai numește și *funcția de punct fix* asociată mpo-ului M .

Teorema 7.2.3.1. Funcția de punct fix asociată unei mpo complete este continuă.

Demonstrație. Fie $M = (A, \leq)$ o mpo completă. Vom arăta că μ este supremumul unui lanț de funcții continue, ceea ce va conduce la faptul că μ este continuă (a se vedea Observația 7.2.1.4).

Fie $F_i : [A \rightarrow A] \rightarrow A$ funcția dată prin $F_i(f) = f^i(\perp_A)$, pentru orice funcție continuă $f : A \rightarrow A$ și $i \geq 0$. Arătăm prin inducție după $i \geq 0$ că F_i este funcție continuă, pentru orice $i \geq 0$.

Funcția F_0 este funcția constantă \perp_A și, deci, este continuă. Dacă presupunem că F_i este continuă, unde $i \geq 0$, atunci

$$F_{i+1}(f) = f^{i+1}(\perp_A) = f(F_i(f)) = \psi(id(f), F_i(f)) = (\psi \circ (id, F_i))(f),$$

pentru orice $f \in [A \rightarrow A]$, ceea ce ne arată că $F_{i+1} = \psi \circ (id, F_i)$, unde id este funcția identitate. Deci, F_{i+1} este continuă fiind compunere de funcții continue.

Mulțimea $L = \{F_i | i \geq 0\}$ este lanț de funcții. În adevăr, pentru orice $0 \leq i \leq j$ are loc

$$F_i(f) = f^i(\perp_A) \leq f^j(\perp_A) = F_j(f),$$

pentru orice $f \in [A \rightarrow A]$.

Supremumul lanțului L este dat prin:

$$\begin{aligned} (\text{sup}(L))(f) &= \text{sup}(L(f)) && \text{(Lema 7.1.3.2)} \\ &= \text{sup}(\{F_i(f) | i \geq 0\}) \\ &= \text{sup}(\{f^i(\perp_A) | i \geq 0\}) \\ &= \mu(f), \end{aligned}$$

pentru orice $f \in [A \rightarrow A]$, ceea ce arată că $\text{sup}(L) = \mu$. Deci μ este continuă. \square

Dacă o funcție 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 funcție continuă la cel mai mic punct fix al ei.

Teorema 7.2.3.2. (Teorema lui Park)

Fie $M = (A, \leq)$ o mpo completă și $f : A \rightarrow A$ o funcție continuă. Dacă există $x \in A$ astfel încât $f(x) \leq x$, atunci $\mu(f) \leq x$.

Demonstrație. Fie $x \in A$ astfel încât $f(x) \leq x$. Monotonia funcției f conduce la

$$f(\perp_A) \leq f(x) \leq x.$$

Inductiv, obținem $f^i(\perp_A) \leq x$, pentru orice $i \geq 0$. Ca urmare,

$$\text{sup}(\{f^i(x) | i \geq 0\}) \leq x,$$

ceea ce ne arată că $\mu(f) \leq x$. \square

Definiția 7.2.3.1. Fie $M = (A, \leq)$ o mpo completă și P un predicat pe A . Spunem că P este *admisibil* dacă are loc

$$(\forall L \subseteq A \text{ lanț nevid})((\forall a \in L)(P(a)) \Rightarrow P(\text{sup}(L))).$$

Următoarea teoremă, datorată lui Dana Scott [68] și numită *Principiul inducției de punct fix*, este o simplă combinație dintre inducția matematică și conceptul de predicat admisibil.

Teorema 7.2.3.3. (Principiul inducției de punct fix)

Fie $M = (A, \leq)$ o mpo completă și $f : A \rightarrow A$ o funcție continuă. Dacă P este predicat pe A astfel încât:

- (1) P este admisibil;
- (2) $P(\perp_A)$;
- (3) $(\forall i \geq 0)(P(f^i(\perp_A)) \Rightarrow P(f^{i+1}(\perp_A)))$,

atunci $P(\mu(f))$.

Demonstrație. În baza proprietăților de la (2) și (3), prin inducție matematică, obținem că are loc $P(f^i(\perp_A))$, pentru orice $i \geq 0$. Cum P este admisibil, deducem că are loc $P(\sup(\{f^i(\perp_A) | i \geq 0\}))$, adică $P(\mu(f))$. \square

Evident că suntem interesați de existența predicatelor admisibile.

Observația 7.2.3.1.

- (1) Nu orice predicat este admisibil. Fie, de exemplu, $P : (\mathbf{N}_\perp \rightarrow \mathbf{N}_\perp) \rightarrow \{0, 1\}$ dat prin

$$P(x) = \begin{cases} 0, & (\forall n \in \mathbf{N})(f(n) \neq \perp) \\ 1, & \text{altfel} \end{cases}$$

și lanțul L din tabelul de mai jos

L	\perp	0	1	2	3	\dots
f_0	\perp	\perp	\perp	\perp	\perp	\dots
f_1	\perp	0	\perp	\perp	\perp	\dots
f_2	\perp	0	0	\perp	\perp	\dots
f_3	\perp	0	0	0	\perp	\dots
\dots	\dots	\dots	\dots	\dots	\dots	\dots
$\sup(L)$	\perp	0	0	0	\dots	\dots

$P(f_i) = 1$ 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 obținut din predicatul P de la (1) prin înlocuirea lui 0 cu 1 și a lui 1 cu 0, atunci constatăm că Q este admisibil (supremum unui lanț de funcții totale pe \mathbf{N} este o funcție totală, conform definiției ordinii parțiale pe funcții și a ordinii parțiale pe \mathbf{N}_\perp). Însă $\neg Q = P$, care nu este admisibil.

Teorema 7.2.3.4. Fie $M = (A, \leq)$ o mpo completă iar P și Q predicate pe A . Dacă P și Q sunt admisibile, atunci $P \vee Q$ și $P \wedge Q$ sunt admisibile.

Demonstrație. Admisibilitatea predicatului $P \wedge Q$ decurge imediat de la admisibilitatea predicatelor P și Q . Dacă elementele unui lanț nevid satisfac $P \wedge Q$, atunci ele satisfac atât P cât și Q . Atunci P și Q sunt satisfăcute și de supremumul lanțului L , ceea ce ne arată că $P \wedge Q$ este admisibil.

Fie $L \subseteq A$ un lanț nevid astfel încât $(P \vee Q)(a)$, pentru orice $a \in L$. Considerăm lăturile $L_1 = \{a \in L | P(a)\}$ și $L_2 = \{a \in L | Q(a)\}$. Dacă unul dintre ele este vid, atunci celălalt este chiar L , iar admisibilitatea predicatelor P și Q conduce la $(P \vee Q)(\sup(L))$. Dacă ambele sunt nevide, atunci unul dintre ele este cofinal în celălalt, ceea ce asigură că supremumul lanțului L este dat de supremumul acestuia. Atunci, admisibilitatea predicatelor P și Q conduce la $(P \vee Q)(\sup(L))$. Deci, $P \vee Q$ este admisibil. \square

Teorema 7.2.3.5. Fie $M = (A, \leq)$ și $M' = (A', \leq')$ mpo complete iar f_i și g_i funcții continue de la A la A' , unde $1 \leq i \leq n$ și $n \geq 1$. Atunci, predicatul $P : A \rightarrow \{0, 1\}$ dat prin

$$P(a) \Leftrightarrow (\forall i)(f_i(a) \leq' g_i(a)),$$

pentru orice $a \in A$, este admisibil.

Demonstrație. Fie $L \subseteq A$ un lanț 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 funcțiile f_i și g_i sunt continue, relația 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. \square

Corolarul 7.2.3.1. Fie $M = (A, \leq)$ și $M' = (A', \leq')$ mpo complete și $f, g : A \rightarrow A$ funcții continue. Atunci, predicatul P dat prin

$$P(a) \Leftrightarrow f(a) = g(a),$$

pentru orice $a \in A$, este admisibil.

Demonstrație. Considerăm predicatele P_1 și P_2 date prin

$$P_1(a) \Leftrightarrow f(a) \leq' g(a)$$

și

$$P_2(a) \Leftrightarrow g(a) \leq' f(a),$$

pentru orice $a \in A$. Conform Teoremei 7.2.3.5, P_1 și P_2 sunt admisibile. Atunci Teorema 7.2.3.4 conduce la faptul că $P_1 \wedge P_2$ este admisibil. Însă $P_1 \wedge P_2 = P$. \square

7.3. Aplicații: semantica și analiza programelor

7.3.1. Semantica programelor

Vom arăta în această secțiune cum putem utiliza aparatul mulțimilor parțial ordonate complete și al funcțiilor continue pentru a descrie semantica limbajelor de programare. Vom considera două clase de programe, *programe recursive* și *programe structurale*, și vom urma, în principal, ideile din [132, 198, 129]. Indicăm însă și [160] pentru mai multe detalii asupra semanticii limbajelor de programare.

7.3.1.1. λ-notație

La începutul anilor 1930 multe din cercetările matematice canalizate pe introducerea unui concept de funcție “efectiv calculabilă” au inceput să se finalizeze datorită efortului conjugat al mai multor matematicieni de renume precum Church, Kleene, Turing, Gödel și alții. Rezultatul este ceea ce numim azi *funcție recursivă*. Un prim pas în definirea acestora a fost făcut de Alonso Church⁶ prin introducerea *λ-notației* și a conceptului de *funcție λ-definibilă* [30] (a se vedea și [31]). Ulterior, λ-notația s-a dovedit o achiziție inestimabilă în studiul limbajelor de programare și a semanticii acestora și, în special, în cadrul limbajelor de programare funcțională.

Descrierea limbajului λ-notației parcurge două mari etape: sintaxa și semantica.

Sintaxa λ-notației se bazează pe utilizarea simbolurilor auxiliare “{”, “}”, “(”, “)”, “[”, “[”, “.” și “λ”, și pe conceptele de *tip*, *bază* și *λ-term* pe care le descriem după cum urmează:

- **Tip.** Fie \mathcal{T}_0 o mulțime ale cărei elemente le numim *tipuri de bază*. *Tipurile peste* \mathcal{T}_0 se definesc inducitiv prin:

- orice tip de bază este tip;
- dacă $\tau_1, \dots, \tau_n, \tau$ sunt tipuri, atunci $(\tau_1, \dots, \tau_n \rightarrow \tau)$ este tip.

Notăm mulțimea astfel definită prin \mathcal{T} ;

- **Bază.** O *bază* pentru λ-notație este un sistem $\mathcal{B} = (\mathcal{T}_0, \mathcal{V}, \mathcal{F})$, unde:

- \mathcal{T}_0 este o mulțime de tipuri de bază;
- \mathcal{V} este o mulțime de *variabile cu tip* (fiecare dintre ele având asociat un unic tip peste \mathcal{T}_0);
- \mathcal{F} este o mulțime de *simboluri funcționale cu tip* (fiecare dintre ele având asociat un unic tip peste \mathcal{T}_0).

⁶Stephen Cole Kleene și Alan Mathison Turing au fost doctoranți ai lui Alonso Church.

În plus, vom presupune următoarele:

- mulțimile \mathcal{V} , \mathcal{F} și \mathcal{T} sunt disjuncte două câte două;
- pentru orice tip există oricâte variabile este nevoie.

Simbolurile funcționale al căror tip va fi din \mathcal{T}_0 vor fi numite și *simboluri constante*. Variabilele vor fi notate prin $x, y, z, \dots, p, q, r, \dots, F, G, H, \dots$, iar simbolurile funcționale prin f, g, h, \dots ;

- **λ-term.** Fie \mathcal{B} o bază pentru λ-notație. *λ-termii peste* \mathcal{B} se definesc prin inducție simultană astfel:

- dacă t este variabilă sau simbol funcțional de tip τ , atunci t este *λ-term de tip* τ ;
- (*aplicație*) dacă u este λ-term de tip $(\tau_1, \dots, \tau_n \rightarrow \sigma)$, iar t_i sunt λ-termi de tip τ_i , $1 \leq i \leq n$, atunci $u(t_1, \dots, t_n)$ este *λ-term de tip* σ ;
- (*abstracție*) dacă u este λ-term de tip σ , iar x_i sunt variabile de tip τ_i , $1 \leq i \leq n$, atunci $[\lambda x_1, \dots, x_n. u]$ este *λ-term de tip* $(\tau_1, \dots, \tau_n \rightarrow \sigma)$.

Exemplul 7.3.1.1. Fie $\mathcal{T}_0 = \{nat, bool\}$. Atunci

$$nat, \text{bool}, (nat, nat \rightarrow nat) \text{ și } (nat, nat \rightarrow bool)$$

sunt tipuri peste \mathcal{T}_0 . Presupunem că x și y sunt variabile de tip *nat*, b este variabilă de tip *bool*, 2 și 3 sunt constante de tip *nat*, iar $+$ și \cdot sunt simboluri funcționale de tip $(nat, nat \rightarrow nat)$. Atunci

$$\cdot(2, x), \cdot(3, y) \text{ și } +(\cdot(2, x), \cdot(3, y))$$

sunt λ-termi de tip *nat*,

$$[\lambda x. \cdot(2, x)] \text{ și } [\lambda y. \cdot(3, y)]$$

sunt λ-termi de tip $(nat \rightarrow nat)$, iar

$$[\lambda x, y. +(\cdot(2, x), \cdot(3, y))]$$

este λ-term de tip $(nat, nat \rightarrow nat)$.

Dată o funcție $f : A_1 \rightarrow A_2$, $x \in A_1$ și $d \in A_2$, vom nota prin $f[x/d]$ funcția 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 notație poate fi extinsă în mod natural la $f[x_1/d_1] \cdots [x_n/d_n]$.

Lema 7.3.1.1. Fie A_1 o mulțime, $x_1, \dots, x_n \in A_1$ unde $n \geq 1$, și (A_2, \leq_2) o mpo completă. Atunci funcția $\psi_{x_1, \dots, x_n} : (A_1 \rightarrow A_2) \times A_2^n \rightarrow (A_1 \rightarrow A_2)$ dată prin

$$\psi_{x_1, \dots, x_n}(f, (d_1, \dots, d_n)) = f[x_1/d_1] \cdots [x_n/d_n],$$

pentru orice $f \in (A_1 \rightarrow A_2)$ și $(d_1, \dots, d_n) \in A_2^n$, este continuă.

Demonstrație. Vom face demonstrația lemei pentru $n = 1$ (aceeași demonstrație poate fi ușor generalizată la cazul $n > 1$).

Fie $L \subseteq (A_1 \rightarrow A_2) \times A_2$ un lanț nevid. Supremum acestui lanț există deoarece A_2 și $(A_1 \rightarrow A_2)$ sunt mpo complete. Mai mult, dacă notăm

$$L_1 = \{f \mid (\exists d)((f, d) \in L)\}$$

și

$$L_2 = \{d \mid (\exists f)((f, d) \in L)\},$$

atunci $\text{sup}(L) = (\text{sup}(L_1), \text{sup}(L_2))$.

Are loc

$$\text{sup}(\psi_{x_1}(L)) = \text{sup}(\{f[x_1/d_1] \mid (f, d_1) \in L\}),$$

iar în baza Lemei 7.1.3.2 obținem

$$\text{sup}(\psi_{x_1}(L))(y) = \begin{cases} \text{sup}(L_1)(y), & \text{dacă } y \neq x_1 \\ \text{sup}(L_2), & \text{altfel,} \end{cases}$$

pentru orice $y \in A_1$.

Pe de altă parte,

$$\psi_{x_1}(\text{sup}(L))(y) = \text{sup}(L_1)[x_1/\text{sup}(L_2)](y) = \begin{cases} \text{sup}(L_1)(y), & \text{dacă } y \neq x_1 \\ \text{sup}(L_2), & \text{altfel,} \end{cases}$$

pentru orice $y \in A_1$.

Ca urmare, $\text{sup}(\psi_{x_1}(L)) = \psi_{x_1}(\text{sup}(L))$, ceea ce arată că ψ_{x_1} este continuă. \square

Semantica λ-notației se bazează pe concepțele de *interpretare a unei baze*, *atribuire* și *funcție semantică*:

• **Interpretare a unei baze.** O *interpretare* a unei baze \mathcal{B} este un cuplu

$$\mathcal{I} = (((D_\tau, \leq_\tau) \mid \tau \in \mathcal{T}_0), \mathcal{I}_0),$$

unde:

– pentru orice $\tau \in \mathcal{T}_0$, (D_τ, \leq_τ) este mpo completă, numită *domeniu tipului* τ .

Pentru tipurile $\tau = (\tau_1, \dots, \tau_n \rightarrow \sigma)$ ce nu sunt de bază, domeniile se definesc inductiv prin

$$[D_{\tau_1} \times \cdots \times D_{\tau_n} \rightarrow D_\sigma];$$

– $\mathcal{I}_0 : \mathcal{F} \rightarrow \bigcup_{\tau \in \mathcal{T}} D_\tau$ este o funcție de *interpretare inițială* 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 -notație și \mathcal{I} o interpretare a ei. O *atribuire* sau *asignare* pentru baza \mathcal{B} sub interpretarea \mathcal{I} este orice funcție

$$\gamma : \mathcal{V} \rightarrow \bigcup_{\tau \in \mathcal{T}} D_\tau$$

astfel încât, pentru orice $x \in \mathcal{V}$, dacă x are tipul τ , atunci $\gamma(x) \in D_\tau$.

Vom nota prin $\Gamma_{\mathcal{B}, \mathcal{I}}$ mulțimea tuturor atribuirilor pentru baza \mathcal{B} sub interpretarea \mathcal{I} . Atunci când \mathcal{B} și \mathcal{I} sunt clare din context, notația $\Gamma_{\mathcal{B}, \mathcal{I}}$ va fi simplificată la Γ ;

• **Funcția semantică a λ -termilor.** Fie \mathcal{B} o bază, \mathcal{I} o interpretare a bazei \mathcal{B} și t un λ -term de tip τ . Dacă t nu conține variabile, atunci, intuitiv, interpretând fiecare element din t obținem un element din D_τ . Dacă însă t conține variabile, atunci, pentru fiecare atribuire a variabilelor, obținem o interpretare a λ -termului t . Deci în acest caz, interpretarea lui t trebuie să fie o funcție ce depinde de atribuirii. Ca urmare, *funcția semantică a lui t* se definește ca fiind funcția $\mathcal{I}(t) : \Gamma \rightarrow D_\tau$ 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 $t = u(t_1, \dots, t_n)$, unde u este de tip $(\tau_1, \dots, \tau_n \rightarrow \sigma)$ iar t_i sunt de tip τ_i , $1 \leq i \leq n$, atunci

$$\mathcal{I}(t)(\gamma) = \mathcal{I}(u)(\gamma)(\mathcal{I}(t_1)(\gamma), \dots, \mathcal{I}(t_n)(\gamma)),$$

pentru orice $\gamma \in \Gamma$;

- dacă $t = [\lambda x_1, \dots, x_n. u]$ este de tip $\tau = (\tau_1, \dots, \tau_n \rightarrow \sigma)$, unde u este de tip σ iar x_i sunt de tip τ_i , $1 \leq i \leq n$, atunci

$$\mathcal{I}(t)(\gamma) : D_{\tau_1} \times \cdots \times D_{\tau_n} \rightarrow D_\sigma$$

$$\mathcal{I}(t)(\gamma)(d_1, \dots, d_n) = \mathcal{I}(u)(\gamma[x_1/d_1] \cdots [x_n/d_n]),$$

pentru orice $\gamma \in \Gamma$ și $(d_1, \dots, d_n) \in D_{\tau_1} \times \cdots \times D_{\tau_n}$ (intuitiv, în acest caz, t denotă o funcție de x_1, \dots, x_n . Atunci, interpretarea lui va depinde doar de atribuirea variabilelor diferite de x_1, \dots, x_n).

În continuare vom stabili câteva proprietăți foarte importante ale mulțimii Γ și ale funcțiilor semantică a λ -termilor.

Propoziția 7.3.1.1. Fie \mathcal{B} o bază și \mathcal{I} o interpretare a ei. Atunci $\Gamma_{\mathcal{B}, \mathcal{I}}$ cu ordinea parțială pe funcții este mpo completă.

Demonstrație. 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 $(D_x, \leq_x) = (D_\tau, \leq_\tau)$. Are loc

$$\Gamma_{\mathcal{B}, \mathcal{I}} = \{\gamma : \mathcal{V} \rightarrow \bigcup_{x \in \mathcal{V}} D_x | (\forall x \in \mathcal{V})(\gamma(x) \in D_x)\} = \prod_{x \in \mathcal{V}} D_x,$$

iar ordinea parțială pe $\Gamma_{\mathcal{B}, \mathcal{I}}$ este întocmai ordinea parțială pe $\prod_{x \in \mathcal{V}} D_x$.

Ca urmare, mpo indușă de $\Gamma_{\mathcal{B}, \mathcal{I}}$ este exact produsul direct al familiei de mpo complete $((D_x, \leq_x) | x \in \mathcal{V})$ și, deci, este mpo completă. \square

Theoremă 7.3.1.1. (Teorema de continuitate a funcției semanticice a λ -termilor)

Fie \mathcal{B} o bază și \mathcal{I} o interpretare a ei. Atunci, pentru orice tip τ și λ -term t de tip τ are loc:

- (1) $\mathcal{I}(t)(\gamma) \in D_\tau$, pentru orice $\gamma \in \Gamma$;
- (2) $\mathcal{I}(t) : \Gamma \rightarrow D_\tau$ este funcție continuă.

Demonstrație. Vom face demonstrația teoremei prin inducție 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 lanț nevid $L \subseteq \Gamma$ și observăm că $\mathcal{I}(t)(L) = L(x)$ este lanț nevid în D_τ . Cum (D_τ, \leq_τ) este mpo completă, există $\text{sup}(L(x))$ și, conform Lemei 7.1.3.2, are loc

$$\text{sup}(L(x)) = (\text{sup}(L))(x).$$

Deci, există $\text{sup}(\mathcal{I}(t)(L))$ și

$$\text{sup}(\mathcal{I}(t)(L)) = \text{sup}(L(x)) = (\text{sup}(L))(x) = \mathcal{I}(t)(\text{sup}(L)),$$

ceea ce ne arată că $\mathcal{I}(t)$ este funcție 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 funcției $\mathcal{I}(t)$ urmează imediat de la faptul că aceasta este funcție constantă.

Cazul 3: $t = u(t_1, \dots, t_n)$, u este de tip $(\tau_1, \dots, \tau_n \rightarrow \tau)$ și t_i este de tip τ_i pentru orice $1 \leq i \leq n$. Presupunem că λ -termii u, t_1, \dots, t_n satisfac (1) și (2) din teorema. Atunci, pentru orice atribuire γ ,

$$\mathcal{I}(t)(\gamma) = \mathcal{I}(u)(\gamma)(\mathcal{I}(t_1)(\gamma), \dots, \mathcal{I}(t_n)(\gamma)),$$

care este un element din D_τ conform definiției funcției de interpretare. De asemenea, putem scrie

$$\mathcal{I}(t) = \psi \circ (\mathcal{I}(u), (\mathcal{I}(t_1), \dots, \mathcal{I}(t_n))),$$

ceea ce ne arată că $\mathcal{I}(t)$ este funcție continuă fiind compunere de funcții continue (conform ipotezei, Corolarului 7.2.1.2 și Corolarului 7.2.1.5).

Cazul 4: $t = [\lambda x_1, \dots, x_n. u]$, u este de tip σ , x_i este de tip τ_i , pentru orice $1 \leq i \leq n$, și $\tau = (\tau_1, \dots, \tau_n \rightarrow \sigma)$. Presupunem că λ -termul u satisfac (1) și (2) din teorema. Fie γ o atribuire. Pentru a arăta că are loc $\mathcal{I}(t)(\gamma) \in D_\sigma$ avem de arătat că $\mathcal{I}(t)(\gamma)$ este funcție continuă de la $D_{\tau_1} \times \dots \times D_{\tau_n}$ la D_σ . Ca urmare, avem de arătat că $\mathcal{I}(t)$ și $\mathcal{I}(t)(\gamma)$ sunt funcții continue. Forma acestor funcții sugerează considerarea unei funcții continue g ale cărei funcții Curry asociate să fie exact aceste funcții. Ca urmare, considerăm funcția

$$g : \Gamma \times (D_{\tau_1} \times \dots \times D_{\tau_n}) \rightarrow D_\sigma$$

dată prin

$$g(\gamma, (d_1, \dots, d_n)) = \mathcal{I}(u)(\gamma[x_1/d_1] \dots [x_n/d_n]),$$

pentru orice γ și $d_i \in D_{\tau_i}$, $1 \leq i \leq n$.

Funcția g este compunere de funcții continue (conform ipotezei și Lemei 7.3.1.1) deoarece ea poate fi scrisă în forma

$$g = \mathcal{I}(u) \circ \psi_{x_1, \dots, x_n}.$$

Deci g este continuă. Atunci funcțiile Curry asociate funcției g sunt continue. Adică funcțiile $\mathcal{I}(t)$ și $\mathcal{I}(t)(\gamma)$, pentru orice atribuire γ , sunt funcții continue. \square

Definiția 7.3.1.1. Fie \mathcal{B} o bază și \mathcal{I} o interpretare a ei. Spunem că o variabilă x apare liber în λ -termul t dacă:

- $t = x \in \mathcal{V}$ sau
- $t = u(t_1, \dots, t_n)$ și x apare liber în unul din termii u, t_1, \dots, t_n sau
- $t = [\lambda x_1, \dots, x_n. u]$, x este diferită de x_1, \dots, x_n și apare liber în u .

Dacă x nu apare liber în t , atunci spunem că x este mărginită în t .

Observația 7.3.1.1. Atragem explicit atenția asupra faptului că într-un λ -term de forma $t = u(t_1, \dots, t_n)$, o variabilă x poate să apară liber într-un λ -term t_i și mărginită în u . În 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 $D_{\text{nat}} = \mathbf{N}_\perp$, $\mathcal{I}_0(3), \mathcal{I}_0(2) \in \mathbf{N}$, x de tip nat și $\mathcal{I}_0(f) : \mathbf{N}_\perp^2 \rightarrow \mathbf{N}_\perp$, 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 .

În 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), obținem $\mathcal{I}(t)(\gamma) = 3 + (2 + \gamma(x))$.

Teorema 7.3.1.2. (Teorema de coincidență)

Fie \mathcal{B} o bază, \mathcal{I} o interpretare a ei și t un λ -term. Atunci, pentru orice două atribuiri γ și γ' ce satisfac $\gamma(x) = \gamma'(x)$ pentru orice variabilă x ce apare liber în t , are loc $\mathcal{I}(t)(\gamma) = \mathcal{I}(t)(\gamma')$.

Demonstrație. Vom face demonstrația teoremei prin inducție structurală asupra λ -termului t .

Cazul 1: $t = x \in \mathcal{V}$. Atunci x este liberă în t și, deci, pentru orice două atribuiri γ și 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 și γ' , are loc

$$\mathcal{I}(t)(\gamma) = \mathcal{I}_0(t) = \mathcal{I}(t)(\gamma').$$

Cazul 3: $t = u(t_1, \dots, t_n)$, u este de tip $(\tau_1, \dots, \tau_n \rightarrow \tau)$ și t_i este de tip τ_i pentru orice $1 \leq i \leq n$. Presupunem că λ -termii u, t_1, \dots, t_n satisfac teorema. Fie γ și γ' două atribuiri ce coincid pe variabilele libere din t . Atunci, ele vor coincide și pe variabilele libere din u, t_1, \dots, t_n , ceea ce conduce la:

$$\begin{aligned} \mathcal{I}(t)(\gamma) &= \mathcal{I}(u)(\gamma)(\mathcal{I}(t_1)(\gamma), \dots, \mathcal{I}(t_n)(\gamma)) \\ &= \mathcal{I}(u)(\gamma')(\mathcal{I}(t_1)(\gamma'), \dots, \mathcal{I}(t_n)(\gamma')) \\ &= \mathcal{I}(t)(\gamma'). \end{aligned}$$

Cazul 4: $t = [\lambda x_1, \dots, x_n. u]$, u este de tip σ , x_i este variabilă de tip τ_i , pentru orice $1 \leq i \leq n$ și $\tau = (\tau_1, \dots, \tau_n \rightarrow \sigma)$. Presupunem că λ -termul u satisfac teorema, și fie γ și γ' două atribuiri ce coincid pe variabilele libere din t . Atunci, $\gamma[x_1/d_1] \cdots [x_n/d_n]$ și $\gamma'[x_1/d_1] \cdots [x_n/d_n]$ vor coincide și pe variabilele libere din u , ceea ce conduce la:

$$\begin{aligned} \mathcal{I}(t)(\gamma)(d_1, \dots, d_n) &= \mathcal{I}(u)(\gamma[x_1/d_1] \cdots [x_n/d_n]) \\ &= \mathcal{I}(u)(\gamma'[x_1/d_1] \cdots [x_n/d_n]) \\ &= \mathcal{I}(t)(\gamma')(d_1, \dots, d_n), \end{aligned}$$

pentru orice $(d_1, \dots, d_n) \in D_{\tau_1} \times \cdots \times D_{\tau_n}$. \square

Corolarul 7.3.1.1. Fie \mathcal{B} o bază, \mathcal{I} o interpretare a ei și t un λ -term. Dacă t nu are variabile libere, atunci $\mathcal{I}(t)(\gamma) = \mathcal{I}(t)(\gamma')$, pentru orice două atribuiri γ și γ' .

Demonstrație. Direct de la Teorema 7.3.1.2 \square

7.3.1.2. Programe recursive

Conceptul de program recursiv a fost introdus de McCarthy în 1963 [139]. Abordarea noastră va urma [132, 129].

Un program recursiv este o mulțime de *ecuații recursive*, fiecare astfel de ecuație fiind alcătuită din 2 termi, unul definind *antetul ecuației*, iar celălalt, *corpul acesteia*. Așa cum vom vedea, termii ce intră în componența ecuațiilor recursive sunt λ -termi peste o bază potrivit aleasă.

Vom considera o mulțime $\mathcal{T}_0 = \{b, d\}$ 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 $\beta_1, \dots, \beta_s, \beta$ sunt tipuri de bază, atunci $(\beta_1, \dots, \beta_s \rightarrow \beta)$ este tip.

O *bază* pentru construcția de programe recursive este un triplet $\mathcal{B} = (\mathcal{T}_0, \mathcal{V}, \mathcal{F})$, unde:

- \mathcal{T}_0 este o mulțime de tipuri de bază (ca mai sus);
- \mathcal{V} este o mulțime de *variabile*. Variabilele sunt împărțite în două clase:
 - variabile de tip d , notate prin x, y, z, \dots (eventual indexate);
 - variabile de tip $(\underbrace{d, \dots, d}_{n \geq 1} \rightarrow d)$. Acestea vor fi numite *variabile funcționale* și vor fi notate prin F, G, H, \dots (eventual indexate);
- \mathcal{F} este o mulțime de *simboluri funcționale*. Aceste simboluri vor fi notate prin f, g, h, \dots (eventual indexate) și vom presupune că printre ele se găsesc și următoarele simboluri funcționale:

simbol funcțional	tip
<i>false</i>	b
<i>true</i>	b
\neg	$(b \rightarrow b)$
$=$	$(d, d \rightarrow b)$
$\vee, \wedge, \rightarrow, \leftrightarrow$	$(b, b \rightarrow b)$
<i>if_then_else</i>	$(b, d, d \rightarrow d)$

Fie \mathcal{B} o bază pentru construcția de programe recursive. Termii utilizati în definirea programelor recursive sunt definiți inductiv astfel:

- orice variabilă de tip d este term de tip d ;
- orice simbol funcțional f de tip b sau d este term de tip b sau, respectiv, d ;
- dacă f este simbol funcțional de tip $(\beta_1, \dots, \beta_s \rightarrow \beta)$, iar t_i sunt termi de tip β_i , $1 \leq i \leq s$, atunci $f(t_1, \dots, t_n)$ este term de tip β ;
- dacă F este variabilă funcțională de tip $(\underbrace{d, \dots, d}_{s \geq 1} \rightarrow d)$, iar t_i sunt termi de tip d , $1 \leq i \leq s$, atunci $F(t_1, \dots, t_n)$ este term de tip d .

O ecuație sau procedură recursivă este o pereche de termi

$$(F(x_1, \dots, x_s), t),$$

unde x_1, \dots, x_s sunt variabile distincte, iar t este un term de tip d ce poate conține orice variabilă funcțională însă, ca variabile de tip d , el poate conține doar x_1, \dots, x_s . Termul $F(x_1, \dots, x_s)$ se numește *antetul ecuației*, iar termul t , *corpul* acesteia. Uzual, ecuația $(F(x_1, \dots, x_s), t)$ se mai notează prin

$$F(x_1, \dots, x_s) \Leftarrow t$$

(\Leftarrow fiind un simbol nou).

Un program recursiv este un cuplu (S, k) , unde:

- S este o mulțime de ecuații recursive

$$S = \{(F_1(x_{11}, \dots, x_{1s_1}), t_1), \dots, (F_n(x_{n1}, \dots, x_{ns_n}), t_n)\}.$$

astfel încât pentru orice $1 \leq i \leq n$, t_i poate conține ca variabile funcționale doar F_1, \dots, F_n ;

- $1 \leq k \leq n$.

Uzual, sistemul (S, k) se notează prin

$$(S, k) \quad \left\{ \begin{array}{l} F_1(x_{11}, \dots, x_{1s_1}) \Leftarrow t_1 \\ \dots \\ F_n(x_{n1}, \dots, x_{ns_n}) \Leftarrow t_n \end{array} \right.$$

Variabila funcțională F_k se numește *variabila funcțională principală*. Atunci când programul recursiv este format doar dintr-o singură ecuație recursivă îl vom nota mai simplu prin

$$(S) \quad F(x_1, \dots, x_s) \Leftarrow t$$

Așa 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 menționate mai sus). O astfel de bază va fi numită *bază pentru programe recursive*.

Exemplul 7.3.1.2. Următoarele construcții sunt programe recursive:

$$(1) \quad (S) \quad F(x) \Leftarrow \text{if } x = 0 \text{ then } 1 \text{ else } x * F(x - 1)$$

(acest program, interpretat peste \mathbf{N}_\perp , aşa cum vom vedea în secțiunea următoare, calculează funcția factorial).

$$(2) \quad (S, 1) \quad \left\{ \begin{array}{l} F_1(x) \Leftarrow \text{if } x = 0 \text{ then } 0 \text{ else } F_2(x - 1) \\ F_2(x) \Leftarrow \text{if } x = 0 \text{ then } 1 \text{ else } F_1(x - 1) \end{array} \right.$$

(acest program calculează restul împărțirii unui număr natural la 2, atunci când este interpretat peste \mathbf{N}_\perp , aşa cum se va arăta în secțiunea următoare).

7.3.1.3. Semantica denotațională a programelor recursive

Fie \mathcal{B} o bază pentru programe recursive și

$$(S, k) \quad \left\{ \begin{array}{l} F_1(x_{11}, \dots, x_{1s_1}) \Leftarrow t_1 \\ \dots \\ F_n(x_{n1}, \dots, x_{ns_n}) \Leftarrow t_n \end{array} \right.$$

un program recursiv. Asociem fiecărei ecuații recursive

$$F_i(x_{i1}, \dots, x_{is_i}) \Leftarrow t_i$$

un λ -term

$$T_i = [\lambda F_1, \dots, F_n. [\lambda x_{i1}, \dots, x_{is_i}. t_i]]$$

al cărui tip este

$$\tau_i = ((\underbrace{d, \dots, d}_{s_1} \rightarrow d), \dots, (\underbrace{d, \dots, d}_{s_n} \rightarrow d) \rightarrow (\underbrace{d, \dots, d}_{s_i} \rightarrow d)),$$

unde $1 \leq i \leq n$.

Fie \mathcal{I} o interpretare a bazei \mathcal{B} . Vom presupune următoarele:

- $D_b = \text{Bool}_\perp$, unde $\text{Bool} = \{0, 1\}$;
- $D_d = D_\perp$, unde D este un domeniu nevid arbitrar dar fixat;
- $\mathcal{I}_0(\text{false}) = 0$ și $\mathcal{I}_0(\text{true}) = 1$;
- $\mathcal{I}_0(f)$ este extensia naturală a simbolului funcțional f , pentru orice

$$f \in \{\neg, =, \vee, \wedge, \rightarrow, \leftrightarrow\}$$

(a se vedea Secțiunea 1.4.2.4). În cazul $f = \text{if_then_else}$, $\mathcal{I}_0(f)$ este extensia din Exemplul 1.4.2.2(3).

Sub această interpretare, domeniul tipului τ_i va fi

$$D_{\tau_i} = [[D_\perp^{s_1} \rightarrow D_\perp] \times \dots \times [D_\perp^{s_n} \rightarrow D_\perp] \rightarrow [D_\perp^{s_i} \rightarrow D_\perp]].$$

Vom nota prin D^* domeniul

$$D^* = [D_\perp^{s_1} \rightarrow D_\perp] \times \dots \times [D_\perp^{s_n} \rightarrow D_\perp].$$

Definiția 7.3.1.2. Fie (S, k) un program recursiv și \mathcal{I} o interpretare ca mai sus. Funcția semantică a programului (S, k) sub interpretarea \mathcal{I} este funcția

$$\phi_{\mathcal{I}}(S, k) : D^* \rightarrow D^*$$

dată prin

$$\phi_{\mathcal{I}}(S, k) = (\mathcal{I}(T_1)(\gamma), \dots, \mathcal{I}(T_n)(\gamma)),$$

unde γ este o atribuire arbitrară.

Observația 7.3.1.2.

- (1) Definiția funcției semantice a unui program recursiv este consistentă, în sensul că ea nu depinde de asignarea aleasă, deoarece λ -termii T_1, \dots, T_n nu au variabile libere (a se vedea Corolarul 7.3.1.1).
- (2) Funcția semantică $\phi_{\mathcal{I}}(S, k)$ este continuă deoarece $pr_i \circ \phi_{\mathcal{I}}(S, k) = \mathcal{I}(T_i)(\gamma)$ este funcție continuă, pentru orice $1 \leq i \leq n$ (Corolarul 7.2.1.2).

Definiția 7.3.1.3. Fie (S, k) un program recursiv și \mathcal{I} o interpretare ca mai sus. Semantica denotațională a programului (S, k) este funcția parțială

$$\mathcal{M}_{\mathcal{I}}(S, k) : D_d^{s_k} \rightsquigarrow D_d$$

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 $a \in D_{\perp}^{s_k}$.

Notățiile $\phi_{\mathcal{I}}(S, k)$ și $\mathcal{M}_{\mathcal{I}}(S, k)$ vor fi simplificate la $\phi_{\mathcal{I}}(S)$, respectiv, $\mathcal{M}_{\mathcal{I}}(S)$ atunci când programul are doar o ecuație. În plus, vom scrie $\phi_{\mathcal{I}}(S) = \mathcal{I}(T)(\gamma)$ în loc de $\phi_{\mathcal{I}}(S) = (\mathcal{I}(T)(\gamma))$, unde T este λ -termul asociat ecuației programului iar γ este o asignare arbitrară.

Vom încheia secțiunea printr-un exemplu de calcul al semanticii denotaționale a unui program recursiv.

Exemplul 7.3.1.3. Fie programul recursiv

$$(S, 1) \quad \begin{cases} F_1(x) \Leftarrow \text{if } x = 0 \text{ then } 0 \text{ else } F_2(x - 1) \\ F_2(x) \Leftarrow \text{if } x = 0 \text{ then } 1 \text{ else } F_1(x - 1) \end{cases}$$

din Exemplul 7.3.1.2(2) interpretat peste \mathbf{N}_{\perp} (adică, $D_{\perp} = \mathbf{N}_{\perp}$).

λ -termii asociati sunt

$$T_1 = [\lambda F_1, F_2. [\lambda x. \text{if } x = 0 \text{ then } 0 \text{ else } F_2(x - 1)]]$$

și

$$T_2 = [\lambda F_1, F_2. [\lambda x. \text{if } x = 0 \text{ then } 1 \text{ else } F_1(x - 1)]].$$

Acești λ -termi au tipurile

$$\tau_1 = \tau_2 = ((\text{nat} \rightarrow \text{nat}), (\text{nat} \rightarrow \text{nat}) \rightarrow (\text{nat} \rightarrow \text{nat})),$$

iar domeniile corespunzătoare sunt

$$D_{\tau_1} = D_{\tau_2} = [[\mathbf{N}_{\perp} \rightarrow \mathbf{N}_{\perp}] \times [\mathbf{N}_{\perp} \rightarrow \mathbf{N}_{\perp}] \rightarrow [\mathbf{N}_{\perp} \rightarrow \mathbf{N}_{\perp}]].$$

Funcția semantică a programului este

$$\phi_{\mathcal{I}}(S, 1) = (\mathcal{I}(T_1)(\gamma), \mathcal{I}(T_2)(\gamma)),$$

unde γ este o atribuire arbitrară, dar fixată.

Acum va trebui să calculăm cel mai mic punct fix al funcției semantică a programului, ceea ce se reduce la calculul supremului lanțului

$$L = \{\phi_{\mathcal{I}}(S, 1)^n(\perp_{(\mathbf{N}_{\perp} \rightarrow \mathbf{N}_{\perp})}, \perp_{(\mathbf{N}_{\perp} \rightarrow \mathbf{N}_{\perp})}) | n \geq 0\},$$

unde $\perp_{(\mathbf{N}_{\perp} \rightarrow \mathbf{N}_{\perp})}$ este cel mai mic element al mpo complete $[\mathbf{N}_{\perp} \rightarrow \mathbf{N}_{\perp}]$, adică funcția ce returnează \perp pentru orice $x \in \mathbf{N}_{\perp}$.

Pentru a putea lucra ușor cu aceste funcții vom face următoarele notații:

- f_0 va desemna funcția $\perp_{(\mathbf{N}_{\perp} \rightarrow \mathbf{N}_{\perp})}$;
- $\phi_{\mathcal{I}}(S, 1)^n(f_0, g_0) = (f_n, g_n)$, pentru orice $n \geq 0$, unde $g_0 = f_0$. În plus, $f_n = \mathcal{I}(T_1)(\gamma)(f_{n-1}, g_{n-1})$ și $g_n = \mathcal{I}(T_2)(\gamma)(f_{n-1}, g_{n-1})$, pentru orice $n \geq 1$.

Pentru determinarea perechilor (f_n, g_n) vom face câteva iterări până când vom putea intui forma acestora. Are loc:

$$\begin{aligned} f_1(k) &= \mathcal{I}(T_1)(\gamma)(f_0, g_0)(k) \\ &= \mathcal{I}(T_1)(\gamma[F_1/f_0][F_2/g_0][x/k]) \\ &= \text{if } k = 0 \text{ then } 0 \text{ else } g_0(k - 1) \\ &= \text{if } k = 0 \text{ then } 0 \text{ else } \perp \end{aligned}$$

$$\begin{aligned} g_1(k) &= \mathcal{I}(T_2)(\gamma)(f_0, g_0)(k) \\ &= \mathcal{I}(T_2)(\gamma[F_1/f_0][F_2/g_0][x/k]) \\ &= \text{if } k = 0 \text{ then } 1 \text{ else } f_0(k - 1) \\ &= \text{if } k = 0 \text{ then } 1 \text{ else } \perp \end{aligned}$$

$$\begin{aligned} f_2(k) &= \mathcal{I}(T_1)(\gamma)(f_1, g_1)(k) \\ &= \mathcal{I}(T_1)(\gamma[F_1/f_1][F_2/g_1][x/k]) \\ &= \text{if } k = 0 \text{ then } 0 \text{ else } g_1(k - 1) \\ &= \text{if } k = 0 \text{ then } 0 \text{ else if } k - 1 = 0 \text{ then } 1 \text{ else } \perp \\ &= \text{if } k = 0 \text{ then } 0 \text{ else if } k = 1 \text{ then } 1 \text{ else } \perp \end{aligned}$$

$$\begin{aligned} g_2(k) &= \mathcal{I}(T_2)(\gamma)(f_1, g_1)(k) \\ &= \mathcal{I}(T_2)(\gamma[F_1/f_1][F_2/g_1][x/k]) \\ &= \text{if } k = 0 \text{ then } 1 \text{ else } f_1(k - 1) \\ &= \text{if } k = 0 \text{ then } 1 \text{ else if } k - 1 = 0 \text{ then } 0 \text{ else } \perp \\ &= \text{if } k = 0 \text{ then } 1 \text{ else if } k = 1 \text{ then } 0 \text{ else } \perp \end{aligned}$$

$$\begin{aligned}
f_3(k) &= \mathcal{I}(T_1)(\gamma)(f_2, g_2)(k) \\
&= \mathcal{I}(T_1)(\gamma[F_1/f_2][F_2/g_2][x/k]) \\
&= \text{if } k = 0 \text{ then } 0 \text{ else } g_2(k - 1) \\
&= \text{if } k = 0 \text{ then } 0 \text{ else if } k - 1 = 0 \text{ then } 1 \text{ else} \\
&\quad \text{if } k - 1 = 1 \text{ then } 0 \text{ else } \perp \\
&= \text{if } k = 0 \text{ then } 0 \text{ else if } k = 1 \text{ then } 1 \text{ else} \\
&\quad \text{if } k = 2 \text{ then } 0 \text{ else } \perp \\
\\
g_3(k) &= \mathcal{I}(T_2)(\gamma)(f_2, g_2)(k) \\
&= \mathcal{I}(T_2)(\gamma[F_1/f_2][F_2/g_2][x/k]) \\
&= \text{if } k = 0 \text{ then } 1 \text{ else } f_2(k - 1) \\
&= \text{if } k = 0 \text{ then } 1 \text{ else if } k - 1 = 0 \text{ then } 0 \text{ else} \\
&\quad \text{if } k - 1 = 1 \text{ then } 1 \text{ else } \perp \\
&= \text{if } k = 0 \text{ then } 1 \text{ else if } k = 1 \text{ then } 0 \text{ else} \\
&\quad \text{if } k = 2 \text{ then } 1 \text{ else } \perp,
\end{aligned}$$

pentru orice $k \in \mathbf{N}_\perp$. În acest moment putem presupune că are loc

$$f_n(k) = \begin{cases} 0, & \text{dacă } k < n \text{ este par} \\ 1, & \text{dacă } k < n \text{ este impar} \\ \perp, & \text{altfel} \end{cases}$$

și

$$g_n(k) = \begin{cases} 1, & \text{dacă } k < n \text{ este par} \\ 0, & \text{dacă } k < n \text{ este impar} \\ \perp, & \text{altfel,} \end{cases}$$

pentru orice $n \geq 1$ și $k \in \mathbf{N}_\perp$. Presupunerea noastră se dovedește a fi corectă, ceea ce poate fi demonstrat cu ușurință prin inducție matematică.

Acum supremumul lanțului L se obține imediat ca fiind

$$\sup(L) = \sup(\{(f_n, g_n) | n \geq 0\}) = (f^*, g^*),$$

unde

$$f^*(k) = \begin{cases} 0, & \text{dacă } k \in \mathbf{N} \text{ este par} \\ 1, & \text{dacă } k \in \mathbf{N} \text{ este impar} \\ \perp, & \text{dacă } k = \perp \end{cases}$$

și

$$g^*(k) = \begin{cases} 1, & \text{dacă } k \in \mathbf{N} \text{ este par} \\ 0, & \text{dacă } k \in \mathbf{N} \text{ este impar} \\ \perp, & \text{dacă } k = \perp, \end{cases}$$

pentru orice $k \in \mathbf{N}_\perp$. Ca urmare, semantica denotațională 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 $k \in \mathbf{N}_\perp$.

7.3.1.4. Programme structurate

O altă clasă importantă de programe, de natură imperativă, este cea a *programelor structurate*⁷. Diferența majoră dintre acestea și programele recursive constă în aceea că programele structurate folosesc o structură specială pentru iterații, numită `while...do`.

O bază pentru construcția programelor structurate este un triplet $\mathcal{B} = (\mathcal{V}, \mathcal{F}, \mathcal{P})$ format din 3 multimi disjuncte între ele, unde:

- \mathcal{V} este o mulțime de *variabile*;
- \mathcal{F} este o mulțime de *simboluri funcționale*, fiecare având asociat o aritate. Simbolurile funcționale de aritate 0 sunt numite și *constante funcționale*;
- \mathcal{P} este o mulțime de *simboluri predicative*, fiecare având asociat o aritate. Simbolurile predicative de aritate 0 sunt numite și *constante propoziționale*.

Termii peste o bază \mathcal{B} se definesc inductiv prin:

- orice variabilă sau constantă (simbol funcțional de aritate 0) este term;
- dacă t_1, \dots, t_n sunt termi și f este simbol funcțional de aritate n , unde $n \geq 1$, atunci $f(t_1, \dots, t_n)$ 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 ” și “ \Leftrightarrow ”, și *simbolurile auxiliare* “(”, “)”, “,” și “.”. Toate aceste simboluri sunt presupuse distințe între ele și distințe de elementele mulțimilor bazei \mathcal{B} . Atunci expresiile logice se definesc inductiv prin:

- simbolurile *true* și *false* sunt expresii logice;
- constantele propoziționale sunt expresii logice;
- dacă t_1 și t_2 sunt termi, atunci $(t_1 = t_2)$ este expresie logică;
- dacă t_1, \dots, t_n sunt termi și P este simbol predicativ n -ar, unde $n \geq 1$, atunci $P(t_1, \dots, t_n)$ este expresie logică;

⁷Programele structurate sunt frecvent întâlnite în literatura de specialitate și sub denumirea de programe *while*.

- dacă e_1 și e_2 sunt expresii logice, atunci $(\neg e_1)$, $(e_1 \vee e_2)$, $(e_1 \wedge e_2)$, $(e_1 \Rightarrow e_2)$ și $(e_1 \Leftrightarrow e_2)$ sunt expresii logice.

Acum, programele structurate peste o bază \mathcal{B} se definesc inductiv, utilizând simbolurile auxiliare “ $::=$ ”, “ $:$ ”, “if”, “then”, “else”, “while”, și “do”, astfel:

- dacă $x \in \mathcal{V}$ și t este term peste \mathcal{B} , atunci $x := t$ este program structurat peste \mathcal{B} ;
- dacă S_1 și S_2 sunt programe structurate peste \mathcal{B} și e este o expresie logică peste \mathcal{B} , atunci $S_1; S_2$, if e then S_1 else S_2 și while e do S_1 sunt programe structurate peste \mathcal{B} .

Simbolurile “if”, “then” și “else”, luate împreună, nu trebuie confundate cu simbolul funcțional if_then_else utilizat în cadrul programelor recursive. Structura indusă de aceste simboluri va fi interpretată cumva similar structurii if_then_else de la programe recursive.

Atunci când S_2 este obținut 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 acțiune 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 convenția adoptată eliminăm această situație ambiguă. O altă metodă de eliminare a acestei ambiguități se poate face prin considerarea unui nou simbol, fi, și utilizarea structurii

if e then S_1 else S_2 fi.

O discuție similară are loc pentru while e do S_1 .

Trebuie să remarcăm că programele structurate nu sunt liber inductiv definite deoarece structura $S_1; S_2; S_3$ are cel puțin două construcții inductive diferite (și astfel de cazuri sunt de fapt singurele posibile ce fac ca definiția programelor structurate să nu fie liber inductivă). O astfel de ambiguitate nu mai poate fi eliminată chiar și de simplu cum am făcut mai sus. În general, construcțiile ce nu sunt liber inductiv definite pot crea probleme referitor la definiția (recursivă) a unei funcții semantice a acestora. În cazul programelor structurate vom arăta că, deși definiția acestora nu este liber inductivă, funcția semantică poate fi definită în mod consistent.

Exemplul 7.3.1.4. Următoarele construcții sunt programe structurate peste o bază potrivit aleasă:

1. while $x > 0$ do $x := x - 1$.
2. $y := 1$; while $\neg(x = 0)$ do $(y := y * x; x := x - 1)$. Acest program structurat calculează funcția factorial atunci când este interpretat peste numere naturale.
3. $z := 0$; while $y \leq x$ do $(z := z + 1; x := x - y)$.

7.3.1.5. Semantica denotațională a programelor structurate

Fie \mathcal{B} o bază pentru programe structurate. O interpretare pentru baza \mathcal{B} este o pereche $\mathcal{I} = (D, \mathcal{I}_0)$ formată dintr-un domeniu nevid D și o funcție de interpretare inițială \mathcal{I}_0 ce satisface:

- $\mathcal{I}_0(f)$ este funcție de la D^n la D , pentru orice $f \in \mathcal{F}$ de aritate $n \geq 0$;
- $\mathcal{I}_0(P)$ este funcție de la D^n la Bool , pentru orice $P \in \mathcal{P}$ de aritate $n \geq 0$, unde $\text{Bool} = \{0, 1\}$ este o mulțime ce conține două elemente distincte.

O atribuire sau asignare a bazei \mathcal{B} sub o interpretare \mathcal{I} este o funcție $\gamma : \mathcal{V} \rightarrow D$. În teoria programării imperative astfel de funcții se mai numesc și stări. O stare furnizează deci valorile variabilelor la un moment dat. Însă, trebuie remarcat că nu se cere ca stările să fie toate accesibile de la starea inițială a programului. Vom nota prin $\Gamma_{\mathcal{B}, \mathcal{I}}$ mulțimea tuturor atribuirilor bazei \mathcal{B} sub interpretarea \mathcal{I} . Notația va fi simplificată la Γ , atunci când \mathcal{B} și \mathcal{I} sunt clare din context. Este bine să avem continuu în vedere că γ reprezintă o stare, iar un program structurat nu face altceva decât să transforme o stare într-o altă stare.

Semantica termilor și expresiilor logice ce ajută la construcția programelor structurate se definește recursiv ca o funcție \mathcal{I} de la mulțimea termilor și expresiilor logice la mulțimea ($\Gamma \rightarrow D$). Astfel, pentru orice $\gamma \in \Gamma$, $\mathcal{I}(t)(\gamma)$ este dată prin:

- $\mathcal{I}(t)(\gamma) = \mathcal{I}_0(t)$, dacă $t \in \mathcal{F}$ este constantă;
- $\mathcal{I}(t)(\gamma) = \gamma(t)$, dacă $t \in \mathcal{V}$ este variabilă;
- $\mathcal{I}(f(t_1, \dots, t_n))(\gamma) = \mathcal{I}_0(f)(\mathcal{I}(t_1)(\gamma), \dots, \mathcal{I}(t_n)(\gamma))$;
- $\mathcal{I}(\text{true})(\gamma) = 1$ și $\mathcal{I}(\text{false})(\gamma) = 0$;
- $\mathcal{I}(p)(\gamma) = \mathcal{I}_0(p)$, pentru orice constantă propozițională 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);
- $\mathcal{I}(P(t_1, \dots, t_n))(\gamma) = \mathcal{I}_0(P)(\mathcal{I}(t_1)(\gamma), \dots, \mathcal{I}(t_n)(\gamma))$;
- $\mathcal{I}(\neg e)(\gamma) = \begin{cases} 1, & \text{dacă } \mathcal{I}(e)(\gamma) = 0 \\ 0, & \text{altfel} \end{cases}$
(egalitatea $\mathcal{I}(e)(\gamma) = 0$ este identitate de elemente în Bool).

În mod similar se definește \mathcal{I} pentru celelalte expresii logice.

Cu aceste elemente pregătitoare putem introduce semantica denotațională a programelor structurate. Înainte de aceasta trebuie să remarcăm că mulțimea atribuirilor (stărilor) nu este mpo completă deoarece nu avem definită nici o relație de ordine

parțială 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 Γ_{\perp} , iar altul este de a transforma D într-o mpo plată D_{\perp} (faptul că D_{\perp} este completă asigură că mulțimea tuturor funcțiilor de la Γ la D_{\perp} este completă, în raport cu ordinea parțială pe funcții indușă de ordinea parțială 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, și anume \perp . Cazul $\gamma = \perp$ va trebui tratat separat deoarece termii și expresiile logice sunt interpretate ca funcții definite pe Γ și nu pe Γ_{\perp} .

Definiția 7.3.1.4. Fie S un program structurat peste o bază \mathcal{B} și \mathcal{I} o interpretare a bazei \mathcal{B} . Funcția semantică a programului S sub interpretarea \mathcal{I} este funcția

$$\phi_{\mathcal{I}}(S) : \Gamma_{\perp} \rightarrow \Gamma_{\perp}$$

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 $S_1; S_2$;
- $\phi_{\mathcal{I}}(S)(\gamma) = \begin{cases} \phi_{\mathcal{I}}(S_1)(\gamma), & \text{dacă } \mathcal{I}(e)(\gamma) = 1 \text{ și } \gamma \neq \perp \\ \phi_{\mathcal{I}}(S_2)(\gamma), & \text{dacă } \mathcal{I}(e)(\gamma) = 0 \text{ și } \gamma \neq \perp \\ \perp, & \text{dacă } \gamma = \perp, \end{cases}$
pentru orice $\gamma \in \Gamma_{\perp}$, dacă S este programul $\text{if } e \text{ then } S_1 \text{ else } S_2$;
- $\phi_{\mathcal{I}}(S) = \mu(F)$, dacă S este programul $\text{while } e \text{ do } S_1$, unde F este funcția

$$F : [\Gamma_{\perp} \rightarrow \Gamma_{\perp}] \rightarrow [\Gamma_{\perp} \rightarrow \Gamma_{\perp}]$$

dată prin

$$F(f)(\gamma) = \begin{cases} (f \circ \phi_{\mathcal{I}}(S_1))(\gamma), & \text{dacă } \mathcal{I}(e)(\gamma) = 1 \text{ și } \gamma \neq \perp \\ \gamma, & \text{dacă } \mathcal{I}(e)(\gamma) = 0 \text{ și } \gamma \neq \perp \\ \perp, & \text{dacă } \gamma = \perp, \end{cases}$$

pentru orice $f \in [\Gamma_{\perp} \rightarrow \Gamma_{\perp}]$ și $\gamma \in \Gamma_{\perp}$.

Definiția funcției semantică pentru structura $\text{while } e \text{ do } S$ se bazează pe observația că dacă scriem informal

$$\text{while } e \text{ do } S = \begin{cases} (\text{while } e \text{ do } S) \circ S, & \text{dacă } e \text{ este adevărată} \\ id, & \text{dacă } e \text{ este falsă} \end{cases}$$

și notăm prin F funcția din membrul drept al egalității, unde id este funcția identitate, atunci $\text{while } e \text{ do } S$ verifică proprietatea

$$F(\text{while } e \text{ do } S) = \text{while } e \text{ do } S.$$

Cu alte cuvinte, $\text{while } e \text{ do } S$ este punct fix al funcției F .

Deoarece definiția programelor structurate nu este liberă, va trebui să arătăm că, în adevăr, $\phi_{\mathcal{I}}(S)$ este funcție. Concomitent vom arăta că aceasta este și continuă.

Teorema 7.3.1.3. Pentru orice program structurat S , $\phi_{\mathcal{I}}(S)$ este funcție continuă.

Demonstrație. Vom face demonstrația prin inducție structurală asupra programului S .

Cazul 1: S este de forma $x := t$. Este clar că $\phi_{\mathcal{I}}(S)$ este funcție. Ea este și continuă deoarece este o extensie naturală (a se vedea Secțiunea 1.4.2.4).

Cazul 2: S este de forma $S_1; S_2$. Presupunem că $\phi_{\mathcal{I}}(S_1)$ și $\phi_{\mathcal{I}}(S_2)$ sunt funcții continue. Cum compunerea de funcții continue conduce la funcții continue, deducem că $\phi_{\mathcal{I}}(S)$ este funcție continuă.

Cazul 3: S este de forma $\text{if } e \text{ then } S_1 \text{ else } S_2$. Presupunem că $\phi_{\mathcal{I}}(S_1)$ și $\phi_{\mathcal{I}}(S_2)$ sunt funcții continue. Este clar că $\phi_{\mathcal{I}}(S)$ este funcție. Ea este și continuă deoarece poate fi considerată ca o compunere a funcțiilor $\phi_{\mathcal{I}}(S_1)$ și $\phi_{\mathcal{I}}(S_2)$ cu extensia naturală la Γ_{\perp} a funcției if_then_else definită pe Γ (Secțiunea 1.4.2.4).

Cazul 4: S este de forma $\text{while } e \text{ do } S_1$. Presupunem că $\phi_{\mathcal{I}}(S_1)$ este funcție continuă. Va fi suficient să arătăm că F este o funcție continuă. Atunci ea va avea un cel mai mic punct fix care va fi o funcție continuă, deoarece este element al mulțimii $[\Gamma_{\perp} \rightarrow \Gamma_{\perp}]$. Ca urmare, aceasta va demonstra atât faptul că $\phi_{\mathcal{I}}$ este bine-definită, cât și faptul că este funcție continuă.

Vom arăta că F este continuă în mod direct. Fie $L \subseteq [\Gamma_{\perp} \rightarrow \Gamma_{\perp}]$ un lanț nevid de funcții. Cum $[\Gamma_{\perp} \rightarrow \Gamma_{\perp}]$ este mpo completă, există $\text{sup}(L)$. Va trebui să arătăm că are loc $F(\text{sup}(L))(\gamma) = \text{sup}(F(L))(\gamma)$, pentru orice $\gamma \in \Gamma_{\perp}$. Vom lua în considerare 3 cazuri (și vom utiliza intens Lema 7.1.3.2):

- $\gamma = \perp$. Atunci $F(\text{sup}(L))(\gamma) = \perp$ (conform definiției funcției F) și $\text{sup}(F(L))(\gamma) = \text{sup}(F(L)(\gamma)) = \text{sup}(\{\perp\}) = \perp$, stabilind astfel egalitatea $F(\text{sup}(L))(\gamma) = \text{sup}(F(L))(\gamma)$;
- $\mathcal{I}(e)(\gamma) = 0$ și $\gamma \neq \perp$. Atunci $F(\text{sup}(L))(\gamma) = \gamma$ (conform definiției funcției F) și $\text{sup}(F(L))(\gamma) = \text{sup}(F(L)(\gamma)) = \text{sup}(\{\gamma\}) = \gamma$, stabilind astfel egalitatea $F(\text{sup}(L))(\gamma) = \text{sup}(F(L))(\gamma)$;

- $\mathcal{I}(e)(\gamma) = 1$ și $\gamma \neq \perp$. Atunci $F(\text{sup}(L))(\gamma) = \text{sup}(L)(\phi_{\mathcal{I}}(S_1)(\gamma))$ (conform definiției funcției F). Cum $\phi_{\mathcal{I}}(S_1)$ este funcție continuă, $F(L) = \{f \circ \phi_{\mathcal{I}}(S_1) | f \in L\}$ este lanț de funcții continue în $[\Gamma_{\perp} \rightarrow \Gamma_{\perp}]$, al cărui suprem este $\text{sup}(F(L)) = \text{sup}(L) \circ \phi_{\mathcal{I}}(S_1)$ (ceea ce este ușor de văzut). Atunci,

$$\begin{aligned} \text{sup}(F(L))(\gamma) &= (\text{sup}(L) \circ \phi_{\mathcal{I}}(S_1))(\gamma) = \text{sup}(L)(\phi_{\mathcal{I}}(S_1)(\gamma)), \\ \text{stabilind astfel egalitatea } F(\text{sup}(L))(\gamma) &= \text{sup}(F(L))(\gamma). \end{aligned}$$

Aceasta încheie demonstrația teoremei⁸. \square

Semantica denotațională a programelor structurate se obține ca și în cazul programelor recursive.

Definiția 7.3.1.5. Fie S un program structurat peste o bază \mathcal{B} și \mathcal{I} o interpretare a bazei \mathcal{B} . Semantica denotațională a programului S este funcția parțială

$$\mathcal{M}_{\mathcal{I}}(S) : \Gamma \rightsquigarrow \Gamma$$

dată prin

$$\mathcal{M}_{\mathcal{I}}(S)(\gamma) = \begin{cases} \phi_{\mathcal{I}}(S)(\gamma), & \text{dacă această valoare nu este } \perp \\ \text{nedefinită,} & \text{altfel,} \end{cases}$$

pentru orice $\gamma \in \Gamma$.

Încheiem secțiunea printr-un exemplu de calcul al semanticii denotaționale a unui program structurat.

Exemplul 7.3.1.5. Fie programul structurat S dat prin

$$y := 1; \text{while } \neg(x = 0) \text{ do } (y := y * x; x := x - 1)$$

și interpretat peste $D = \mathbb{N}$ cu interpretarea uzualea a operatorilor \neg , $*$ și $-$ (mai mult, vom nota $\mathcal{I}_0(*)$ tot prin $*$, $\mathcal{I}_0(\neg)$ tot prin \neg și $\mathcal{I}_0(n)$ tot prin n , pentru orice $n \geq 0$).

Fie $\gamma \in \Gamma_{\perp}$. Dacă $\gamma = \perp$, atunci $\phi_{\mathcal{I}}(S)(\gamma) = \perp$. Altfel

$$\begin{aligned} \phi_{\mathcal{I}}(S)(\gamma) &= \phi_{\mathcal{I}}(\text{while } \neg(x = 0) \text{ do } (y := y * x; x := x - 1))(\phi_{\mathcal{I}}(y := 1)(\gamma)) \\ &= \phi_{\mathcal{I}}(\text{while } \neg(x = 0) \text{ do } (y := y * x; x := x - 1))(\gamma[y/\mathcal{I}_0(1)]) \\ &= \phi_{\mathcal{I}}(\text{while } \neg(x = 0) \text{ do } (y := y * x; x := x - 1))(\gamma[y/1]) \\ &= \mu(F)(\gamma[y/1]), \end{aligned}$$

unde F este funcția dată prin:

- $F(f)(\gamma') = f(\phi_{\mathcal{I}}(y := y * x; x := x - 1)(\gamma'))$, dacă $\mathcal{I}(\neg(x = 0))(\gamma') = 1$ și $\gamma' \neq \perp$;

⁸Utilizând λ-notația se poate da o altă demonstrație faptului că F este continuă, arătând că F este interpretarea unui λ-term T ca cel de mai jos [129]:

$$T = [\lambda f. [\lambda \gamma. \text{if } \gamma = \gamma \text{ then if } E \text{ then } f(T_1) \text{ else } \gamma \text{ else } t]]$$

În cadrul acestui λ-term, t este un λ-term arbitrar, interpretarea lui T_1 trebuie să fie $\phi_{\mathcal{I}}(S_1)$, iar interpretarea expresiei E trebuie să fie extensia naturală a funcției $\mathcal{I}(e)$ (reamintim că $\mathcal{I}(e)$ este definită pe Γ și nu pe Γ_{\perp}). În plus, `if_then_else` din cadrul acestui λ-term T trebuie interpretat ca în Secțiunea 1.4.2.4.

Deși această soluție ar părea mai simplă, ea ridică multe probleme referitoare la readaptarea funcțiilor la domeniul Γ_{\perp} . Din punctul nostru de vedere, demonstrația deja adoptată este de preferat.

- $F(f)(\gamma') = \gamma'$, dacă $\mathcal{I}(\neg(x = 0))(\gamma') = 0$ și $\gamma' \neq \perp$;
- $F(f)(\gamma') = \perp$, dacă $\gamma' = \perp$,

pentru orice $f \in [\Gamma_{\perp} \rightarrow \Gamma_{\perp}]$ și $\gamma' \in \Gamma_{\perp}$.

Dacă explicităm și mai mult funcția F obținem că aceasta este dată prin:

- $F(f)(\gamma') = f(\gamma'[y/\gamma'(y) * \gamma'(x)][x/\gamma'(x) - 1])$, dacă $\gamma'(x) > 0$;
- $F(f)(\gamma') = \gamma'$, dacă $\gamma'(x) = 0$;
- $F(f)(\gamma') = \perp$, dacă $\gamma' = \perp$,

pentru orice $f \in [\Gamma_{\perp} \rightarrow \Gamma_{\perp}]$ și $\gamma' \in \Gamma_{\perp}$ (pentru simplificarea scrierii s-a renunțat la “ $\gamma' \neq \perp$ ” atunci când $\gamma(x)$ este comparat cu numere naturale, înțelegând că γ' nu poate fi \perp în aceste cazuri).

Fie $f_0 = \perp_{(\Gamma_{\perp} \rightarrow \Gamma_{\perp})}$ cel mai mic element al mpo complete $[\Gamma_{\perp} \rightarrow \Gamma_{\perp}]$, adică funcția dată prin $f_0(\gamma') = \perp$, pentru orice $\gamma' \in \Gamma_{\perp}$. Calculul celui mai mic punct fix al funcției F se reduce la calculul supremului lanțului $L = \{F^n(f_0) | n \geq 0\}$. Ca urmare, vom determina întâi elementele acestui lanț. Are loc:

$$\begin{aligned} F(f_0)(\gamma') &= \begin{cases} f_0(\gamma'[y/\gamma'(y) * \gamma'(x)][x/\gamma'(x) - 1]), & \text{dacă } \gamma'(x) > 0 \\ \gamma', & \text{dacă } \gamma'(x) = 0 \\ \perp, & \text{dacă } \gamma' = \perp \end{cases} \\ &= \begin{cases} \perp, & \text{dacă } \gamma' = \perp \text{ sau } \gamma'(x) > 0 \\ \gamma', & \text{dacă } \gamma'(x) = 0, \end{cases} \end{aligned}$$

pentru orice $\gamma' \in \Gamma_{\perp}$.

Pentru a ușura calculul funcției $F^2(f_0)$, notăm

$$\gamma'' = \gamma'[y/\gamma'(y) * \gamma'(x)][x/\gamma'(x) - 1].$$

Are loc:

$$\begin{aligned} F^2(f_0)(\gamma') &= F(F(f_0))(\gamma') \\ &= \begin{cases} F(f_0)(\gamma''[y/\gamma''(y) * \gamma''(x)][x/\gamma''(x) - 1]), & \text{dacă } \gamma''(x) > 0 \\ \gamma', & \text{dacă } \gamma''(x) = 0 \\ \perp, & \text{dacă } \gamma'' = \perp \end{cases} \\ &= \begin{cases} \perp, & \text{dacă } \gamma'' = \perp \text{ sau } \gamma''(x) > 0 \\ \gamma''[y/\gamma''(y) * \gamma''(x)][x/\gamma''(x) - 1], & \text{dacă } \gamma''(x) = 0 \\ \gamma', & \text{dacă } \gamma''(x) = 0 \end{cases} \\ &= \begin{cases} \perp, & \text{dacă } \gamma' = \perp \text{ sau } \gamma'(x) > 1 \\ \gamma'[y/\gamma'(y) * \gamma'(x)][x/\gamma'(x) - 1], & \text{dacă } \gamma'(x) = 1 \\ \gamma', & \text{dacă } \gamma'(x) = 0, \end{cases} \end{aligned}$$

pentru orice $\gamma' \in \Gamma_{\perp}$.

Să mai facem un pas și să calculăm $F^3(f_0)$:

$$\begin{aligned} F^3(f_0)(\gamma') &= F(F^2(f_0))(\gamma') \\ &= \begin{cases} F^2(f_0)(\gamma'[y/\gamma'(y) * \gamma'(x)][x/\gamma'(x) - 1]), & \text{dacă } \gamma'(x) > 0 \\ \gamma', & \text{dacă } \gamma'(x) = 0 \\ \perp, & \text{dacă } \gamma' = \perp \end{cases} \\ &= \begin{cases} \perp, & \text{dacă } \gamma' = \perp \text{ sau } \gamma''(x) > 1 \\ \gamma''[y/\gamma''(y) * \gamma''(x)][x/\gamma''(x) - 1], & \text{dacă } \gamma''(x) = 1 \\ \gamma'', & \text{dacă } \gamma''(x) = 0 \\ \gamma', & \text{dacă } \gamma'(x) = 0 \end{cases} \\ &= \begin{cases} \perp, & \text{dacă } \gamma' = \perp \text{ sau } \gamma'(x) > 2 \\ \gamma'[y/\gamma'(y) * \gamma'(x) * (\gamma'(x) - 1)][x/\gamma'(x) - 2], & \text{dacă } \gamma'(x) = 2 \\ \gamma'[y/\gamma'(y) * \gamma'(x)][x/\gamma'(x) - 1], & \text{dacă } \gamma'(x) = 1 \\ \gamma', & \text{dacă } \gamma'(x) = 0, \end{cases} \end{aligned}$$

pentru orice $\gamma' \in \Gamma_{\perp}$.

Prinț-o simplă inducție matematică obținem:

$$F^{n+1}(f_0)(\gamma') = \begin{cases} \perp, & \text{dacă } \gamma' = \perp \text{ sau } \gamma'(x) > n \\ \gamma'[y/\gamma'(y) * j * \dots * 2 * 1][x/0], & \text{dacă } 1 \leq \gamma'(x) = j \leq n \\ \gamma', & \text{dacă } \gamma'(x) = 0, \end{cases}$$

pentru orice $\gamma' \in \Gamma_{\perp}$ și $n \geq 1$.

Atunci poate fi ușor observat că are loc:

$$\mu(F)(\gamma') = \begin{cases} \perp, & \text{dacă } \gamma' = \perp \\ \gamma'[y/\gamma'(y) * n * \dots * 2 * 1][x/0], & \text{dacă } \gamma'(x) = n \geq 1, \\ \gamma', & \text{dacă } \gamma'(x) = 0, \end{cases}$$

pentru orice $\gamma' \in \Gamma_{\perp}$. Înlocuind γ' cu $\gamma[y/1]$ obținem

$$\mu(F)(\gamma[y/1]) = \begin{cases} \perp, & \text{dacă } \gamma = \perp \\ \gamma[y/1 * n * \dots * 2 * 1][x/0], & \text{dacă } \gamma(x) = n \geq 1, \\ \gamma[y/1], & \text{dacă } \gamma(x) = 0, \end{cases}$$

pentru orice $\gamma \in \Gamma_{\perp}$. Atunci, semantica programului nostru va fi:

$$\mathcal{M}_{\mathcal{I}}(S)(\gamma) = \begin{cases} \gamma[y/1 * n * \dots * 2 * 1][x/0], & \text{dacă } \gamma(x) = n \geq 1 \\ \gamma[y/1], & \text{dacă } \gamma(x) = 0, \end{cases}$$

pentru orice $\gamma \neq \perp$.

De exemplu, dacă alegem γ astfel încât $\gamma(x) = 0$, atunci

$$\mathcal{M}_{\mathcal{I}}(S)(\gamma)(y) = 1,$$

iar dacă alegem γ astfel încât $\gamma(x) = 5$, atunci

$$\mathcal{M}_{\mathcal{I}}(S)(\gamma)(y) = 1 * 5 * 4 * 3 * 2 * 1.$$

7.3.2. Analiza și verificarea programelor

Una din chestiunile fundamentale în teoria elaborării programelor o constituie studiul corectitudinii acestora. Informal, spunem că un program este corect dacă el face exact ceea ce indică specificațiile acestuia. Din punct de vedere formal, corectitudinea unui program se definește prin intermediul a două predicate P_i și P_o . Să presupunem că \mathcal{B} este o bază de construcție de programe (recursive sau structurate), \mathcal{I} este o interpretare a bazei \mathcal{B} și Γ este mulțimea atribuirilor (stărilor). Presupunem că P_i și P_o sunt predicate peste Γ . Atunci spunem că un program S peste baza \mathcal{B}

(1) este *parțial corect relativ la P_i și P_o* dacă are loc

$$(\forall \gamma \in \Gamma)(P_i(\gamma) \wedge \mathcal{M}_{\mathcal{I}}(S)(\gamma) \downarrow \Rightarrow P_o(\mathcal{M}_{\mathcal{I}}(S)(\gamma)));$$

(2) se termină relativ la P_i dacă are loc

$$(\forall \gamma \in \Gamma)(P_i(\gamma) \Rightarrow \mathcal{M}_{\mathcal{I}}(S)(\gamma) \downarrow);$$

(3) este *total corect relativ la P_i și P_o* dacă are loc

$$(\forall \gamma \in \Gamma)(P_i(\gamma) \Rightarrow \mathcal{M}_{\mathcal{I}}(S)(\gamma) \downarrow \wedge P_o(\mathcal{M}_{\mathcal{I}}(S)(\gamma))).$$

Așa cum se vede, corectitudinea totală implică atât terminare cât și corectitudine parțială. În general, toate probleme sunt *nedecidabile*, adică nu există un algoritm care, primind la intrare un program și predicatele P_i și/sau P_o , să se opreasă întotdeauna și să ne spună dacă programul respectiv este sau nu parțial corect (se termină, este sau nu total corect).

Două abordări clasice în studiul corectitudinii programelor sunt *testarea* și *verificarea*. Testarea programului înseamnă execuția acestuia pe o mulțime critică de date de intrare, înregistrarea rezultatelor și compararea acestora cu ceea ce ar fi trebuit să fie. Dacă în urma unor astfel de teste rezultatele obținute coincid cu ceea ce teoretic ar fi trebuit să se obțină, atunci “crește încrederea” în corectitudinea programului. Însă asta nu înseamnă că programul este corect.

Verificarea programelor, spre deosebire de testare, ne spune clar dacă programul este sau nu (parțial) corect. Ea se realizează luând în considerare semantica programului (operatională, denotațională, axiomatică sau orice alt tip de semantică definită formal pentru programul în cauză) și apelând la tehnici formale (matematice) pentru demonstrarea corectitudinii. Spre deosebire de testare, verificarea are un grad de automatizare foarte redus. Demonstrațiile de corectitudine se fac în principal manual. Anumiți pași pot fi automatizați dar, din punct de vedere practic, la un nivel scăzut.

O altă abordare, care înglobează o bună parte din aspectul corectitudinii, o constituie *analiza programelor*. Analiza programelor, spre deosebire de testare și verificare, conduce și la *optimizare de cod*.

Scopul acestei secțiuni este de a face o foarte simplă introducere în teoria analizei și verificării programelor.

7.3.2.1. Analiza programelor

Prin *analiza programelor* se înțelege procesul automat de analiză a comportării unui program, proces realizat prin intermediul unui alt program numit *program de analiză* care este presupus a fi corect din toate punctele de vedere. Comportarea unui program este în strânsă legătură cu spațiul stărilor programului (mulțimea tuturor stărilor acestuia). Programe foarte simple pot avea spații de stări foarte mari. De exemplu, un program ce utilizează doar 80 de variabile Booleene poate conduce la un spațiu de stări de cel puțin 2^{80} stări. Analiza unui astfel de spațiu prin parcurgerea acestuia stare cu stare este de nerealizat din punct de vedere practic⁹.

Aceasta ne spune că pentru analiza programelor trebuie aplicate tehnici care să evite parcurgerea tuturor stărilor programului. Evident, astfel de tehnici depind de proprietățile pe care dorim să le analizăm.

În principal, există două clase mari de tehnici de analiză: *analiză statică* și *analiză dinamică*. În cadrul analizei statice programele nu sunt executate efectiv, ci sunt analizate pentru a se obține diverse informații pe baza structurii sintactice a acestora. Programul de analiză poate simula o execuție parțială a programului dat, așa cum vom vedea în cadrul analizei statice pe care o vom descrie complet în această secțiune.

Există foarte multe tipuri de analiză statică (pentru detalii recomandăm [159]). Să considerăm câteva exemple ușoare care ne vor conduce apoi la tehnica de *analiză a dependențelor funcționale*. Presupunem că într-un program avem următorul cuplu de instrucțiuni:

$$x := 5; \quad y := x * x + 25.$$

Într-un astfel de caz observăm că y va fi 50, ceea ce face ca această structură să poată fi simplificată la

$$x := 5; \quad y := 50$$

obținându-se astfel un cod mai eficient. O analiză care se ocupă cu astfel de probleme poartă denumirea de *analiză a propagării constantelor*.

Să presupunem acum că într-un program avem o structură de forma

$$y := x * x + 25; \quad \text{while } y \leq 0 \text{ do } \dots$$

Evident, într-un astfel de caz nu este necesară testarea variabilei y deoarece aceasta va fi întotdeauna strict pozitivă. Ca urmare, secvența de instrucțiuni corespunzătoare variantei " $y \leq 0$ " poate fi eliminată. O analiză care se ocupă cu detectarea de astfel de probleme poartă denumirea de *analiză de semn*. O astfel de analiză este însotită de o tehnică de eliminare de cod conducând astfel la optimizarea acestuia.

⁹Numărul 2^{80} este foarte mare. Pentru a avea o idee asupra acestui număr să considerăm următorul calcul simplu. Un an are 31536000 secunde, ceea ce este mai puțin de 2^{29} . Dacă am avea un program care să poată verifica 10^9 stări pe secundă, ceea ce nu depășește 2^{36} , atunci programul nu ar putea verifica prin analiză exhaustivă (stare cu stare) mai mult de 2^{65} stări într-un an.

O analiză foarte importantă este cea prin care se stabilește dacă există legături, numite *dependențe funcționale*, între valorile finale ale variabilelor de ieșire și valorile inițiale ale variabilelor de intrare. Ne-existența unei legături între valoarea finală a unei variabile de ieșire și valorile inițiale ale variabilelor de intrare atrage atenția programatorului că anumite părți din program pot fi eronate. Dacă considerăm programul

$$y := 1; \quad \text{while } \neg(x = 0) \text{ do } (y := y * x; \quad x := x - 1)$$

în care x este variabilă de intrare, iar y este de ieșire, atunci este ușor de văzut că există o dependență funcțională între y și x . Este instructiv de remarcat că dacă eliminăm prima instrucțiune din program, atunci valoarea finală a variabilei y va depinde nu numai de valoarea inițială a variabilei x dar și de valoarea inițială a ei însăși.

Vom descrie în cele ce urmează tehnica de analiză a dependențelor funcționale pe programe structurate definite ca în Secțiunea 7.3.1.4. Vom presupune că orice program are fixată încă de la început o împărțire a variabilelor în *variabile de intrare*, *variabile de ieșire* și *variabile locale*. Anumite variabile de ieșire pot fi variabile de intrare. Vom utiliza două simboluri speciale, *OK* și *D?*, pentru a specifica următoarele:

- *OK* specifică faptul că o valoare (obținută prin evaluarea unei expresii sau ca fiind valoarea unei variabile la un moment dat) depinde clar de valorile inițiale ale variabilelor de intrare, și numai de acestea;
- *D?* specifică faptul că o valoare (obținută prin evaluarea unei expresii sau ca fiind valoarea unei variabile la un moment dat) poate depinde de valorile inițiale ale unor variabile care nu sunt de intrare.

Fie $\mathbf{P} = \{OK, D?\}$. Structurăm această mulțime ca o latice completă considerând $D?$ ca fiind cel mai mare element. Notăm prin \leq ordinea parțială (care de fapt este totală) pe această latice.

Ideea de bază este următoarea. Inițial (în starea inițială), toate variabilele de intrare au asociată valoarea *OK*, iar restul variabilelor *D?*. Pe măsură ce se execută o nouă instrucțiune, starea curentă a programului se modifică și, odată cu ea se modifică și proprietățile *OK* și *D?* asociate variabilelor. Dacă în starea finală a programului toate variabilele de ieșire vor avea valoarea *OK*, atunci între valorile lor finale și valorile inițiale ale variabilelor de intrare există o dependență funcțională. Altfel, pot exista variabile de ieșire a căror valoare finală să depindă de valoarea inițială a unor variabile care nu sunt de intrare.

Decorarea variabilelor cu unul dintre aceste două simboluri, *OK* sau *D?*, se face prin intermediul conceptului de *p-stare*. Înainte de a introduce acest concept, considerăm o nouă variabilă notată *on-track* care va da informații asupra structurii de control a programului.

O *p-stare* este o funcție $\psi : \mathcal{V} \cup \{\text{on-track}\} \rightarrow \mathbf{P}$. Vom nota prin Ψ mulțimea tuturor p-stărilor. Această mulțime, cu ordinea uzuală pe funcții, este împre completă deoarece \mathbf{P} este latice completă.

Starea care asociază OK fiecărei variabile va fi notată prin $INIT$, iar cea care asociază $D?$ fiecărei variabile va fi notată prin $LOST$. $INIT$ este cel mai mic element, iar $LOST$ este cel mai mare element, al mpo complete Ψ . Datează o p-stare ψ , prin $OK(\psi)$ vom nota mulțimea tuturor variabilelor care au asociat OK în starea ψ . Altfel spus, $OK(\psi) = \psi^{-1}(OK)$. O p-stare ψ este numită *proprie* dacă $\text{on-track} \in OK(\psi)$; altfel, ea este numită *improprie*.

Analiza se definește recursiv ca o funcție de interpretare, numită *p-interpretare*, după structura programului (în manieră similară semanticii denotaționale). Primul pas este de a defini (recursiv) *funcția de p-interpretare a termilor și expresiilor logice*, notată $p\mathcal{I}$, care este o funcție de la mulțimea termilor și expresiilor logice la mulțimea $(\Psi \rightarrow \mathbf{P})$. Astfel, dacă $\psi \in \Psi$, atunci $p\mathcal{I}(t)(\psi)$ este definită prin:

- $p\mathcal{I}(t)(\psi) = \begin{cases} OK, & \text{dacă } \psi \text{ este proprie} \\ D?, & \text{altfel,} \end{cases}$
dacă $t \in \mathcal{F}$ este constantă;
 - $p\mathcal{I}(t)(\psi) = \begin{cases} \psi(t), & \text{dacă } \psi \text{ este proprie} \\ D?, & \text{altfel,} \end{cases}$
dacă $t \in \mathcal{V} \cup \{\text{on-track}\}$ este variabilă;
 - $p\mathcal{I}(f(t_1, \dots, t_n))(\psi) = \sup(\{p\mathcal{I}(t_1)(\psi), \dots, p\mathcal{I}(t_n)(\psi)\});$
 - $p\mathcal{I}(\text{true})(\psi), p\mathcal{I}(\text{false})(\psi)$ și $p\mathcal{I}(p)(\psi)$, pentru orice constantă propozițională p , se definesc exact ca în cazul constantelor;
 - $p\mathcal{I}(t_1 = t_2)(\psi) = \sup(\{p\mathcal{I}(t_1)(\psi), p\mathcal{I}(t_2)(\psi)\});$
 - $p\mathcal{I}(P(t_1, \dots, t_n))(\psi) = \sup(\{p\mathcal{I}(t_1)(\psi), \dots, p\mathcal{I}(t_n)(\psi)\});$
 - $p\mathcal{I}(\neg e)(\psi) = p\mathcal{I}(e)(\psi);$
 - $p\mathcal{I}(e_1 \circ e_2)(\psi) = \sup(\{p\mathcal{I}(e_1)(\psi), p\mathcal{I}(e_2)(\psi)\}),$ pentru orice $\circ \in \{\vee, \wedge, \Rightarrow, \Leftrightarrow\}$
- (supremumul este considerat în laticea \mathbf{P}).

Acum, *funcția de p-interpretare a programelor* este funcția

$$p\phi_{\mathcal{I}}(S) : \Psi \rightarrow \Psi$$

dată prin:

- $p\phi_{\mathcal{I}}(S)(\psi) = \psi[x/p\mathcal{I}(t)(\psi)],$ pentru orice $\psi \in \Psi$, dacă S este programul $x := t$;
- $p\phi_{\mathcal{I}}(S) = p\phi_{\mathcal{I}}(S_2) \circ p\phi_{\mathcal{I}}(S_1),$ dacă S este programul $S_1; S_2$;
- $p\phi_{\mathcal{I}}(S)(\psi) = \begin{cases} \sup(\{p\phi_{\mathcal{I}}(S_1)(\psi), p\phi_{\mathcal{I}}(S_2)(\psi)\}), & \text{dacă } p\mathcal{I}(e)(\psi) = OK \\ LOST, & \text{altfel,} \end{cases}$
pentru orice $\psi \in \Psi$, dacă S este programul $\text{if } e \text{ then } S_1 \text{ else } S_2$;

- $p\phi_{\mathcal{I}}(S) = \mu(F),$ dacă S este programul $\text{while } e \text{ do } S_1$, unde F este funcția

$$F : [\Psi \rightarrow \Psi] \rightarrow [\Psi \rightarrow \Psi]$$

dată prin

$$F(f)(\psi) = \begin{cases} \sup(\{f(p\phi_{\mathcal{I}}(S_1)(\psi)), \psi\}), & \text{dacă } p\mathcal{I}(e)(\psi) = OK \\ LOST, & \text{altfel,} \end{cases}$$

pentru orice $f \in [\Psi \rightarrow \Psi]$ și $\psi \in \Psi$

(supremumul este considerat în mpo completă Ψ).

Așa cum se vede, funcția de p-interpretare a unui program se definește în manieră similară funcției semantică a programului (Secțiunea 7.3.1.5). Aceasta este natural deoarece funcția de p-interpretare a programului evaluează programul relativ la anumite valori asociate suplimentar variabilelor (și nu neapărat la valorile uzuale ale acestora).

Ca și în cazul funcției semantică a programelor structurate, se poate arăta cu ușurință că funcția de p-interpretare este bine definită și continuă.

Următorul algoritm implementează ideile prezentate mai sus.

Analiză dependențe funcționale

input: un program structurat S , o mulțime $I \subseteq \mathcal{V}$ de variabile de intrare și o mulțime $O \subseteq \mathcal{V}$ de variabile de ieșire;

output: “Yes” (există dependențe funcționale) sau “No?” (ar putea să nu existe dependențe funcționale);

begin

1. Fie ψ_0 astfel încât $OK(\psi_0) = I \cup \{\text{on-track}\}$;
 2. Fie $\psi_f := p\phi_{\mathcal{I}}(S)(\psi_0)$;
 3. **if** $O \cup \{\text{on-track}\} \subseteq OK(\psi_f)$ **then** “Yes” **else** “No?”
- end.**

Vom încheia secțiunea printr-un exemplu de aplicare a algoritmului de mai sus (p-stările vor fi restricționate doar la variabilele programului în cauză).

Exemplul 7.3.2.1. Considerăm programul structurat din Exemplul 7.3.1.5. Dacă ψ_0 este p-starea inițială, ca în algoritmul de analiză a dependențelor funcționale, atunci:

$$\begin{aligned} p\phi_{\mathcal{I}}(S)(\psi_0) &= p\phi_{\mathcal{I}}(\text{while } \neg(x = 0) \text{ do} \\ &\quad (y := y * x; x := x - 1))(p\phi_{\mathcal{I}}(y := 1)(\psi_0)) \\ &= p\phi_{\mathcal{I}}(\text{while } \neg(x = 0) \text{ do} (y := y * x; x := x - 1))(\psi_0[y/OK]) \\ &= \mu(F)(\psi_0[y/OK]), \end{aligned}$$

unde $F(f)(\psi)$ este dată prin:

- $F(f)(\psi) = \sup(\{f(p\phi_I(y := y * x; x := x - 1)(\psi)), \psi\})$, dacă este satisfăcută relația $p\mathcal{I}(\neg(x = 0))(\psi) = OK$;
- $F(f)(\psi) = LOST$, altfel,

pentru orice $f \in [\Psi \rightarrow \Psi]$ și $\psi \in \Psi$.

Dacă explicităm și mai mult funcția F obținem că aceasta este dată prin

- $F(f)(\psi) = \sup(\{f(\psi), \psi\})$, dacă $\psi(x) = OK$ și $\psi(\text{on-track}) = OK$;
- $F(f)(\psi) = LOST$, altfel,

pentru orice $f \in [\Psi \rightarrow \Psi]$ și $\psi \in \Psi$.

Fie $f_0 = \perp_{(\Psi \rightarrow \Psi)}$ cel mai mic element al mpo complete $[\Psi \rightarrow \Psi]$, adică funcția dată prin $f_0(\psi) = INIT$, pentru orice $\psi \in \Psi$. Calculul celui mai mic punct fix al funcției F se reduce la calculul supremumului lanțului $L = \{F^n(f_0)|n \geq 0\}$. Ca urmare, vom determina întâi elementele acestui lanț. Are loc:

$$F(f_0)(\psi) = \begin{cases} \psi, & \text{dacă } \psi(x) = OK \text{ și } \psi(\text{on-track}) = OK \\ LOST, & \text{altfel} \end{cases}$$

$$F^2(f_0)(\psi) = \begin{cases} \psi, & \text{dacă } \psi(x) = OK \text{ și } \psi(\text{on-track}) = OK \\ LOST, & \text{altfel,} \end{cases}$$

pentru orice ψ .

Deoarece $F(f_0) = F^2(f_0)$, supremumul lanțului este $F(f_0)$, care este deci și cel mai mic punct fix al funcției F .

Atunci

$$p\phi_I(S)(\psi_0)(y) = \mu(F)(\psi_0[y/OK])(y) = OK,$$

ceea ce ne arată că există o dependență funcțională între valoarea inițială a lui x și valoarea finală a variabilei y .

Pentru programul din Exemplul 7.3.2.1 au fost necesare doar două iterații pentru obținerea celui mai mic punct fix al funcției F asociate structurii while în cauză. Evident, acesta este un caz particular dar, în general, ne putem aștepta la un număr finit de iterații pentru atingerea celui mai mic punct fix, sau nu? În cazul determinării semanticii denotaționale răspunsul este negativ și este dat chiar de același program (Exemplul 7.3.1.5). În cazul analizei statice a programelor, datorită particularității p-stărilor, se poate arăta că cel mai mic punct fix poate fi atins după un număr finit de iterații.

Pentru prezentarea acestui rezultat definim întâi *variabilele libere* ale unui program structurat, în variantă inductivă, astfel:

- $FV(x := t) = \{x\} \cup FV(t)$;
- $FV(S_1; S_2) = FV(S_1) \cup FV(S_2)$;
- $FV(\text{if } b \text{ then } S_1 \text{ else } S_2) = FV(b) \cup FV(S_1) \cup FV(S_2)$;

- $FV(\text{while } b \text{ do } S) = FV(b) \cup FV(S)$.

Facem acum observația că pentru aplicarea algoritmului de analiză a dependențelor funcționale ne putem limita doar la o mulțime finită de variabile, și anume cele ale programului în cauză (p-stările au fost definite ca funcții de la $\mathcal{V} \cup \{\text{on-track}\}$ la \mathbf{P} , iar \mathcal{V} poate fi infinită deoarece ea este mulțimea de variabile peste care se construiesc programele structurate). Restrângând mulțimea de variabile asupra căreia ne îndreptăm atenția va trebui să restricționăm p-stările. Mai exact, dacă X este o submulțime a mulțimii \mathcal{V} , atunci vom defini restricția unei p-stări ψ la X ca fiind p-starea ψ_X dată prin:

$$\psi_X(x) = \begin{cases} \psi(x), & \text{dacă } x \in X \cup \{\text{on-track}\} \\ \psi(\text{on-track}), & \text{altfel,} \end{cases}$$

pentru orice $x \in \mathcal{V} \cup \{\text{on-track}\}$. Fie $\Psi_X = \{\psi_X | \psi \in \Psi\}$.

Vom nota cu $p\mathcal{I}_X$ și $p\phi_X$ funcțiile de p-interpretare obținute în același mod ca și funcțiile $p\mathcal{I}$ și $p\phi$ dar cu singura diferență că Ψ este înlocuită cu Ψ_X . În rest, păstrăm aceleași notații.

Următoarea teoremă constituie rezultatul central în ceea ce privește optimalitatea algoritmului. Demonstrația ei poate fi găsită în [160].

Teorema 7.3.2.1.

- (1) Pentru orice program $\text{while } b \text{ do } S$ are loc

$$p\phi_X(\text{while } b \text{ do } S) = F^{m+1}(f_0),$$

unde $X = FV(\text{while } b \text{ do } S)$, f_0 este cel mai mic element al mpo complete $[\Psi_X \rightarrow \Psi_X]$ și $m = |X|$.

- (2) Există un program $\text{while } b \text{ do } S$ pentru care are loc

$$p\phi_X(\text{while } b \text{ do } S) \neq F^{m-1}(f_0),$$

unde $X = FV(\text{while } b \text{ do } S)$, f_0 este cel mai mic element al mpo complete $[\Psi_X \rightarrow \Psi_X]$ și $m = |X|$.

7.3.2.2. Verificarea programelor

Așa cum s-a menționat, verificarea programelor se face luând în considerare un anumit tip de semantică asociată programelor și exploatajând diverse proprietăți ale semanticii în cauză. Evident, noi vom considera semantică denotațională (pentru că aceasta a fost dezvoltată în secțiunile anterioare). În cadrul acesteia, demonstrațiile de corectitudine se bazează în principal pe următoarele tehnici:

- principiul inducției de punct fix;

- teorema lui Park;
- proprietatea semantică de a fi cel mai mic punct fix al unei funcții continue, printre mulțimea tuturor punctelor fixe ale acelei funcții.

Vom exemplifica fiecare din aceste tehnici. Pentru mai multe tehnici și exemple, cititorul este îndrumat către [129].

Exemplul 7.3.2.2. (Corectitudine parțială pentru programe recursive)

Fie S programul recursiv

$$(S) \quad F(x, y) \Leftarrow \text{if } x = y \text{ then } 1 \text{ else } (y + 1) * F(x, y + 1)$$

interpretat peste \mathbf{N}_\perp (cu interpretările inițiale uzuale ale operatorilor, extinse natural acolo unde este necesar).

Dorim să demonstrăm că are loc

$$(\forall m \in \mathbf{N})(\mathcal{M}_I(S)(m, 0) \Downarrow \Rightarrow \mathcal{M}_I(S)(m, 0) = m!).$$

Vom face aceasta prin inducție de punct fix, demonstrând că are loc relația

$$\mu(\phi_I(S)) \leq f,$$

unde $f : \mathbf{N}_\perp \times \mathbf{N}_\perp \rightarrow \mathbf{N}_\perp$ este funcția dată prin

$$f(m, n) = \begin{cases} m!/n!, & \text{dacă } m, n \in \mathbf{N} \text{ și } m \geq n \\ \perp, & \text{altfel,} \end{cases}$$

pentru orice $m, n \in \mathbf{N}_\perp$ (cu convenția $0! = 1$).

Înainte de toate, remarcăm că predicatul ce definește proprietatea pe care dorim să o demonstrăm este admisibil, iar funcția semantică a programului este dată prin $\phi_I(S) = I(T)(\gamma)$, unde T este λ -termul asociat ecuației programului, iar γ este o atribuire arbitrară.

Parcurgem acum următorii pași:

- *Baza inducției:* $f_0 \leq f$, unde $f_0 : \mathbf{N}_\perp^2 \rightarrow \mathbf{N}_\perp$ este funcția total nedefinită. Aceasta este satisfăcută în mod trivial;
- *Pasul inductiv:* Presupunem că $g \leq f$, unde $g \in [\mathbf{N}_\perp^2 \rightarrow \mathbf{N}_\perp]$. Vom arăta că are loc

$$\phi_I(S)(g) \leq f,$$

adică,

$$(*) \quad \text{if } m = n \text{ then } 1 \text{ else } (n + 1) * g(m, n + 1) \leq f(m, n),$$

pentru orice $m, n \in \mathbf{N}_\perp$. Avem de analizat următoarele trei cazuri:

1. dacă $m = \perp$ sau $n = \perp$, atunci $(*)$ este îndeplinită în mod trivial;

2. dacă $m, n \in \mathbf{N}$ și $m = n$, atunci $(*)$ este satisfăcută deoarece

$$1 \leq m!/m! = 1;$$

3. dacă $m, n \in \mathbf{N}$ și $m \neq n$, atunci avem de arătat că are loc

$$(n + 1) * g(m, n + 1) \leq f(m, n).$$

Conform ipotezei, $g(m, n + 1) \leq f(m, n + 1)$. Înmulțind această relație cu $(n + 1)$ obținem inegalitatea pe care trebuie să o demonstrează.

Ca urmare, în baza Principiului inducției de punct fix, are loc

$$\mu(\phi_I(S)) \leq f,$$

care conduce la

$$\mu(\phi_I(S))(m, 0) \leq m!/0! = m!,$$

pentru orice $m \in \mathbf{N}$. De aici urmează proprietatea cerută.

Vom exemplifica acum modul de utilizare a Teoremei lui Park.

Exemplul 7.3.2.3. (Corectitudine parțială pentru programe recursive)

Fie S programul recursiv

$$(S) \quad F(x, y) \Leftarrow \text{if } x = y \text{ then } 1 \text{ else } (y + 2) * (y + 1) * F(x, y + 2)$$

interpretat peste \mathbf{N}_\perp (cu interpretările inițiale uzuale ale operatorilor, extinse natural acolo unde este necesar).

Dorim să demonstrăm că are loc

$$(\forall m, n \in \mathbf{N})(m \geq n \wedge \mathcal{M}_I(S)(m, n) \Downarrow \Rightarrow \mathcal{M}_I(S)(m, n) = m!/n!).$$

Dacă notăm prin $f : \mathbf{N}_\perp \times \mathbf{N}_\perp \rightarrow \mathbf{N}_\perp$ funcția dată prin

$$f(m, n) = \begin{cases} m!/n!, & \text{dacă } m, n \in \mathbf{N} \text{ și } m \geq n \\ \perp, & \text{altfel,} \end{cases}$$

pentru orice $m, n \in \mathbf{N}_\perp$, atunci proprietatea de mai sus se reduce la a arăta că are loc $\mu(\phi_I(S)) \leq f$. În baza Teoremei lui Park, această ultimă relație se reduce la a arăta că are loc $\phi_I(S)(f) \leq f$. Ca urmare, vom arăta că are loc:

$$\text{if } m = n \text{ then } 1 \text{ else } (n + 2) * (n + 1) * f(m, n + 2) \leq f(m, n),$$

pentru orice $m, n \in \mathbf{N}_\perp$. Pentru aceasta vom considera următoarele cazuri:

1. $m = \perp$ sau $n = \perp$. În acest caz proprietatea este trivial satisfăcută;
2. $m, n \in \mathbf{N}$ și $m = n$. Atunci ambii membri ai inegalității sunt 1;

3. $m, n \in \mathbf{N}$ și $m < n$. În acest caz, ambii membri ai inegalității sunt \perp ;
 4. $m, n \in \mathbf{N}$ și $m = n + 1$. În acest caz, membrul stâng al inegalității este \perp ;
 5. $m, n \in \mathbf{N}$ și $m \geq n + 2$. În acest caz, membrul stâng al inegalității este $(n+2) * (n+1) * f(m, n+2)$ care evident satisfacă inegalitatea
- $$(n+2) * (n+1) * f(m, n+2) \leq m!/n! = f(m, n).$$

Deci $\phi_{\mathcal{I}}(S)(f) \leq f$, iar în baza Teoremei lui Park obținem $\mu(\phi_{\mathcal{I}}(S)) \leq f$, care încheie demonstrația proprietății cerute.

Vom exemplifica acum metoda verificării programelor bazată pe proprietatea semantică denotaționale de a fi cel mai mic punct fix al unei funcții continue.

Exemplul 7.3.2.4. (Corectitudine parțială pentru programe structurate)

Fie programul structurat S dat prin

$y := 1; \text{while } \neg(x = 0) \text{ do } (y := y * x; x := x - 1)$

și interpretat peste $D = \mathbf{N}$ cu interpretarea uzualea a operatorilor \neg , $*$ și $-$ (mai mult, vom nota $\mathcal{I}_0(*)$ tot prin $*$, $\mathcal{I}_0(-)$ tot prin $-$ și $\mathcal{I}_0(n)$ tot prin n , pentru orice $n \geq 0$).

Vom arăta că programul S verifică:

$$(\forall \gamma \in \Gamma)(\mathcal{M}_{\mathcal{I}}(S)(\gamma) \downarrow \Rightarrow \mathcal{M}_{\mathcal{I}}(S)(\gamma)(y) = x!).$$

Aceasta se reduce la a arăta că are loc

$$(\forall \gamma \in \Gamma)(\phi_{\mathcal{I}}(S)(\gamma) \leq \gamma(x)!).$$

Cum pentru orice $\gamma \neq \perp$ are loc $\phi_{\mathcal{I}}(S)(\gamma) = \mu(F)(\gamma[y/1])$ (a se vedea Exemplul 7.3.1.5), proprietatea de mai sus se reduce la a arăta că are loc

$$(\forall \gamma \in \Gamma)(\mu(F)(\gamma[y/1]) \leq \gamma(x)!).$$

Fie funcția $g : \Gamma_{\perp} \rightarrow \Gamma_{\perp}$ dată prin

$$g(\gamma') = \begin{cases} \gamma'[y/\gamma'(y) * \gamma'(x)!][x/0], & \text{dacă } \gamma'(x) > 0 \\ \gamma', & \text{dacă } \gamma'(x) = 0 \\ \perp, & \text{dacă } \gamma' = \perp, \end{cases}$$

pentru orice $\gamma' \in \Gamma_{\perp}$.

Dacă arătăm că g este punct fix al funcției F , atunci obținem $\mu(F) \leq g$, de la care va urma proprietatea cerută. Reamintim întâi că F este dată prin:

- $F(f)(\gamma') = f(\gamma'[y/\gamma'(y) * \gamma'(x)][x/\gamma'(x) - 1]),$ dacă $\gamma'(x) > 0$;
- $F(f)(\gamma') = \gamma',$ dacă $\gamma'(x) = 0;$

- $F(f)(\gamma') = \perp,$ dacă $\gamma' = \perp,$

pentru orice $f \in [\Gamma_{\perp} \rightarrow \Gamma_{\perp}]$ și $\gamma' \in \Gamma_{\perp}.$

Atunci, notând $\gamma'' = \gamma'[y/\gamma'(y) * \gamma'(x)][x/\gamma'(x) - 1]$, pentru orice γ' , obținem:

$$\begin{aligned} F(g)(\gamma') &= \begin{cases} g(\gamma'[y/\gamma'(y) * \gamma'(x)][x/\gamma'(x) - 1]), & \text{dacă } \gamma'(x) > 0 \\ \gamma', & \text{dacă } \gamma'(x) = 0 \\ \perp, & \text{dacă } \gamma' = \perp \end{cases} \\ &= \begin{cases} \gamma''[y/\gamma''(y) * \gamma''(x)!][x/0], & \text{dacă } \gamma''(x) > 0 \\ \gamma'', & \text{dacă } \gamma''(x) = 0 \\ \gamma', & \text{dacă } \gamma'(x) = 0 \\ \perp, & \text{dacă } \gamma' = \perp \end{cases} \\ &= \begin{cases} \gamma'[y/\gamma'(y) * \gamma'(x) * (\gamma'(x) - 1)!][x/0], & \text{dacă } \gamma'(x) > 1 \\ \gamma'[y/\gamma'(y) * \gamma'(x)][x/\gamma'(x) - 1], & \text{dacă } \gamma'(x) = 1 \\ \gamma', & \text{dacă } \gamma'(x) = 0 \\ \perp, & \text{dacă } \gamma' = \perp \end{cases} \\ &= \begin{cases} \gamma'[y/\gamma'(y) * \gamma'(x)!][x/0], & \text{dacă } \gamma'(x) > 0 \\ \gamma', & \text{dacă } \gamma'(x) = 0 \\ \perp, & \text{dacă } \gamma' = \perp \end{cases} \\ &= g(\gamma'). \end{aligned}$$

Aceasta încheie demonstrația de corectitudine parțială.

Capitolul 8

Algebre universale

Structurile algebrice de semigrup, monoid, grup, inel, corp și spațiu vectorial, dar nu numai acestea, au în comun următoarea caracteristică foarte generală: toate consideră mulțimi pe care se definesc una sau mai multe operații ce satisfac diverse axiome. Pentru toate aceste structuri există multe concepte ce se introduc în mod similar, cum ar fi conceptul de substructură (submulțime închisă la operațiile structurii), homomorfism, congruență (relație de echivalență compatibilă cu operațiile structurii) etc. Este natural a pune problema găsirii unei structuri algebrice care să generalizeze toate aceste structuri și sub umbrela căreia să poată fi dezvoltată o serie de concepte și rezultate comune. O primă încercare în găsirea unei astfel de structuri a fost făcută în Secțiunea 1.4.4. Ea a fost suficientă la momentul respectiv pentru prezentarea într-un cadru general a unor concepte similare referitoare la semigrupuri, monoizi etc. Însă, aşa cum se va vedea mai departe, acele structuri nu erau altceva decât algebrelor universale, teoria acestora fiind subiectul acestui capitol.

Cel care a punctat clar pentru prima dată necesitatea unui astfel de studiu general și comparativ a fost matematicianul englez Alfred North Whitehead în lucrarea sa *A Treatise on Universal Algebra* din 1898 [220].¹

“Such algebras have intrinsic value for separate detailed study; also they are worthy of a comparative study, for the sake of the light thereby thrown on the general theory of symbolic reasoning, and on algebraic symbolism in particular. The comparative study necessarily presupposes some previous separate study, comparison being impossible without knowledge.”

Chiar dacă Whitehead a recunoscut necesitatea introducerii algebrelor universale, el nu a produs nici un rezultat în această direcție. Primele rezultate au fost publicate de G. Birkhoff abia în 1933² și, urmând direcția trasată de Birkhoff, până

¹A.N. Whitehead menționează în lucrarea sa că subiectul acesta a fost inspirat mult din lucrările lui W.R. Hamilton și A. De Morgan, iar terminologia de algebră universală ar fi fost introdusă de J.J. Sylvester.

²Acest interval de 35 de ani este oarecum justificat prin aceea că multe ramuri ale algebrei nu erau încă suficient dezvoltate pentru a permite un studiu comparativ consistent.

prin anii 1950 s-au scris multe lucrări ce au tratat aspecte legate de algebrelor libere, homomorfisme, congruențe, subalgebrelor etc.

O dată cu dezvoltarea logicii matematice, prin anii 1950, s-a creat posibilitatea aplicării acesteia în studiul algebrelor universale. Este vorba despre o serie de lucrări foarte importante datorate lui A. Tarski, C.C. Chang, L.A. Henkin, H.J. Keisler, D. Scott, A.I. Mal'cev etc. Algebrele multi-sortate, aşa cum vor fi prezentate în acest capitol, au fost introduse de P.J. Higgins în 1963 [88].

Primele aplicații ale algebrelor universale în informatică le întâlnim în teoria limbajelor formale și a automatelor în perioada anilor 1961-1963, prin lucrările lui M.P. Schützenberger care, în principal, reformulează algebric rezultatele obținute anterior de S.C. Kleene, J. Myhill, M. Rabin, D. Scott, A. Nerode etc. Anii 1965-1967 aduc noi rezultate obținute pe cale algebrică privitoare la descompunerea și sinteza automatelor; este vorba despre teoria inițiată de Krohn și Rhodes (1965) și despre studiile lui Eilenberg și Wright (1967). Aplicabilitatea algebrelor universale în teoria limbajelor de programare apare începând cu anul 1975 prin unele lucrări nepublicate ale lui S. Zilles despre utilizarea algebrelor multisortate și a ecuațiilor pentru a modela tipurile de date. Abordarea lui Zilles este dezvoltată de către J.V. Guttag și J.J. Horning (1975-1978) și, în mod definitiv, de către J.A. Goguen, J.W. Thatcher, E.G. Wagner și J.B. Wright³. Foarte pe larg vorbind, putem spune că, astăzi, algebrele universale au aplicații fundamentale în teoria automatelor, teoria tipurilor de date, studiul sintaxei și semanticii limbajelor de programare, programare funcțională, concurență și paralelism, verificarea programelor și verificarea hardware.

Referințele standard pentru studiul algebrelor universale, pe care le recomandăm tuturor acelora care doresc să aprofundeze acest domeniu, sunt [77, 36, 22, 142, 13]. Printre contribuțiiile românești la teoria algebrelor universale și a aplicațiilor acesteia în informatică menționăm pe cele aduse de Teodor Rus (actualmente, profesor la University of Iowa) și pe cele aduse de școala formată și condusă de profesorul Virgil Emil Căzănescu de la Universitatea din București. Acestuia din urmă îi datorăm forma finală a acestui capitol, atât prin sugestia de a prezenta algebrele universale permitând suporturi vide și supraîncărcarea operatorilor, cât și prin indicarea unor rezultate ce au simplificat sau au dat eleganță unor demonstrații. În plus, Secțiunea 8.9.1 a fost scrisă la sugestia acestuia.

8.1. Structuri sortate

Scopul acestei secțiuni este de a pregăti notația necesară în lucrul cu algebre universale. Vom lucra cu familii indexate de mulțimi, relații și funcții. Mulțimea de index va fi notată întotdeauna prin S care, în teoria algebrelor universale, se numește

³Grupul format din acești patru cercetători, J.A. Goguen, J.W. Thatcher, E.G. Wagner și J.B. Wright, este cunoscut în informatică ca fiind *grupul ADJ*.

mulțime de sorturi, elementele ei fiind numite sorturi. Utilizarea terminologiei de “sort” pentru elementele mulțimii S se bazează pe faptul că în aplicații practice elementele acestei mulțimi indică, mai mult sau mai puțin vizibil în mod direct, natura elementelor mulțimilor pe care le indexeză. De exemplu, alegerea $S = \{nat, bool\}$ sugerează că în cadrul familiei $(A_s | s \in S)$, A_{nat} este mulțimea numerelor naturale (dar nu forțează ca A_{nat} să fie aceasta). În acest fel putem gândi că elementele cu care se lucrează sunt “sortate” după tipul (natura) lor, permisând elemente cu mai mult de un tip (adică, mulțimile componente ale familiei nu sunt, în mod necesar, disjuncte).

Mulțimi sortate. În teoria algebrelor universale, familiile S -indexate se mai numesc și mulțimi S -sortate. Ca urmare, vom utiliza alternativ aceste două terminologii și, atunci când S se subînțelege din context, ea va fi omisă din terminologia în cauză.

În lucrul cu astfel de familii vom utiliza notații și terminologii specifice. Astfel, dacă $A = (A_s | s \in S)$ și $B = (B_s | s \in S)$ sunt familiile S -indexate (mulțimi S -sortate), atunci vom nota:

- $\emptyset = (\emptyset | s \in S)$ (în stânga egalității “ \emptyset ” desemnează familia vidă, pe când în dreapta egalității “ \emptyset ” desemnează mulțimea vidă. Distincția va fi întotdeauna clară din context, astfel că nu va fi nevoie să folosim notații distincte pentru a desemna familia ale cărei componente sunt toate mulțimea vidă, și mulțimea vidă);
- $A = \emptyset$ dacă $A_s = \emptyset$, pentru orice $s \in S$ (în lucrul cu mulțimi sortate, $A = \emptyset$ nu înseamnă că A este familia vidă, adică familia care nu are nici un element – a se vedea Secțiunea 1.2.4. Atragem atenția în mod expres că această convenție va funcționa numai pentru mulțimi sortate. Adică, atunci când vom spune că o familie de mulțimi sortate $(A^i | i \in I)$ este vidă, vom înțelege că $I = \emptyset$);
- $A \neq \emptyset$ dacă $A_s \neq \emptyset$, pentru orice $s \in S$;
- $A = B$ dacă $A_s = B_s$, pentru orice $s \in S$;
- $A \subseteq B$ dacă $A_s \subseteq B_s$, pentru orice $s \in S$;
- $A \subset B$ dacă $A_s \subset B_s$, pentru orice $s \in S$.

Operațiile uzuale cu familiile S -indexate se definesc pe componente. De exemplu,

- $A \cup B = ((A \cup B)_s | s \in S)$, unde $(A \cup B)_s = A_s \cup B_s$, pentru orice $s \in S$;
- $A \cap B$, $A - B$ și $A \times B$ se definesc similar reunii.

Produsul cartezian $A \times B$ poate fi extins la un număr finit de mulțimi S -sortate. Pentru cazul unei familii arbitrară de familii S -sortate, $(A^i | i \in I)$, definim produsul direct, pe componente, în mod ușor:

$$\prod_{i \in I} A^i = ((\prod_{i \in I} A^i)_s | s \in S),$$

unde

$$\left(\prod_{i \in I} A^i\right)_s = \{f | f : I \rightarrow \bigcup_{i \in I} A_s^i \wedge (\forall i \in I)(f(i) \in A_s^i)\} = \prod_{i \in I} A_s^i,$$

pentru orice sort s . În cazul $I = \emptyset$, $(\prod_{i \in I} A^i)_s = \{\emptyset\}$.

Dacă S^* este monoidul liber generat de S cu unitatea λ , iar $A = (A_s | s \in S)$ este o familie S -indexată, atunci vom nota:

- $A_w = \{\emptyset\}$, dacă $w = \lambda$,
- $A_w = A_{s_1} \times \dots \times A_{s_n}$, dacă $w = s_1 \dots s_n \in S^+$.

Această notație este în concordanță cu notația uzuală pentru produs cartezian. De exemplu, dacă $S = \{s\}$, atunci A_λ este corespondenta mulțimii A_s^0 , iar A_{s^n} , unde $n \geq 1$, este corespondenta mulțimii A_s^n .

Fie $A = (A_s | s \in S)$ o familie S -indexată și $(w, s) \in S^* \times S$. Dacă $f : A_w \rightarrow A_s$ este o funcție și $B \subseteq A$, atunci:

- pe lângă notația standard $f|_{B_w}$ ce desemnează restricția funcției f la B_w , vom utiliza și notația $f|_B$ care ne dă informații mai multe asupra contextului în care se lucrează, specificându-se familia B . Remarcăm că, pentru $w = \lambda$, $f|_{B_w}$ este chiar f deoarece $B_w = A_w = \{\emptyset\}$;
- pe lângă notația standard $f(B_w)$ ce desemnează imaginea mulțimii B_w prin f , vom utiliza și notația $f(B)$ care ne dă informații mai multe asupra contextului în care se lucrează, specificându-se familia B ;
- vom folosi și notația $f(B) \subseteq B$ pentru $f(B) \subseteq B_s$. Această notație are avantajul că nu cere specificarea sortului s ; ea nu trebuie confundată, și credem că nu poate fi confundată cu incluziunea între familii S -indexate.

Relații sortate. Relațiile binare pe familii S -indexate $A = (A_s | s \in S)$ sunt familii S -indexate de relații binare $\rho = (\rho_s | s \in S)$ astfel încât ρ_s este relație binară pe A_s , pentru orice $s \in S$. Astfel de relații se mai numesc și *relații S -sortate*. Atunci când diferența dintre "relație" și "relație S -sortată" se intenționează din context, vom omite termenul "S-sortată".

O relație ρ pe $A = (A_s | s \in S)$ este numită *relație de echivalență pe A* dacă fiecare ρ_s este relație de echivalență pe A_s . Dacă ρ este relație de echivalență pe A , atunci putem defini *familia cât sau factor A/ρ* dată prin $A/\rho = (A_s/\rho_s | s \in S)$.

Incluziunea, reuniunea, diferența, intersecția și compunerea de relații sortate se definesc pe componente. Relația ι_A este dată prin

$$\iota_A = (\iota_{A_s} | s \in S).$$

Evident, orice relație de echivalență ρ pe A satisface $\iota_A \subseteq \rho \subseteq A \times A$.

Restricția unei relații sortate ρ pe A la $B \subseteq A$ se face, de asemenea, pe componente. Adică $\rho|_B = (\rho_s|_{B_s} | s \in S)$.

Funcții sortate. Funcțiile sunt cazuri particulare de relații binare și, deci, *funcțiile S -sortate* vor fi cazuri particulare de relații S -sortate. Date două familii S -sortate $A = (A_s | s \in S)$ și $B = (B_s | s \in S)$, o *funcție S -sortată* de la A la B este o familie de funcții $f = (f_s | s \in S)$ astfel încât f_s este funcție de la A_s la B_s , pentru orice $s \in S$. Vom nota $f : A \rightarrow B$ pentru a specifica că f este funcție S -sortată de la A la B . Atunci când diferența dintre "funcție" și "funcție S -sortată" se intenționează din context, vom omite termenul "S-sortată".

Familiiile de funcții proiecție asociate unui produs cartezian $A \times B$ vor fi noteate $pr_1 = (pr_{1,s} | s \in S)$ și $pr_2 = (pr_{2,s} | s \in S)$. În cazul familiilor indexate $(A^i | i \in I)$, acestea vor fi noteate prin $pri = (pr_{i,s} | s \in S)$, pentru orice $i \in I$.

Concepțele de injectivitate, surjectivitate, bijectivitate, compunere de funcții, imagine și imagine inversă a unei subfamilii printr-o funcție S -sortată, restricție a unei funcții la o subfamilie, se introduc pe componente (pentru fiecare sort în parte). De exemplu, dacă $f : A \rightarrow B$ este o funcție S -sortată și $C \subseteq A$, atunci $f(C) = (f_s(C_s) | s \in S)$. Funcția identică pe A , notată 1_A , este $1_A = (1_{A_s} | s \in S)$.

Fie A și B două familii S -sortate de mulțimi și $f = (f_s | s \in S)$ o funcție de la A la B . Pentru orice $w \in S^*$ definim funcția $f_w : A_w \rightarrow B_w$ astfel:

- dacă $w = \lambda$, atunci $f_w(\emptyset) = \emptyset$;
- dacă $w = s \in S$, atunci $f_w(a) = f_s(a)$, pentru orice $a \in A_s$;
- dacă $w = s_1 \dots s_n \in S^+$ și $n > 1$, atunci

$$f_w(a_1, \dots, a_n) = (f_{s_1}(a_1), \dots, f_{s_n}(a_n)),$$

pentru orice $(a_1, \dots, a_n) \in A_w$.

În cazul în care f este funcția proiecție pr_i , notația f_w va deveni $pr_{i,w}$.

Predicale S -sortate sunt cazuri particulare de funcții S -sortate. Astfel, un predicator S -sortat pe o familie A este o familie de predicate $P = (P_s | s \in S)$, unde P_s este un predicator pe A_s , pentru orice $s \in S$.

Închideri. Teoria închiderii, prezentată în Secțiunea 1.3, poate fi ușor generalizată la cazul mulțimilor S -sortate. Pentru aceasta avem de considerat *relații r de tip $(w, s) \in S^* \times S$ pe V*, ce sunt submulțimi ale produsului cartezian $V_w \times V_s$, unde $V = (V_s | s \in S)$. În cazul $w = \lambda$, r va fi considerată ca submulțime a lui V_s (în loc de submulțime a mulțimii $\{\emptyset\} \times V_s$).

Dacă $A \subseteq V$, atunci prin $r(A_w)$ vom nota mulțimea

$$r(A_w) = \{b \in V_s | (\exists a \in A_w)((a, b) \in r)\}.$$

Dacă $w = \lambda$, atunci $r(A_w) = r$, conform convenției adoptate mai sus. Vom utiliza frecvent notația $r(A)$ în loc de $r(A_w)$, așa cum am făcut și la funcții.

Spunem că A este *închisă* la o familie \mathcal{R} de relații cu tip dacă A este închisă la orice relație $r \in \mathcal{R}$, adică $r(A) \subseteq A_s$ (presupunând că r are tipul (w, s)).

Închiderea familiei A la \mathcal{R} , notată prin $\mathcal{R}[\![A]\!]$, este cea mai mică familie ce include A și este închisă la \mathcal{R} . Ca în demonstrația Teoremei 1.3.1.1 se arată că $\mathcal{R}[\![A]\!]$ există, este unică și poate fi obținută ca supremumul (reuniunea) șirului de mulțimi S -sortate ($B^m | m \geq 0$) dat prin:

- $B^0 = A$;
- $B^{m+1} = B^m \cup (\bigcup_{r \in \mathcal{R} \text{ de tip } (w, s)} r(B^m) | s \in S)$, pentru orice $m \geq 0$

(atragem atenția că B^m nu reprezintă produsul cartezian al familiei B cu ea însăși de m ori, ci este o notație pentru mulțimea construită la pasul m pornind cu A).

Principiul inducției structurale poate fi și el adaptat ușor la cazul mulțimilor sortate. Dacă $B = \mathcal{R}[\![A]\!]$ iar $P = (P_s | s \in S)$ este un predicat S -sortat ce satisfacă

- (1) $P_s(a)$, pentru orice $s \in S$ și $a \in A_s$;
- (2) $(P_{s_1}(a_1) \wedge \dots \wedge P_{s_n}(a_n)) \Rightarrow P_s(a)$, pentru orice $r \in \mathcal{R}$ de tip $(s_1 \dots s_n, s)$, $(a_1, \dots, a_n) \in B_{s_1 \dots s_n}$ și $a \in B_s$ cu $((a_1, \dots, a_n), a) \in r$ (în cazul $s_1 \dots s_n = \lambda$, proprietatea P se înțelege a fi satisfăcută de orice $a \in r$),

atunci $P_s(a)$, pentru orice $s \in S$ și $a \in B_s$.

Fie $B = \mathcal{R}[\![A]\!]$. Pentru orice $s \in S$ și $a \in B_s$, elementul a admite cel puțin o construcție inductivă de la A și \mathcal{R} ,

$$a_0, a_1, \dots, a_k = a,$$

unde, pentru orice $i \leq k$, are loc una din următoarele proprietăți:

- există un sort s_i astfel încât $a_i \in A_{s_i}$, sau
- există $i_1, \dots, i_n < i$ și $r \in \mathcal{R}$ de tip $(s_{i_1} \dots s_{i_n}, s_i)$ astfel încât $a_{i_j} \in B_{s_{i_j}}$, pentru orice $1 \leq j \leq n$, și $((a_{i_1}, \dots, a_{i_n}), a_i) \in r$ (în cazul $s_1 \dots s_n = \lambda$ înțelegem că $a_i \in r$). În plus, dacă $i = k$, atunci $s_i = s$.

Atunci când orice element $b \in B_s$, pentru orice s , admite o unică construcție inductivă de la A și \mathcal{R} , vom spune că B este liber inductiv definită. Într-un astfel de caz, orice funcție de la A la C poate fi extinsă la o unică funcție de la B la C (ceea ce constituie un echivalent al Teoremei recursiei din Secțiunea 1.3.3 pentru mulțimi S -sortate).

8.2. Signaturi și algebrelle

O algebră universală este formată dintr-o familie de mulțimi $(A_s | s \in S)$, unde S este o mulțime de index nevidă, și o mulțime de operații ce acționează pe aceste mulțimi. Dacă $\circ : A_{s_1} \times \dots \times A_{s_n} \rightarrow A_s$ este o astfel de operație, atunci domeniul ei este

complet specificat de secvența $s_1 \dots s_n \in S^*$, iar codomeniul ei, de elementul $s \in S$. Perechea $(s_1 \dots s_n, s) \in S^* \times S$ va fi numită *tipul* operației \circ .

În studiul general al algebrelor universale nu vom fi interesați de specificarea concretă a operațiilor sau de proprietățile particulare ale acestora, ci numai de tipul lor. O astfel de specificație de tip va purta denumirea de *signatură*.

Definiția 8.2.1. Fie S o mulțime nevidă. Se numește *signatură S -sortată* orice familie $(S^* \times S)$ -indexată de mulțimi $\Sigma = (\Sigma_{w,s} | (w, s) \in S^* \times S)$.

Fie $\Sigma = (\Sigma_{w,s} | (w, s) \in S^* \times S)$ o signatură S -sortată. Elementele mulțimii S sunt numite în mod ușual *sorturi* (a se vedea Secțiunea 8.1), iar elementele mulțimilor $\Sigma_{w,s}$ sunt numite *simboluri de operații* sau *simboluri de funcții* sau *simboluri funcționale*. Dacă $\sigma \in \Sigma_{w,s}$, atunci secvența w este numită *aritatea* sau *tipul domeniului* simbolului funcțional σ , s este numit *sortul* sau *tipul codomeniului* lui σ , iar cuplul (w, s) , *tipul* lui σ . Simbolurile funcționale de aritate λ ce au tipul codomeniului $s \in S$ sunt numite *simboluri constante de sort* s . În mod ușual, mulțimile $\Sigma_{\lambda,s}$ sunt renotate prin Σ_s .

Signaturile S -sortate pentru care $|S| = 1$ se mai numesc și *signaturi unisortate*. În contrast, signaturile S -sortate în care S este arbitrară dar nevidă se mai numesc *signaturi multisortate*.

Observația 8.2.1. Fie $\Sigma = (\Sigma_{w,s} | (w, s) \in S^* \times S)$ o signatură S -sortată.

- (1) Familia de mulțimi Σ poate să nu fie disjunctă. Aceasta ne spune că pot exista simboluri funcționale ce au mai mult de un tip. Nu trebuie să privim aceasta cu suspiciune. De exemplu, în matematica de zi cu zi utilizăm simbolul “+” pentru a desemna adunarea numerelor întregi, a numerelor reale sau a numerelor complexe. Această proprietate, numită *supraîncărcarea operatorilor*⁴, se dovedește a fi foarte importantă și în informatică, în special în cadrul tehnicilor de parsare pentru limbajele de programare moderne aşa cum ar fi Ada, în programarea orientată pe obiecte, în cadrul tehnicilor de rafinare a reprezentării datelor etc. Chiar și limbajele de programare clasice permit supraîncărcarea operatorilor utilizând, de exemplu, același simbol funcțional pentru adunarea atât a numerelor întregi, cât și a numerelor reale.

Pentru a evita posibilele ambiguități ce pot apărea prin referirea la simbolurile funcționale ale signaturii Σ vom folosi exprimări de genul “ $\sigma \in \Sigma_{w,s}$ ”, punând astfel clar în evidență tipul sub care se consideră simbolul funcțional σ . Atunci când specificarea tipului nu este importantă vom folosi notația “ $\sigma \in \Sigma$ ” pentru a spune că σ este un simbol funcțional al signaturii Σ .

- (2) Dacă Σ este signatură unisortată, de exemplu $S = \{s\}$, atunci tipul (w, s) poate fi specificat mai simplu indicând doar $|w|$. Este potrivit atunci a renota mulțimile $\Sigma_{w,s}$ prin $\Sigma_{|w|}$.

⁴Termenul în limba engleză este cel de *overloading*.

- (3) Dacă Σ este o familie disjunctă, caz în care vom mai spune că Σ este o *signatură disjunctă*, atunci putem defini funcția

$$\tau : \bigcup_{(w,s) \in S^* \times S} \Sigma_{w,s} \rightarrow S^* \times S$$

prin $\tau(\sigma) = (w, s)$, pentru orice $(w, s) \in S^* \times S$ și $\sigma \in \Sigma_{w,s}$. Signatura Σ este atunci complet specificată de cuplul $(\bigcup_{(w,s) \in S^* \times S} \Sigma_{w,s}, \tau)$. Funcția τ poartă denumirea de *funcția de tip* a signaturii.

Reciproc, orice pereche (X, τ) formată dintr-o mulțime X și o funcție τ de la X la $S^* \times S$ induce o familie $(S^* \times S)$ -indexată disjunctă de submulțimi ale lui X

$$(X_{w,s} | (w, s) \in S^* \times S),$$

unde $X_{w,s} = \{\sigma \in X | \tau(\sigma) = (w, s)\}$, pentru orice $(w, s) \in S^* \times S$. Această familie poate fi privită ca o signatură S -sortată disjunctă.

Ca urmare a acestei echivalențe, vom prefera ca signaturile disjuncte să fie private ca mulțimi de simboluri funcționale și nu ca familiile de mulțimi de simboluri funcționale. Notația " $\sigma \in \Sigma$ " va fi atunci notația clasică de apartenență a simbolului funcțional σ la mulțimea Σ .

Am clasificat astfel signaturile (unisortate, multisortate) în disjuncte și nedisjuncte. În [77, 36, 22, 13] sunt considerate numai signaturi unisortate disjuncte, în timp ce în [142, 130] este considerat cazul general.

Prezentăm în continuare câteva exemple de signaturi.

Exemplul 8.2.1.

- (1) Fie $S = \{s\}$ și

- $\Sigma_s = \{\sigma_3\}$;
- $\Sigma_{s,s} = \{\sigma_2\}$;
- $\Sigma_{ss,s} = \{\sigma_1, \sigma_4\}$;
- $\Sigma_{w,s} = \emptyset$, în celelalte cazuri.

Structura astfel definită este o signatură unisortată disjunctă. Conform Observației 8.2.1(2), am fi putut nota Σ_s prin Σ_0 , $\Sigma_{s,s}$ prin Σ_1 , și $\Sigma_{ss,s}$ prin Σ_2 .

- (2) Fie $S = \{nat\}$ și

- $\Sigma_{nat} = \{zero\}$;
- $\Sigma_{nat,nat} = \{succ\}$;
- $\Sigma_{w,s} = \emptyset$, în celelalte cazuri.

Structura astfel definită este o signatură unisortată disjunctă. Notațiile utilizate sunt oarecum specifice informaticii: *nat* sugerează un domeniu de numere naturale, *zero* sugerează numărul 0, iar *succ* sugerează funcția succesor.

- (3) Fie $S = \{bool\}$ și

- $\Sigma_{bool} = \{true, false\}$;
- $\Sigma_{bool,bool} = \{not\}$;
- $\Sigma_{bool bool,bool} = \{and\}$;
- $\Sigma_{w,s} = \emptyset$, în celelalte cazuri.

Structura astfel definită este o signatură unisortată disjunctă. *bool* sugerează un domeniu de numere naturale, *true* sugerează valoarea de adevăr “adevărat”, *false* sugerează valoarea de adevăr “fals”, *not* sugerează negația, iar *and* sugerează conjuncția.

- (4) Fie $S = \{nat, bool\}$ și

- $\Sigma_{nat} = \{zero\}$;
- $\Sigma_{nat,nat} = \{succ\}$;
- $\Sigma_{bool} = \{true, false\}$;
- $\Sigma_{bool,bool} = \{not\}$;
- $\Sigma_{bool bool,bool} = \{and\}$;
- $\Sigma_{nat nat,nat} = \{add, mult\}$;
- $\Sigma_{nat nat,bool} = \{leq\}$;
- $\Sigma_{w,s} = \emptyset$, în celelalte cazuri.

Structura astfel definită este o signatură multisortată disjunctă. *add* sugerează adunarea numerelor naturale, *mult* sugerează înmulțirea numerelor naturale, iar *leq* sugerează operatorul \leq (restul simbolurilor pot fi considerate ca la (2) și (3)).

Definiția 8.2.2. O signatură S -sortată Σ este numită *finită* dacă mulțimea

$$\{(\sigma, w, s) | (w, s) \in S^* \times S \wedge \sigma \in \Sigma_{w,s}\}$$

este finită.

Exemplul 8.2.2. Toate signaturile din Exemplul 8.2.1 sunt signaturi finite.

Definiția 8.2.3. Fie Σ și Σ' două signaturi S -și, respectiv, S' -sortate. Spunem că Σ este o *subsignatură* a signurii Σ' dacă au loc relațiile $S \subseteq S'$ și $\Sigma_{w,s} \subseteq \Sigma'_{w,s}$, pentru orice $(w, s) \in S^* \times S$.

Exemplul 8.2.3. Signaturile din Exemplul 8.2.1(2)(3) sunt subsignaturi ale signaturii din Exemplul 8.2.1(4).

Definiția 8.2.4. Fie Σ și Σ' două signaturi S -șii, respectiv, S' -sortate. Se numește *homomorfism* sau *morfism* de la Σ la Σ' orice pereche (φ, ψ) formată dintr-o funcție $\varphi : S \rightarrow S'$ și o familie $(S^* \times S)$ -indexată de funcții $\psi = (\psi_{w,s} | (w, s) \in S^* \times S)$ cu proprietatea că, pentru orice $(w, s) \in S^* \times S$, $\psi_{w,s}$ este funcție de la $\Sigma_{w,s}$ la $\Sigma'_{\varphi^*(w,s)}$, unde φ^* este extensia homomorfă a funcției φ la $S^* \times S$ (adică

- $\varphi^*(\lambda, s) = (\lambda, \varphi(s)),$
- $\varphi^*(s_1 \cdots s_n, s) = (\varphi(s_1) \cdots \varphi(s_n), \varphi(s)),$

pentru orice $s_1 \cdots s_n \in S^+$ and $s \in S$).

Observația 8.2.2. Dacă Σ este o signatură S -sortată disjunctă, iar Σ' este o signatură S' -sortată disjunctă, atunci homomorfismele de la Σ la Σ' pot fi definite ca perechi de funcții (φ, ψ) ce satisfac $\tau' \circ \psi = \varphi^* \circ \tau$, unde $\varphi : S \rightarrow S'$, $\psi : \Sigma \rightarrow \Sigma'$, iar τ și τ' sunt funcțiile de tip ale celor două signaturi (Observația 8.2.1(3)).

Proprietatea " $\tau' \circ \psi = \varphi^* \circ \tau$ " poate fi exprimată simplu prin "diagrama din Figura 8.1 comută".

$$\begin{array}{ccc} \Sigma & \xrightarrow{\psi} & \Sigma' \\ \tau \downarrow & & \downarrow \tau' \\ S^* \times S & \xrightarrow{\varphi^*} & S'^* \times S' \end{array}$$

Figura 8.1: Homomorfism de signaturi disjuncte

Evident, dacă (φ, ψ) și (φ', ψ') sunt morfisme de signaturi de la Σ la Σ' și, respectiv, de la Σ' la Σ'' , atunci putem defini *compunerea lor* prin

$$(\varphi, \psi) \circ (\varphi', \psi') = (\varphi' \circ \varphi, \psi''),$$

unde $\psi'' = (\psi''_{w,s} | (w, s) \in S^* \times S)$ și $\psi''_{w,s} = \psi'_{\varphi^*(w,s)} \circ \psi_{w,s}$, pentru orice $(w, s) \in S^* \times S$. Este ușor de verificat că această definiție este consistentă în sensul că, pentru orice $(w, s) \in S^* \times S$, $\psi''_{w,s}$ este funcție de la $\Sigma_{w,s}$ la $\Sigma''_{(\varphi' \circ \varphi)^*(w,s)}$. Ca urmare, $(\varphi, \psi) \circ (\varphi', \psi')$ este morfism de la Σ la Σ'' .

În cazul signaturilor disjuncte compunerea de homomorfisme este ilustrată în diagrama din Figura 8.2 (a se vede Observația 8.2.2).

Conceptul central în teoria algebrelor universale este cel de *algebră universală*. O algebră universală nu este altceva decât o "interpretare" particulară a unei signaturi, interpretare ce se obține prin asocierea de domenii sorturilor și de funcții simbolurilor funcționale.

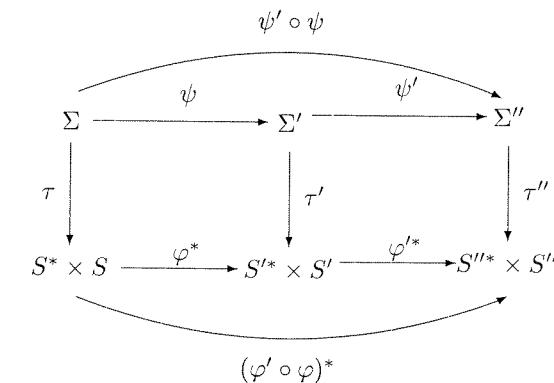


Figura 8.2: Componere de homomorfisme de signaturi disjuncte

Definiția 8.2.5. Fie Σ o signatură S -sortată. O *algebră universală de signatură* Σ , sau Σ -algebră, este un cuplu $\mathbf{A} = (A, \Sigma^A)$, unde:

- (1) $A = (A_s | s \in S)$ este o familie S -indexată de multimi;
- (2) $\Sigma^A = (\Sigma_{w,s}^A | (w, s) \in S^* \times S)$, $\Sigma_{w,s}^A = \{\sigma^A | \sigma \in \Sigma_{w,s}\}$, iar σ^A este o funcție (totală) de la A_w la A_s , pentru orice $(w, s) \in S^* \times S$ și $\sigma \in \Sigma_{w,s}$.

Dacă $\mathbf{A} = (A, \Sigma^A)$ este o Σ -algebră, multimile A_s vor fi numite *suportul de sort* s al Σ -algebrei, iar elementele multimilor $\Sigma_{w,s}^A$, *operațiile* sau *funcțiile* Σ -algebrei. Operațiile din Σ_s^A se mai numesc și *operații constante de sort* s sau *constante de sort* s .

Atragem din nou atenția asupra faptului că o signatură Σ poate avea simboluri funcționale cu mai mult de un tip. În acest caz, notația σ^A este ambiguă. Însă, specificând că σ este în $\Sigma_{w,s}$, vom înțelege implicit că în respectivul context, σ^A este considerată ca fiind definită pe A_w și cu valori în A_s . În acest mod credeam că tot ceea ce va urma în acest capitol poate fi înțeles fără dificultăți, păstrând o notație suplă (o notație de genul (σ, w, s) pentru simbolul funcțional σ și (σ^A, w, s) pentru σ^A poate fi un candidat ideal în evitarea ambiguităților posibile [130]). Însă, o astfel de notație este greoai, în timp ce notația deja adoptată, manipulată cu atenție, nu are cum să creeze ambiguități.

Observația 8.2.3. Fie $\mathbf{A} = (A, \Sigma^A)$ o Σ -algebră, unde Σ este o signatură S -sortată.

- (1) Dacă $\sigma \in \Sigma_s$, atunci σ^A este o funcție (totală) de la $\{\emptyset\}$ la A_s . Totalitatea funcției σ^A ne spune că A_s nu poate fi vidă (altfel σ^A ar trebui să fie funcția vidă, însă funcția vidă este totală numai dacă domeniul ei de definiție este multimea vidă). Cum A_s este nevidă, $\sigma^A \in \Sigma_s^A$ poate fi identificată prin elementul $\sigma^A(\emptyset) \in A_s$, așa cum am făcut și în Secțiunea 1.2.3 cu operațiile 0-are. Aceasta ne

permite să scriem simplificat $\sigma^A \in A_s$ și, deci, $\Sigma_s^A \subseteq A_s$, pentru orice $s \in S$ pentru care $\Sigma_s \neq \emptyset$ (adică, constantele de sort s sunt elemente ale suportului A_s).

- (2) Operațiile algebrei \mathbf{A} pot fi împărțite în 3 clase în funcție de tipul lor, astfel:
 - operații constante, induse de simboluri funcționale constante;
 - operații vide, induse de simboluri funcționale de tip $(s_1 \cdots s_n, s)$, unde $n \geq 1$ și cel puțin una din mulțimile A_{s_1}, \dots, A_{s_n} este vidă (dacă toate aceste mulțimi sunt nevide, atunci totalitatea operației induse conduce la faptul că A_s este nevidă și, deci, operația indusă este nevidă);
 - operații nevide, induse de simboluri funcționale de tip $(s_1 \cdots s_n, s)$, unde $n \geq 1$ și toate mulțimile A_{s_1}, \dots, A_{s_n} sunt nevide (operațiile constante sunt nevide, dar pe acestea le-am grupat într-o clasă separată pentru a face distincția între ele și celelalte operații nevide);
- (3) În unele abordări, cum ar fi [142], se cere ca mulțimile suport ale Σ -algebrelor să fie nevide.

Atunci când signatura Σ este subînteleasă din context, vom înlocui terminologia de “ Σ -algebră” prin cea de “algebră”. Algebrele peste signaturi unisortate (multisortate) se mai numesc *algebrelor unisortate (multisortate)*.

Algebrele vor fi notate prin **A**, **B**, **C** etc., înțelegând implicit că familia mulțimilor suport este notată prin A , B , C etc., iar familia mulțimilor de operații prin Σ^A , Σ^B , Σ^C etc. În cazul algebrelor unisortate vom înlocui familia mulțimilor suport prin unica mulțime a acestei familii, eliminând indexul indus de unicul sort. Adică, într-un astfel de caz, o algebră **A** va fi un cuplu (A, Σ^A) format dintr-o mulțime A și o familie de mulțimi de funcții pe A , $\Sigma^A = \{\Sigma_n^A | n \geq 0\}$, astfel încât pentru orice $n \geq 0$ și $\sigma \in \Sigma_n$, σ^A este funcție de la A^n la A . Mai mult, dacă Σ^A nu conține “prea multe” elemente, atunci le putem lista separat unul după altul, după mulțimea A , având grijă să specificăm tipul operațiilor în caz de nedisjunctivitate. De exemplu,

$$\mathbf{A} = (A, (\sigma_1^A, w, s), (\sigma_1^A, w', s'), \sigma_2^A, \sigma_3^A)$$

poate desemna o algebră unisortată cu 4 operații: primele două operații sunt bazate pe un același simbol funcțional dar cu tipuri diferite $(w, s) \neq (w', s')$, ultimele două operații având exact câte un tip (specificat prin signatură).

Exemplul 8.2.4.

- (1) Considerăm signatura din Exemplul 8.2.1(1) și:

- $A_s = R$, unde R este o mulțime arbitrară nevidă;
- $\sigma_1^A = +$, unde $+$ este o operație binară pe R ;
- $\sigma_2^A = -$, unde $-$ este o operație unară pe R ;

- $\sigma_3^A = 0$, unde 0 este o operație nulară pe R ;
- $\sigma_4^A = \cdot$, unde \cdot este o operație binară pe R .

Ceea ce am obținut este o algebră universală. Dacă operațiile de mai sus satisfac proprietățile din Exemplul 1.4.4.1(5), atunci algebra astfel definită este un inel. Ca urmare, signatura din Exemplul 8.2.1(1) poate fi gândită ca o signatură a tuturor inelelor deoarece acestea pot fi obținute prin interpretări corespunzătoare ale acestei signaturi.

- (2) Semigrupurile, monoizii, grupurile, inelelele, corporile, spațiile vectoriale, toate sunt cazuri particulare de algebrelor universale. În cazul spațiilor vectoriale trebuie considerate două sorturi, $S = \{s_1, s_2\}$, unul pentru a defini corpul peste care se construiește spațiul vectorial și unul pentru spațiul vectorial în cauză. Simbolurile funcționale se aleg în mod similar cazului precedent.
- (3) Considerăm signatura din Exemplul 8.2.1(2) și:

- $A_{nat} = \mathbb{N}$;
- $zero^A = 0$, unde 0 este numărul natural zero;
- $succ^A = Succ$, unde $Succ$ este funcția $Succ : \mathbb{N} \rightarrow \mathbb{N}$ dată prin $Succ(n) = n + 1$, pentru orice $n \in \mathbb{N}$.

Ceea ce am obținut este o algebră universală.

Invităm cititorul să interpreze și signaturile din Exemplul 8.2.1(3)(4) după cum sugerează numele operațiilor acestora (dar nu neapărat).

Observația 8.2.4. Fie **A** o Σ -algebră. Mulțimile $\Sigma_{w,s}^A$ sunt formate din funcții totale. Există cazuri în care nu toate operațiile algebrei pot fi totale (atunci când o algebră universală se dorește a modela o anumită structură). Astfel de cazuri nu vor fi considerate în lucrarea noastră, dar cititorul interesat poate consulta [171].

Definiția 8.2.6. O algebră **A** este numită *vidă* dacă $A_s = \emptyset$, pentru orice $s \in S$.

Observația 8.2.5. Este ușor de văzut că o Σ -algebră vidă poate exista doar dacă Σ nu conține simboluri constante. Mai mult, dacă Σ nu conține simboluri constante atunci există exact o Σ -algebră vidă.

Σ -algebra vidă, atunci când există, are numai operații vide.

Definiția 8.2.7. O algebră **A** este numită *trivială* dacă $|A_s| \leq 1$, pentru orice $s \in S$.

Observația 8.2.6. Fie Σ o signatură.

- (1) Orice familie de mulțimi $A = (A_s | s \in S)$ cu proprietatea $|A_s| = 1$, pentru orice $s \in S$, poate fi organizată ca o Σ -algebră care, conform definiției, este trivială.

- (2) Dacă Σ nu are simboluri constante, atunci Σ -algebra vidă este algebră trivială de signatură Σ .

Pentru orice signatură Σ există Σ -algebrelor triviale.

Exemplul 8.2.5. Monoizii triviali, grupurile triviale, inelele triviale, spațiile vectoriale triviale, sunt cazuri particulare de algebrelor universale triviale.

8.3. Subalgebrelor. Inducție structurală

Așa cum am spus în Secțiunea 8.1, dacă $B \subseteq A$ este o subfamilie S -sortată a familiei A iar $\sigma \in \Sigma_{w,s}$ este un simbol funcțional de tip (w, s) , atunci vom nota restricția operației σ^A la $B_w \subseteq A_w$ prin $\sigma^A|_B$ în loc de $\sigma^A|_{B_w}$, ori de câte ori nu va fi necesar să specificăm w . Atunci când $w = \lambda$, $B_w = \{\emptyset\} = A_w$ și $\sigma^A|_{B_w} = \sigma^A$.

Atragem din nou atenția asupra faptului că un simbol funcțional σ poate avea mai multe tipuri. Însă atunci când vom scrie " $\sigma \in \Sigma_{w,s}$ " vom înțelege că (w, s) este tipul sub care se consideră simbolul funcțional σ .

Definiția 8.3.1. O Σ -algebră \mathbf{B} este numită *subalgebră* a Σ -algebrelor \mathbf{A} , și notăm $\mathbf{B} \leq \mathbf{A}$, dacă $B \subseteq A$ și $\sigma^B = \sigma^A|_B$, pentru orice tip (w, s) și $\sigma \in \Sigma_{w,s}$.

Dacă \mathbf{B} este subalgebră a algebrelor \mathbf{A} și, în plus, $\mathbf{B} \neq \mathbf{A}$, atunci vom spune că \mathbf{B} este *subalgebră proprie* a algebrelor \mathbf{A} și vom nota $\mathbf{B} < \mathbf{A}$.

Mulțimea tuturor subalgebrelor algebrelor \mathbf{A} va fi notată prin $\text{Sub}(\mathbf{A})$. Această mulțime este nevidă deoarece conține cel puțin algebra \mathbf{A} (existența algebrelor vide depinde direct de signatură, motiv pentru care nu putem afirma că aceasta este, întotdeauna, subalgebră a algebrelor \mathbf{A}).

Exemplul 8.3.1. Privind semigrupurile ca algebrelor universale, subsemigrupurile sunt subalgebrelor în semigrupuri. Același lucru este valabil și pentru submonoizi, subgrupuri etc.

Definiția 8.3.2. Fie \mathbf{A} o algebră și $B \subseteq A$. Spunem că B este *închisă în* \mathbf{A} dacă $\sigma^A(B) \subseteq B$, pentru orice tip (w, s) și $\sigma \in \Sigma_{w,s}$.

Observația 8.3.1. Fie \mathbf{A} o algebră.

- (1) În Definiția 8.3.2 este suficient să cerem " $\sigma^A(B) \subseteq B$ " doar pentru operațiile σ^A nevide (inclusând și operațiile constante), ceea ce vom face pe parcursul întregului capitol.
- (2) Dacă $B \subseteq A$ este închisă în \mathbf{A} , atunci $\Sigma_s^A \subseteq B_s$, pentru orice $s \in S$ (adică, orice subfamilie închisă în \mathbf{A} conține constantele algebrelor \mathbf{A}).

- (3) Suportul oricărei subalgebrelor \mathbf{B} a algebrelor \mathbf{A} este închis în \mathbf{A} . Reciproc, dacă $B \subseteq A$ este închisă în \mathbf{A} , atunci ea induce o subalgebră $\mathbf{B} = (B, \Sigma^B)$, unde operațiile σ^B sunt date prin $\sigma^B = \sigma^A|_B$, pentru orice tip (w, s) și $\sigma \in \Sigma_{w,s}$.
- (4) Dacă $(B^i | i \in I)$ este o familie nevidă de subfamilii ale lui A ce sunt închise în \mathbf{A} , atunci $\bigcap_{i \in I} B^i$ este închisă în \mathbf{A} .
- (5) Dacă $(\mathbf{B}^i | i \in I)$ este o familie nevidă de subalgebrelor ale algebrelor \mathbf{A} , atunci $\bigcap_{i \in I} B^i$ induce o subalgebră a algebrelor \mathbf{A} , notată $\bigcap_{i \in I} \mathbf{B}^i$ și numită *intersecția familiei* $(\mathbf{B}^i | i \in I)$.

Propoziția 8.3.1. Fie \mathbf{A} , \mathbf{B} și \mathbf{C} algebrelor astfel încât $\mathbf{B} \leq \mathbf{A}$. Atunci $\mathbf{C} \leq \mathbf{B}$ dacă și numai dacă $\mathbf{C} \leq \mathbf{A}$ și $C \subseteq B$.

Demonstrație. Să presupunem întâi că $\mathbf{C} \leq \mathbf{B}$. Atunci $C \subseteq B$, și cum $B \subseteq A$, urmează că $C \subseteq A$. Pentru orice tip (w, s) și $\sigma \in \Sigma$, $\sigma^C = \sigma^B|_C$ și $\sigma^B = \sigma^A|_B$. Cum $C \subseteq B$, deducem $\sigma^C = \sigma^A|_C$. Deci $\mathbf{C} \leq \mathbf{A}$.

Reciproc, pentru orice tip (w, s) și $\sigma \in \Sigma_{w,s}$ are loc

$$\sigma^C = \sigma^A|_C \subseteq \sigma^A|_B = \sigma^B,$$

ceea ce arată că $\sigma^C = \sigma^B|_C$. Combinând aceasta cu incluziunea $C \subseteq B$ obținem $\mathbf{C} \leq \mathbf{B}$. \square

Fie \mathbf{A} o algebră și $X \subseteq A$. Considerăm închiderea familiei X la operațiile algebrelor \mathbf{A} , fie aceasta $\langle X \rangle_{\mathbf{A}}$. Evident, $\langle X \rangle_{\mathbf{A}}$ este închisă în \mathbf{A} și ea este intersecția tuturor subfamilierilor $B \subseteq A$ ce includ X și sunt închise în \mathbf{A} . De fapt, putem scrie

$$\langle X \rangle_{\mathbf{A}} = \bigcup_{m \geq 0} B^m,$$

unde

- $B^0 = X$;
- $B^{m+1} = B^m \cup (\bigcup_{w \in S^*, \sigma \in \Sigma_{w,s}} \sigma^A(B^m) | s \in S)$, pentru orice $m \geq 0$.

Observăm că B^1 conține toate constantele algebrelor \mathbf{A} .

Pentru orice $s \in S$ și $a \in \langle X \rangle_{\mathbf{A}}, s$, elementul a admite cel puțin o construcție inductivă de la X și Σ^A ,

$$a_0, a_1, \dots, a_k = a,$$

unde, pentru orice $i \leq k$, are loc una din următoarele proprietăți:

- există un sort s_i astfel încât $a_i \in X_{s_i}$, sau
- există $i_1, \dots, i_n < i$ și $\sigma \in \Sigma_{s_{i_1} \dots s_{i_n}, s_i}$ astfel încât $a_{i_j} \in \langle X \rangle_{\mathbf{A}}, s_{i_j}$, pentru orice $1 \leq j \leq n$, și $a_i = \sigma^A(a_{i_1}, \dots, a_{i_n})$ (în cazul $s_1 \dots s_n = \lambda$ înțelegem că $a_i = \sigma^A$). În plus, dacă $i = k$, atunci $s_i = s$.

Conform Observației 8.3.1(3), $\langle X \rangle_{\mathbf{A}}$ definește o subalgebră a algebrei \mathbf{A} , notată $\langle X \rangle_{\mathbf{A}} = (\langle X \rangle_{\mathbf{A}}, \Sigma^{\langle X \rangle_{\mathbf{A}}})$ și numită *subalgebra generată de X*. Ea este intersecția tuturor subalgebrelor algebrei \mathbf{A} ce includ X și, deci, este cea mai mică subalgebră (în sensul incluziunii) a algebrei \mathbf{A} ce include X .

Direct de la această discuție obținem:

Propoziția 8.3.2. Fie \mathbf{A} o algebră. Atunci, pentru orice subfamilie $X \subseteq A$ are loc $\langle X \rangle_{\mathbf{A}} = \langle (X_s \cup \Sigma_s^A | s \in S) \rangle_{\mathbf{A}}$.

Propoziția 8.3.3. Fie \mathbf{A} și \mathbf{B} algebrelle și $X, Y \subseteq A$.

- (1) Dacă $X \subseteq Y$, atunci $\langle X \rangle_{\mathbf{A}} \subseteq \langle Y \rangle_{\mathbf{A}}$, și deci $\langle X \rangle_{\mathbf{A}} \leq \langle Y \rangle_{\mathbf{A}}$.
- (2) Dacă $\mathbf{B} \leq \mathbf{A}$ și $X \subseteq B$, atunci $\langle X \rangle_{\mathbf{A}} \subseteq B$, și deci $\langle X \rangle_{\mathbf{A}} \leq \mathbf{B}$.

Demonstrație. (1) urmează direct de la definițiile conceptelor de închidere și subalgebră, iar (2) urmează de la (1) considerând $Y = B$ și remarcând că B este închisă în \mathbf{A} . \square

Un caz important de subalgebră generată de o parte X a sa este cel în care $\langle X \rangle_{\mathbf{A}} = A$. În acest caz spunem că \mathbf{A} este *generată de X*.

Evident, orice algebră \mathbf{A} este generată de o parte a sa (măcar de $X = A$), iar Propoziția 8.3.2 ne spune că \mathbf{A} este generată de familia vidă dacă și numai dacă este generată de familia multimilor de constante ale ei.

Propoziția 8.3.4. Multimea $Sub(\mathbf{A})$ a tuturor subalgebrelor unei algebrelle \mathbf{A} formează latice completă în raport cu incluziunea.

Demonstrație. Intersecția oricărei familii nevide de subalgebre ale unei algebrelle \mathbf{A} este subalgebră a algebrei \mathbf{A} . Deoarece $\mathbf{A} \in Sub(\mathbf{A})$, obținem imediat că $Sub(\mathbf{A})$ este inf-semilatice completă ce are cel mai mare element. Atunci, în baza Teoremei 1.4.3.1, $Sub(\mathbf{A})$ este latice completă.⁵ \square

Principiul inducției structurale pentru multimi sortate definite inductiv (Secțiunea 8.1) poate fi utilizat și în cazul algebrelor universale.

Teorema 8.3.1. (Principiul inducției structurale pentru algebrelle universale) Fie \mathbf{A} o algebră generată de o subfamilie X a sa. Dacă $P = (P_s | s \in S)$ este o proprietate astfel încât:

- (1) $P_s(x)$, pentru orice $s \in S$ și $x \in X_s \cup \Sigma_s^A$;
- (2) $(P_{s_1}(a_1) \wedge \dots \wedge P_{s_n}(a_n)) \Rightarrow P_s(\sigma^A(a_1, \dots, a_n))$, pentru orice $(s_1 \dots s_n, s) \in S^+ \times S$, $\sigma \in \Sigma_{s_1 \dots s_n, s}$ și $(a_1, \dots, a_n) \in A_{s_1 \dots s_n}$,

⁵Propoziția 8.3.4 poate fi demonstrată și direct observând că supremumul unei familii nevide $(B^i | i \in I)$ de subalgebre ale algebrei \mathbf{A} este subalgebra $\langle \bigcup_{i \in I} B^i \rangle_{\mathbf{A}}$.

atunci $P_s(a)$, pentru orice $s \in S$ și $a \in A_s$.⁶

Definiția 8.3.3. O algebră \mathbf{A} este numită *minimală* dacă nu are subalgebre proprii.

Propoziția 8.3.5. O algebră \mathbf{A} este minimală dacă și numai dacă \mathbf{A} este generată de familia vidă (echivalent, dacă și numai dacă \mathbf{A} este generată de familia multimilor de constante ale ei).

Demonstrație. Dacă presupunem că \mathbf{A} este minimală dar nu poate fi generată de \emptyset , ci numai de subfamilii nevide X ce conțin și simboluri funcționale neconstante, atunci relația $\langle \emptyset \rangle_{\mathbf{A}} \subset \langle X \rangle_{\mathbf{A}} = A$ conduce la faptul că \mathbf{A} are subalgebre proprii, ceea ce constituie o contradicție.

Reciproc, dacă presupunem că \mathbf{A} este generată de familia vidă (echivalent, de familia constanțelor ei) dar are subalgebre proprii \mathbf{B} , cum orice subalgebră include familia multimilor de constante ale algebrei \mathbf{A} , urmează că închiderea familiei \mathbf{B} în \mathbf{A} este A , contrazicând faptul că \mathbf{B} este închisă în \mathbf{A} și $\mathbf{B} \subset A$. \square

8.4. Congruențe și algebrelle cât

8.4.1. Definiții. Exemple. Proprietăți de bază

Dată o algebră \mathbf{A} , vom nota prin $E(\mathbf{A})$ mulțimea tuturor relațiilor de echivalență pe suportul algebrei. Evident, $E(\mathbf{A})$ este latice completă, ceea ce se obține ca în Exemplul 1.4.3.3. O relație de echivalență pe o algebră ce ia în considerare și operațiile algebrei, nu numai suportul ei, va fi numită congruență. Înainte de a introduce acest concept, reamintim că dacă A este o mulțime, atunci relația vidă este relație de echivalență pe A dacă și numai dacă $A = \emptyset$. În plus, într-un astfel de caz, mulțimea cât \emptyset/\emptyset este \emptyset .

Definiția 8.4.1.1. Fie \mathbf{A} o algebră și $\rho = (\rho_s | s \in S)$ o relație de echivalență pe A . Spunem că ρ este *congruență* în algebră \mathbf{A} dacă pentru orice $(s_1 \dots s_n, s) \in S^+ \times S$ și $\sigma \in \Sigma_{s_1 \dots s_n, s}$ are loc

$$(\forall a_1, b_1, \dots, a_n, b_n)((\forall 1 \leq i \leq n)(a_i, b_i \in A_{s_i} \wedge a_i \rho_{s_i} b_i))$$

⁶În unele lucrări, cerințele (1) și (2) se pot întâlni și în varianta

- (1') $P_s(x)$, pentru orice $s \in S$ și $x \in X_s$;
- (2') $(P_{s_1}(a_1) \wedge \dots \wedge P_{s_n}(a_n)) \Rightarrow P_s(\sigma^A(a_1, \dots, a_n))$, pentru orice $(s_1 \dots s_n, s) \in S^* \times S$, $\sigma \in \Sigma_{s_1 \dots s_n, s}$ și $(a_1, \dots, a_n) \in A_{s_1 \dots s_n}$.

Într-o astfel de exprimare, la (2') trebuie înțeles că proprietatea P se cere a fi satisfăcută și de toate constantele algebrei \mathbf{A} . În varianta aleasă de noi, acest fapt este precizat clar la punctul (1).

$$\Rightarrow \sigma^A(a_1, \dots, a_n) \rho_s \sigma^A(b_1, \dots, b_n)).$$

Observația 8.4.1.1. Fie \mathbf{A} o Σ -algebră și ρ o congruență în \mathbf{A} .

- (1) Trebuie remarcat că proprietatea din Definiția 8.4.1.1 este implicit satisfăcută de funcții σ^A vide deoarece, într-un astfel de caz, cel puțin una din mulțimile A_{s_1}, \dots, A_{s_n} este vidă. Aceasta face ca formula

$$(\forall 1 \leq i \leq n)(a_i, b_i \in A_{s_i} \wedge a_i \rho_{s_i} b_i)$$

să fie falsă și, astfel, întreaga formulă să fie adevărată. Aceasta ne permite ca verificarea proprietății de congruență să fie întotdeauna făcută doar pentru funcții σ^A nevide, ceea ce vom face pe parcursul întregului capitol.

De asemenea, trebuie remarcat că are loc:

$$\sigma^A = \emptyset \Leftrightarrow (\exists 1 \leq i \leq n)(\rho_{s_i} = \emptyset) \Leftrightarrow (\exists 1 \leq i \leq n)(A_{s_i} = \emptyset).$$

- (2) Proprietatea din Definiția 8.4.1.1 spune că ρ este "compatibilă" cu operațiile algebrei. Această compatibilitate poate fi vizualizată ca în Figura 8.3.

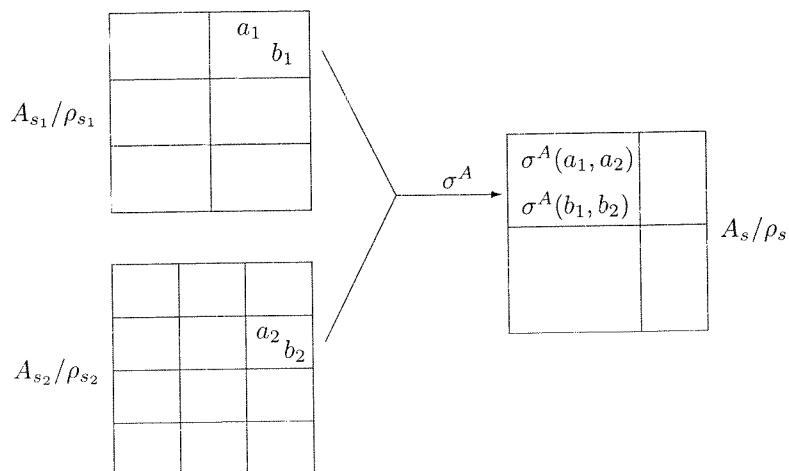


Figura 8.3: Compatibilitatea congruențelor cu operațiile algebrei

Vom nota prin $Con(\mathbf{A})$ mulțimea tuturor congruențelor algebrei \mathbf{A} . Este ușor de văzut că $\iota_A = (\iota_{A_s} | s \in S)$ și $A \times A = (A_s \times A_s | s \in S)$ sunt congruențe în \mathbf{A} , iar relația vidă $\emptyset = (\emptyset | s \in S)$ este congruență în \mathbf{A} doar dacă \mathbf{A} este algebra vidă (caz în care $\iota_A = \emptyset$). În plus, pentru orice $\rho \in Con(\mathbf{A})$ are loc:

$$\iota_A \subseteq \rho \subseteq A \times A.$$

Fie \mathbf{A} o algebră și $\rho \in Con(\mathbf{A})$. Considerăm familia

$$A/\rho = (A_s/\rho_s | s \in S)$$

pe care o înzestrăm cu o structură de Σ -algebră considerând, pentru fiecare tip (w, s) și $\sigma \in \Sigma$, funcția $\sigma^{A/\rho}$ dată prin:

- $\sigma^{A/\rho} = [\sigma^A]_{\rho_s}$, dacă $w = \lambda$;
- $\sigma^{A/\rho}([a_1]_{\rho_{s_1}}, \dots, [a_n]_{\rho_{s_n}}) = [\sigma^A(a_1, \dots, a_n)]_{\rho_s}$, pentru orice $a_1 \in A_{s_1}, \dots, a_n \in A_{s_n}$, dacă $w = s_1 \dots s_n \in S^+$ și σ^A este nevidă;
- $\sigma^{A/\rho} = \emptyset$, dacă $\sigma^A = \emptyset$ (dacă σ^A este funcția vidă atunci cel puțin una din mulțimile ce intervin în construcția domeniului ei este vidă și, deci, mulțimea cât corespunzătoare este mulțimea vidă ceea ce face ca $\sigma^{A/\rho}$ să fie funcția vidă).

Definiția funcțiilor $\sigma^{A/\rho}$ nu depinde de reprezentanții de clasă. În adevăr, dacă $a_i \rho_{s_i} b_i$, pentru orice $1 \leq i \leq n$, faptul că ρ este congruență conduce la

$$\sigma^A(a_1, \dots, a_n) \rho_s \sigma^A(b_1, \dots, b_n)$$

și, deci, $[\sigma^A(a_1, \dots, a_n)]_{\rho_s} = [\sigma^A(b_1, \dots, b_n)]_{\rho_s}$, de unde urmează imediat

$$\sigma^{A/\rho}([a_1]_{\rho_{s_1}}, \dots, [a_n]_{\rho_{s_n}}) = \sigma^{A/\rho}([b_1]_{\rho_{s_1}}, \dots, [b_n]_{\rho_{s_n}}).$$

Ca urmare, dacă notăm $\Sigma^{A/\rho} = (\Sigma_{w,s}^{A/\rho} | (w, s) \in S^* \times S)$, unde $\Sigma_{w,s}^{A/\rho} = \{\sigma^{A/\rho} | \sigma \in \Sigma_{w,s}\}$, pentru orice $(w, s) \in S^* \times S$, atunci cuplul $(A/\rho, \Sigma^{A/\rho})$ este Σ -algebră. Această algebră va fi notată prin \mathbf{A}/ρ și va fi numită *algebra cât sau factor indușă de \mathbf{A} și ρ* .

Exemplul 8.4.1.1. Fie $\mathbf{G} = (G, \circ, ', e)$ un grup (pe care îl privim ca algebră universală). Atunci au loc următoarele proprietăți:

- (1) Pentru orice congruență $\rho \in Con(\mathbf{G})$, $[e]_\rho$ este subgrup normal al lui \mathbf{G} . În plus,

$$a \rho b \Leftrightarrow a \circ b' \in [e]_\rho,$$

pentru orice $a, b \in G$.

În adevăr, pentru orice $x \in G$ și $a \in [e]_\rho$ au loc relațiile

$$\begin{aligned} a \in [e]_\rho &\Leftrightarrow a \rho e \\ &\Leftrightarrow a \rho (x' \circ e \circ x) \\ &\Leftrightarrow a \in x' \circ [e]_\rho \circ x, \end{aligned}$$

care conduc la $[e]_\rho = x' \circ [e]_\rho \circ x$, ceea ce ne arată că $[e]_\rho$ este subgrup normal al grupului \mathbf{G} .

Echivalența " $a \rho b \Leftrightarrow a \circ b' \in [e]_\rho$ " se obține cu ușurință pe baza faptului că ρ este congruență:

$$\begin{aligned} a \rho b &\Leftrightarrow (a \circ b') \rho (b \circ b') \\ &\Leftrightarrow (a \circ b') \rho e \\ &\Leftrightarrow a \circ b' \in [e]_\rho. \end{aligned}$$

(2) Pentru orice subgrup normal H al grupului \mathbf{G} , relația ρ dată prin

$$a \rho b \Leftrightarrow a \circ b' \in H,$$

pentru orice $a, b \in G$, este congruență în \mathbf{G} (a se vedea Secțiunea 4.3). În plus, $H = [e]_\rho$. În adevăr:

$$\begin{aligned} a \in H &\Leftrightarrow a \circ e' \in H \\ &\Leftrightarrow a \rho e \\ &\Leftrightarrow a \in [e]_\rho. \end{aligned}$$

Că urmare a acestor proprietăți putem spune că funcția ce asociază unei congruențe ρ subgrupul normal $[e]_\rho$ stabilește o bijecție între $Con(\mathbf{G})$ și mulțimea subgrupurilor normale ale grupului \mathbf{G} . Vom justifica doar injectivitatea acestei funcții. Fie ρ_1 și ρ_2 două congruențe. Presupunem că are loc $[e]_{\rho_1} = [e]_{\rho_2}$. Atunci,

$$\begin{aligned} a \rho_1 b &\Leftrightarrow a \circ b' \in [e]_{\rho_1} \\ &\Leftrightarrow a \circ b' \in [e]_{\rho_2} \\ &\Leftrightarrow a \rho_2 b, \end{aligned}$$

ceea ce arată că $\rho_1 = \rho_2$ și, astfel, stabilește injectivitatea funcției.

Propoziția 8.4.1.1. Pentru orice algebră \mathbf{A} , $Con(\mathbf{A})$ este sublatice completă a laticei complete $E(\mathbf{A})$.

Demonstrație. Evident, $Con(\mathbf{A}) \subseteq E(\mathbf{A})$. În plus:

- $A \times A \in Con(\mathbf{A})$, și
- pentru orice submulțime nevidă $\mathcal{C} \subseteq Con(\mathbf{A})$, $\bigcap \mathcal{C}$ este congruență în \mathbf{A} .

Atunci, conform Teoremei 1.4.3.1, $Con(\mathbf{A})$ este latice completă. \square

Propoziția 8.4.1.2. Fie \mathbf{A} și \mathbf{B} algebri și $\rho \in Con(\mathbf{A})$. Dacă $\mathbf{B} \leq \mathbf{A}$, atunci $\rho|_B \in Con(\mathbf{B})$.

Demonstrație. Afirmația din propoziție urmează cu ușurință de la faptul că $\rho|_B$ este echivalentă pe B , B este închisă în \mathbf{A} și ρ este congruență în \mathbf{A} . \square

Dată o algebră \mathbf{A} , $B \subseteq A$ și $\rho \in Con(\mathbf{A})$, vom nota prin B^ρ familia

$$B_s^\rho = \{a \in A_s | [a]_{\rho_s} \cap B_s \neq \emptyset\},$$

pentru orice $s \in S$. Figura 8.4 reprezintă grafic modul de construcție al mulțimii B_s^ρ . Pătratele punctate sunt clase de echivalență ce intersectează B_s (reprezentată ca un pătrat desenat cu linie continuă). Reunind toate clasele de echivalență se obține B_s^ρ . Evident, dacă $A_s = \emptyset$, atunci $B_s^\rho = \emptyset$.

Propoziția 8.4.1.3. Fie \mathbf{A} o algebră, $B \subseteq A$ și $\rho \in Con(\mathbf{A})$. Dacă B este închisă în \mathbf{A} , atunci B^ρ este închisă în \mathbf{A} .

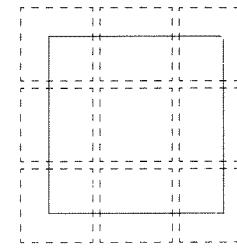


Figura 8.4: Construcția mulțimii B_s^ρ

Demonstrație. Fie $(s_1 \cdots s_n, s) \in S^+ \times S$, $\sigma \in \Sigma_{s_1 \cdots s_n, s}$ și $a_i \in B_{s_i}^\rho$, pentru orice $1 \leq i \leq n$. Atunci, $[a_i]_{\rho_{s_i}} \cap B_{s_i} \neq \emptyset$. Fie $a'_i \in [a_i]_{\rho_{s_i}} \cap B_{s_i}$, pentru orice $1 \leq i \leq n$.

Deoarece B este închisă în \mathbf{A} , urmează $\sigma^A(a'_1, \dots, a'_n) \in B_s$. Pe de altă parte,

$$\sigma^A(a_1, \dots, a_s) \in [\sigma^A(a'_1, \dots, a'_n)]_{\rho_s},$$

ceea ce arată că $\sigma^A(a_1, \dots, a_s) \in B_s^\rho$. Adăugând la aceasta și faptul că B^ρ conține toate constantele algebrei \mathbf{A} deoarece B le conține, obținem că B^ρ este închisă în algebra \mathbf{A} . \square

Atunci când B^ρ este închisă în \mathbf{A} , subalgebra generată va fi notată prin \mathbf{B}^ρ .

Propoziția 8.4.1.4. Fie \mathbf{A} o algebră, $X \subseteq A$ și $\rho \in Con(\mathbf{A})$. Dacă \mathbf{A} este generată de X , atunci \mathbf{A}/ρ este generată de X/ρ , unde $X/\rho = (X_s/\rho_s | s \in S)$ și $X_s/\rho_s = \{[a]_{\rho_s} | a \in X_s\}$, pentru orice $s \in S$.

Demonstrație. Este ușor de arătat că au loc următoarele proprietăți:

- orice element ce are o construcție inductivă de la X/ρ și $\Sigma^{A/\rho}$ este în A/ρ , și
- orice element din A/ρ are o construcție inductivă de la X/ρ și $\Sigma^{A/\rho}$.

De aici urmează afirmația din propoziție. \square

Atragem atenția asupra faptului că X/ρ în Propoziția 8.4.1.4 nu este, în mod necesar, familia cât indușă de X și ρ .

Reamintim că dacă $\rho, \theta \in E(A)$ și $\rho \subseteq \theta$, unde A este o mulțime, atunci $\theta/\rho \in E(A/\rho)$ (a se vedea Propoziția 1.2.2.2(1)).

Extindem definiția relației θ/ρ și pentru cazul în care ρ și θ sunt relații de echivalență pe familii S -sortate de mulțimi. Extensia se face, evident, pe componente. Adică, $\theta/\rho = (\theta_s/\rho_s | s \in S)$. Toate proprietățile stabilite în Propoziția 1.2.2.2 se transferă și la astfel de relații.

Propoziția 8.4.1.5. Fie \mathbf{A} o algebră și ρ, θ, θ_1 și θ_2 congruențe în \mathbf{A} ce satisfac $\rho \subseteq \theta \cap \theta_1 \cap \theta_2$. Atunci au loc următoarele proprietăți:

- (1) θ/ρ este congruență în \mathbf{A}/ρ ;
- (2) orice congruență în \mathbf{A}/ρ este de forma θ'/ρ , unde $\theta' \in Con(\mathbf{A})$ și $\rho \subseteq \theta'$;
- (3) $\rho/\rho = \iota_{A/\rho}$;
- (4) $A^2/\rho = (A/\rho)^2$ (A^2 este relația binară $A \times A$ care, evident, include ρ);
- (5) $\theta_1 \subset \theta_2$ dacă și numai dacă $\theta_1/\rho \subset \theta_2/\rho$;
- (6) $\theta_1 \neq \theta_2$ dacă și numai dacă $\theta_1/\rho \neq \theta_2/\rho$.

Demonstrație. (1) Conform Propoziției 1.2.2.2(1), ne rămâne de arătat că, pentru orice $(s_1 \cdots s_n, s) \in S^+ \times S$, $\sigma \in \Sigma_{s_1 \cdots s_n, s}$ și $[a_i]_{\rho_{s_i}}, [b_i]_{\rho_{s_i}} \in A_{s_i}/\rho_{s_i}$ cu $[a_i]_{\rho_{s_i}} \theta_{s_i} / \rho_{s_i} [b_i]_{\rho_{s_i}}$, pentru orice $1 \leq i \leq n$, are loc

$$\sigma^{A/\rho}([a_1]_{\rho_{s_1}}, \dots, [a_n]_{\rho_{s_n}}) \theta_s / \rho_s \sigma^{A/\rho}([b_1]_{\rho_{s_1}}, \dots, [b_n]_{\rho_{s_n}}).$$

Însă relațiile $[a_i]_{\rho_{s_i}} \theta_{s_i} / \rho_{s_i} [b_i]_{\rho_{s_i}}$ sunt echivalente cu $a_i \theta_{s_i} b_i$, pentru orice i , de unde urmează

$$\sigma^A(a_1, \dots, a_n) \theta_s / \rho_s \sigma^A(b_1, \dots, b_n),$$

ca urmare a faptului că θ este congruență în algebra \mathbf{A} . Dar atunci definiția relației θ/ρ conduce la

$$[\sigma^A(a_1, \dots, a_n)]_{\rho_s} \theta_s / \rho_s [\sigma^A(b_1, \dots, b_n)]_{\rho_s},$$

de la care deducem

$$\sigma^{A/\rho}([a_1]_{\rho_{s_1}}, \dots, [a_n]_{\rho_{s_n}}) \theta_s / \rho_s \sigma^{A/\rho}([b_1]_{\rho_{s_1}}, \dots, [b_n]_{\rho_{s_n}}).$$

Deci $\theta/\rho \in Con(\mathbf{A}/\rho)$.

(2) se obține ca în Propoziția 1.2.2.2(2), arătând suplimentar că θ' este congruență în \mathbf{A} , iar (3), (4), (5) și (6) urmează de la Propoziția 1.2.2.2 și de la faptul că congruențele în \mathbf{A} sunt cazuri particulare de echivalențe pe A . \square

Următorul corolar, cunoscut sub denumirea de *Teorema de corespondență*, prezintă un rezultat cu multiple aplicații. Menționăm întâi că intervalul dintre două congruențe se definește în mod ușual ca fiind intervalul dintre două elemente într-o mulțime parțial ordonată (a se vedea Definiția 1.4.1.3(4)).

Corolarul 8.4.1.1. (Teorema de corespondență)

Fie \mathbf{A} o algebră și $\rho \in Con(\mathbf{A})$. Atunci funcția

$$h : [\rho, A^2] \rightarrow Con(\mathbf{A}/\rho)$$

dată prin

$$h(\theta) = \theta/\rho,$$

pentru orice $\theta \in [\rho, A^2]$, este izomorfism de latici $([\rho, A^2], \subseteq)$ fiind sublattice a laticei $Con(\mathbf{A})$.

Demonstrație. Surjectivitatea urmează de la Propoziția 8.4.1.5(2), injectivitatea de la Propoziția 8.4.1.5(6), iar compatibilitatea funcției h cu incluziunea de la Propoziția 8.4.1.5(5). \square

Izomorfismul din Corolarul 8.4.1.1 poate fi vizualizat grafic ca în Figura 8.5.

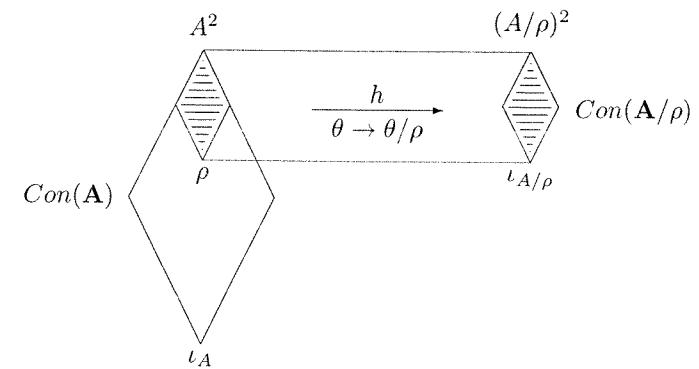


Figura 8.5: Reprezentare grafică a Teoremei de corespondență

Definiția 8.4.1.2. O algebră \mathbf{A} este numită *simplă* dacă singurele ei congruențe sunt ι_A și A^2 .

Evident, algebrelor triviale sunt simple.

Propoziția 8.4.1.6. Fie \mathbf{A} o algebră și $\rho \in Con(\mathbf{A})$. \mathbf{A}/ρ este simplă dacă și numai dacă $\rho = A^2$ sau ρ este maximală în $Con(\mathbf{A}) - A^2$.

Demonstrație. Să presupunem întâi că \mathbf{A}/ρ este simplă. Atunci singurele ei congruențe sunt $\iota_{A/\rho}$ și $(A/\rho)^2$.

Evident, are loc $\rho = A^2$ sau $\rho \neq A^2$. Dacă presupunem că $\rho \neq A^2$ și ρ nu este maximală în $Con(\mathbf{A}) - A^2$, atunci există $\theta \in Con(\mathbf{A})$ astfel încât $\rho \subset \theta \subset A^2$. De la aceasta, în baza Propoziției 8.4.1.5, urmează

$$\rho/\rho \subset \theta/\rho \subset A^2/\rho,$$

adică

$$\iota_{A/\rho} \subset \theta/\rho \subset (A/\rho)^2,$$

ceea ce arată că \mathbf{A}/ρ nu este simplă, ceea ce constituie o contradicție.

Reciproc, dacă $\rho = A^2$, atunci \mathbf{A}/ρ este trivială, și deci este simplă.

Presupunem acum că ρ este maximală în $Con(\mathbf{A}) - A^2$ dar \mathbf{A}/ρ nu este simplă. Atunci există $\bar{\theta} \in Con(\mathbf{A}/\rho)$ astfel încât $\iota_{A/\rho} \subset \bar{\theta} \subset (A/\rho)^2$. Propoziția 8.4.1.5(2) ne spune că există o congruență θ în \mathbf{A} astfel încât $\bar{\theta} = \theta/\rho$, iar inclusiunile de mai sus, în baza aceleiași propoziții, conduc la $\rho \subset \theta \subset A^2$. Deci ρ nu este maximală în $Con(\mathbf{A}) - A^2$, ceea ce constituie o contradicție. \square

Vom încheia secțiunea printr-un rezultat important ce face legătura între congruențe și subalgebrelor. Orice congruență într-o algebră \mathbf{A} este o subfamilie a familiei $A \times A$. Este natural să ne întrebă dacă nu cumva această subfamilie definește o subalgebră a unei algebrelor cu suportul $A \times A$, și reciproc.

Teorema 8.4.1.1. Pentru orice Σ -algebră \mathbf{A} există o Σ' -algebră \mathbf{B} cu suportul $A \times A$ astfel încât există o corespondență bijectivă între $Con(\mathbf{A})$ și $Sub(\mathbf{B})$.

Demonstrație. Fie $B = A \times A$. Considerăm o signatură Σ' și definim Σ'^B astfel (elementele signaturii Σ' vor fi deduse din cele ce urmează):

- $\sigma^B((a_1, b_1), \dots, (a_n, b_n)) = (\sigma^A(a_1, \dots, a_n), \sigma^A(b_1, \dots, b_n))$, pentru orice $\sigma \in \Sigma$ de tip $(s_1 \cdots s_n, s) \in S^+ \times S$ și orice $a_i, b_i \in A_{s_i}$, $1 \leq i \leq n$ (dacă $\sigma^A = \emptyset$, atunci $\sigma^B = \emptyset$);
- $\sigma_{s,a}^B = (a, a)$, pentru orice $s \in S$ și $a \in A_s$ (dacă $A_s \neq \emptyset$);
- $\alpha_s^B((a, b)) = (b, a)$, pentru orice $s \in S$ și $a, b \in A_s$ (dacă $A_s \neq \emptyset$);
- $\beta_s^B((a, b), (c, d)) = \begin{cases} (a, d), & \text{dacă } b = c \\ (a, b), & \text{altfel,} \end{cases}$ pentru orice $s \in S$ și $a, b, c, d \in A_s$ (dacă $A_s \neq \emptyset$).

Este clar că $\mathbf{B} = (B, \Sigma'^B)$ este o algebră. Vom arăta că, pentru orice $X \subseteq B$, X este închisă în \mathbf{B} dacă și numai dacă $X \in Con(\mathbf{A})$, ceea ce va încheia demonstrația teoremei.

Fie $X \subseteq B$ închisă în \mathbf{B} . Deoarece X este închisă la $\sigma_{s,a}^B$, α_s^B și β_s^B , pentru orice $\sigma \in \Sigma$, $s \in S$ și $a \in A_s$, deducem că X este echivalentă pe A , iar faptul că X este închisă la σ^B , pentru orice $\sigma \in \Sigma$ de tip $(w, s) \in S^+ \times S$, conduce la $X \in Con(\mathbf{A})$.

Reciproc, demonstrația decurge similar celei de mai sus. \square

8.4.2. Congruențe principale

Dacă \mathbf{A} este o algebră și $X = (X_s | s \in S) \subseteq A \times A$, atunci vom nota prin

$$\equiv_{\mathbf{A}}^X = (\equiv_{\mathbf{A}}^{X_s} | s \in S)$$

congruență (pe \mathbf{A}) generată de X . Evident, $\equiv_{\mathbf{A}}^X$ este cea mai mică congruență pe \mathbf{A} ce include X sau, cu alte cuvinte, $\equiv_{\mathbf{A}}^X$ este intersecția tuturor congruențelor pe \mathbf{A} ce includ X . Dacă X are proprietatea $|X_s| = 1$, pentru orice $s \in S$ cu $A_s \neq \emptyset$, atunci $\equiv_{\mathbf{A}}^X$ va fi numită *congruență principală*.

Propoziția 8.4.2.1. Fie \mathbf{A} o algebră. Atunci au loc următoarele proprietăți:

- (1) $(\equiv_{\mathbf{A}}^{X_s} | s \in S) = (\equiv_{\mathbf{A}}^{Y_s} | s \in S)$, pentru orice X și Y cu proprietatea $|X_s| = 1$, $|Y_s| = 1$ și, dacă $X_s = \{(a, b)\}$ atunci $Y_s \subseteq \{(a, b), (b, a)\}$, pentru orice $s \in S$ cu $A_s \neq \emptyset$;
- (2) $(\equiv_{\mathbf{A}}^{X_s} | s \in S) = sup(\{(\equiv_{\mathbf{A}}^{Y_s} | s \in S) | (\forall s \in S)(Y_s \subseteq X_s \wedge |Y_s| \leq 1)\})$, pentru orice $X \subseteq A$ cu proprietatea că X_s este finită pentru orice $s \in S$;
- (3) $\rho = sup(\{(\equiv_{\mathbf{A}}^{Y_s} | s \in S) | (\forall s \in S)(Y_s \subseteq \rho_s \wedge |Y_s| \leq 1)\})$, pentru orice $\rho \in Con(\mathbf{A})$ (supremumul este considerat în laticea completă a congruențelor algebrei \mathbf{A}).

Demonstrație. (1) urmează direct de la proprietatea de simetrie pe care trebuie să o satisfacă orice relație de echivalență.

(2) Fie $Y_s \subseteq X_s$ cu $|Y_s| \leq 1$, pentru orice $s \in S$. Cum

$$(\equiv_{\mathbf{A}}^{Y_s} | s \in S) \subseteq (\equiv_{\mathbf{A}}^{X_s} | s \in S),$$

urmează că

$$sup(\{(\equiv_{\mathbf{A}}^{Y_s} | s \in S) | (\forall s \in S)(Y_s \subseteq X_s \wedge |Y_s| \leq 1)\}) \subseteq (\equiv_{\mathbf{A}}^{X_s} | s \in S).$$

Pe altă parte, $X \subseteq sup(\{(\equiv_{\mathbf{A}}^{Y_s} | s \in S) | (\forall s \in S)(Y_s \subseteq X_s \wedge |Y_s| \leq 1)\})$, ceea ce arată că

$$(\equiv_{\mathbf{A}}^{X_s} | s \in S) \subseteq sup(\{(\equiv_{\mathbf{A}}^{Y_s} | s \in S) | (\forall s \in S)(Y_s \subseteq X_s \wedge |Y_s| \leq 1)\})$$

(deoarece $sup(\{(\equiv_{\mathbf{A}}^{Y_s} | s \in S) | (\forall s \in S)(Y_s \subseteq X_s \wedge |Y_s| \leq 1)\})$ este închisă, fiind congruență). Am demonstrat astfel (2).

(3) Au loc relațiile:

$$\rho \subseteq sup(\{(\equiv_{\mathbf{A}}^{Y_s} | s \in S) | (\forall s \in S)(Y_s \subseteq \rho_s \wedge |Y_s| \leq 1)\}) \subseteq \rho.$$

Ca urmare, are loc (3). \square

8.5. Homomorfisme de algebrelor

8.5.1. Definiții. Exemple. Proprietăți de bază

Înainte de a introduce conceptul de homomorfism de algebrelor, reamintim că funcția vidă este funcție de la A la B doar dacă A este mulțimea vidă. Funcția vidă de la \emptyset

la B este injectivă; ea este surjectivă (deci și bijectivă) doar dacă $B = \emptyset$.

Definiția 8.5.1.1. Fie \mathbf{A} și \mathbf{B} algebrelle. O funcție $h : A \rightarrow B$ este numită *homomorfism de la \mathbf{A} la \mathbf{B}* dacă are loc

$$(\forall a)(a \in A_w \Rightarrow h_s(\sigma^A(a)) = \sigma^B(h_w(a))),$$

pentru orice $(w, s) \in S^* \times S$ și $\sigma \in \Sigma_{w,s}$.

Observația 8.5.1.1. Fie \mathbf{A}, \mathbf{B} și h ca în Definiția 8.5.1.1.

- (1) Fie $(s_1 \cdots s_n, s) \in S^+ \times S$ și $\sigma \in \Sigma_{s_1 \cdots s_n, s}$. Dacă cel puțin una din mulțimile A_{s_1}, \dots, A_{s_n} este multimea vidă, atunci formula " $a \in A_{s_1 \cdots s_n}$ " este falsă ceea ce face ca întreaga formulă să fie adevărată. Deci proprietatea din Definiția 8.5.1.1 este implicit satisfăcută dacă σ^A este funcția vidă. Aceasta ne permite ca verificarea proprietății de homomorfism să fie întotdeauna făcută doar pentru funcții σ^A nevide, ceea ce vom face pe parcursul întregii lucrări.

Remarcăm, de asemenea, că are loc:

$$(\exists 1 \leq i \leq n)(h_{s_i} = \emptyset) \Leftrightarrow (\exists 1 \leq i \leq n)(A_{s_i} = \emptyset) \Leftrightarrow \sigma^A = \emptyset$$

și

$$h_s = \emptyset \Leftrightarrow A_s = \emptyset \Rightarrow \sigma^A = \emptyset.$$

Ca urmare, dacă A_{s_1}, \dots, A_{s_n} sunt nevide, atunci A_s este nevidă (deoarece σ^A este funcție de la $A_{s_1 \cdots s_n}$ la A_s) și, deci, $B_{s_1}, \dots, B_{s_n}, B_s$ sunt nevide (deoarece h este funcție de la A la B). Ca urmare, relația " $h_s(\sigma^A(a)) = \sigma^B(h_w(a))$ " din Definiția 8.5.1.1 este consistentă.

- (2) Pentru orice sort s și constantă $\sigma^A \in \Sigma_s^A$ are loc

$$h_s(\sigma^A(\emptyset)) = \sigma^B(h_\lambda(\emptyset)) = \sigma^B(\emptyset)$$

(ultima egalitate are loc ca urmare a convenție acceptată în Secțiunea 8.1). Această relație ne spune că homomorfismele "păstrează" constantele. Cum pentru orice σ și algebră \mathbf{A} am convenit să notăm $\sigma^A(\emptyset)$ prin σ^A , relația de mai sus se poate simplifica la

$$h_s(\sigma^A) = \sigma^B.$$

- (3) Proprietatea din Definiția 8.5.1.1 poate fi simplu exprimată prin "diagrama din Figura 8.6 comută". Ea exprimă *compatibilitatea* homomorfismului h cu operațiile celor două algebrelle.

Vom nota prin $\text{Hom}(\mathbf{A}, \mathbf{B})$ mulțimea tuturor homomorfismelor de la \mathbf{A} la \mathbf{B} .

În mod ușual, homomorfismele injective sunt numite *monomorfisme* sau *scufundări*, cele surjective, *epimorfisme*, iar cele bijective, *izomorfisme*. Homomorfismele

$$\begin{array}{ccc} A_w & \xrightarrow{\sigma^A} & A_s \\ h_w \downarrow & & \downarrow h_s \\ B_w & \xrightarrow{\sigma^B} & B_s \end{array}$$

Figura 8.6: Proprietatea de homomorfism

(izomorfisme) de la o algebră \mathbf{A} la ea însăși sunt numite *endomorfisme (automorfisme)* iar mulțimea lor este notată prin $\text{End}(\mathbf{A})$ ($\text{Aut}(\mathbf{A})$). Este clar că are loc

$$\text{Aut}(\mathbf{A}) \subseteq \text{End}(\mathbf{A}) \subseteq M(\mathbf{A}),$$

unde $M(\mathbf{A})$ reprezintă mulțimea tuturor funcțiilor de la A la A (a se compara cu notația din Exemplul 3.1.2). Funcția identică pe A este automorfism al algebrei \mathbf{A} .

Două algebrelle \mathbf{A} și \mathbf{B} se numesc *izomorfe* dacă există un izomorfism h între ele și, într-un astfel de caz, vom nota $\mathbf{A} \cong \mathbf{B}$ (sau $\mathbf{A} \stackrel{h}{\cong} \mathbf{B}$, dacă dorim să punem în evidență izomorfismul h).

Exemplul 8.5.1.1. Homomorfismele de semigrupuri, monoizi, grupuri etc. întâlnite în capitolile anterioare sunt cazuri particulare de homomorfisme de algebrelle universale.

Următoarea propoziție, a cărei demonstrație este lăsată în seama cititorului, prezintă câteva simple proprietăți ale homomorfismelor.

Propoziția 8.5.1.1. Fie \mathbf{A}, \mathbf{B} și \mathbf{C} algebrelle.

- (1) Compunere de homomorfisme (izomorfisme) este homomorfism (izomorfism).
- (2) Dacă f este izomorfism de la \mathbf{A} la \mathbf{B} , atunci f^{-1} este izomorfism de la \mathbf{B} la \mathbf{A} .
- (3) f este izomorfism de la \mathbf{A} la \mathbf{B} dacă și numai dacă există un homomorfism g de la \mathbf{B} la \mathbf{A} astfel încât $f \circ g = 1_B$ și $g \circ f = 1_A$.
- (4) Dacă h este homomorfism (monomorfism) de la \mathbf{A} la \mathbf{B} și $\mathbf{C} \leq \mathbf{A}$, atunci $h|_C$ este homomorfism (monomorfism) de la \mathbf{C} la \mathbf{B} .

Propoziția 8.5.1.2. Fie \mathbf{A} și \mathbf{B} algebrelle, f și g homomorfisme de la \mathbf{A} la \mathbf{B} , $X \subseteq A$ și $Y \subseteq B$.

- (1) Dacă X este închisă în \mathbf{A} , atunci $f(X)$ este închisă în \mathbf{B} .
- (2) Dacă Y este închisă în \mathbf{B} , atunci $f^{-1}(Y)$ este închisă în \mathbf{A} .
- (3) Familia

$$eq(f, g) = (\{a \in A_s | f_s(a) = g_s(a)\} | s \in S)$$

este închisă în \mathbf{A} .

Demonstrație. Vom demonstra doar (1). Pentru orice $(w, s) \in S^* \times S$, $\sigma \in \Sigma_{w,s}$ și $x \in X_w$ are loc

$$\sigma^B(f_w(x)) = f_s(\sigma^A(x)) \in f_s(X_s),$$

ceea ce arată că $f(X)$ este închisă în \mathbf{B} . \square

Fie $f : A \rightarrow B$ un homomorfism de la algebra \mathbf{A} la algebra \mathbf{B} . Conform Propoziției 8.5.1.2(1), $f(A)$ este închisă în \mathbf{B} și, deci, definește o subalgebră a algebrei \mathbf{B} . Această subalgebră va fi numită *imaginăa homomorfă* a algebrei \mathbf{A} prin f și va fi notată prin $f(\mathbf{A})$. Similar, $f^{-1}(B)$ definește o subalgebră a algebrei \mathbf{A} . Această subalgebră va fi numită *imaginăa homomorfă inversă* a algebrei \mathbf{B} prin f și va fi notată prin $f^{-1}(\mathbf{B})$.

Familia $eq(f, g)$ din Propoziția 8.5.1.2(3) poartă denumirea de *familia sau multi-mea de egalitate a homomorfismelor* f și g .

Corolarul 8.5.1.1. Fie $f : A \rightarrow B$ un homomorfism de la algebra \mathbf{A} la algebra \mathbf{B} .

- (1) Dacă $\mathbf{C} \leq \mathbf{A}$, atunci $f(\mathbf{C}) \leq \mathbf{B}$.
- (2) Dacă $\mathbf{C} \leq \mathbf{B}$, atunci $f^{-1}(\mathbf{C}) \leq \mathbf{A}$.

Corolarul 8.5.1.2. Fie \mathbf{A} și \mathbf{B} algebrelle, f și g homomorfisme de la \mathbf{A} la \mathbf{B} și $X \subseteq A$. Dacă $f|_X = g|_X$ atunci $f|_{\langle X \rangle_A} = g|_{\langle X \rangle_A}$.

Demonstrație. Dacă $f|_X = g|_X$, atunci $X \subseteq eq(f, g)$. Cum $eq(f, g)$ este închisă în \mathbf{A} (Propoziția 8.5.1.2(3)), urmează $\langle X \rangle_A \subseteq eq(f, g)$, ceea ce conduce la $f|_{\langle X \rangle_A} = g|_{\langle X \rangle_A}$. \square

Direct de la Corolarul 8.5.1.2 obținem:

Corolarul 8.5.1.3. Fie \mathbf{A} și \mathbf{B} algebrelle, f și g homomorfisme de la \mathbf{A} la \mathbf{B} și $X \subseteq A$. Dacă X generează \mathbf{A} și $f|_X = g|_X$, atunci $f = g$.

Corolarul 8.5.1.3 ne spune că dacă X generează \mathbf{A} și o funcție $f : X \rightarrow B$ se poate extinde la un homomorfism $f' : A \rightarrow B$, atunci această extensie este unică. De cele mai multe ori f' se notează tot prin f .

Corolarul 8.5.1.4. Dacă \mathbf{A} este o Σ -algebră ce poate fi generată de \emptyset , atunci unicul endomorfism al ei este 1_A .

Demonstrație. Orice endomorfism f al unei algebrelle \mathbf{A} ce poate fi generată de \emptyset coincide cu endomorfismul 1_A pe \emptyset . Ca urmare, $f = 1_A$ (Corolarul 8.5.1.3). \square

Corolarul 8.5.1.5. Pentru orice Σ -algebră minimală \mathbf{A} are loc

$$End(\mathbf{A}) = Aut(\mathbf{A}) = \{1_A\}.$$

Demonstrație. Algebrele minimale sunt generate de \emptyset (Propoziția 8.3.5). Ca urmare, corolarul urmează de la Corolarul 8.5.1.4. \square

Corolarul 8.5.1.6. Fie \mathbf{A} și \mathbf{B} două Σ -algebrelle minimale. Dacă există două homomorfismele $f : A \rightarrow B$ și $g : B \rightarrow A$, atunci f și g sunt izomorfisme inverse unul altuiu.

Demonstrație. $g \circ f$ este endomorfism al algebrei \mathbf{A} , iar $f \circ g$ este endomorfism al algebrei \mathbf{B} . Conform Corolarului 8.5.1.5, $g \circ f = 1_A$ și $f \circ g = 1_B$. Deci, f și g sunt izomorfisme inverse unul altuiu (Propoziția 8.5.1.1(3)). \square

Corolarul 8.5.1.7. Fie \mathbf{A} și \mathbf{B} algebrelle, $f : A \rightarrow B$ un homomorfism și $X \subseteq A$. Atunci $f(\langle X \rangle_A) = \langle f(X) \rangle_B$.

Demonstrație. Incluziunea $f(X) \subseteq f(\langle X \rangle_A)$, în baza faptului că $f(\langle X \rangle_A)$ este închisă în \mathbf{B} (Propoziția 8.5.1.2(1)), conduce la $\langle f(X) \rangle_B \subseteq f(\langle X \rangle_A)$.

Incluziunea $f(X) \subseteq \langle f(X) \rangle_B$ conduce la $X \subseteq f^{-1}(\langle f(X) \rangle_B)$ de la care, în baza faptului că $f^{-1}(\langle f(X) \rangle_B)$ este închisă în \mathbf{A} (Propoziția 8.5.1.2(2)), obținem $\langle X \rangle_A \subseteq f^{-1}(\langle f(X) \rangle_B)$ și, deci, $f(\langle X \rangle_A) \subseteq \langle f(X) \rangle_B$.

Combinând cele două incluziuni obținute mai sus, deducem proprietatea din enunțul corolarului. \square

Observația 8.5.1.2. Corolarul 8.5.1.7 poate fi demonstrat și direct pornind de la relația $\langle X \rangle_A = \bigcup_{n \geq 0} X^n$, unde familiile X^n sunt definite ca în Secțiunea 8.3. Prin inducție matematică se arată că are loc $f(X^m) \subseteq \langle f(X) \rangle_B$, pentru orice $m \geq 0$. Ca urmare,

$$f(\langle X \rangle_A) = f\left(\bigcup_{m \geq 0} X^m\right) = \bigcup_{m \geq 0} f(X^m) \subseteq \langle f(X) \rangle_B.$$

Reciproc, cum $\langle X \rangle_A$ este închisă în \mathbf{A} și f este homomorfism, $f(\langle X \rangle_A)$ este închisă în \mathbf{B} . Dar atunci, $f(X) \subseteq f(\langle X \rangle_A)$ conduce la $\langle f(X) \rangle_B \subseteq f(\langle X \rangle_A)$.

Corolarul 8.5.1.8. Fie \mathbf{A} și \mathbf{B} algebrelle, $f : A \rightarrow B$ un homomorfism și $X \subseteq A$. Dacă X generează \mathbf{A} , atunci $f(X)$ generează $f(\mathbf{A})$.

Demonstrație. Corolarul 8.5.1.7 și faptul că X generează \mathbf{A} conduc la $\langle f(X) \rangle_B = f(\langle X \rangle_A) = f(\mathbf{A})$. \square

Teorema 8.5.1.1. (Teorema de descompunere a homomorfismelor)
Fie $h : A \rightarrow B$ un homomorfism de la algebra \mathbf{A} la algebra \mathbf{B} . Atunci există o algebră \mathbf{C} , un epimorfism $f : A \rightarrow C$ și un monomorfism $g : C \rightarrow B$ astfel încât $h = g \circ f$. În plus, pentru orice algebră \mathbf{C}' , epimorfism $f' : A \rightarrow C'$ și monomorfism $g' : C' \rightarrow B$ astfel încât $h = g' \circ f'$, există un unic homomorfism $d : C \rightarrow C'$ astfel încât $f' = d \circ f$ și $g = g' \circ d$ (a se vedea diagrama din Figura 8.7).

Demonstrație. În baza Teoremei 1.2.3.1, aplicate pentru fiecare sort în parte, avem:

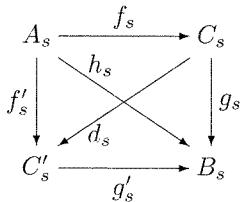


Figura 8.7: Descompunerea homomorfismelor

- funcția $f : A \rightarrow C$ dată prin $f_s(a) = h_s(a)$, pentru orice sort s și $a \in A_s$, unde $C = h(A)$, este surjectivă;
- funcția $g : C \rightarrow B$ dată prin $g_s(c) = c$, pentru orice sort s și $c \in C_s$, este injectivă;
- $h = g \circ f$;
- dacă C' , f' și g' sunt ca în enunțul teoremei, atunci funcția $d : C \rightarrow C'$ dată prin $d_s(c) = f'_s(a)$, unde $f_s(a) = c$, pentru orice $s \in S$ și $c \in C_s$, este unică funcție ce satisface $f' = d \circ f$ și $g = g' \circ d$.

Ceea ce ne rămâne de arătat este că f , g și d sunt homomorfisme. Aceasta este imediat pentru f și g . Pentru d , fie $(w, s) \in S^* \times S$, $\sigma \in \Sigma_{w,s}$ și $c \in C_w$. Avem de arătat că are loc:

$$d_s(\sigma^C(c)) = \sigma^{C'}(d_w(c)).$$

Conform definiției funcției d are loc:

$$d_s(\sigma^C(c)) = f'_s(a), \text{ unde } a \in A_s \text{ și } f_s(a) = \sigma^C(c)$$

și

$$\sigma^{C'}(d_w(c)) = \sigma^{C'}(f'_w(a)), \text{ unde } a \in A_w \text{ și } f_w(a) = c.$$

Obținem atunci:

$$\begin{aligned} \sigma^{C'}(d_w(c)) &= \sigma^{C'}(f'_w(a)) \\ &= f'_s(\sigma^A(a)) \\ &= d_s(\sigma^C(f_w(a))) \\ &= d_s(\sigma^C(c)) \end{aligned}$$

(a treia egalitate urmează în baza faptului că $f_s(\sigma^A(a)) = \sigma^C(f_w(a))$). Am arătat astfel că d este homomorfism. \square

8.5.2. Structura algebrică a mulțimilor $End(\mathbf{A})$ și $Aut(\mathbf{A})$

Prin $M(\mathbf{A})$ am notat mulțimea tuturor funcțiilor de la algebra \mathbf{A} la ea însăși. Împreună cu operația de compunere, aceasta formează monoid. În plus, au loc următoarele proprietăți:

Propoziția 8.5.2.1. Fie \mathbf{A} o algebră.

- (1) Mulțimea endomorfismelor algebrei \mathbf{A} formează monoid în raport cu compunerea (unitatea fiind 1_A), ce este submonoid al monoidului $M(\mathbf{A})$.
- (2) Mulțimea automorfismelor algebrei \mathbf{A} formează grup în raport cu compunerea (unitatea fiind 1_A).

Monoidul (grupul) $End(\mathbf{A})$ ($Aut(\mathbf{A})$) se numește *monoidul endomorfismelor* (*grupul automorfismelor*) algebrei \mathbf{A} .

Propoziția 8.5.2.1 admite și o reciprocă, în sensul următor.

Teorema 8.5.2.1. Orice monoid este izomorf cu monoidul endomorfismelor unei algebrelor unisortate⁷.

Demonstrație. Fie $\mathbf{M} = (A, \cdot, e)$ un monoid. Definim algebra unisortată $\mathbf{A} = (A, \Sigma^A)$, unde $\Sigma_1 = \{\sigma_a | a \in A\}$, $\Sigma_n = \emptyset$, pentru orice $n \neq 1$, și

$$\sigma_a^A(x) = x \cdot a,$$

pentru orice $a, x \in A$ (semnul operației \cdot va fi omis în cele ce urmează).

Vom arăta că $\mathbf{M} \cong End(\mathbf{A})$. Pentru aceasta vom arăta întâi că are loc

- (i) $End(\mathbf{A}) = \{f_a | a \in A\}$, unde $f_a : A \rightarrow A$ este dată prin $f_a(x) = ax$, pentru orice $a, x \in A$.

Pentru orice $a, b, x \in A$ are loc

$$f_a(\sigma_b^A(x)) = f_a(xb) = axb = \sigma_b^A(ax) = \sigma_b^A(f_a(x)),$$

ceea ce arată că $f_a \in End(\mathbf{A})$, pentru orice $a \in A$.

Fie acum $f \in End(\mathbf{A})$. Considerăm $a = f(e)$ și obținem

$$f(x) = f(\sigma_x^A(e)) = \sigma_x^A(f(e)) = \sigma_x^A(a) = ax = f_a(x),$$

pentru orice $x \in A$, ceea ce arată că orice endomorfism al algebrei \mathbf{A} este de forma f_a cu $a \in A$. Am arătat astfel că are loc (i).

Un simplu exercițiu ne arată că au loc și următoarele proprietăți:

- (ii) pentru orice $a, b \in A$, $f_a = f_b$ dacă și numai dacă $a = b$;
- (iii) pentru orice $a, b \in A$, $f_a \circ f_b = f_{a \cdot b}$;

⁷Acest rezultat apare pentru prima dată independent, într-o versiune preliminară a celei de a treia ediții a lucrării [13] (semnat de A.G. Waterman) și în notele nepublicate ale lui G. Grätzer, "Some Results on Universal Algebras", din 1962 (a se vedea [77], pag. 68). Prima demonstrație publicată este datorată lui M. Armbrust și J. Schmidt [2]. Ideea demonstrației este însă conținută în lucrarea [11] a lui Birkhoff.

Implicit, acest rezultat poate fi găsit și în [97], iar pe de altă parte el este caz particular al Lemei lui Yoneda din teoria categoriilor [232].

(iv) $f_e = 1_A$.

Fie acum $\varphi : A \rightarrow End(\mathbf{A})$ dată prin $\varphi(a) = f_a$, pentru orice $a \in A$. Relațiile (i) și (ii) ne spun că φ este bijecție, iar (iii) și (iv), că φ este homomorfism. Ca urmare, φ este izomorfism. \square

Corolarul 8.5.2.1. Pentru orice grup \mathbf{G} există o algebră \mathbf{A} astfel încât

$$\mathbf{G} \cong Aut(\mathbf{A}) = End(\mathbf{A}).$$

Demonstrație. Fie $\mathbf{G} = (G, \cdot', e)$ un grup. \mathbf{G} este și monoid și, utilizând construcția din Teorema 8.5.2.1, obținem $\mathbf{G} \cong End(\mathbf{A})$. Ceea ce ne mai rămâne de arătat este că orice endomorfism al acestei algebrelor este și automorfism.

Fie $f \in End(\mathbf{A})$. Atunci, există $a \in A$ astfel încât $f = f_a$. Au loc relațiile

$$f_a \circ f_{a'} = f_{aa'} = f_e = 1_A = f_e = f_{a'a} = f_{a'} \circ f_a,$$

care arată că f_a este bijecție și, deci, automorfism. \square

8.5.3. Homomorfisme și congruențe

Știm că funcțiile injective păstrează relațiile de echivalență (Propoziția 1.2.2.1). Acest rezultat poate fi extins la congruențe în algebrelor. În primul rând, dacă ρ este o relație pe $A = (A_s | s \in S)$ și $f : A \rightarrow B$ este o funcție, atunci vom nota prin $f(\rho)$ relația $f(\rho) = (f_s(\rho_s) | s \in S)$, unde $f_s(\rho_s) = \{(f_s(a), f_s(b)) | (a, b) \in \rho_s\}$, pentru orice $s \in S$. Propoziția 1.2.2.1 poate fi reformulată și pentru cazul relațiilor sortate.

Propoziția 8.5.3.1. Fie \mathbf{A} și \mathbf{B} două algebrelor și $f : A \rightarrow B$ un monomorfism. Atunci, pentru orice congruență ρ în \mathbf{A} , $f(\rho)$ este congruență în $f(\mathbf{A})$.

Demonstrație. Conform Propoziției 1.2.2.1, aplicată pe sorturi, ceea ce ne rămâne de arătat este că $f(\rho)$ este compatibilă cu operațiile algebrei $f(\mathbf{A})$.

Fie $(s_1 \cdots s_n, s) \in S^+ \times S$, $\sigma \in \Sigma_{s_1 \cdots s_n, s}$ și $(x_i, y_i) \in f_{s_i}(\rho_{s_i})$, pentru orice $1 \leq i \leq n$. Atunci există $(a_i, b_i) \in \rho_{s_i}$ cu $f_{s_i}(a_i) = x_i$ și $f_{s_i}(b_i) = y_i$, pentru orice $1 \leq i \leq n$. Cum ρ este congruență în \mathbf{A} , deducem că are loc $\sigma^A(a_1, \dots, a_n) \rho_s \sigma^A(b_1, \dots, b_n)$, de unde urmează

$$f_s(\sigma^A(a_1, \dots, a_n)) f_s(\rho_s) f_s(\sigma^A(b_1, \dots, b_n)).$$

Folosind faptul că f este homomorfism, deducem

$$\sigma^B(f_{s_1}(a_1), \dots, f_{s_n}(a_n)) f_s(\rho_s) \sigma^B(f_{s_1}(b_1), \dots, f_{s_n}(b_n)),$$

care este echivalentă cu $\sigma^B(x_1, \dots, x_n) f_s(\rho_s) \sigma^B(y_1, \dots, y_n)$. Dar această ultimă relație ne arată că $\sigma^{f(A)}(x_1, \dots, x_n) f_s(\rho_s) \sigma^{f(A)}(y_1, \dots, y_n)$. Deci, $f(\rho)$ este compatibilă cu operațiile algebrei $f(\mathbf{A})$. \square

Remarkăm, în demonstrația Propoziției 8.5.3.1, că injectivitatea funcției f este utilizată doar pentru a asigura că $f(\rho)$ este relație de echivalență (a se vedea Propoziția 1.2.2.1).

Corolarul 8.5.3.1. Fie \mathbf{A} și \mathbf{B} două algebrelor și $f : A \rightarrow B$ un izomorfism. Atunci, pentru orice congruență ρ în \mathbf{A} , $f(\rho)$ este congruență în \mathbf{B} .

Este ușor de văzut că are loc și reciproca Corolarului 8.5.3.1. Adică, orice congruență în \mathbf{B} este de forma $f(\rho)$, unde ρ este o congruență în \mathbf{A} .

Fie $f : A \rightarrow B$ o funcție S -sortată. Nucleul acestei funcții, notat $ker(f)$, este definit prin $ker(f) = (ker(f_s) | s \in S)$, unde $ker(f_s) = \{(a, b) | f_s(a) = f_s(b)\}$, pentru orice $s \in S$. Următoarea propoziție este imediat de verificat.

Propoziția 8.5.3.2. Pentru orice algebrelor \mathbf{A} și \mathbf{B} și homomorfism $h : A \rightarrow B$, $ker(h) \in Con(\mathbf{A})$.

Fie \mathbf{A} o algebră și $\rho \in Con(\mathbf{A})$. Vom nota prin $f_\rho : A \rightarrow A/\rho$ funcția $(f_\rho)_s(a) = [a]_{\rho_s}$, pentru orice $s \in S$ și $a \in A_s$. Este clar că f_ρ este bine definită. Mai mult, se arată cu ușurință că are loc:

Propoziția 8.5.3.3. Pentru orice algebră \mathbf{A} și congruență $\rho \in Con(\mathbf{A})$, f_ρ este epimorfism.

f_ρ este numită *epimorfismul natural induș de ρ* . Proprietatea de universalitate a mulțimii cât (Teorema 1.2.3.2) poate fi reformulată pentru algebrelor, astfel.

Teorema 8.5.3.1. (Proprietatea de universalitate a algebrei cât)

Pentru orice algebrelor \mathbf{A} și \mathbf{B} , homomorfism $f : A \rightarrow B$ și congruență ρ în \mathbf{A} ce satisfacă $\rho \subseteq ker(f)$, există un unic homomorfism $g : A/\rho \rightarrow B$ ce satisfacă $f = g \circ f_\rho$. În plus, dacă $\rho = ker(f)$, atunci g este monomorfism, iar dacă f este epimorfism, atunci g este epimorfism.

Demonstrație. Ca în demonstrația Teoremei 1.2.3.2, pe componente, se arată că funcția g dată prin $g_s([a]_{\rho_s}) = f_s(a)$, pentru orice $s \in S$ și $a \in A_s$, satisfac teorema. Noi ne vom limita la a arăta doar că g este homomorfism.

Fie $\sigma \in \Sigma_{s_1 \cdots s_n, s}$ și $a_i \in A_{s_i}$, pentru orice $1 \leq i \leq n$. Atunci

$$\begin{aligned} g_s(\sigma^{A/\rho}([a_1]_{\rho_{s_1}}, \dots, [a_n]_{\rho_{s_n}})) &= g_s([\sigma^A(a_1, \dots, a_n)]_{\rho_s}) \\ &= f_s(\sigma^A(a_1, \dots, a_n)) \\ &= \sigma^B(f_{s_1}(a_1), \dots, f_{s_n}(a_n)) \\ &= \sigma^B(g_{s_1}([a_1]_{\rho_{s_1}}), \dots, g_{s_n}([a_n]_{\rho_{s_n}})) \end{aligned}$$

(în cazul $\sigma \in \Sigma_s$, $g_s(\sigma^{A/\rho}) = g_s([\sigma^A]_{\rho_s}) = f_s(\sigma^A) = \sigma^B$). Ca urmare, g este homomorfism. \square

Teorema 8.5.3.1 are consecințe similare Teoremei 1.2.3.2. Una dintre acestea va fi prezentată în secțiunea următoare sub forma “primei teoreme de izomorfism”.

Fie \mathbf{A} o algebră și $\rho \in Con(\mathbf{A})$. Dacă $\rho = \iota_A$, atunci $[a]_{\rho_s} \cap [b]_{\rho_s} = \emptyset$, pentru orice $s \in S$ și $a, b \in A_s$ cu $a \neq b$. Ca urmare, funcția f_ρ este injectivă, ceea ce conduce, prin Propoziția 8.5.3.3, la faptul că f_ρ este izomorfism. Deci, $\mathbf{A} \cong \mathbf{A}/\rho$. În cazul în care algebra \mathbf{A} este minimală are loc și reciprocă.

Propoziția 8.5.3.4. Fie \mathbf{A} o algebră minimală și $\rho \in Con(\mathbf{A})$. Dacă $\mathbf{A} \cong \mathbf{A}/\rho$, atunci $\rho = \iota_A$.

Demonstrație. Dacă \mathbf{A} este minimală, atunci ea este generată de constantele ei. Pentru orice izomorfism $f : A \rightarrow A/\rho$, f și f_ρ coincid pe constante. Ca urmare, Corolarul 8.5.1.3 conduce la faptul că $f = f_\rho$, ceea ce înseamnă că f_ρ este funcție injectivă. Dar atunci ρ nu poate fi decât ι_A . \square

8.5.4. Teoreme de izomorfism

Teoremele de izomorfism pe care le prezentăm în această secțiune generalizează teoremele de izomorfism întâlnite în teoria grupurilor (Sectiunea 4.3) și inelelor (Sectiunea 5.2).

Teorema 8.5.4.1. (Prima teoremă de izomorfism⁸⁾

Fie \mathbf{A} și \mathbf{B} două algebrelor și $f : A \rightarrow B$ un epimorfism. Atunci există un izomorfism $h : A/\ker(f) \rightarrow B$ astfel încât $f = h \circ f_{\ker(f)}$.

Demonstrație. Direct de la Teorema 8.5.3.1. \square

Teorema 8.5.4.1 poate fi vizualizată grafic ca în Figura 8.8. “Celulele” mulțimii A reprezintă submulțimile maximale de elemente ce au aceeași imagine prin f (de fapt, ele sunt clase de echivalență în raport cu $\ker(f)$). Echivalența $\ker(f)$ “compactifică” aceste submulțimi, producând $A/\ker(f)$ ce poate fi pusă în corespondență bijectivă cu B .

Exemplul 8.5.4.1. Prima teoremă de izomorfism din teoria grupurilor este caz particular al Teoremei 8.5.4.1. În adevăr, considerând algebrelor \mathbf{A} și \mathbf{B} ca fiind grupurile G_1 și G_2 , obținem $G_1/\ker(f) \cong G_2$, pentru orice epimorfism f de la G_1 la G_2 .

Teorema 8.5.4.2. (A doua teoremă de izomorfism)

Fie \mathbf{A} o algebră și $\rho, \theta \in Con(\mathbf{A})$ astfel încât $\rho \subseteq \theta$. Atunci $(\mathbf{A}/\rho)/(\theta/\rho) \cong \mathbf{A}/\theta$.

⁸Este întâlnită, în mod ușual, și sub denumirea de *Teorema de homomorfism* sau cu enunțul simplificat “Dacă $f : \mathbf{A} \rightarrow \mathbf{B}$ este un epimorfism de algebrelor, atunci $\mathbf{A}/\ker(f) \cong \mathbf{B}$ ”.

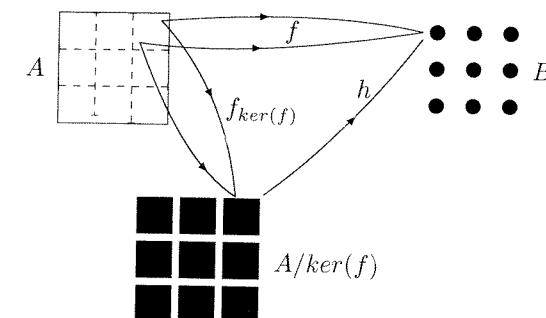


Figura 8.8: Reprezentare grafică a primei teoreme de izomorfism

Demonstrație. Funcția h dată prin $h_s([a]_{\rho_s}/\theta_s/\rho_s) = [a]_{\theta_s}$, pentru orice $s \in S$ și $a \in A_s$, stabilește un izomorfism între $(\mathbf{A}/\rho)/(\theta/\rho)$ și \mathbf{A}/θ . \square

A doua teoremă de izomorfism poate fi vizualizată grafic ca în diagrama din Figura 8.9. Clasele de echivalență în raport cu ρ sunt reprezentate grafic prin linie punctată, iar clasele de echivalență în raport cu θ/ρ și în raport cu θ sunt reprezentate grafic prin linie continuă.

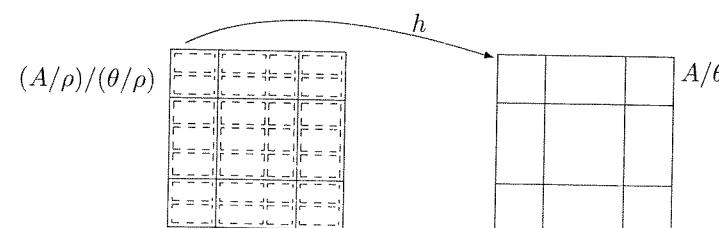


Figura 8.9: Reprezentare grafică a celei de a doua teoreme de izomorfism

Exemplul 8.5.4.2. A treia teoremă de izomorfism din teoria grupurilor este caz particular al Teoremei 8.5.4.2. În adevăr, să considerăm algebra \mathbf{A} ca fiind grupul G , ρ ca fiind congruența indușă de subgrupul normal N_1 , și θ ca fiind congruența indușă de subgrupul normal N_2 . Deoarece N_1 este subgrup al lui N_2 , obținem $\rho \subseteq \theta$. Atunci, Teorema 8.5.4.2 conduce la $(\mathbf{A}/\rho)/(\theta/\rho) \cong A/\theta$, care este de fapt a treia teoremă de izomorfism din teoria grupurilor.

Teorema 8.5.4.3. (A treia teoremă de izomorfism)

Fie \mathbf{A} o algebră, $\mathbf{B} \leq \mathbf{A}$ și $\rho \in Con(\mathbf{A})$. Atunci $\mathbf{B}/(\rho|_{\mathbf{B}}) \cong \mathbf{B}^\rho/(\rho|_{\mathbf{B}^\rho})$.

Demonstrație. Funcția h dată prin $h_s([b]_{\rho_s|_{B_s}}) = [b]_{\rho_s|_{B_s^\rho}}$, pentru orice $s \in S$ și $b \in B$, satisface teorema. \square

A treia teoremă de izomorfism poate fi vizualizată grafic ca în diagrama din Figura 8.10. Clasele de echivalență în raport cu $\rho|_B$ sunt reprezentate grafic prin linie continuă, iar clasele de echivalență în raport cu $\rho|_{B^\rho}$ sunt reprezentate grafic prin linie punctată (acestea fiind peste mulțimea B^ρ).

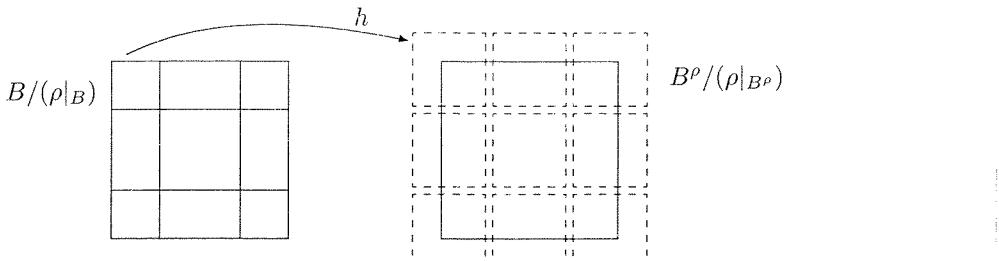


Figura 8.10: Reprezentare grafică a celei de a treia teoreme de izomorfism

Exemplul 8.5.4.3. A doua teoremă de izomorfism din teoria grupurilor este caz particular al Teoremei 8.5.4.3. În adevăr, să considerăm algebra **A** ca fiind grupul G , **B** ca fiind subgrupul H al lui G , și ρ ca fiind congruența indușă de subgrupul normal N . Atunci $N \cap H$ este chiar $\rho|_B$, iar NH este B^ρ . Conform Teoremei 8.5.4.3, $\mathbf{B}/(\rho|_B) \cong B^\rho/(\rho|_{B^\rho})$, care este a doua teoremă de izomorfism din teoria grupurilor.

8.6. Produse de algebrelor

Una dintre construcțiile de bază în teoria algebrelor universale este cea de produs de algebrelor.

8.6.1. Produse directe de algebrelor

Produsul cartezian al mulțimilor suport a două algebrelor universale poate fi înzestrat cu o structură de algebră universală, într-un mod foarte natural.

Definiția 8.6.1.1. *Produsul (cartezian)* a două algebrelor **A** și **B**, notat $\mathbf{A} \times \mathbf{B}$, este algebra $\mathbf{A} \times \mathbf{B} = (A \times B, \Sigma^{A \times B})$ unde:

(1) $A \times B = ((A \times B)_s | s \in S)$ și $(A \times B)_s = A_s \times B_s$, pentru orice $s \in S$;

(2) $\Sigma^{A \times B} = \{\sigma^{A \times B} | \sigma \in \Sigma\}$;

(3) pentru orice $(w, s) \in S^* \times S$, $\sigma \in \Sigma_{w,s}$ și $c \in (A \times B)_w$, are loc

$$\sigma^{A \times B}(c) = (\sigma^A(pr_{1,w}(c)), \sigma^B(pr_{2,w}(c))),$$

unde pr_1 și pr_2 sunt familiile de funcții proiecție asociate produsului $A \times B$.

Remarcăm că, pentru orice simbol constantă σ , $\sigma^{A \times B} = (\sigma^A, \sigma^B)$.

Propoziția 8.6.1.1. Fie **A** și **B** algebrelor. Atunci au loc următoarele proprietăți:

- (1) pr_1 și pr_2 sunt homomorfisme de la $\mathbf{A} \times \mathbf{B}$ la **A** și, respectiv, **B**. Dacă $A \times B \neq \emptyset$, atunci pr_1 și pr_2 sunt epimorfisme;
- (2) pentru orice algebră **C** și homomorfisme $f : C \rightarrow A$ și $g : C \rightarrow B$, există un unic homomorfism $h : C \rightarrow A \times B$ astfel încât $f = pr_1 \circ h$ și $g = pr_2 \circ h$;
- (3) funcțiile $f : A \rightarrow A \times A$ și $g : B \rightarrow B \times B$ date prin $f_s(a) = (a, a)$ și $g_s(b) = (b, b)$, pentru orice $s \in S$, $a \in A_s$ și $b \in B_s$, sunt monomorfisme.

Demonstrație. Vom demonstra (1) și (2), (3) rămânând în seama cititorului.

(1) Pentru orice $(w, s) \in S^* \times S$, $\sigma \in \Sigma_{w,s}$ și $c \in (A \times B)_w$, are loc

$$pr_{1,s}(\sigma^{A \times B}(c)) = pr_{1,s}(\sigma^A(pr_{1,w}(c)), \sigma^B(pr_{2,w}(c))) = \sigma^A(pr_{1,w}(c)),$$

ceea ce arată că pr_1 este homomorfism. În mod similar se arată că pr_2 este homomorfism. Evident, dacă $A \times B \neq \emptyset$, atunci pr_1 și pr_2 sunt surjecții.

(2) Definim h prin

$$h_s(c) = (f_s(c), g_s(c)),$$

pentru orice sort s și $c \in C_s$ (dacă $C_s = \emptyset$, atunci $h_s = \emptyset$). Se verifică imediat că are loc $f = pr_1 \circ h$ și $g = pr_2 \circ h$.

Arătăm că h este homomorfism. Fie $(w, s) \in S^* \times S$, $\sigma \in \Sigma_{w,s}$ și $c \in C_w$. Atunci:

$$\begin{aligned} h_s(\sigma^C(c)) &= (f_s(\sigma^C(c)), g_s(\sigma^C(c))) \\ &= (\sigma^A(f_w(c)), \sigma^B(g_w(c))) \\ &= \sigma^{A \times B}(f_w(c), g_w(c)) \\ &= \sigma^{A \times B}(h_w(c)), \end{aligned}$$

ceea ce arată că h este homomorfism.

Unicitatea homomorfismului h este imediată: dacă presupunem că h' este un alt homomorfism ce verifică proprietatea din propoziție, atunci relațiile

$$pr_1 \circ h = f = pr_1 \circ h' \text{ și } pr_2 \circ h = g = pr_2 \circ h'$$

conduc imediat la $h = h'$. \square

Proprietatea din Propoziția 8.6.1.1(2), numită *proprietatea de universalitate a produsului de algebrelor*, poate fi simplu exprimată prin “pentru orice algebră \mathbf{C} și homomorfisme $f : C \rightarrow A$ și $g : C \rightarrow B$, există un unic homomorfism $h : C \rightarrow A \times B$ astfel încât diagrama din Figura 8.11 comută”.

$$\begin{array}{ccccc} & & A \times B & & \\ & \swarrow f & \downarrow h & \searrow g & \\ C & & & & B \end{array}$$

Figura 8.11: Proprietatea de universalitate a produsului de algebrelor

Evident, produsul cartezian a două algebrelor poate fi generalizat, în mod natural, la $n \geq 2$ algebrelor. În cazul în care $\mathbf{A}_1 = \dots = \mathbf{A}_n = \mathbf{A}$, produsul $\mathbf{A}_1 \times \dots \times \mathbf{A}_n$ se notează simplificat prin \mathbf{A}^n și este numit *puterea a n-a a algebrelor* \mathbf{A} . Uzual, definim \mathbf{A}^1 prin \mathbf{A} .

Demonstrația următoarei propoziții rămâne în seama cititorului.

Propoziția 8.6.1.2. Fie $\mathbf{A}, \mathbf{A}', \mathbf{B}, \mathbf{B}'$ și \mathbf{C} algebrelor. Atunci au loc următoarele proprietăți:

- (1) $\mathbf{A} \times \mathbf{B} \cong \mathbf{B} \times \mathbf{A}$;
- (2) $\mathbf{A} \times (\mathbf{B} \times \mathbf{C}) \cong (\mathbf{A} \times \mathbf{B}) \times \mathbf{C} \cong \mathbf{A} \times \mathbf{B} \times \mathbf{C}$;
- (3) dacă $\mathbf{A} \cong \mathbf{A}'$ și $\mathbf{B} \cong \mathbf{B}'$, atunci $\mathbf{A} \times \mathbf{B} \cong \mathbf{A}' \times \mathbf{B}'$.

Generalizăm acum produsul cartezian al unui număr finit de algebrelor la o familie arbitrară nevidă $(\mathbf{A}^i | i \in I)$ de algebrelor. Reamintim întâi că familia $\prod_{i \in I} \mathbf{A}^i$ este definită ca fiind

$$\prod_{i \in I} \mathbf{A}^i = ((\prod_{i \in I} \mathbf{A}^i)_s | s \in S),$$

unde, pentru orice $s \in S$, $(\prod_{i \in I} \mathbf{A}^i)_s$ este mulțimea

$$(\prod_{i \in I} \mathbf{A}^i)_s = \{f | f : I \rightarrow \bigcup_{i \in I} \mathbf{A}_s^i \wedge (\forall i \in I)(f(i) \in \mathbf{A}_s^i)\} = \prod_{i \in I} \mathbf{A}_s^i$$

(dacă există $i \in I$ astfel încât $\mathbf{A}_s^i = \emptyset$, atunci $(\prod_{i \in I} \mathbf{A}^i)_s = \emptyset$).

Pentru $(w, s) \in S^* \times S$ și $\sigma \in \Sigma_{w,s}$ definim funcția

$$\sigma^{\prod_{i \in I} \mathbf{A}^i} : (\prod_{i \in I} \mathbf{A}^i)_w \rightarrow \prod_{i \in I} \mathbf{A}_s^i$$

prin

$$\sigma^{\prod_{i \in I} \mathbf{A}^i}(c)(j) = \begin{cases} \sigma^{A^j}, & \text{dacă } w = \lambda \\ \sigma^{A^j}(pr_{1,w}(c)(j), \dots, pr_{|w|,w}(c)(j)), & \text{altfel,} \end{cases}$$

pentru orice $c \in (\prod_{i \in I} \mathbf{A}^i)_w$ și $j \in I$.

Familia mulțimilor tuturor acestor funcții va fi notată prin $\Sigma^{\prod_{i \in I} \mathbf{A}^i}$. Este clar atunci că $(\prod_{i \in I} \mathbf{A}^i, \Sigma^{\prod_{i \in I} \mathbf{A}^i})$ este algebră.

Definiția 8.6.1.2. Numim *produsul direct* al familiei nevide de algebrelor $(\mathbf{A}^i | i \in I)$, algebră notată $\prod_{i \in I} \mathbf{A}^i$ și definită prin

$$\prod_{i \in I} \mathbf{A}^i = (\prod_{i \in I} \mathbf{A}^i, \Sigma^{\prod_{i \in I} \mathbf{A}^i}).$$

În cazul $\mathbf{A}^i = \mathbf{A}$, pentru orice $i \in I$, produsul direct al familiei $(\mathbf{A}^i | i \in I)$ este numit *puterea directă* a algebrelor \mathbf{A} , ce se mai notează prin \mathbf{A}^I .

Similar Propoziției 8.6.1.1 putem demonstra următoarele.

Propoziția 8.6.1.3. Fie $(\mathbf{A}^i | i \in I)$ o familie nevidă de algebrelor. Atunci au loc următoarele proprietăți:

- (1) pentru orice $i \in I$, pr_i este homomorfism de la algebra $\prod_{i \in I} \mathbf{A}^i$ la \mathbf{A}^i . Dacă $\prod_{i \in I} \mathbf{A}^i \neq \emptyset$, atunci pr_i este epimorfism, pentru orice $i \in I$;
- (2) pentru orice algebră \mathbf{C} și homomorfism $f^i : \mathbf{C} \rightarrow \mathbf{A}^i$, unde $i \in I$, există un unic homomorfism $h : \mathbf{C} \rightarrow \prod_{i \in I} \mathbf{A}^i$ astfel încât $f^i = pr_i \circ h$;
- (3) funcția $f^i : A^i \rightarrow \prod_{j \in I} A^{i,j}$ dată prin $f_s^i(a)(j) = a$, pentru orice $s \in S$, $j \in I$ și $a \in A_s^{i,j}$, unde $A^{i,j} = A^i$, pentru orice $j \in I$, este monomorfism.

Demonstrație. Ne vom limita doar la a defini homomorfismul h de la (2). Aceasta este dat prin $h_s(a)(i) = f_s^i(a)$, pentru orice $s \in S$, $a \in A_s$ și $i \in I$. \square

Proprietatea din Propoziția 8.6.1.3(2) este extensia *proprietății de universalitate a produsului de algebrelor* (Propoziția 8.6.1.1(2)) la o familie nevidă de algebrelor.

Observația 8.6.1.1. Homomorfismul h din demonstrația Propoziției 8.6.1.3(2) este monomorfism dacă și numai dacă $\bigcap_{i \in I} \ker(f^i) = \iota_A$.

În adevăr, să presupunem că $\bigcap_{i \in I} \ker(f^i) = \iota_A$. Dacă ar exista s și $a, b \in A_s$ astfel încât $a \neq b$ și $h_s(a) = h_s(b)$, atunci am obține $f_s^i(a) = f_s^i(b)$, pentru orice $i \in I$, ceea ce ar arăta că $(a, b) \in \bigcap_{i \in I} \ker(f^i)$, contrazicând ipoteza.

Reciproca se obține în manieră similară.

Fie $n \geq 1$ un număr natural și $I = \{1, \dots, n\}$. Pentru orice familie de multimi $(A^i | i \in I)$, funcția

$$h : \prod_{i \in I} A^i \rightarrow A^1 \times \dots \times A^n$$

dată prin $h(f) = (f(1), \dots, f(n))$, pentru orice $f \in \prod_{i \in I} A^i$, este bijecție. Atunci, obținem cu ușurință:

Propoziția 8.6.1.4. Fie $I = \{1, \dots, n\}$ și $(\mathbf{A}^i | i \in I)$ o familie de algebrelle, unde $n \geq 1$.

Atunci

$$\prod_{i \in I} \mathbf{A}^i \cong \mathbf{A}^1 \times \dots \times \mathbf{A}^n.$$

Următoarea propoziție poate fi demonstrată cu ușurință de cititor.

Propoziția 8.6.1.5. Fie $(\mathbf{A}^i | i \in I)$ și $(\mathbf{B}^i | i \in I)$ două familii nevide de algebrelle și $(h^i | i \in I)$ o familie de homomorfisme, unde h^i este de la \mathbf{A}^i la \mathbf{B}^i , pentru orice $i \in I$.

Atunci funcția $h : \prod_{i \in I} A^i \rightarrow \prod_{i \in I} B^i$ dată prin

$$h_s(f)(i) = h^i_s(f(i)),$$

pentru orice $s \in S$, $f \in \prod_{i \in I} A^i_s$ și $i \in I$, este homomorfism.

Ultimul rezultat al acestei secțiuni se referă la algebra cât indușă de un produs de congruențe.

Propoziția 8.6.1.6. Fie $(\mathbf{A}^i | i \in I)$ o familie nevidă de algebrelle și $(\rho^i | i \in I)$ o familie de congruențe, unde $\rho^i \in \text{Con}(\mathbf{A}^i)$, pentru orice $i \in I$. Atunci $\prod_{i \in I} \rho^i$ este congruență în algebra $\prod_{i \in I} \mathbf{A}^i$ și are loc

$$(\prod_{i \in I} \mathbf{A}^i) / (\prod_{i \in I} \rho^i) \cong \prod_{i \in I} (\mathbf{A}^i / \rho^i).$$

Demonstrație. Se verifică cu ușurință că $\prod_{i \in I} \rho^i \in \text{Con}(\prod_{i \in I} \mathbf{A}^i)$.

Familia $(h_s | s \in S)$ dată prin

$$h_s([f]_{\prod_{i \in I} \rho^i})(i) = [f(i)]_{\rho^i},$$

pentru orice $s \in S$, $f \in (\prod_{i \in I} A^i)_s$ și $i \in I$, stabilește un izomorfism între algebrelle $(\prod_{i \in I} \mathbf{A}^i) / (\prod_{i \in I} \rho^i)$ și $\prod_{i \in I} (\mathbf{A}^i / \rho^i)$. \square

8.6.2. Algebrelle decompozabile

Definiția 8.6.2.1. Spunem că o algebră \mathbf{A} este *decompozabilă* dacă există două algebrelle netriviale \mathbf{B} și \mathbf{C} astfel încât $\mathbf{A} \cong \mathbf{B} \times \mathbf{C}$. Altfel, \mathbf{A} este numită *idecompozabilă*.

Dacă algebra \mathbf{A} satisfacă $\mathbf{A} \cong \mathbf{B} \times \mathbf{C}$, atunci vom spune că perechea de algebrelle (\mathbf{B}, \mathbf{C}) formează o *descompunere* a algebrelui \mathbf{A} . Evident, putem vorbi de descompuneri ale unei algebrelle în mai mult de două algebrelle.

Observația 8.6.2.1. Deoarece algebrelle considerate de noi pot avea și multimi suport vide, aceasta face posibil ca algebrelle triviale să poată fi descompuse în algebrelle netriviale. De exemplu, dacă considerăm algebrelle date prin

	s_1	s_2	s_3	s_4
A	{a}	\emptyset	\emptyset	\emptyset
B	{a}	{b, c}	\emptyset	\emptyset
C	{a}	\emptyset	{b, c}	\emptyset

unde $b \neq c$ iar operațiile sunt $\sigma_1^A = \sigma_1^B = \sigma_1^C = a$ și $\sigma_2, \sigma_3 \in \Sigma_{s_2 s_3, s_4}$ cu

$$\sigma_2^A = \sigma_2^B = \sigma_2^C = \sigma_3^A = \sigma_3^B = \sigma_3^C = \emptyset,$$

atunci constatăm că \mathbf{A} este trivială, \mathbf{B} și \mathbf{C} sunt netriviale, dar $\mathbf{A} \cong \mathbf{B} \times \mathbf{C}$.

Observăm că descompuneri ca în Observația 8.6.2.1 pot apărea atunci când unul din domeniile algebrelui care se descompune este mulțimea vidă.

Definiția 8.6.2.2. Spunem că o descompunere $\mathbf{A} \cong \mathbf{B} \times \mathbf{C}$ a algebrelui \mathbf{A} este *impropriă* dacă există un sort s astfel încât $A_s = \emptyset$ și $B_s \cup C_s \neq \emptyset$. Dacă descompunerea $\mathbf{A} \cong \mathbf{B} \times \mathbf{C}$ nu este impropriă, atunci vom spune că este *propriă*.

Descompunerea algebrelui \mathbf{A} din Observația 8.6.2.1 este impropriă.

Lema 8.6.2.1. Dacă o algebră admite descompuneri, atunci admite descompuneri proprii.

Demonstrație. Fie \mathbf{A} o algebră și $\mathbf{A} \cong \mathbf{B} \times \mathbf{C}$ o descompunere a ei. Dacă această descompunere nu este proprie, atunci definim două algebrelle \mathbf{B}' și \mathbf{C}' astfel:

- pentru orice sort s , dacă $A_s = \emptyset$, atunci $B'_s = \emptyset = C'_s$; altfel, $B'_s = B_s$ și $C'_s = C_s$;
- pentru orice tip (w, s) și $\sigma \in \Sigma_{w, s}$, dacă $A_w = \emptyset$ sau $A_s = \emptyset$, atunci $\sigma^{B'} = \sigma^{C'} = \emptyset$; altfel, $\sigma^{B'} = \sigma^B$ și $\sigma^{C'} = \sigma^C$ (observăm că dacă $A_s = \emptyset$, atunci $\sigma^A = \emptyset$ și, deci, $A_w = \emptyset$). Ca urmare, există un sort s' în secvența w cu proprietatea $A_{s'} = \emptyset$. Aceasta face ca $B'_{s'}$ și $C'_{s'}$ să fie \emptyset și, deci, definiția operațiilor $\sigma^{B'}$ și $\sigma^{C'}$ ca fiind \emptyset este consistentă).

Este ușor de văzut că aceste construcții sunt corecte, $\mathbf{A} \cong \mathbf{B}' \times \mathbf{C}'$ și această descompunere este proprie. \square

Lema 8.6.2.1 ne permite să considerăm numai descompuneri proprii. Ca urmare, când vom spune că \mathbf{A} este *propriu decompozabilă* vom înțelege că există două algebrelle

netriviale \mathbf{B} și \mathbf{C} astfel încât $\mathbf{A} \cong \mathbf{B} \times \mathbf{C}$ și această descompunere este proprie. Dacă \mathbf{A} nu este propriu decompozabilă, atunci vom spune că este *propriu idecompozabilă*.

Orice algebră trivială este propriu idecompozabilă (ca urmare, algebra \mathbf{A} din Observația 8.6.2.1, fiind trivială, este propriu idecompozabilă).

Congruențele induse de homomorfismele proiecție asociate unui produs de algebrelor au o proprietate interesantă ce permite obținerea unor rezultate de caracterizare a algebrelor propriu decompozabile.

Definiția 8.6.2.3. O pereche de congruențe (ρ, θ) ale unei algebrelor \mathbf{A} este numită *pereche de congruențe factor* dacă au loc proprietățile:

$$(1) \quad \rho \wedge \theta = \iota_A;$$

$$(2) \quad \rho \vee \theta = A^2;$$

$$(3) \quad \rho \text{ și } \theta \text{ comută, adică } \rho \circ \theta = \theta \circ \rho$$

(\wedge reprezintă infimumul, iar \vee reprezintă supremumul, în laticea $Con(\mathbf{A})$).

Propoziția 8.6.2.1. Pentru orice două algebrelor \mathbf{A} și \mathbf{B} , $(ker(pr_1), ker(pr_2))$ este pereche de congruențe factor a algebriei $\mathbf{A} \times \mathbf{B}$.

Demonstrație. Pentru orice $s \in S$ și $(a, b), (c, d) \in A_s \times B_s$ are loc

$$(a, b) ker(pr_{1,s})(c, d) \Leftrightarrow a = c$$

și

$$(a, b) ker(pr_{2,s})(c, d) \Leftrightarrow b = d.$$

Ca urmare,

$$\begin{aligned} (a, b) (ker(pr_{1,s}) \cap ker(pr_{2,s})) (c, d) &\Leftrightarrow a = c \wedge b = d \\ &\Leftrightarrow (a, c) = (b, d), \end{aligned}$$

ceea ce arată că $ker(pr_{1,s}) \cap ker(pr_{2,s}) = \iota_{A_s \times B_s} = \iota_{(A \times B)_s}$.

Pentru orice sort s și $(a, b), (c, d) \in A_s \times B_s$, are loc

$$(a, b) (ker(pr_{1,s}) \vee ker(pr_{2,s})) (a, d) (ker(pr_{1,s}) \vee ker(pr_{2,s})) (c, d),$$

ceea ce arată că are loc $(A_s \times B_s)^2 \subseteq ker(pr_{1,s}) \vee ker(pr_{2,s})$, de unde urmează $(A_s \times B_s)^2 = ker(pr_{1,s}) \vee ker(pr_{2,s})$.

Dacă $(a, b) (ker(pr_{1,s}) \circ ker(pr_{2,s})) (c, d)$, atunci

$$(a, b) ker(pr_{2,s}) (c, b) ker(pr_{1,s}) (c, d),$$

adică $(a, b) (ker(pr_{2,s}) \circ ker(pr_{1,s})) (c, d)$, și reciproc.

Ca urmare, $(ker(pr_1), ker(pr_2))$ este pereche de congruențe factor pentru algebra $\mathbf{A} \times \mathbf{B}$. \square

Teorema 8.6.2.1. Fie \mathbf{A} o algebră. Atunci au loc următoarele proprietăți:

- (1) pentru orice descompunere $\mathbf{A} \cong \mathbf{B} \times \mathbf{C}$ și izomorfism h de la $\mathbf{B} \times \mathbf{C}$ la \mathbf{A} , perechea $(h(ker(pr_1)), h(ker(pr_2)))$ este pereche de congruențe factor a algebrei \mathbf{A} , unde pr_1 și pr_2 sunt homomorfismele proiecție asociate produsului $\mathbf{B} \times \mathbf{C}$;
- (2) pentru orice pereche (ρ, θ) de congruențe factor a algebrei \mathbf{A} , $\mathbf{A} \cong \mathbf{A}/\rho \times \mathbf{A}/\theta$.

Demonstrație. (1) urmează de la Propoziția 8.6.2.1 prin simple verificări pe care le lăsăm în seama cititorului.

(2) Fie (ρ, θ) o pereche de congruențe factor a algebrei \mathbf{A} . Considerăm funcția $h : A \rightarrow A/\rho \times A/\theta$ dată prin

$$h_s(a) = ([a]_{\rho_s}, [a]_{\theta_s}),$$

pentru orice $s \in S$ și $a \in A_s$.

Verificăm că h este injectivă. Are loc:

$$\begin{aligned} h_s(a) = h_s(b) &\Rightarrow ([a]_{\rho_s}, [a]_{\theta_s}) = ([b]_{\rho_s}, [b]_{\theta_s}) \\ &\Leftrightarrow [a]_{\rho_s} = [b]_{\rho_s} \wedge [a]_{\theta_s} = [b]_{\theta_s} \\ &\Leftrightarrow a \rho_s b \wedge a \theta_s b \\ &\Rightarrow a = b, \end{aligned}$$

pentru orice $s \in S$ și $a, b \in A_s$, ceea ce stabilește injectivitatea funcției h (ultima implicație urmează de la faptul că $\rho \wedge \theta = \iota_A$).

Stabilim surjectivitatea funcției h . Fie $s \in S$ și $([a]_{\rho_s}, [b]_{\theta_s}) \in A_s/\rho_s \times A_s/\theta_s$. Conform proprietăților perechii (ρ, θ) , există $c \in A_s$ astfel încât $a \rho_s c \theta_s b$. Atunci

$$h_s(c) = ([c]_{\rho_s}, [c]_{\theta_s}) = ([a]_{\rho_s}, [b]_{\theta_s}).$$

Deci h este surjectivă.

Arătăm că h este homomorfism și încheiem astfel demonstrația teoremei. Fie $(s_1 \cdots s_n, s) \in S^* \times S$, $\sigma \in \Sigma_{s_1 \cdots s_n, s}$ și $(a_1, \dots, a_n) \in A_{s_1 \cdots s_n}$. Are loc:

$$\begin{aligned} h_s(\sigma^A(a_1, \dots, a_n)) &= ([\sigma^A(a_1, \dots, a_n)]_{\rho_s}, [\sigma^A(a_1, \dots, a_n)]_{\theta_s}) \\ &= (\sigma^{A/\rho}([a_1]_{\rho_{s_1}}, \dots, [a_n]_{\rho_{s_n}}), \sigma^{A/\theta}([a_1]_{\theta_{s_1}}, \dots, [a_n]_{\theta_{s_n}})) \\ &= \sigma^{A/\rho \times A/\theta}(([a_1]_{\rho_s}, [a_1]_{\theta_s}), \dots, ([a_n]_{\rho_s}, [a_n]_{\theta_s})) \\ &= \sigma^{A/\rho \times A/\theta}(h_{s_1}(a_1), \dots, h_{s_n}(a_n)). \end{aligned}$$

Teorema este demonstrată. \square

Corolarul 8.6.2.1. O algebră \mathbf{A} este propriu decompozabilă dacă și numai dacă admite cel puțin o pereche de congruențe factor diferite de congruența de egalitate ι_A și de congruența totală \mathbf{A}^2 .

Demonstrație. Dacă \mathbf{A} este propriu decompozabilă, atunci există două algebrelle netriviale \mathbf{B} și \mathbf{C} astfel încât $\mathbf{A} \cong \mathbf{B} \times \mathbf{C}$ și această descompunere este proprie. Fie h un izomorfism de la $\mathbf{B} \times \mathbf{C}$ la \mathbf{A} . Perechea $(h(\ker(pr_1)), h(\ker(pr_2)))$ este pereche de congruențe factor ale algebrei \mathbf{A} (Teorema 8.6.2.1(1)). Mai mult, aceste congruențe sunt diferite de congruența de egalitate și de cea totală deoarece \mathbf{B} și \mathbf{C} sunt netriviale.

Reciproc, dacă (ρ, θ) este o pereche de congruențe factor ale algebrei \mathbf{A} , diferite de congruența de egalitate și de cea totală, atunci $\mathbf{A} \cong \mathbf{A}/\rho \times \mathbf{A}/\theta$ (Teorema 8.6.2.1(2)), iar \mathbf{A}/ρ și \mathbf{A}/θ sunt netriviale. În plus, este ușor de văzut că descompunerea este proprie. Deci \mathbf{A} este propriu decompozabilă. \square

8.6.3. Produse subdirecte de algebrelle

Definiția 8.6.3.1. Spunem că o algebră \mathbf{A} este un *produs subdirect* al unei familii nevidă de algebrelle $(\mathbf{A}^i | i \in I)$, și notăm $\mathbf{A} \leq_{SP} \prod_{i \in I} \mathbf{A}^i$, dacă $\mathbf{A} \leq \prod_{i \in I} \mathbf{A}^i$ și $A^i = pr_i(A)$, pentru orice $i \in I$.

Informal, putem spune că o algebră \mathbf{A} este un produs subdirect al unei familii de algebrelle $(\mathbf{A}^i | i \in I)$ dacă \mathbf{A} este o subfamilie de traiectorii ale produsului direct $\prod_{i \in I} \mathbf{A}^i$, închisă la operațiile algebrei produs direct, și care acoperă, pe componente, fiecare algebră \mathbf{A}^i .

Definiția 8.6.3.2. Se numește *reprezentare subdirectă* a algebrei \mathbf{A} într-o familie nevidă de algebrelle $(\mathbf{A}^i | i \in I)$ orice monomorfism $h : \mathbf{A} \rightarrow \prod_{i \in I} \mathbf{A}^i$ ce satisfac proprietatea $h(\mathbf{A}) \leq_{SP} \prod_{i \in I} \mathbf{A}^i$.

Teorema 8.6.3.1. Fie \mathbf{A} o algebră.

- (1) Dacă \mathbf{A} admite o reprezentare subdirectă h într-o familie nevidă de algebrelle $(\mathbf{A}^i | i \in I)$, atunci familia de congruențe $(\ker(pr_i \circ h) | i \in I)$ verifică proprietățile $\mathbf{A}/\ker(pr_i \circ h) \cong \mathbf{A}^i$, pentru orice $i \in I$, și $\bigcap_{i \in I} \ker(pr_i \circ h) = \iota_A$.
- (2) Dacă $(\rho^i | i \in I)$ este o familie nevidă de congruențe a algebrei \mathbf{A} astfel încât $\bigcap_{i \in I} \rho^i = \iota_A$, atunci \mathbf{A} admite o reprezentare subdirectă în $(\mathbf{A}/\rho^i | i \in I)$.

Demonstrație. (1) Deoarece $h(\mathbf{A}) \leq_{SP} \prod_{i \in I} \mathbf{A}^i$, are loc $pr_i(h(\mathbf{A})) = \mathbf{A}^i$, pentru orice $i \in I$. Ca urmare, $pr_i \circ h$ este epimorfism de la \mathbf{A} la \mathbf{A}^i , iar prima teoremă de izomorfism conduce la $\mathbf{A}/\ker(pr_i \circ h) \cong \mathbf{A}^i$, pentru orice $i \in I$.

Pentru orice $s \in S$ și $a, b \in A_s$ are loc:

$$\begin{aligned} (a, b) \in \bigcap_{i \in I} \ker(pr_{i,s} \circ h_s) &\Leftrightarrow (\forall i \in I)(h_s(a)(i) = h_s(b)(i)) \\ &\Leftrightarrow h_s(a) = h_s(b) \\ &\Leftrightarrow a = b \end{aligned}$$

(ultima echivalență folosește faptul că h_s este injecție). Deci, $\bigcap_{i \in I} \ker(pr_i \circ h) = \iota_A$.

(2) Fie $f_{\rho_i} : A \rightarrow A/\rho_i$ epimorfismul natural induș de ρ_i , pentru orice i . Proprietatea de universalitate a produsului direct $\prod_{i \in I} A/\rho_i$ (Propoziția 8.6.1.3(2)) conduce la existența unui unic homomorfism $h : A \rightarrow \prod_{i \in I} A/\rho_i$ ce satisfacă $f_{\rho_i} = pr_i \circ h$, pentru orice i . Cum f_{ρ_i} este epimorfism, obținem $pr_i(h(A)) = A/\rho_i$, pentru orice i . În plus, ipoteza $\bigcap_{i \in I} \rho^i = \iota_A$ conduce la faptul că h este monomorfism. Deci h este o reprezentare subdirectă a algebrei A în $(\mathbf{A}/\rho^i | i \in I)$. \square

Corolarul 8.6.3.1. Fie \mathbf{A} o algebră și $(\mathbf{A}^i | i \in I)$ o familie nevidă de algebrelle. \mathbf{A} admite o reprezentare subdirectă în familia $(\mathbf{A}^i | i \in I)$ dacă și numai dacă există o familie nevidă de congruențe $(\rho^i | i \in I)$ în \mathbf{A} astfel încât $\mathbf{A}/\rho^i \cong \mathbf{A}^i$, pentru orice $i \in I$, și $\bigcap_{i \in I} \rho^i = \iota_A$.

Demonstrație. Implicația în sens direct urmează de la Teorema 8.6.3.1(1).

Reciproc, de la Teorema 8.6.3.1(2) urmează că există o reprezentare subdirectă h a algebrei \mathbf{A} în familia $(\mathbf{A}/\rho^i | i \in I)$. Ipotezele corolarului ne asigură că există un izomorfism între algebrelle $\prod_{i \in I} \mathbf{A}/\rho^i$ și $\prod_{i \in I} \mathbf{A}^i$. Pentru orice astfel de izomorfism φ , $\varphi \circ h$ este reprezentare subdirectă a algebrei \mathbf{A} în familia $(\mathbf{A}^i | i \in I)$. \square

Evident, ne putem pune problema găsirii unor modalități de determinare a unor produse subdirecte a unei familii de algebrelle. Nu există nici un rezultat general în acest sens. Pentru răspunsuri parțiale la această chestiune, cititorul este îndrumat către [63].

8.7. Algebrelle libere

Algebrelle libere generalizează conceptul de monoid (grup, inel etc.) liber. Pe lângă importanța acestora în matematică, algebrelle libere au o importanță deosebită în informatică, în special în teoria tipurilor abstrakte de date.

8.7.1. Algebrelle de termi

Termii sunt structuri sintactice cu care lucrăm în mod ușual. De exemplu, $x + 2$, unde x este o variabilă iar 2 este o constantă, este un term. Aceasta este scris utilizând notația infix; în notație postfix el are forma $+(x, 2)$. Dacă “+” este considerat ca simbol funcțional atât de tip $(nat\ nat, nat)$ cât și de tip $(real\ real, real)$, atunci expresia $+(x, 2)$ este ambiguă. Putem să o dezambiguizăm dacă pentru fiecare simbol utilizat specificăm și sortul lui. De exemplu, o scriere de tipul $+.nat(x.nat, 2.nat)$ ne spune clar că + este considerat de tip $(nat\ nat, nat)$, x este variabilă de tip nat , iar 2 este o constantă de tip nat .

Pornind de la aceste remarci vom încerca în continuare să introducem riguroș conceptual de term. Fie Σ o signatură S -sortată și $X = (X_s | s \in S)$ o familie disjunctă de variabile (adică $X_s \cap X_{s'} = \emptyset$, pentru orice $s, s' \in S$ cu $s \neq s'$). În tot ceea ce urmează vom înțelege implicit că familia de variabile X care se consideră este disjunctă, iar când vom spune că X este disjunctă de Σ vom înțelege că are loc

$$X_s \cap \bigcup_{(w,s) \in S^* \times S} \Sigma_{w,s} = \emptyset,$$

pentru orice $s \in S$.

Vom nota prin $\Sigma.S$ familia

$$\Sigma.S = ((\Sigma.S)_{w,s} | (w, s) \in S^* \times S),$$

unde $(\Sigma.S)_{w,s} = \{\sigma.s | \sigma \in \Sigma_{w,s}\}$, pentru orice $(w, s) \in S^* \times S$. $(\Sigma.S)_{\lambda,s}$ este notată simplificat prin $(\Sigma.S)_s$. Observăm că $\sigma.s \in (\Sigma.S)_{w,s} \cap (\Sigma.S)_{w',s}$, pentru orice $\sigma \in \Sigma_{w,s} \cap \Sigma_{w',s}$. Ca urmare, $\Sigma.S$ poate fi o familie nedisjunctă dacă Σ este nedisjunctă. Însă familiile $((\Sigma.S)_s | s \in S)$ și X vor fi considerate întotdeauna disjuncte.

Definiția 8.7.1.1. Fie Σ o signatură și X o familie de variabile, disjunctă de Σ . Termii de sort s peste Σ și X se definesc inductiv prin:

- (1) pentru orice $s \in S$ și $x \in X_s$, x este term de sort s ;
- (2) pentru orice $(s_1 \cdots s_n, s) \in S^* \times S$, $\sigma \in \Sigma_{s_1 \cdots s_n, s}$ și orice term t_i de sort s_i , $1 \leq i \leq n$, $\sigma.s(t_1, \dots, t_n)$ este term de sort s (în cazul $s_1 \cdots s_n = \lambda$ vom înțelege că $\sigma.s$ este term de sort s).

Vom nota prin $T_\Sigma(X)_s$ mulțimea tuturor termilor de sort s peste Σ și X , iar prin $T_\Sigma(X)$, familia $T_\Sigma(X) = (T_\Sigma(X)_s | s \in S)$. Atunci când $X = \emptyset$, vom simplifica notația la $T_\Sigma = (T_\Sigma.s | s \in S)$.

Termii ce nu conțin variabile se numesc *termi de bază*.

Exemplul 8.7.1.1.

- (1) Considerând signatura din Exemplul 8.2.1(2) și $X_{nat} = \{x, y\}$, următoarele structurile sintactice sunt termi de sort *nat*: x , y , $zero.nat$, $succ.nat(x)$, ...
- (2) Considerând signatura din Exemplul 8.2.1(4), $X_{nat} = \{x, y\}$ și $X_{bool} = \{z\}$, $and.bool(true.bool, z)$ este term de sort *bool*, iar $add.nat(zero.nat, y)$ este term de sort *nat*.

Observația 8.7.1.1. Fie Σ o signatură și X o familie de variabile, disjunctă de Σ .

- (1) Termii sunt construcții sintactice peste familiile $\Sigma.S$ și X . Dacă considerăm mulțimea $\bigcup \Sigma.S \cup \bigcup X \cup \{((), ,)\}$, atunci termii sunt cuvinte peste această mulțime, adică elemente ale monoidului liber generat de această mulțime (dar nu orice element al acestui monoid este term).

- (2) $T_\Sigma(X)$ este familie disjunctă de mulțimi, ceea ce se poate verifica cu ușurință în baza faptului că X este o familie disjunctă de mulțimi și a modului de construcție a termilor.
- (3) $T_\Sigma(X)_s$ poate fi nevidă chiar dacă $(\Sigma.S)_s \cup X_s = \emptyset$. De exemplu, dacă σ este de tip (s', s) cu $s' \neq s$ și există cel puțin un term t de sort s' , atunci $\sigma.s(t)$ va fi de sort s .

Deoarece mulțimea termilor este definită inductiv, principiul inducției structurale poate fi folosit în acest caz.

Teorema 8.7.1.1. (Principiul inducție structurale pentru termi)

Fie Σ o signatură și X o familie de variabile, disjunctă de Σ . Dacă $P = (P_s | s \in S)$ este un predicat astfel încât:

- (1) $P_s(t)$, pentru orice $s \in S$ și $t \in (\Sigma.S)_s \cup X_s$;
- (2) $(P_{s_1}(t_1) \wedge \dots \wedge P_{s_n}(t_n)) \Rightarrow P_s(\sigma.s(t_1, \dots, t_n))$, pentru orice $(s_1 \cdots s_n, s) \in S^+ \times S$, $\sigma \in \Sigma_{s_1 \cdots s_n, s}$ și $t_i \in T_\Sigma(X)_{s_i}$, $1 \leq i \leq n$,

atunci $P_s(t)$, pentru orice $s \in S$ și $t \in T_\Sigma(X)_s$.

$T_\Sigma(X)$ poate fi structurată ca o Σ -algebră considerând operațiile $\sigma^{T_\Sigma(X)}$ date prin:

- $\sigma^{T_\Sigma(X)}(t_1, \dots, t_n) = \sigma.s(t_1, \dots, t_n)$, pentru orice $(s_1 \cdots s_n, s) \in S^+ \times S$, $\sigma \in \Sigma_{s_1 \cdots s_n, s}$ și $t_i \in T_\Sigma(X)_{s_i}$, $1 \leq i \leq n$;
- $\sigma^{T_\Sigma(X)}(\emptyset) = \sigma.s$, pentru orice $s \in S$ și $\sigma \in \Sigma_s$.

Algebra $\mathbf{T}_\Sigma(X) = (T_\Sigma(X), \Sigma^{T_\Sigma(X)})$ astfel obținută se numește *algebra termilor peste Σ și X* sau *Σ -algebra termilor peste X* . În cazul $X = \emptyset$, această algebră este notată prin \mathbf{T}_Σ și este numită *algebra termilor de bază peste Σ* sau *Σ -algebra termilor de bază*.

Observația 8.7.1.2.

- (1) Fie Σ o signatură și X o familie de variabile, disjunctă de Σ . Este trivial de văzut că $T_\Sigma(X)$ este liber inductiv definită de X și constructorii corespunzători asociați simbolurilor funcționale. Adică orice term admite exact o construcție inductivă de la X și $\Sigma.S$ (cititorul poate verifica aceasta printr-o simplă inducție structurală).
- (2) Pentru orice signatură Σ , algebra \mathbf{T}_Σ este minimală deoarece este generată de familia vidă (echivalent, de familia constantelor ei).

Homomorfismele f de la $\mathbf{T}_\Sigma(X)$ la o Σ -algebră \mathbf{A} satisfac

$$f_s(\sigma.s(t_1, \dots, t_n)) = f_s(\sigma^{T_\Sigma(X)}(t_1, \dots, t_n)) = \sigma^A(f_{s_1}(t_1), \dots, f_{s_n}(t_n)),$$

pentru orice $(s_1 \cdots s_n, s) \in S^+ \times S$, $\sigma \in \Sigma_{s_1 \cdots s_n, s}$ și $t_i \in T_\Sigma(X)_{s_i}$, $1 \leq i \leq n$, și

$$f_s(\sigma \cdot s) = f_s(\sigma^{T_\Sigma(X)}(\emptyset)) = \sigma^A(f_s(\emptyset)) = \sigma^A(\emptyset) = \sigma^A,$$

pentru orice $s \in S$ și $\sigma \in \Sigma_s$.

Este clar că f este complet definită de restricția ei la X .

Teorema 8.7.1.2. (Principiul recursiei algebrice finitare)

Fie Σ o signatură și X o familie de variabile, disjunctă de Σ . Atunci, pentru orice algebră \mathbf{A} și orice funcție $f : X \rightarrow A$, există un unic homomorfism $\bar{f} : T_\Sigma(X) \rightarrow A$ ce extinde f (adică, $\bar{f}_s(x) = f_s(x)$, pentru orice $s \in S$ și $x \in X_s$).

Demonstrație. Direct de la Observația 8.7.1.2(1) și Teorema recursiei pentru mulțimi S -sortate (Secțiunea 8.1). \square

Funcțiile $f : X \rightarrow A$ (ca în Teorema 8.7.1.2) se mai numesc și *atribuiră* sau *asignări*.

Ele atribuie variabilelor elemente dintr-o algebră \mathbf{A} . Unica extensie a unei atribuirii f la un homomorfism de la $T_\Sigma(X)$ la \mathbf{A} nu face altceva decât să interpreteze termii din $T_\Sigma(X)$ în algebra \mathbf{A} , având ca suport o atribuire dată. Din acest motiv, aceste homomorfisme mai sunt numite și *funcții de interpretare* sau *interpretări* sau *funcții de evaluare* sau *evaluări*. Vom nota atribuirile, cu precădere, prin γ , iar unica lor extensie prin $\bar{\gamma}$.

Mulțimea tuturor atribuirilor în algebra \mathbf{A} va fi notată prin $\Gamma(X, \mathbf{A})$.

Corolarul 8.7.1.1. Fie Σ o signatură și X o familie de variabile, disjunctă de Σ . Atunci, pentru orice algebră \mathbf{A} există un unic homomorfism de la T_Σ la \mathbf{A} .

Demonstrație. Direct de la Teorema 8.7.1.2, considerând funcția vidă ca fiind unică atribuire de la $X = \emptyset$ la A . \square

Unicul homomorfism din Corolarul 8.7.1.1 se notează în mod frecvent prin f_A sau eval_A .

Observația 8.7.1.3. Combinând Corolarul 8.7.1.1 cu Observația 8.7.1.2 obținem că o Σ -algebră \mathbf{A} este minimală dacă și numai dacă există un unic epimorfism de la T_Σ la \mathbf{A} .

Observația 8.7.1.4. Fie Σ o signatură și X o familie de variabile, disjunctă de Σ . Ar putea părea natural de a defini termii utilizând Σ în loc de $\Sigma \cdot S$, în mod similar Definiției 8.7.1.1 dar renunțând la construcția “ $\cdot s$ ”. De exemplu, în loc de $\sigma \cdot s(t_1, \dots, t_n)$ să scriem $\sigma(t_1, \dots, t_n)$. Problemele care apar cu o astfel de definiție sunt însă următoarele. Să presupunem că $\sigma \in \Sigma_{s_1, s} \cap \Sigma_{s_2, s}$, $s_1 \neq s_2$ și $\sigma_1 \in \Sigma_{s_1} \cap \Sigma_{s_2}$. Atunci, $\sigma(s_1)$ este term de sort s . Mai mult, el reprezintă doi termi de sort s , unul în care σ_1 este de sort s_1 și unul în care σ_1 este de sort s_2 . Însă în mulțimea $T_\Sigma(X)_s$ el va avea doar o apariție, în timp ce prin Definiția 8.7.1.1 ambele variante ale acestui term sunt considerate, $\sigma \cdot s(s_1)$ și $\sigma \cdot s(s_2)$.

O simplificare ca cea sugerată mai sus în notația termilor este legitimă dacă signatura Σ este familie disjunctă de mulțimi. Putem însă accepta o astfel de simplificare notațională chiar dacă Σ nu este familie disjunctă de mulțimi, cu condiția ca de fiecare dată să specificăm clar tipul simbolurilor funcționale σ ce definesc termii $\sigma \cdot s(t_1, \dots, t_n)$. Aceasta va fi de fapt presupunerea sub care vom lucra mai departe. Astfel, termii peste Σ și X de sort $s \in S$ vor fi variabile $x \in X_s$, constante $\sigma \in \Sigma_s$ sau structuri de formă $\sigma(t_1, \dots, t_n)$, unde $(s_1 \cdots s_n, s) \in S^+ \times S$, $\sigma \in \Sigma_{s_1 \cdots s_n, s}$ și t_i este term de sort s_i , pentru orice $1 \leq i \leq n$.

Familia variabilelor unui term t , $\text{Var}(t) = (\text{Var}(t)_s | s \in S)$, se definește inductiv prin:

- dacă $t = x \in X_s$, atunci

$$\text{Var}(t)_{s'} = \begin{cases} \{x\}, & \text{dacă } s' = s \\ \emptyset, & \text{altfel,} \end{cases}$$

pentru orice $s' \in S$;

- dacă $t = \sigma \in \Sigma_s$, atunci $\text{Var}(t)_{s'} = \emptyset$, pentru orice $s' \in S$;

- dacă $t = \sigma(t_1, \dots, t_n)$, unde $\sigma \in \Sigma_{s_1 \cdots s_n, s}$, $n \geq 1$ și $t_i \in T_\Sigma(X)_{s_i}$, pentru orice $1 \leq i \leq n$, atunci

$$\text{Var}(t) = \bigcup_{i=1}^n \text{Var}(t_i).$$

Corolarul 8.7.1.2. (Teorema coincidenței)

Fie Σ o signatură și X o familie de variabile, disjunctă de Σ . Atunci, pentru orice algebră \mathbf{A} și orice atribuire $\gamma, \gamma' : X \rightarrow A$ are loc

$$(\forall s \in S)(\forall t \in T_\Sigma(X)_s)(\gamma|_{\text{Var}(t)} = \gamma'|_{\text{Var}(t)} \Rightarrow \bar{\gamma}_s(t) = \bar{\gamma}'_s(t)).$$

Demonstrație. Corolarul poate fi obținut direct de la Corolarul 8.5.1.3 printr-o definire corespunzătoare a unor funcții, sau prin inducție structurală. \square

Corolarul 8.7.1.3. Orice algebră este imaginea homomorfă a unei algebrelor de termi.

Demonstrație. Fie \mathbf{A} o algebră. Atunci există $X \subseteq A$ care să genereze \mathbf{A} . Considerăm X' o familie echivalentă cu X astfel încât X' este disjunctă de Σ , și fie $f : X' \rightarrow X$ o funcție bijectivă (aceasta există ca urmare a alegerii lui X'). f se poate extinde la o funcție $f' : X' \rightarrow A$ prin $f'_s(x) = f_s(x)$, pentru orice $s \in S$ și $x \in X_s$, iar Teorema 8.7.1.2 asigură existența unui unic homomorfism h de la $T_\Sigma(X')$ la \mathbf{A} ce extinde f' . Atunci

$$h(T_\Sigma(X')) = h(\langle X' \rangle) = \langle h(X') \rangle_{\mathbf{A}} = \langle f'(X') \rangle_{\mathbf{A}} = \langle X \rangle_{\mathbf{A}} = \mathbf{A},$$

ceea ce arată că \mathbf{A} este imaginea homomorfă a algebrelor $T_\Sigma(X')$ (prima egalitate urmează de la faptul că $T_\Sigma(X')$ este generată de X' , iar a doua de la Corolarul 8.5.1.7). \square

Corolarul 8.7.1.4. Orice algebră este izomorfă cu algebra căt a unei algebrelor de termi.

Demonstrație. Fie \mathbf{A} o algebră. Conform Corolarului 8.7.1.3 există o algebră de termi $\mathbf{T}_\Sigma(X)$ și un epimorfism $f : \mathbf{T}_\Sigma(X) \rightarrow \mathbf{A}$. În baza Teoremei 8.5.4.1, $\mathbf{T}_\Sigma(X)/\ker(f) \cong \mathbf{A}$. \square

Corolarul 8.7.1.5. (Proprietatea de proiectivitate a algebrelor de termi)

Fie Σ o signatură, \mathbf{A} și \mathbf{B} două Σ -algebrelor și $f : A \rightarrow B$ un epimorfism. Atunci, pentru orice familie de variabile X disjunctă de Σ și orice homomorfism h de la $\mathbf{T}_\Sigma(X)$ la B , există un homomorfism g de la $\mathbf{T}_\Sigma(X)$ la A astfel încât $f \circ g = h$.

Demonstrație. Fie $g' : X \rightarrow A$ o funcție ce satisfacă $g'_s(x) \in f^{-1}(h_s(x))$, pentru orice sort s și $x \in X_s$. Conform Teoremei 8.7.1.2, g' se extinde la un unic homomorfism g de la $\mathbf{T}_\Sigma(X)$ la A . În plus, g satisfacă proprietatea din corolar. \square

Proprietatea de proiectivitate poate fi vizualizată ca în Figura 8.12. Ea spune că orice algebră de termi ce poate fi proiectată în codomeniul unui epimorfism poate fi proiectată și în domeniul acestuia și cele două proiecții sunt funcțional corelate prin epimorfismul în cauză⁹.

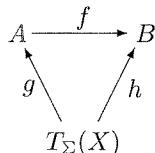


Figura 8.12: Proprietatea de proiectivitate a algebrelor de termi

Atribuirile $f : X \rightarrow \mathbf{T}_\Sigma(Y)$ într-o algebră de termi se mai numesc și *substituții*. Ele se extind unic la homomorfisme $\bar{f} : \mathbf{T}_\Sigma(X) \rightarrow \mathbf{T}_\Sigma(Y)$. În unele lucrări, substituțiile sunt definite ca endomorfisme, aceasta presupunând $Y = X$.

Teorema 8.7.1.3. (Teorema substituției)

Fie Σ o signatură iar X și Y familii de variabile, disjuncte de Σ . Pentru orice substituție $f : X \rightarrow \mathbf{T}_\Sigma(Y)$ și atribuire $\gamma : Y \rightarrow A$, unde \mathbf{A} este o Σ -algebră, are loc

$$(\forall s \in S)(\forall t \in \mathbf{T}_\Sigma(Y)_s)(\bar{\gamma}_s(\bar{f}_s(t)) = \bar{\gamma}'_s(t)),$$

unde $\gamma' : X \rightarrow A$ este atribuirea dată prin

$$\gamma'_s(x) = \bar{\gamma}_s(f_s(x)),$$

pentru orice $s \in S$ și $x \in X_s$.

⁹ Este clar că proprietatea de proiectivitate poate fi generalizată prin înlocuirea algebrelor de termi prin orice algebră liberă generată de o parte a sa.

Demonstrație. Homomorfismele $\bar{\gamma} \circ \bar{f}$ și $\bar{\gamma}'$ coincid pe X . Atunci teorema urmează de la Corolarul 8.5.1.3. \square

Teorema substituție vrea să spună că a face substituții și apoi a interpreta rezultatul este echivalent cu a face deodată interpretări potrivit alese.

Fie $f : \mathbf{T}_\Sigma(X) \rightarrow \mathbf{T}_\Sigma(Y)$ o substituție și t un term de sort s . De multe ori, în loc de $\bar{f}_s(t)$ se scrie

$$t[x_1/t_1, \dots, x_n/t_n],$$

presupunând că x_1, \dots, x_n sunt toate variabilele din t , x_i și t_i sunt de sort s_i și $\bar{f}_{s_i}(x_i) = t_i$, pentru orice $1 \leq i \leq n$.

8.7.2. Algebrelibere. Definiții și proprietăți de bază

O algebră de termi $\mathbf{T}_\Sigma(X)$ are două proprietăți fundamentale:

1. este generată de X și, mai mult, orice element al ei are exact o construcție inductivă de la X și Σ ;
2. orice funcție $f : X \rightarrow A$, unde \mathbf{A} este o algebră, se extinde în mod unic la un homomorfism de la $\mathbf{T}_\Sigma(X)$ la \mathbf{A} .

Aceste proprietăți sunt cruciale în studiul algebrelor universale și în aplicațiile acestora în diverse domenii.

Cea de a doua proprietate se referă la clasa tuturor algebrelor de o același natură. Există situații care cer restrângerea la o subclasă \mathcal{K} de algebrelor (de același signatură). În acest caz, este posibil ca $\mathbf{T}_\Sigma(X)$ să nu mai fie în această clasă. Vom investiga în continuare cazul în care putem găsi într-o clasă \mathcal{K} de algebrelor o algebră cu proprietăți similare algebrelui $\mathbf{T}_\Sigma(X)$.

Prin *clasă de Σ -algebrelor* vom înțelege o colecție nevidă \mathcal{K} de Σ -algebrelor. O clasă \mathcal{K} de algebrelor este numită *netrivială* dacă pentru orice sort $s \in S$ există $\mathbf{A} \in \mathcal{K}$ astfel încât $|A_s| \geq 2$.

Definiția 8.7.2.1. Fie \mathcal{K} o clasă de Σ -algebrelor și X o familie S -sortată.

- (1) Spunem că o algebră $\mathbf{A} \in \mathcal{K}$ este *\mathcal{K} -liberă relativ la o funcție* $\varphi : X \rightarrow A$ dacă pentru orice algebră $\mathbf{B} \in \mathcal{K}$ și orice funcție $f : X \rightarrow B$ există un unic homomorfism h de la \mathbf{A} la \mathbf{B} ce satisfacă $f = h \circ \varphi$.
- (2) Spunem că o algebră $\mathbf{A} \in \mathcal{K}$ este *\mathcal{K} -inițială* dacă \mathbf{A} este \mathcal{K} -liberă relativ la $\varphi : \emptyset \rightarrow A$.

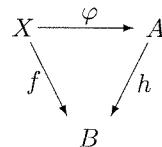


Figura 8.13: Proprietatea din Definiția 8.7.2.1(1)

Proprietatea din Definiția 8.7.2.1(1) se exprimă adesea și în varianta “pentru orice algebră $\mathbf{B} \in \mathcal{K}$ există un unic homomorfism de la \mathbf{A} la \mathbf{B} ce face comutativă diagrama din Figura 8.13”, și se mai numește *proprietatea de universalitate a algebrei \mathbf{A} relativ la $\varphi : X \rightarrow A$ și \mathcal{K}* . Menționăm că o algebră \mathbf{A} poate satisface această proprietate de universalitate fără a fi în clasa \mathcal{K} ; dacă ea este și în clasa \mathcal{K} , atunci este \mathcal{K} -liberă relativ la $\varphi : X \rightarrow A$.

Observația 8.7.2.1. Fie \mathcal{K} o clasă de Σ -algebrelor și $\mathbf{A} \in \mathcal{K}$.

- (1) În cazul în care $\varphi : X \rightarrow A$ este funcția inclusiune ($X \subseteq A$ și $\varphi_s(x) = x$, pentru orice sort s și $x \in X_s$), \mathbf{A} este \mathcal{K} -liberă relativ la $\varphi : X \rightarrow A$ dacă pentru orice algebră $\mathbf{B} \in \mathcal{K}$ și orice funcție $f : X \rightarrow B$, există un unic homomorfism h de la \mathbf{A} la \mathbf{B} ce extinde f .
- (2) \mathbf{A} este \mathcal{K} -inițială dacă pentru orice $\mathbf{B} \in \mathcal{K}$ există un unic homomorfism h de la \mathbf{A} la \mathbf{B} .

Atunci când \mathcal{K} este clasa tuturor algebrelor, vom simplifica terminologia de la \mathcal{K} -liberă (\mathcal{K} -inițială) la liberă (inițială). Uneori, vom mai adăuga “liberă (inițială) în clasa tuturor algebrelor de aceeași semnatură”. O altă simplificare a terminologiei o vom face atunci când funcția φ este funcția inclusiune. În acest caz vom spune că \mathbf{A} este \mathcal{K} -liberă (liberă) relativ la X .

Exemplul 8.7.2.1. Fie Σ o semnatură și X o familie de variabile, disjunctă de Σ .

- (1) Σ -algebra $\mathbf{T}_\Sigma(X)$ este liberă în clasa tuturor algebrelor de semnatură Σ , relativ la X . Aceasta urmează imediat de la Observația 8.7.2.1(1) și Teorema 8.7.1.2.
- (2) Σ -algebra \mathbf{T}_Σ este inițială în clasa tuturor algebrelor de semnatură Σ . Aceasta urmează imediat de la Observația 8.7.2.1(2) și Corolarul 8.7.1.1.

Dual conceptului de algebră inițială este cel de algebră finală.

Definiția 8.7.2.2. Fie \mathcal{K} o clasă de Σ -algebrelor și $\mathbf{A} \in \mathcal{K}$. \mathbf{A} este \mathcal{K} -finală dacă pentru orice $\mathbf{B} \in \mathcal{K}$ există un unic homomorfism h de la \mathbf{B} la \mathbf{A} .

Atunci când \mathcal{K} este clasa tuturor algebrelor, vom simplifica terminologia de la \mathcal{K} -finală la cea de finală.

Exemplul 8.7.2.2. Orice Σ -algebră ce are suportul de sort s de cardinal 1, pentru orice sort s , este finală.

Vom prezenta în continuare câteva proprietăți de bază ale algebrelor libere.

Propoziția 8.7.2.1. Fie \mathbf{A} o algebră \mathcal{K} -liberă relativ la $\varphi : X \rightarrow A$. Dacă \mathcal{K} este netrivială, atunci φ este funcție injectivă.

Demonstrație. Presupunem, prin contradicție, că ar exista un sort s și două elemente $x_1, x_2 \in X_s$ cu $x_1 \neq x_2$ și $\varphi_s(x_1) = \varphi_s(x_2)$.

Deoarece \mathcal{K} este netrivială, există o algebră $\mathbf{B} \in \mathcal{K}$ pentru care $|B_s| \geq 2$. Fie $b_1, b_2 \in B_s$ cu $b_1 \neq b_2$, și fie $f : X \rightarrow B$ o funcție ce satisface $f_s(x_1) = b_1$ și $f_s(x_2) = b_2$.

Proprietatea algebrei \mathbf{A} de a fi \mathcal{K} -liberă relativ la $\varphi : X \rightarrow A$ conduce la existența unui unic homomorfism h de la \mathbf{A} la \mathbf{B} ce satisface $f = h \circ \varphi$. Dar atunci,

$$b_1 = f_s(x_1) = h_s(\varphi_s(x_1)) = h_s(\varphi_s(x_2)) = f_s(x_2) = b_2,$$

ceea ce constituie o contradicție. \square

Ideea de demonstrație din Propoziția 8.7.2.1 poate fi ilustrată grafic ca în Figura 8.14.

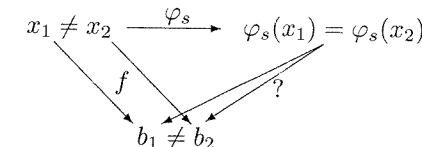


Figura 8.14: Ilustrare grafică a demonstrației Propoziției 8.7.2.1

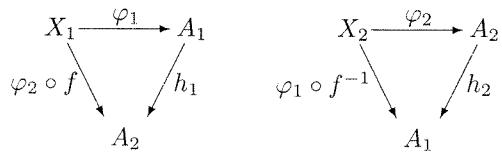
Propoziția 8.7.2.2. Fie \mathcal{K} o clasă de algebrelor. Dacă $\mathbf{A}_1 \in \mathcal{K}$ este \mathcal{K} -liberă relativ la $\varphi_1 : X_1 \rightarrow A_1$, $\mathbf{A}_2 \in \mathcal{K}$ este \mathcal{K} -liberă relativ la $\varphi_2 : X_2 \rightarrow A_2$, iar X_1 și X_2 sunt echipotente, atunci $\mathbf{A}_1 \cong \mathbf{A}_2$.

Demonstrație. Fie $f : X_1 \rightarrow X_2$ o funcție bijectivă. Considerăm diagramele din Figura 8.15, unde h_1 și h_2 sunt unicele homomorfisme ce satisfac

$$(*) \begin{cases} \varphi_2 \circ f = h_1 \circ \varphi_1 \\ \varphi_1 \circ f^{-1} = h_2 \circ \varphi_2 \end{cases}$$

Aceste două relații sunt echivalente cu relațiile $(**)$

$$(**) \begin{cases} \varphi_2 = (h_1 \circ h_2) \circ \varphi_2 \\ \varphi_1 \circ f^{-1} = h_2 \circ \varphi_2. \end{cases}$$

Figura 8.15: Proprietatea de universalitate pentru \mathbf{A}_1 și \mathbf{A}_2

În adevăr, relațiile (*) conduc la

$$\varphi_2 = h_1 \circ (\varphi_1 \circ f^{-1}) = h_1 \circ (h_2 \circ \varphi_2) = (h_1 \circ h_2) \circ \varphi_2,$$

obținându-se (**). În mod similar se arată că (**) conduce la (*).

Cum h_1 și h_2 sunt unicele homomorfisme care satisfac relațiile de la (*), ele vor fi și unicele care satisfac relațiile de la (**). Însă $\varphi_2 = 1_{A_2} \circ \varphi_2$. Ca urmare, $h_1 \circ h_2 = 1_{A_2}$.

În mod similar se arată că relațiile (*) sunt echivalente cu relațiile (***)

$$(***) \begin{cases} \varphi_2 \circ f = h_1 \circ \varphi_1 \\ \varphi_1 = (h_2 \circ h_1) \circ \varphi_1, \end{cases}$$

ceea ce va conduce la $h_2 \circ h_1 = 1_{A_1}$.

Combinând acum $h_1 \circ h_2 = 1_{A_2}$ și $h_2 \circ h_1 = 1_{A_1}$ obținem că h_1 și h_2 sunt inverse unul altuia. Deci ele sunt izomorfisme. \square

Propoziția 8.7.2.3. Fie \mathbf{A} o algebră \mathcal{K} -liberă relativ la $\varphi : X \rightarrow A$. Dacă \mathcal{K} este închisă la subalgebre (adică, orice subalgebră a unei algebrelor din \mathcal{K} este tot în \mathcal{K}), atunci \mathbf{A} este generată de $\varphi(X)$.

Demonstrație. $\langle \varphi(X) \rangle_{\mathbf{A}}$ este subalgebră a algebrelor \mathbf{A} și, apelând la ipoteză, este în clasa \mathcal{K} .

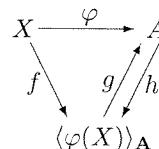


Figura 8.16: Demonstrația Propoziției 8.7.2.3

Fie $f : X \rightarrow \langle \varphi(X) \rangle_{\mathbf{A}}$ dată prin $f_s(x) = \varphi_s(x)$, pentru orice sort s și $x \in X_s$, și fie h unicul homomorfism asigurat de proprietatea de universalitate a algebrelor \mathbf{A} relativ la $\varphi : X \rightarrow A$. El satisface

$$(*) \quad f = h \circ \varphi$$

(a se vedea diagrama din Figura 8.16).

Fie funcția $g : \langle \varphi(X) \rangle_{\mathbf{A}} \rightarrow A$ dată prin $g_s(a) = a$, pentru orice sorts s și $a \in \langle \varphi(X) \rangle_{\mathbf{A}}_s$. Observăm că g este funcție injectivă și satisfacă $g \circ f = \varphi$.

Vom arăta că (*) este echivalentă cu (**) dată prin

$$(**) \quad \varphi = (g \circ h) \circ \varphi$$

În adevăr, dacă pornim de la (*) obținem

$$\varphi = g \circ f = g \circ (h \circ \varphi) = (g \circ h) \circ \varphi$$

și, deci, are loc (**).

Reciproc, fie g' un invers la stânga al funcției g (g are un invers la stânga fiind injectivă). Atunci, $g \circ f = \varphi$ conduce la $f = g' \circ \varphi$. Utilizând (**) obținem

$$f = g' \circ \varphi = (g' \circ g) \circ h \circ \varphi = h \circ \varphi,$$

ceea ce furnizează (*).

Cum h este unicul homomorfism ce satisfac (*), iar (*) este echivalentă cu (**), h este unicul homomorfism ce satisfac (**). Însă, 1_A satisfac $\varphi = 1_A \circ \varphi$. Ca urmare, $1_A = g \circ h$, ceea ce ne arată că g este funcție surjectivă. Combinând cu injectivitatea acesteia, obținem că g este izomorfism de la $\langle \varphi(X) \rangle_{\mathbf{A}}$ la \mathbf{A} , demonstrând astfel propoziția. \square

8.7.3. Teorema lui Birkhoff de existență a algebrelor libere

În această secțiune vom demonstra un rezultat fundamental asupra existenței algebrelor libere într-o clasă \mathcal{K} de algebrelor.

Fie \mathcal{K} o clasă de algebrelor. Vom nota prin:

- $S(\mathcal{K})$, închiderea clasei \mathcal{K} la subalgebre, adică cea mai mică clasă de algebrelor ce include \mathcal{K} și este închisă la subalgebre (pentru orice $\mathbf{A} \in S(\mathcal{K})$ și $\mathbf{B} \leq \mathbf{A}$, $\mathbf{B} \in S(\mathcal{K})$);
- $H(\mathcal{K})$, închiderea clasei \mathcal{K} la homomorfisme, adică cea mai mică clasă de algebrelor ce include \mathcal{K} și este închisă la homomorfisme (pentru orice $\mathbf{A} \in H(\mathcal{K})$ și homomorfism h definit pe \mathbf{A} , $h(\mathbf{A}) \in H(\mathcal{K})$);
- $I(\mathcal{K})$, închiderea clasei \mathcal{K} la izomorfisme, adică cea mai mică clasă de algebrelor ce include \mathcal{K} și este închisă la izomorfisme;
- $P(\mathcal{K})$, închiderea clasei \mathcal{K} la produse directe, adică cea mai mică clasă de algebrelor ce include \mathcal{K} și este închisă la produse directe (produsul direct al oricărei familii nevide de algebrelor din \mathcal{K} este algebră în \mathcal{K}).

Putem privi S , H , I și P ca niște operatori pe clase de algebrelor. Ei pot fi combinați pentru a produce închideri multiple. De exemplu, $IS(\mathcal{K})$ va desemna închiderea clasei

\mathcal{K} la subalgebre și izomorfisme. Atragem însă atenția că închiderea este realizată simultan și nu întâi la un operator și apoi la celălalt.

Fie $\mathbf{T}_\Sigma(X)$ o algebră de termi și \mathcal{K} o clasă de algebrelle. Clasa \mathcal{K} induce, în mod natural, următoarele relații binare pe $\mathbf{T}_\Sigma(X)$:

$$1. \quad \equiv^{\mathcal{K}} = \bigcap \{ \ker(h) \mid h \in \text{Hom}(\mathbf{T}_\Sigma(X), \mathbf{A}), \mathbf{A} \in \mathcal{K} \}.$$

$\equiv^{\mathcal{K}}$ pune în relație toți termii care se interpretează la fel, prin homomorfisme, în orice algebră din \mathcal{K} ;

$$2. \quad \approx^{\mathcal{K}} = \bigcap \{ \rho \mid \rho \in \text{Con}(\mathbf{T}_\Sigma(X)), \mathbf{T}_\Sigma(X)/\rho \in IS(\mathcal{K}) \}.$$

$\approx^{\mathcal{K}}$ pune în relație toți termii care se găsesc în orice relație de congruență care structurează $\mathbf{T}_\Sigma(X)$ ca o algebră cât din $IS(\mathcal{K})$;

$$3. \quad \sim^{\mathcal{K}} = (\sim_s^{\mathcal{K}} \mid s \in S), \text{ unde}$$

$$\sim_s^{\mathcal{K}} = \{(t, t') \in T_\Sigma(X)_s^2 \mid (\forall \mathbf{A} \in \mathcal{K})(\forall \gamma : X \rightarrow A)(\bar{\gamma}_s(t) = \bar{\gamma}_s(t'))\},$$

pentru orice $s \in S$.

$$\text{Evident, putem scrie } \sim^{\mathcal{K}} = \bigcap \{ \ker(\bar{\gamma}) \mid \gamma \in \Gamma(X, \mathbf{A}), \mathbf{A} \in \mathcal{K} \}.$$

$\sim^{\mathcal{K}}$ pune în relație toți termii care se interpretează la fel, prin asignări, în orice algebră din \mathcal{K} .

Vom face câteva remarcări generale, cu caracter intuitiv, asupra acestor relații, după care vom demonstra unele proprietăți de bază ale lor. Cum orice atribuire conduce la un homomorfism și orice homomorfism provine dintr-o atribuire (restrictia homomorfismului la X), relațiile $\equiv^{\mathcal{K}}$ și $\sim^{\mathcal{K}}$ trebuie să fie identice. Relația $\approx^{\mathcal{K}}$ este intersecția tuturor congruențelor $\ker(h)$ în $\mathbf{T}_\Sigma(X)$, pe cînd $\approx^{\mathcal{K}}$ este intersecția “tuturor” congruențelor în $\mathbf{T}_\Sigma(X)$. Însă orice congruență ρ în $\mathbf{T}_\Sigma(X)$ este de tipul $\ker(h)$ dacă alegem $h : \mathbf{T}_\Sigma(X) \rightarrow \mathbf{T}_\Sigma(X)/\rho$ dat prin

$$h_s(t) = [t]_{\rho_s},$$

pentru orice sort s și term t de sort s . Ca urmare, $\equiv^{\mathcal{K}}$ și $\approx^{\mathcal{K}}$ trebuie să fie și ele identice (cu cerința suplimentară ca $\mathbf{T}_\Sigma(X)/\rho$ să fie în \mathcal{K}).

Propoziția 8.7.3.1. Fie $\mathbf{T}_\Sigma(X)$ o algebră de termi și \mathcal{K} o clasă de Σ -algebrelle. Atunci, au loc următoarele proprietăți:

$$(1) \quad \equiv^{\mathcal{K}}, \approx^{\mathcal{K}} \text{ și } \sim^{\mathcal{K}} \text{ sunt congruențe în } \mathbf{T}_\Sigma(X);$$

$$(2) \quad \equiv^{\mathcal{K}} = \approx^{\mathcal{K}} = \sim^{\mathcal{K}}.$$

Demonstrație. (1) Toate aceste relații sunt intersecții de congruențe în $\mathbf{T}_\Sigma(X)$ și, deci, sunt congruențe în $\mathbf{T}_\Sigma(X)$.

(2) Arătăm că are loc $\equiv^{\mathcal{K}} = \approx^{\mathcal{K}}$. Pentru aceasta este suficient să observăm că orice homomorfism $h : T_\Sigma(X) \rightarrow A$, unde $\mathbf{A} \in \mathcal{K}$, este extensia homomorfă a atribuirii $\gamma_h : X \rightarrow A$ dată prin $\gamma_h = h|_X$.

Arătăm că are loc $\approx^{\mathcal{K}} = \sim^{\mathcal{K}}$. Pentru orice homomorfism $h : T_\Sigma(X) \rightarrow A$, unde $\mathbf{A} \in \mathcal{K}$, $\ker(h)$ este congruență în algebră $\mathbf{T}_\Sigma(X)$ și

$$\mathbf{T}_\Sigma(X)/\ker(h) \cong h(\mathbf{T}_\Sigma(X)).$$

Cum $h(\mathbf{T}_\Sigma(X)) \leq \mathbf{A}$, deducem că are loc $\mathbf{T}_\Sigma(X)/\ker(h) \in IS(\mathcal{K})$. Deci $\approx^{\mathcal{K}} \subseteq \equiv^{\mathcal{K}}$.

Reciproc, dacă ρ este o congruență în $\mathbf{T}_\Sigma(X)$, funcția $h : T_\Sigma(X) \rightarrow T_\Sigma(X)/\rho$ dată prin

$$h_s(t) = [t]_{\rho_s},$$

pentru orice sort s și term t de sort s , este homomorfism ce satisfacă $\ker(h) = \rho$. Dacă presupunem în plus că $\mathbf{T}_\Sigma(X)/\rho \in IS(\mathcal{K})$, iar $\mathbf{A} \in \mathcal{K}$ și $f : \mathbf{T}_\Sigma(X)/\rho \rightarrow \mathbf{A}$ este un monomorfism, atunci $f \circ h$ este homomorfism de la $\mathbf{T}_\Sigma(X)$ la \mathbf{A} ce verifică $\ker(f \circ h) = \rho$. Deci orice congruență ρ ce intervene în definirea relației $\approx^{\mathcal{K}}$ este de tipul $\ker(h')$, unde h' este un homomorfism de la $\mathbf{T}_\Sigma(X)$ la o algebră din \mathcal{K} . Deci $\equiv^{\mathcal{K}} \subseteq \approx^{\mathcal{K}}$. \square

Lema 8.7.3.1. Fie $\mathbf{T}_\Sigma(X)$ o algebră de termi și \mathcal{K} o clasă de algebrelle. Atunci $\mathbf{T}_\Sigma(X)/\equiv^{\mathcal{K}}$ satisfacă proprietatea de universalitate relativ la $X/\equiv^{\mathcal{K}}$ și \mathcal{K} (reamintim că pentru orice sort s , $X_s/\equiv_s^{\mathcal{K}}$ este mulțimea tuturor claselor de echivalență din $T_\Sigma(X)_s/\equiv_s^{\mathcal{K}}$ ce au cel puțin un reprezentant în X_s).

Demonstrație. Fie $\mathbf{A} \in \mathcal{K}$ și $f : X/\equiv^{\mathcal{K}} \rightarrow \mathbf{A}$. Va trebui să arătăm că există un unic homomorfism $h : T_\Sigma(X) \rightarrow \mathbf{A}$ ce satisfacă $f = h \circ \varphi$, unde φ este funcția incluziune de la $X/\equiv^{\mathcal{K}}$ la $\mathbf{T}_\Sigma(X)/\equiv^{\mathcal{K}}$.

Considerăm $\alpha : X \rightarrow X/\equiv^{\mathcal{K}}$ dată prin $\alpha_s(x) = [x]_{\equiv_s^{\mathcal{K}}}$, pentru orice sort s și $x \in X_s$, și funcția $f' = f \circ \alpha$ (a se vedea diagrama din Figura 8.17).

$$\begin{array}{ccccc} X & \xrightarrow{\alpha} & X/\equiv^{\mathcal{K}} & \xrightarrow{\varphi} & T_\Sigma(X)/\equiv^{\mathcal{K}} \\ & \searrow f' & \downarrow f & \nearrow h & \\ & & A & & \end{array}$$

Figura 8.17: Demonstrația Lemei 8.7.3.1

Vom arăta că funcția h dată prin

$$h_s([t]_{\equiv_s^{\mathcal{K}}}) = \bar{f}'_s(t),$$

pentru orice sort s și term t de sort s , satisfacă lema:

1. h este bine definită. Dacă $t \equiv_s^{\mathcal{K}} t'$, atunci $t \sim_s^{\mathcal{K}} t'$, și deci $\bar{f}'_s(t) = \bar{f}'_s(t')$, deoarece f' este o atribuire;
2. h este homomorfism. Pentru orice termi t_i de sort s_i , $1 \leq i \leq n$, și orice $\sigma \in \Sigma_{s_1 \dots s_n, s}$, au loc relațiile:

$$\begin{aligned} h_s(\sigma^{T_\Sigma(X)/\equiv^{\mathcal{K}}}([t_1]_{\equiv_{s_1}^{\mathcal{K}}}, \dots, [t_n]_{\equiv_{s_n}^{\mathcal{K}}})) &= h_s([\sigma(t_1, \dots, t_n)]_{\equiv_s^{\mathcal{K}}}) \\ &= \bar{f}'_s(\sigma(t_1, \dots, t_n)) \\ &= \bar{f}'_s(\sigma^{T_\Sigma(X)}(t_1, \dots, t_n)) \\ &= \sigma^A(\bar{f}'_{s_1}(t_1), \dots, \bar{f}'_{s_n}(t_n)) \\ &= \sigma^A(h_{s_1}([t_1]_{\equiv_{s_1}^{\mathcal{K}}}), \dots, h_{s_n}([t_n]_{\equiv_{s_n}^{\mathcal{K}}})), \end{aligned}$$

ceea ce arată că h este homomorfism;

3. h satisfacă $f = h \circ \varphi$. Pentru orice sort s și $x \in X_s$ au loc relațiile

$$h_s([x]_{\equiv_s^{\mathcal{K}}}) = \bar{f}'_s(x) = f'_s(x) = f_s(\alpha_s(x)) = f_s([x]_{\equiv_s^{\mathcal{K}}}),$$

ceea ce arată că $f = h \circ \varphi$;

4. h este unic cu proprietățile de mai sus. Algebra $T_\Sigma(X)/\equiv^{\mathcal{K}}$ este generată de $X/\equiv^{\mathcal{K}}$ (Propoziția 8.4.1.4) și h extinde f . Conform Corolarului 8.5.1.3, h este unic.

Demonstrația este încheiată. \square

Lema 8.7.3.2. Fie $T_\Sigma(X)$ o algebră de termi și \mathcal{K} o clasă de algebrelor. Dacă clasa \mathcal{K} satisfacă $\mathcal{K} = ISP(\mathcal{K})$, atunci $T_\Sigma(X)/\equiv^{\mathcal{K}} \in \mathcal{K}$.

Demonstrație. Conform Propoziției 8.7.3.1, $T_\Sigma(X)/\equiv^{\mathcal{K}} = T_\Sigma(X)/\approx^{\mathcal{K}}$. Notăm $\approx^{\mathcal{K}} = \bigcap_{i \in I} \rho^i$, și fie h funcția de la $T_\Sigma(X)/\approx^{\mathcal{K}}$ la $\prod_{i \in I} T_\Sigma(X)/\rho^i$ dată prin:

$$h_s([t]_{\approx_s^{\mathcal{K}}})(i) = [t]_{\rho^i_s},$$

pentru orice sort s , term t de sort s și $i \in I$.

Este trivial de verificat că h este bine definită și, în plus, este monomorfism. Atunci are loc:

$$T_\Sigma(X)/\equiv^{\mathcal{K}} = T_\Sigma(X)/\approx^{\mathcal{K}} \cong h(T_\Sigma(X)/\approx^{\mathcal{K}}) \leq \prod_{i \in I} T_\Sigma(X)/\rho^i,$$

ceea ce ne arată că $T_\Sigma(X)/\equiv^{\mathcal{K}} \in \mathcal{K}$. \square

Următoarea teoremă este datorată lui Birkhoff [10].

Teorema 8.7.3.1. (Teorema de existență a algebrelor libere)

Orice clasă \mathcal{K} de algebrelor încisă la subalgebrelor, izomorfisme și produse directe admite algebrelor \mathcal{K} -libere.

Demonstrație. Fie \mathcal{K} o clasă de algebrelor cu proprietatea $\mathcal{K} = ISP(\mathcal{K})$. Conform Lemelor 8.7.3.1 și 8.7.3.2, algebra $T_\Sigma(X)/\equiv^{\mathcal{K}} \in \mathcal{K}$ este \mathcal{K} -liberă relativ la $X/\equiv^{\mathcal{K}}$, pentru orice X . \square

Corolarul 8.7.3.1. (Teorema de existență a algebrelor inițiale)

Orice clasă \mathcal{K} de algebrelor încisă la subalgebrelor, izomorfisme și produse directe admite algebrelor \mathcal{K} -inițiale.

Demonstrație. În demonstrația Teoremei 8.7.3.1 se consideră $X = \emptyset$. \square

8.8. Logică ecuațională

Termii sunt structuri sintactice prin care putem manipula obiecte sau combinații de obiecte ale unei algebrelor. Ecuațiile sunt perechi de termi ce definesc proprietăți ale obiectelor sau operațiilor unei algebrelor. De exemplu, validitatea unei ecuații de forma $x + y = y + x$ într-un grup aditiv ne spune că acest grup este comutativ.

Pentru a putea raționa într-un mod corespunzător cu termii și ecuații avem nevoie de un formalism adecvat prin care să putem exprima ce înseamnă că o ecuație este satisfăcută într-o algebră și cum putem deduce noi ecuații (proprietăți) de la ecuații (proprietăți) date (de exemplu, $x \cdot 0 = 0$ este o ecuație validă în orice inel și această ecuație se poate obține “sintactic” de la ecuațiile ce definesc inelul).

8.8.1. Ecuații și modele

In tot ceea ce urmează în această secțiune, atunci când se va considera o familie S -sortată de variabile, se va presupune că aceasta este o familie disjunctă, nevidă și disjunctă de orice semnatură S -sortată.

Definiția 8.8.1.1. Fie Σ o semnatură și X o familie de variabile. Se numește *ecuație de sort s peste Σ și X* , unde $s \in S$, orice pereche de termi (t, t') de sort s .

Atunci când termii unei ecuații sunt termi de bază, vom mai spune că ecuația este *ecuație de bază*.

Ecuațiile (t, t') se mai notează și prin $t = t'$ însă, în acest caz, cititorul nu trebuie să confundă ecuația cu o egalitate (identitate) de termi.

Vom nota prin $Eq(\Sigma, X)_s$ mulțimea tuturor ecuațiilor de sort s peste Σ și X , și prin $Eq(\Sigma, X) = (Eq(\Sigma, X)_s | s \in S)$ familia acestor mulțimi de ecuații.

Observația 8.8.1.1. Unii autori notează ecuațiile prin $(\forall X)t = t'$, specificând astfel și familia de variabile peste care se consideră ecuația. Cum în abordarea noastră

X este considerată dată și fixată pentru tot ceea ce urmează mai departe, putem renunța la " $(\forall X)$ " scriind ecuațiile în forma simplificată " $t = t'$ ". Mai mult, vom specifica clar acolo unde este necesar peste ce signatură și familie de variabile se consideră mulțimea de ecuații sau ecuația în cauză.

Definiția 8.8.1.2. Fie Σ o signatură și X o familie de variabile. Spunem că o ecuație $t = t'$ de sort s este *validă* în algebra \mathbf{A} , și notăm $\mathbf{A} \models t = t'$, dacă $\bar{\gamma}_s(t) = \bar{\gamma}_s(t')$, pentru orice $\gamma \in \Gamma(X, \mathbf{A})$.

Dacă $\mathbf{A} \models t = t'$, atunci vom mai spune că algebra \mathbf{A} este *model* al ecuației $t = t'$. Clasa tuturor modelelor unei mulțimi E de ecuații va fi notată prin $Mod(E)$. Adică, $Mod(E)$ este colecția tuturor algebrelor \mathbf{A} (de aceeași signatură Σ) ce satisfac $\mathbf{A} \models t = t'$, pentru orice $t = t' \in E$.

Vom folosi și notația $\mathbf{A} \models E$ ($\mathcal{K} \models E$) pentru a denota faptul că \mathbf{A} (algebrele din \mathcal{K}) este model a (sunt modele ale) tuturor ecuațiilor din E .

Definiția 8.8.1.3. O clasă \mathcal{K} de algebri se numește *clasă ecuațională* dacă există o mulțime E de ecuații astfel încât $\mathcal{K} = Mod(E)$.

Dacă \mathcal{K} este clasă ecuațională, atunci orice mulțime E de ecuații ce satisfac $\mathcal{K} = Mod(E)$ se numește *mulțime de axiome a clasei \mathcal{K}* , iar \mathcal{K} se mai spune că este *axiomatizabilă prin E* .

Exemplul 8.8.1.1. Fie signatura Σ din Exemplul 8.2.1(1) și $X = \{x, y, z, \dots\}$ o mulțime de variabile disjunctă de Σ .

Următoarele perechi de termi sunt ecuații (de sort s):

1. $\sigma_1(\sigma_1(x, y), z) = \sigma_1(x, \sigma_1(y, z));$
2. $\sigma_1(x, \sigma_3) = x$ și $\sigma_1(\sigma_3, x) = x;$
3. $\sigma_1(x, \sigma_2(x)) = \sigma_3$ și $\sigma_1(\sigma_2(x), x) = \sigma_3;$
4. $\sigma_1(x, y) = \sigma_1(y, x);$
5. $\sigma_4(\sigma_4(x, y), z) = \sigma_4(x, \sigma_4(y, z));$
6. $\sigma_4(x, \sigma_1(y, z)) = \sigma_1(\sigma_4(x, y), \sigma_4(x, z))$ și
 $\sigma_4(\sigma_1(x, y), z) = \sigma_1(\sigma_4(x, z), \sigma_4(y, z)).$

Este clar că ecuațiile de mai sus sunt valide în orice inel. Ca urmare, clasa inelelor este o clasă ecuațională, fiind axiomatizabilă prin mulțimea de ecuații de mai sus.

Definiția 8.8.1.4. Fie E o mulțime de ecuații peste Σ și X . Spunem că ecuația $t = t'$ este *consecință semantică* a mulțimii E de ecuații, și notăm $E \models t = t'$, dacă $Mod(E) \models t = t'$.

Exemplul 8.8.1.2. Fie E mulțimea de ecuații din Exemplul 8.8.1.1. Atunci ecuația $\sigma_4(x, \sigma_3) = \sigma_3$ este consecință semantică a acestei mulțimi E de ecuații (a se vedea Propoziția 5.1.1(1)).

Observația 8.8.1.2. Fie $t = t'$ o ecuație de sort s peste Σ și X , \mathbf{A} o Σ -algebră și $\gamma : X \rightarrow A$ este o asignare.

Considerăm signatura $\Sigma \cup X$ obținută prin adăugarea elementelor familiei X la Σ , fiecare variabilă fiind privită ca o constantă de sort corespunzător. Sub asignarea γ , algebra \mathbf{A} poate fi privită acum ca o $(\Sigma \cup X)$ -algebră considerând că, pentru orice sort s și $x \in X_s$, x^A este elementul $\gamma_s(x) \in A_s$. În acest context vom nota algebra \mathbf{A} prin \mathbf{A}^γ . Mai mult, peste signatura $\Sigma \cup X$, termii t și t' sunt termi fără variabile (adică, elemente ale algebrei $\mathbf{T}_{\Sigma \cup X}$). Dacă $eval_{\mathbf{A}^\gamma}$ este unicul homomorfism de la $\mathbf{T}_{\Sigma \cup X}$ la \mathbf{A}^γ , atunci obținem următoarele echivalențe:

$$\bar{\gamma}_s(t) = \bar{\gamma}_s(t') \Leftrightarrow eval_{\mathbf{A}^\gamma}(t) = eval_{\mathbf{A}^\gamma}(t') \Leftrightarrow \mathbf{A}^\gamma \models t = t'.$$

Pentru a evita orice ambiguitate posibilă vom folosi notația $\models_{\Sigma, X}$ pentru a specifica clar că în stânga și în dreapta ei se găsesc structuri (algebre, ecuații) peste Σ și X . Atunci, echivalențele de mai sus conduc la

$$\mathbf{A} \models_{\Sigma, X} t = t' \Leftrightarrow (\forall \gamma : X \rightarrow A)(\mathbf{A}^\gamma \models_{\Sigma \cup X, \emptyset} t = t').$$

Este interesant de remarcat că pentru orice $(\Sigma \cup X)$ -algebră \mathbf{B} există o Σ -algebră \mathbf{A} și o asignare $\gamma : X \rightarrow A$ astfel încât $\mathbf{B} = \mathbf{A}^\gamma$ și

$$\mathbf{B} \models_{\Sigma \cup X, \emptyset} t = t' \Leftrightarrow \bar{\gamma}_s(t) = \bar{\gamma}_s(t').$$

In adevară, dacă considerăm $A_s = B_s$, pentru orice s , $\sigma^A = \sigma^B$, pentru orice $\sigma \in \Sigma$, și $\gamma_s(x) = x^B$, pentru orice sort s și $x \in X_s$, atunci are loc echivalența de mai sus.

Fie \tilde{X} o copie a familiei X de variabile, disjunctă de Σ (adică, $\tilde{X}_s = \{\tilde{x} | x \in X_s\}$, pentru orice $s \in S$). Vom nota prin \tilde{E} mulțimea de ecuații obținute din E prin înlocuirea fiecărei variabile x din aceste ecuații prin variabila \tilde{x} . Atunci, combinând relațiile de mai sus, obținem:

$$Mod_{\Sigma, X}(E) \models_{\Sigma, X} t = t' \Leftrightarrow Mod_{\Sigma \cup X, \tilde{X}}(\tilde{E}) \models_{\Sigma \cup X, \tilde{X}} t = t'$$

$(Mod_{\Sigma, X}(E)$ în stânga echivalenței reprezintă clasa tuturor modelelor ecuațiilor din E private ca ecuații peste Σ și X , în timp ce $Mod_{\Sigma \cup X, \tilde{X}}(\tilde{E})$ în dreapta echivalenței reprezintă clasa tuturor modelelor ecuațiilor din E private ca ecuații peste $\Sigma \cup X$ și \tilde{X}). Rescriind această echivalență obținem:

$$E \models_{\Sigma, X} t = t' \Leftrightarrow \tilde{E} \models_{\Sigma \cup X, \tilde{X}} t = t'$$

(în stânga echivalenței, ecuațiile din E sunt peste Σ și X , iar în dreapta echivalenței, ecuațiile din \tilde{E} sunt peste $\Sigma \cup X$ și \tilde{X}).

Rezultatul obținut în Observația 8.8.1.2 constituie subiectul următoarei teoreme.

Teorema 8.8.1.1. (Teorema constantelor)

Fie E o mulțime de ecuații și $t = t'$ o ecuație, ambele peste Σ și X . Atunci,

$$E \models_{\Sigma, X} t = t' \Leftrightarrow \tilde{E} \models_{\Sigma \cup X, \tilde{X}} t = t'.$$

Următoarele două probleme sunt considerate probleme centrale ale logicii ecuaționale:

1. *Problema deducției.* Există un sistem de reguli de deducție (inferență) care să permită generarea de consecințe semantice ale unei mulțimi E de ecuații?

Dacă un astfel de sistem de deducție există, atunci el va fi numit *corect* dacă nu generează ecuații ce nu sunt consecințe semantice ale mulțimii E de ecuații, și *complet*, dacă poate genera orice consecință semantică a mulțimii E de ecuații.

2. *Problema axiomatizării.* Ce clase de algebri pot fi axiomatizate?

Vom da răspuns la aceste două probleme în secțiunile următoare.

8.8.2. Deducție ecuațională

Regulile de deducție pe care le vom prezenta în cele ce urmează, numite și regulile de deducție ale logicii ecuaționale, sunt bazate pe proprietățile egalității: reflexivitate, simetrie, tranzitivitate și substituție.

Definiția 8.8.2.1. Fie Σ o signură și X o familie de variabile. Următoarele reguli de deducție sunt numite *regulile de deducție ale logicii ecuaționale* peste Σ și X :

1. (reflexivitate)

$$\overline{t = t}$$

pentru orice sort s și $t \in T_{\Sigma}(X)_s$;

2. (simetrie)

$$\frac{t = t'}{t' = t}$$

pentru orice sort s și $t, t' \in T_{\Sigma}(X)_s$;

3. (tranzitivitate)

$$\frac{t = t', t' = t''}{t = t''}$$

pentru orice sort s și $t, t', t'' \in T_{\Sigma}(X)_s$;

4. (substituție)

$$\frac{t = t', t_0 = t_1}{t[x/t_0] = t'[x/t_1]}$$

pentru orice $s, s' \in S$, $t, t' \in T_{\Sigma}(X)_s$, $x \in X_{s'}$ și $t_0, t_1 \in T_{\Sigma}(X)_{s'}$.

Prima regulă de deducție ne spune că ecuația $t = t$ poate fi dedusă întotdeauna. Cea de a doua regulă ne spune că de la ecuația $t = t'$ putem deduce $t' = t$. În mod similar se interpretează și celelalte două reguli.

Definiția 8.8.2.2. Fie Σ o signură, X o familie de variabile, E o mulțime de ecuații și e o ecuație. Se numește *deducție a ecuației e de la mulțimea E de ecuații* orice secvență finită de ecuații

$$e_1, \dots, e_n$$

ce satisfac:

1. $e = e_n$;
2. pentru orice i , $1 \leq i \leq n$, are loc una din următoarele proprietăți:

(a) $e_i \in E$, sau

(b) e_i este obținută de la ecuații precedente din sir folosind una din regulile de deducție ale logicii ecuaționale. De exemplu, folosirea reflexivității înseamnă că e_i este de forma $t = t$, iar folosirea regulii substituției înseamnă că există $j, k < i$ astfel încât e_j este de forma $t = t'$, e_k este de forma $t_0 = t_1$, iar e_i este de forma $t[x/t_0] = t'[x/t_1]$, unde x este o variabilă de același sort ca și t_0 și t_1 .

Definiția 8.8.2.3. Fie E o mulțime de ecuații și e o ecuație. Spunem că e este consecință sintactică a mulțimii E de ecuații, și notăm $E \vdash e$, dacă există o deducție a ecuației e de la mulțimea E de ecuații.

Exemplul 8.8.2.1. Rescriem ecuațiile din Exemplul 8.8.1.1 folosind notații infix și simboluri uzuale pentru operațiile unui inel:

$$(i1) x + (y + z) = (x + y) + z;$$

$$(i2) x + 0 = x \text{ și } 0 + x = x;$$

$$(i3) x + (-x) = 0 \text{ și } (-x) + x = 0;$$

$$(i4) x + y = y + x;$$

$$(i5) x \cdot (y \cdot z) = (x \cdot y) \cdot z;$$

$$(i6) x \cdot (y + z) = x \cdot y + x \cdot z \text{ și } (x + y) \cdot z = x \cdot y + y \cdot z.$$

Vom arăta că ecuația $x \cdot 0 = 0$ este deducibilă de la $E = \{i1, \dots, i6\}$:

$$(e1) 0 = 0 \\ (\text{în baza reflexivității})$$

$$(e2) x = x + 0 \\ (\text{de la (i2), folosind simetrie})$$

- (e3) $0 = 0 + 0$
(de la (e2), folosind substituție cu (e1))
- (e4) $x \cdot y = x \cdot y$
(în baza reflexivității)
- (e5) $x \cdot 0 = x \cdot (0 + 0)$
(de la (e4), folosind substituție cu (e3))
- (e6) $x \cdot (0 + z) = x \cdot 0 + x \cdot z$
(de la (i6), folosind substituție cu (e1))
- (e7) $x \cdot (0 + 0) = x \cdot 0 + x \cdot 0$
(de la (e6), folosind substituție cu (e1))
- (e8) $x \cdot 0 = x \cdot 0 + x \cdot 0$
(de la (e5) și (e7), folosind tranzitivitate)
- (e9) $-x \cdot 0 + y = -x \cdot 0 + y$
(în baza reflexivității)
- (e10) $-x \cdot 0 + x \cdot 0 = -x \cdot 0 + (x \cdot 0 + x \cdot 0)$
(de la (e9), folosind substituție cu (e8))
- (e11) $x \cdot 0 = x \cdot 0$
(în baza reflexivității)
- (e12) $-x \cdot 0 + x \cdot 0 = 0$
(de la (i3), folosind substituție cu (e11))
- (e13) $0 = -x \cdot 0 + x \cdot 0$
(de la (e12), folosind simetrie)
- (e14) $0 = -x \cdot 0 + (x \cdot 0 + x \cdot 0)$
(de la (e13) și (e10), folosind tranzitivitate)
- (e15) $-x \cdot 0 = -x \cdot 0$
(în baza reflexivității)
- (e16) $-x \cdot 0 + (y + z) = (-x \cdot 0 + y) + z$
(de la (i1), folosind substituție cu (e15))
- (e17) $-x \cdot 0 + (x \cdot 0 + z) = (-x \cdot 0 + x \cdot 0) + z$
(de la (e16), folosind substituție cu (e11))
- (e18) $-x \cdot 0 + (x \cdot 0 + x \cdot 0) = (-x \cdot 0 + x \cdot 0) + x \cdot 0$
(de la (e17), folosind substituție cu (e11))
- (e19) $0 = (-x \cdot 0 + x \cdot 0) + x \cdot 0$
(de la (e18) și (e14), folosind tranzitivitate)

- (e20) $y + x \cdot 0 = y + x \cdot 0$
(în baza reflexivității)
- (e21) $(-x \cdot 0 + x \cdot 0) + x \cdot 0 = 0 + x \cdot 0$
(de la (e20), folosind substituție cu (e12))
- (e22) $0 + x \cdot 0 = x \cdot 0$
(de la (i2), folosind substituție cu (e11))
- (e23) $(-x \cdot 0 + x \cdot 0) + x \cdot 0 = x \cdot 0$
(de la (e21) și (e22), folosind tranzitivitate)
- (e24) $0 = x \cdot 0$
(de la (e19) și (e23), folosind tranzitivitate)
- (e25) $x \cdot 0 = 0$
(de la (e24), folosind simetrie).

Dacă analizăm cu atenție Exemplul 8.8.2.1, constatăm că anumiți pași de substituție, cel puțin din punct de vedere conceptual, pot fi înglobați într-un singur pas. De exemplu, pașii (e16), (e17) și (e18) ar putea fi condensați în unul singur,

$$-x \cdot 0 + (x \cdot 0 + x \cdot 0) = (-x \cdot 0 + x \cdot 0) + x \cdot 0,$$

ca fiind obținut direct de la (i1) prin aplicarea unei substituții. Dar, pentru a putea face aceasta avem nevoie de următorul rezultat.

Propoziția 8.8.2.1. Fie Σ o signatură, X o familie de variabile, E o mulțime de ecuații, t și t' termi de sort s , $x_i \in \text{Var}(t)_{s_i} \cup \text{Var}(t')_{s_i}$ variabilă de sort s_i , iar t_i și t'_i termi de sort s_i , pentru orice $1 \leq i \leq n$. Dacă variabilele x_1, \dots, x_n sunt distințe două câte două, $E \vdash t = t'$ și $E \vdash t_i = t'_i$, pentru orice $1 \leq i \leq n$, atunci

$$E \vdash t[x_1/t_1, \dots, x_n/t_n] = t'[x_1/t'_1, \dots, x_n/t'_n].$$

Demonstrație. Dacă $n = 1$, atunci propoziția urmează direct de la regula substituției.

Prin inducție matematică, ne putem restrânge la cazul $n = 2$. Ideea de bază este de a aplica substituție cu x_1 și $t_1 = t'_1$, și apoi cu x_2 și $t_2 = t'_2$. Aceasta este posibil dacă x_2 nu este variabilă în t_1 sau t'_1 ; altfel, termul $t[x_1/t_1][x_2/t_2]$ poate fi diferit de $t[x_1/t_1, x_2/t_2]$. Însă, mergând pe aceeași idee, putem redenumi variabilele lui t_1 și t'_1 astfel încât acestea să fie diferențiate de toate variabilele din t , t' , t_2 și t'_2 , aplicând succesiv cele două substituții și, în final, redenumim variabilele rezultatului aşa cum au fost ele inițial.

Fie $x \in \text{Var}(t_1)_{s'} \cup \text{Var}(t'_1)_{s'}$ o variabilă de sort s' . Dacă ea apare în t , t' , t_2 sau t'_2 , atunci considerăm o nouă variabilă y de sort s' ce nu este în t , t' , t_1 , t'_1 , t_2 și t'_2 . Redenumirea variabilei x în y se face aplicând regula substituției pentru $t_1 = t'_1$ cu x și $y = y$ (astfel, obținem $E \vdash t_1[x/y] = t'_1[x/y]$).

Fie \bar{t}_1 și \bar{t}'_1 termii obținuți prin redenumirea variabilelor acestora aşa cum s-a menționat mai sus. Aplicând regula substituției de două ori, obținem

$$E \vdash t[x_1/\bar{t}_1][x_2/t_2] = t'[x_1/\bar{t}'_1][x_2/t'_2].$$

Redenumind înapoi variabilele aşa cum au fost ele la început, obținem

$$E \vdash t[x_1/t_1, x_2/t_2] = t'[x_1/t'_1, x_2/t'_2],$$

ceea ce încheie demonstrația propoziției. \square

Corolarul 8.8.2.1. Fie Σ o semnatură, X o familie de variabile, E o mulțime de ecuații, s un sort, iar t și t' termi de sort s . Atunci, au loc următoarele proprietăți:

- (1) dacă $E \vdash t = t'$ și f este o substituție, atunci $E \vdash f_s(t) = f_s(t')$;
- (2) dacă f și g sunt substituții cu proprietatea $E \vdash f_{s'}(x) = g_{s'}(x)$, pentru orice sort s' și $x \in \text{Var}(t)_{s'} \cup \text{Var}(t')_{s'}$, atunci $E \vdash f_s(t) = g_s(t)$.

Demonstrație. Direct de la Propoziția 8.8.2.1. \square

Definiția 8.8.2.4. Fie \mathbf{A} o algebră și ρ o congruență în \mathbf{A} . Spunem că ρ este *congruență total invariantă* dacă pentru orice endomorfism $h : A \rightarrow A$ are loc

$$(\forall s \in S)(\forall a, b \in A_s)(a \rho_s b \Rightarrow h_s(a) \rho_s h_s(b)).$$

Dată o mulțime E de ecuații peste Σ și X , E induce o relație binară pe $T_\Sigma(X)$,

$$=_E = (=_{E,s} | s \in S),$$

prin

$$t =_{E,s} t' \Leftrightarrow E \vdash t = t',$$

pentru orice sort s și termi t și t' de sort s .

Putem arăta că $=_E$ este congruență total invariantă.

Propoziția 8.8.2.2. Dată o mulțime E de ecuații peste Σ și X , relația $=_E$ este congruență total invariantă.

Demonstrație. Faptul că $=_E$ este relație de echivalență se obține imediat de la regulile de reflexivitate, simetrie și tranzitivitate ale logicii ecuaționale.

Să presupunem că are loc $t_i =_{E,s_i} t'_i$, unde t_i și t'_i sunt termi de sort s_i , pentru orice $1 \leq i \leq n$ ($n \geq 1$). Fie $\sigma \in \Sigma_{s_1 \dots s_n, s}$. Considerăm termul $t = \sigma(x_1, \dots, x_n)$, unde x_i este variabilă de sort s_i , pentru orice $1 \leq i \leq n$, iar x_1, \dots, x_n sunt distințe două câte două. Fie substituțiile f și g date prin $f_{s_i}(x_i) = t_i$ și $g_{s_i}(x_i) = t'_i$, pentru orice $1 \leq i \leq n$. In baza Corolarului 8.8.2.1(2) obținem

$$E \vdash \sigma(t_1, \dots, t_n) = \sigma(t'_1, \dots, t'_n),$$

ceea ce ne arată că $=_E$ este o congruență.

Corolarul 8.8.2.1(1) ne spune că $=_E$ este închisă la substituții. Ca urmare, $=_E$ este congruență total invariantă. \square

Demonstrația Propoziției 8.8.2.2 arată necesitatea cerinței $X_s \neq \emptyset$, pentru orice sort s , considerată la începutul Secțiunii 8.8.

Importanța congruențelor total invariante constă în următoarele.

Propoziția 8.8.2.3. Fie ρ o congruență în algebra $\mathbf{T}_\Sigma(X)$ și $t = t'$ o ecuație de sort s , unde $s \in S$. Dacă ρ este total invariantă, atunci are loc

$$\mathbf{T}_\Sigma(X)/\rho \models t = t' \Leftrightarrow (t, t') \in \rho_s.$$

Demonstrație. Fie ρ o congruență în algebra $\mathbf{T}_\Sigma(X)$. Facem întâi observația că orice atribuire

$$f : X \rightarrow T_\Sigma(X)/\rho$$

poate fi descompusă $f = h \circ g$, unde $g : X \rightarrow T_\Sigma(X)$ este dată prin $g_s(x) = t$, iar $h : T_\Sigma(X) \rightarrow T_\Sigma(X)/\rho$ este dată prin $h_s(t) = [t]_{\rho_s}$, pentru orice sort s și $x \in X_s$, presupunând că $f_s(x) = [t]_{\rho_s}$.

Să presupunem acum că are loc

$$\mathbf{T}_\Sigma(X)/\rho \models t = t'.$$

Că urmare, $\bar{f}_s(t) = \bar{f}_s(t')$, pentru orice atribuire $f : X \rightarrow T_\Sigma(X)/\rho$. Utilizând o descompunere ca mai sus pentru f , obținem

$$[\bar{g}_s(t)]_{\rho_s} = [\bar{g}_s(t')]_{\rho_s}.$$

In particular, pentru g dată prin $g_s(x) = x$, obținem $[t]_{\rho_s} = [t']_{\rho_s}$, ceea ce arată că t și t' sunt în relația ρ_s .

Reciproc, presupunem că $(t, t') \in \rho_s$. Vom arăta că pentru orice $g : X \rightarrow T_\Sigma(X)$ are loc $[\bar{g}_s(t)]_{\rho_s} = [\bar{g}_s(t')]_{\rho_s}$. Aceasta este însă imediat deoarece ρ este congruență total invariantă.

Dacă considerăm $h : T_\Sigma(X) \rightarrow T_\Sigma(X)/\rho$ dată ca mai sus, atunci obținem

$$h_s(\bar{g}_s(t)) = h_s(\bar{g}_s(t')).$$

Cum orice atribuire se poate scrie în forma $h \circ g$ (așa cum a fost arătat mai sus), urmează că are loc $\mathbf{T}_\Sigma(X)/\rho \models t = t'$. \square

Corolarul 8.8.2.2. Fie E o mulțime de ecuații peste Σ și X și $t = t'$ o ecuație. Atunci

$$\mathbf{T}_\Sigma(X)/=_E \models t = t' \Leftrightarrow E \vdash t = t'.$$

Demonstrație. De la Propozițiile 8.8.2.2 și 8.8.2.3. \square

Următorul rezultat fundamental stabilește corectitudinea și completitudinea logicii ecuaționale [10].

Teorema 8.8.2.1. (Corectitudinea și completitudinea logicii ecuaționale)

Fie E o mulțime de ecuații peste Σ și X și $t = t'$ o ecuație peste Σ și X . Atunci

$$E \models t = t' \Leftrightarrow E \vdash t = t'.$$

Demonstrație. Presupunem că are loc $E \models t = t'$. Cum $E \vdash E$, Corolarul 8.8.2.2 conduce la $\mathbf{T}_\Sigma(X)/=_E \models t = t'$ care, combinată cu ipoteza, furnizează

$$\mathbf{T}_\Sigma(X)/=_E \models t = t'.$$

Aplicând din nou Corolarul 8.8.2.2, obținem $E \vdash t = t'$.

Reciproc, presupunem că are loc $E \vdash t = t'$. Atunci există o deducție

$$e_1, \dots, e_n$$

de la E , unde e_n este ecuația $t = t'$. Vom arăta prin inducție că, pentru orice i , $Mod(E) \models e_i$. Fie $\mathbf{A} \in Mod(E)$. Considerăm următoarele cazuri:

- e_i se obține din ecuații precedente în sir, prin regula reflexivității, simetriei sau tranzitivității. Atunci, presupunând că aceste ecuații sunt valide în \mathbf{A} , obținem imediat că e_i este validă în \mathbf{A} ;
- e_i se obține din e_j și e_k prin regula substituției, unde $j, k < i$. Presupunem că e_j este o ecuație de sort s de forma $t_1 = t_2$, e_k este o ecuație de sort s' de forma $t_3 = t_4$, și fie x o variabilă de sort s' cu proprietatea că e_i este ecuația $t_1[x/t_3] = t_2[x/t_4]$. Presupunem că ecuațiile e_j și e_k sunt valide în \mathbf{A} .

Notăm prin f și g substituțiile date prin

$$f_{s'}(x) = t_3, \quad g_{s'}(x) = t_4 \quad \text{și} \quad f_{s''}(y) = g_{s''}(y) = y,$$

pentru orice s'' și y cu $s'' \neq s'$ sau $y \neq x$ (dar cel puțin una din ele). Atunci

$$t_1[x/t_3] = \bar{f}_s(t_1)$$

și

$$t_2[x/t_4] = \bar{g}_s(t_2).$$

Pentru orice atribuire $\gamma : X \rightarrow A$,

$$\bar{\gamma}_s(t_1[x/t_3]) = \bar{\gamma}_s(\bar{f}_s(t_1)) = \bar{\gamma}'_s(t_1)$$

și

$$\bar{\gamma}_s(t_2[x/t_4]) = \bar{\gamma}_s(\bar{g}_s(t_2)) = \bar{\gamma}''_s(t_2),$$

unde γ' și γ'' se obțin ca în Teorema 8.7.1.3. Arătăm că $\gamma' = \gamma''$, ceea ce va încheia demonstrația în baza faptului că $t_1 = t_2$ este ecuație validă în \mathbf{A} .

Au loc egalitățile:

$$\begin{aligned} \gamma'_{s'}(x) &= \bar{\gamma}_{s'}(f_{s'}(x)) \\ &= \bar{\gamma}_{s'}(t_3) \\ &= \bar{\gamma}_{s'}(t_4) \\ &= \bar{\gamma}_{s'}(g_{s'}(x)) \\ &= \gamma''_{s'}(x), \end{aligned}$$

egalitatea a treia urmând de la ipoteza inductivă conform căreia ecuația $t_3 = t_4$ este validă în \mathbf{A} . Pentru $s'' \neq s'$ sau $y \neq x$ (dar cel puțin una din ele) au loc egalitățile:

$$\begin{aligned} \gamma'_{s''}(y) &= \bar{\gamma}_{s''}(f_{s''}(y)) \\ &= \bar{\gamma}_{s''}(y) \\ &= \bar{\gamma}_{s''}(g_{s''}(y)) \\ &= \gamma''_{s''}(y). \end{aligned}$$

Deci, $\gamma' = \gamma''$.

Teorema este demonstrată. \square

Observația 8.8.2.1. Corolarul 8.8.2.1 permite împărțirea celei de a patra reguli de deducție a logicii ecuaționale în două reguli:

4. (congruență)

$$\frac{(\forall s)(\forall x \in X_s)(f_s(x) = g_s(x))}{f_s(t) = g_s(t)}$$

pentru orice substituții f și g , $s \in S$ și $t \in T_\Sigma(X)_s$;

5. (substituție)

$$\frac{t = t'}{f_s(t) = f_s(t')}$$

pentru orice sort s , ecuație $t = t'$ de sort s și substituție f .

Aceasta este varianta ce poate fi întâlnită în multe lucrări de specialitate, în special în lucrările membrilor grupului ADJ.

Observația 8.8.2.2. Teorema de corectitudine și completitudine poate fi extinsă la *ecuații conditionate*, adică ecuații ce se cer să fie satisfăcute atunci când anumite ecuații asociate lor sunt satisfăcute. Asupra acestui aspect vom reveni în paragraful dedicat algebrelor cu sorturi ordonate din Secțiunea 8.9.2.3.

8.8.3. Axiomatizare

Vom răspunde acum la cea de a doua întrebare pusă în Secțiunea 8.8.1.

Definiția 8.8.3.1. Se numește *varietate* orice clasă nevidă de algebrelor (de aceeași signatură) închisă la subalgebrelor, homomorfismele și produsele directe.

Teorema 8.8.3.1. Fie Σ o signatură, X o familie de variabile și E o mulțime de ecuații. Atunci, $Mod(E)$ este varietate.

Demonstrație. Dacă o ecuație este validă într-o algebră \mathbf{A} , atunci ea este validă și în orice subalgebră a algebrelor \mathbf{A} (asignările într-o subalgebră a algebrelor \mathbf{A} sunt asignări și în \mathbf{A}). Deci, $Mod(E)$ este închisă la subalgebrelor.

Dacă o ecuație este validă într-o algebră \mathbf{A} , atunci ea este validă și în orice imaginea homomorfă $h(\mathbf{A})$ a algebrelor \mathbf{A} (asignările în $h(\mathbf{A})$ se obțin din asignării în \mathbf{A} cărora li se aplică apoi h). Deci, $Mod(E)$ este închisă la homomorfismele.

Inchiderea la produs direct este puțin mai delicată. Fie $t = t' \in E$ o ecuație de sort s și $(\mathbf{A}^i | i \in I)$ o familie nevidă de algebrelor din $Mod(E)$. Fie $\gamma : X \rightarrow \prod_{i \in I} A^i$ o atribuire. Pentru a arăta că are loc $\bar{\gamma}_s(t) = \bar{\gamma}_s(t')$ va trebui să arătăm că are loc

$$\bar{\gamma}_s(t)(i) = \bar{\gamma}_s(t')(i),$$

pentru orice $i \in I$. Însă $\bar{\gamma}_s(t)(i)$ este $\bar{\gamma}_s^i(t)$, unde $\bar{\gamma}_s^i(x) = \gamma_s(x)(i)$, pentru orice $x \in X_s$ și $i \in I$. Cum algebra \mathbf{A}^i este model pentru $t = t'$, aceasta ne spune că are loc

$$\bar{\gamma}_s^i(t) = \bar{\gamma}_s^i(t'),$$

ceea ce conduce la $\bar{\gamma}_s(t) = \bar{\gamma}_s(t')$. Deci $Mod(E)$ este închisă la produse directe. \square

Corolarul 8.8.3.1. Fie Σ o signatură, X o familie de variabile și E o mulțime de ecuații. Atunci au loc următoarele proprietăți:

- (1) $\mathbf{T}_\Sigma(X)/=_E$ este algebră liberă în clasa $Mod(E)$;
- (2) $\mathbf{T}_\Sigma/=_E$ este algebră inițială în clasa $Mod(E)$

Demonstrație. (1) urmează direct de la Teorema 8.8.3.1 și Teorema 8.7.3.1, iar (2) de la Teorema 8.8.3.1 și Corolarul 8.7.3.1. \square

Prezentăm acum un rezultat fundamental datorat lui Birkhoff [10].

Teorema 8.8.3.2. (Teorema varietății)

O clasă de algebrelor (de aceeași signatură) este axiomatizabilă dacă și numai dacă este varietate.

Demonstrație. Dacă \mathcal{K} este axiomatizabilă, atunci Teorema 8.8.3.1 conduce direct la faptul că ea este varietate.

Reciproc, presupunem că \mathcal{K} este varietate. Fie X o familie de variabile astfel încât X_s este infinită, pentru orice sort s . Considerăm E mulțimea tuturor ecuațiilor valide în orice algebră $\mathbf{A} \in \mathcal{K}$ și arătăm că $\mathcal{K} = Mod(E)$, ceea ce ne va spune că \mathcal{K} este axiomatizabilă (prin E).

In primul rând remarcăm că mulțimea E de ecuații este “maximală” în sensul că pentru orice altă ecuație e peste Σ și Y , unde Y este o familie de variabile, are loc

$$(*) \quad \mathcal{K} \models e \Leftrightarrow Mod(E) \models e.$$

In adevăr, putem redenumi variabilele ecuației e astfel încât toate variabilele ei să fie în X ; fie e' o ecuație obținută în acest mod. Atunci este clar că $\mathcal{K} \models e$ dacă și numai dacă $\mathcal{K} \models e'$, și $Mod(E) \models e$ dacă și numai dacă $Mod(E) \models e'$. In plus, $\mathcal{K} \models e'$ dacă și numai dacă $Mod(E) \models e'$ (deoarece $e' \in E$).

Acum ne vom concentra spre a arăta că are loc $\mathcal{K} = Mod(E)$. Incluziunea $\mathcal{K} \subseteq Mod(E)$ urmează direct de la definiția mulțimii E de ecuații.

Fie acum $\mathbf{A} \in Mod(E)$. Pentru a arăta că $\mathbf{A} \in \mathcal{K}$ vom arăta că \mathbf{A} este imaginea homomorfă a unei algebrelor libere din \mathcal{K} . Cum \mathcal{K} este varietate, \mathbf{A} va fi în \mathcal{K} . Fie Y o familie de variabile astfel încât $|Y_s| \geq |A_s|$, pentru orice sort s , și fie f o funcție surjectivă (arbitrară) de la Y la \mathbf{A} . Această funcție se extinde la un unic homomorfism \bar{f} de la $\mathbf{T}_\Sigma(Y)$ la \mathbf{A} și, $\mathbf{A} = \bar{f}(\mathbf{T}_\Sigma(Y))$.

Considerăm congruența $\sim^{Mod(E)}$ indusă de $Mod(E)$ în $\mathbf{T}_\Sigma(Y)$. Atunci incluziunea

$$\sim^{Mod(E)} \subseteq \ker(\bar{f}),$$

în baza celei de a doua teoreme de homomorfism, conduce la:

$$\begin{aligned} \mathbf{A} &= \bar{f}(\mathbf{T}_\Sigma(Y)) \\ &\cong \mathbf{T}_\Sigma(Y)/\ker(\bar{f}) \\ &\cong (\mathbf{T}_\Sigma(Y)/\sim^{Mod(E)})/(\ker(\bar{f})/\sim^{Mod(E)}). \end{aligned}$$

Însă relația $(*)$ conduce la

$$(\mathbf{T}_\Sigma(Y)/\sim^{Mod(E)})/(\ker(\bar{f})/\sim^{Mod(E)}) \cong (\mathbf{T}_\Sigma(Y)/\sim^K)/(\ker(\bar{f})/\sim^K),$$

iar $\mathbf{T}_\Sigma(Y)/\sim^K$ este algebră \mathcal{K} -liberă. Mai mult,

$$h(\mathbf{T}_\Sigma(Y)/\sim^K) \cong (\mathbf{T}_\Sigma(Y)/\sim^K)/(\ker(\bar{f})/\sim^K),$$

unde h este homomorfismul dat prin

$$h_s([t]_{\sim^K}) = [[t]_{\sim^K}]_{\ker(\bar{f})_s/\sim^K},$$

pentru orice sort s și term t de sort s .

Toate acestea, cumulate, ne spun că algebra \mathbf{A} este izomorfă cu imaginea homomorfă a unei algebrelor \mathcal{K} -libre. Deci $\mathbf{A} \in \mathcal{K}$. \square

8.9. Aplicații: semantica limbajelor de programare și specificare algebrică a tipurilor abstracte de date

In această secțiune vom prezenta două aplicații majore ale algebrelor universale multisortate în informatică. Prima aplicație constă în utilizarea algebrelor multisortate în specificarea semanticii limbajelor de programare, iar a doua aplicație constă în utilizarea algebrelor universale multisortate ca formalism de specificare algebrică a tipurilor de date.

8.9.1. Semantica limbajelor de programare

În contextul utilizării algebrelor universale multisortate în specificarea semanticii unui limbaj de programare [72], semnatura Σ specifică sorturile și simbolurile de operații folosite, iar algebra inițială T_Σ furnizează sintaxa limbajului de programare. Fiecare term al algebrei T_Σ este un program corect construit (o expresie corect construită) conform unor reguli a priori date. Orice altă algebră A în clasa tuturor Σ -algebrelor este un posibil *domeniu semantic*. Unicul homomorfism $eval_A$ de la algebra inițială T_Σ la domeniul semantic A asociază o *semantică* (un *înțeles*) fiecărei construcții sintactice (term/program) a algebrei T_Σ . În acest context, $eval_A$ se mai numește și *homomorfism semantic*.

Metoda astfel descrisă de specificare a semanticii unui limbaj de programare poartă denumirea de *metoda semantică a algebrei inițiale* (MSAI). Vom descrie și exemplificăm pe larg această metodă urmând [72] și [217].

8.9.1.1. Signatura asociată unei gramici independente de context

În general, sintaxa unui limbaj de programare este specificată în notație BNF¹⁰, ceea ce este echivalent cu specificarea printr-o gramatică independentă de context. Ca urmare a acestui fapt, ne vom ocupa întâi de descrierea metodei semanticii algebrei inițiale pentru astfel de gramici. Menționăm încă de la început că, în baza definiției, gramicile Chomsky sunt finite. Pentru un plus de generalitate, vom elimina această restricție și vom defini *gramaticile independente de context* ca fiind 4-uple $G = (V, T, N_0, P)$, unde V și T sunt multimi nevide și disjuncte ale căror elemente se

¹⁰BNF este un acronim pentru "Backus-Naur Form" ce constituie o notație utilizată pentru prima dată de John Backus pentru descrierea sintaxei limbajului ALGOL 58. Ulterior, într-o formă puțin modificată, această notație a fost utilizată și de Peter Naur pentru specificarea sintaxei limbajului ALGOL 60 [154].

numesc (*simboluri*) *neterminale* și, respectiv, *terminale*, N_0 este un simbol neterminat numit *axiomă gramaticii*, iar $P \subseteq V \times (V \cup T)^*$ este o mulțime nevidă ale cărei elemente se numesc *reguli*. În mod ușual, regulile $(N, w) \in P$ se notează în forma $N \rightarrow w$. În plus, vom presupune că acestea sunt etichetate injectiv (reguli distincte sunt etichetate distinct) în varianta

$$p : N \rightarrow w,$$

unde p este o etichetă ce nu este element al mulțimii $V \cup T$ (etichetarea are doar scop tehnic, fiind necesară pentru o mai ușoară referire la regulile gramaticii).

Relația de derivare indușă de o gramatică independentă de context G este relația binară $\Rightarrow_G \subseteq (V \cup T)^* \times (V \cup T)^*$ dată prin

$$u \Rightarrow_G v \Leftrightarrow (\exists u_1, u_2 \in (V \cup T)^*)(u = u_1 N u_2 \wedge v = u_1 w u_2 \wedge N \rightarrow w \in P),$$

pentru orice $u, v \in (V \cup T)^*$. Închiderea reflexivă și tranzitivă a acestei relații este notată prin \Rightarrow_G^* . *Limbajul generat de* G , notat $L(G)$, este definit prin

$$L(G) = \{w \in T^* | N_0 \Rightarrow_G^* w\}.$$

Fie N un neterminat al gramaticii G . O *derivare* în G de la N sau *N-derivare* în G este o secvență de forma

$$N \Rightarrow_G w_1 \Rightarrow_G \cdots \Rightarrow_G w_n,$$

unde $w_i \in (V \cup T)^*$, pentru orice $1 \leq i \leq n$. Atunci când $N = N_0$ vom spune mai simplu că aceasta este o *derivare* în G , iar când $w_n \in T^*$ vom spune că este o *N-derivare terminală* în G . În cel de-al doilea caz terminologia este justificată prin aceea că w_n este format numai din simboluri terminale (și, astfel, *N-derivarea* nu mai poate fi continuată). Dacă în *N-derivarea* de mai sus neterminatul care se scrie în oricare din w_1, \dots, w_{n-1} este întotdeauna ales ca fiind cel mai din stânga neterminat (în cadrul respectivului cuvânt), atunci *N-derivarea* se numește *N-derivare extrem stângă*. Este cunoscut faptul că limbajul unei gramici independente de context poate fi obținut numai prin derivări terminale extrem stângi.

Dată o *N-derivare terminală* $w_0 = N \Rightarrow_G w_1 \Rightarrow_G \cdots \Rightarrow_G w_n$, orice secvență $p_1 \cdots p_n$ de etichete de reguli cu proprietatea că pasul i al derivării poate fi realizat prin regula p_i , poartă denumirea de *cuvânt de N-control* al derivării¹¹. Dacă regula p_i are proprietatea că scrie cel mai din stânga neterminat al lui w_{i-1} , pentru orice $1 \leq i \leq n$, atunci $p_1 \cdots p_n$ este numit *cuvânt de N-control la stânga*. Este clar că o *N-derivare terminală* poate avea mai multe cuvinte de *N-control*, și orice cuvânt de *N-control* poate conduce la mai mult de o *N-derivare terminală*. Însă *N-derivările* terminale extrem stângi conduc la un unic cuvânt de *N-control la stânga*, și reciproc. Ca urmare, *N-derivările* extrem stângi pot fi identificate prin cuvinte de *N-control* la stânga.

¹¹În literatura de specialitate aceste cuvinte de control mai sunt întâlnite și sub denumirea de *cuvinte Szilard*.

Exemplul 8.9.1.1. Mai jos este prezentată o gramatică independentă de context ale cărei reguli sunt etichetate (axioma gramaticii este N_0):

$$\begin{aligned} V &= \{N_0, N_1, N_2\} \\ T &= \{a, b\} \\ p_1 : N_0 &\rightarrow aN_1N_2 \\ p_2 : N_0 &\rightarrow bN_2a \\ p_3 : N_1 &\rightarrow N_0N_1 \\ p_4 : N_1 &\rightarrow a \\ p_5 : N_2 &\rightarrow b \end{aligned}$$

Derivarea extrem stângă

$$N_0 \Rightarrow aN_1N_2 \Rightarrow aaN_2 \Rightarrow aab$$

este identificată prin cuvântul de control la stânga $p_1p_4p_5$.

Orice gramatică independentă de context G induce o signură $\Sigma(G)$ astfel [72]:

- fiecărui neterminal N i se asociază un sort s_N . Fie S mulțimea tuturor acestor sorturi;
- fiecărei reguli

$$p : N \rightarrow u_0N_1u_1 \cdots u_kN_ku_{k+1},$$

unde $u_0, \dots, u_{k+1} \in T^*$ și $N, N_1, \dots, N_k \in V$, i se asociază un simbol funcțional σ_p de tip $(s_{N_1} \cdots s_{N_k}, s_N)$. Regulilor de forma

$$p : N \rightarrow u,$$

unde $u \in T^*$, li se asociază simboluri funcționale constante σ_p de sort s_N .

Constatăm că trecerea de la gramatică la signură se face cu “pierdere de informație”: simbolurile terminale nu sunt luate în calcul, iar axioma gramaticii, în cadrul signurii, nu este evidențiată.

Exemplul 8.9.1.2. Signura asociată gramaticii din Exemplul 8.9.1.1 este descrisă mai jos.

$$\begin{array}{ll} V = \{N_0, N_1, N_2\} & S = \{s_{N_0}, s_{N_1}, s_{N_2}\} \\ T = \{a, b\} & \\ p_1 : N_0 &\rightarrow aN_1N_2 & \sigma_{p_1} \text{ de tip } (s_{N_1}s_{N_2}, s_{N_0}) \\ p_2 : N_0 &\rightarrow bN_2a & \sigma_{p_2} \text{ cu tip } (s_{N_2}, s_{N_0}) \\ p_3 : N_1 &\rightarrow N_0N_1 & \sigma_{p_3} \text{ de tip } (s_{N_0}s_{N_1}, s_{N_1}) \\ p_4 : N_1 &\rightarrow a & \sigma_{p_4} \text{ de tip } (\lambda, s_{N_1}) \\ p_5 : N_2 &\rightarrow b & \sigma_{p_5} \text{ de tip } (\lambda, s_{N_2}) \end{array}$$

Dată o gramatică independentă de context G , evidențiem următoarele 3 algebrelor în clasa algebrelor de signură $\Sigma(G)$:

1. **$\Sigma(G)$ -algebra termilor de bază $\mathbf{T}_{\Sigma(G)}$.** Aceasta este algebră inițială în clasa tuturor algebrelor de signură $\Sigma(G)$;

2. **$\Sigma(G)$ -algebra derivărilor extrem stângi.** Această algebră, notată prin $\mathbf{D}(G)$, este definită astfel:

- pentru orice sort s_N , $D(G)_{s_N}$ este mulțimea cuvintelor de N -control la stânga;
- pentru orice regulă $p : N \rightarrow u_0N_1u_1 \cdots u_kN_ku_{k+1}$, operația $\sigma_p^{D(G)}$ de tip $(s_{N_1} \cdots s_{N_k}, s_N)$ pe argumentele $\alpha_1, \dots, \alpha_k$ produce cuvântul $p\alpha_1 \cdots \alpha_k$ (care este cuvânt de N -control la stânga);
- pentru orice regulă $p : N \rightarrow u$, operația $\sigma_p^{D(G)}$ de sort s_N este identificată prin cuvântul format doar din eticheta p ;

3. **$\Sigma(G)$ -algebra cuvintelor peste T .** Această algebră, notată prin $\mathbf{F}(G)$, este definită astfel:

- pentru orice sort s_N , $F(G)_{s_N}$ este mulțimea T^* ;
- pentru orice regulă $p : N \rightarrow u_0N_1u_1 \cdots u_kN_ku_{k+1}$, operația $\sigma_p^{F(G)}$ de tip $(s_{N_1} \cdots s_{N_k}, s_N)$ pe argumentele $w_1, \dots, w_k \in T^*$ produce cuvântul $u_0w_1u_1 \cdots u_kw_ku_{k+1} \in T^*$;
- pentru orice regulă $p : N \rightarrow u$, operația $\sigma_p^{F(G)}$ de sort s_N este identificată prin cuvântul $u \in T^*$;

Este cât se poate de clar că orice term $t \in \mathbf{T}_{\Sigma(G), s_N}$ definește un unic cuvânt de N -control la stânga, și reciproc. Atunci obținem cu ușurință următorul rezultat:

Teorema 8.9.1.1. Pentru orice gramatică independentă de context, algebră $\mathbf{D}(G)$ a derivărilor extrem stângi ale gramaticii G este algebră inițială în clasa tuturor algebrelor de signură $\Sigma(G)$.

Deoarece $\mathbf{T}_{\Sigma(G)}$ este algebră inițială în clasa tuturor $\Sigma(G)$ -algebrelor, există un unic homomorfism $eval_{\mathbf{F}(G)}$ de la $\mathbf{T}_{\Sigma(G)}$ la $\mathbf{F}(G)$. Acest homomorfism asociază fiecărui term t de sort s_N al algebrelor $\mathbf{T}_{\Sigma(G)}$ un unic cuvânt $w = eval_{\mathbf{F}(G)}(t)$ peste T ce poate fi obținut printr-o N -derivare terminală (nu neapărat unică), termul t specificând regulile ce trebuie aplicate, precum și o ordine (partială) de aplicare a acestora pentru a obține o astfel de N -derivare. Reciproc, orice N -derivare terminală conduce la un term (nu neapărat unic) de sort s_N ce definește respectiva derivare și cuvântul peste T definit de ea. Ca o consecință, $L(G) = eval_{\mathbf{F}(G)}(\mathbf{T}_{\Sigma(G), s_{N_0}})$.

Există o foarte strânsă legătură între injectivitatea homomorfismului $eval_{\mathbf{F}(G)}$ și N -neambiguitatea gramaticii G (G este numită N -ambiguă dacă există cel puțin două N -derivări terminale extrem stângi distincte ce conduc la același cuvânt peste T ; G este numită N -neambiguă dacă nu este N -ambiguă).

Teorema 8.9.1.2. Fie G o gramatică independentă de context. G este N -neambiguă pentru orice N dacă și numai dacă $\text{eval}_{\mathbf{F}(G)}$ este homomorfism injectiv.

Demonstrație. G este N -neambiguă pentru orice N dacă și numai dacă unicul homomorfism de la $\mathbf{D}(G)$ la $\mathbf{F}(G)$ este injectiv. Atunci teorema urmează de la faptul că $\mathbf{T}_{\Sigma(G)}$ și $\mathbf{D}(G)$ sunt izomorfe. \square

Gramatica G furnizează sintaxa unei mulțimi de expresii, și anume a mulțimii $L(G)$ (fiecare cuvânt al lui $L(G)$ poate fi o expresie aritmetică sau logică de un anumit tip, un program structurat etc.). Dat $w \in L(G)$, există cel puțin un term t astfel încât $\text{eval}_{\mathbf{F}(G)}(t) = w$. Termul t ne arată ce reguli pot fi aplicate, și cum pot fi aplicate, pentru a obține o derivare în G care să conducă la w . Ca urmare, t poate fi mai *succinct* decât w (deoarece $\Sigma(G)$ nu ia în considerare terminalele gramaticii). Spunem într-un astfel de caz că t este o *sintaxă abstractă* a lui w , iar w este *sintaxă concretă* a lui t . Ca urmare,

- G furnizează sintaxa concretă a unei mulțimi de expresii;
- $\Sigma(G)$ furnizează sintaxa abstractă a expresiilor definite de G ;
- $\mathbf{T}_{\Sigma(G)}$ este *algebra expresiilor definite* de G , în sintaxă abstractă.

Trecerea de la sintaxa abstractă la cea concretă se face printr-un unic homomorfism.

Orice $\Sigma(G)$ -algebră \mathbf{A} acționează ca domeniu semantic pentru $\mathbf{T}_{\Sigma(G)}$. Unicul homomorfism $\text{eval}_{\mathbf{A}}$ asociază fiecărui term t semantică lui în \mathbf{A} . Dacă *parse* este o

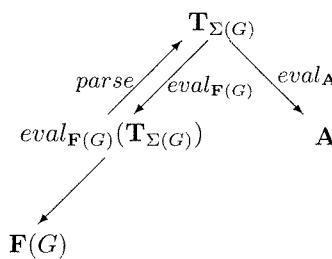


Figura 8.18: Sintaxă concretă, sintaxă abstractă, semantică

funcție ce asociază unui element w din $\text{eval}_{\mathbf{F}(G)}(\mathbf{T}_{\Sigma(G)})$ un term t ce este o sintaxă abstractă pentru w (a se vedea Figura 8.18), atunci are loc

$$\text{eval}_{\mathbf{F}(G)} \circ \text{parse} = 1_{\text{eval}_{\mathbf{F}(G)}(\mathbf{T}_{\Sigma(G)})}.$$

Dacă $\text{eval}_{\mathbf{F}(G)}$ este injectivă, atunci are loc și

$$\text{parse} \circ \text{eval}_{\mathbf{F}(G)} = 1_{\mathbf{T}_{\Sigma(G)}}.$$

Vom discuta în continuare câteva modalități de asociere a unui domeniu semantic pentru algebra $\mathbf{T}_{\Sigma(G)}$. Reamintim întâi că fiecarei reguli $p : N \rightarrow w$ a unei gramaticii independente de context G i se poate asocia un *arbore local ordonat*¹² astfel:

- arborele are rădăcina etichetată cu N ;
- dacă $w = \lambda$, atunci există exact un succesor al nodului rădăcină, etichetat cu λ , iar aceste două noduri sunt singurele noduri ale arborelui;
- dacă $w \neq \lambda$, atunci nodul rădăcină are exact $|w|$ succesiuni direcții etichetați în ordine de la stânga la dreapta prin $x_1, \dots, x_n \in V \cup T$, unde $w = x_1 \cdots x_n$. Nodurile diferite de rădăcină nu au succesiuni.

Arborele asociat regulii $N_0 \rightarrow aN_1N_2$ este reprezentat grafic în Figura 8.19(a).

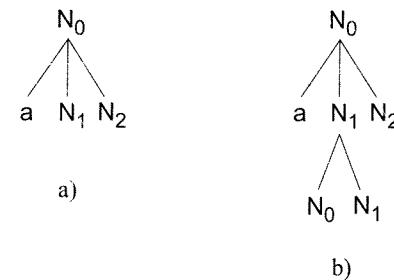


Figura 8.19: a) Arborele asociat regulii $N_0 \rightarrow aN_1N_2$; b) Arbore asociat derivării $N_0 \Rightarrow_G aN_1N_2 \Rightarrow_G aN_0N_1N_2$

Oricarei N -derivări i se poate asocia cel puțin un *arbore local ordonat*. Dacă N -derivarea are doar un pas, atunci singurul arbore asociat este de fapt arborele asociat regulii utilizate la pasul respectiv. Dacă d este un arbore asociat unei N -derivări $N \xrightarrow{*} w_1N'w_2$, atunci înlocuind nodul frunză din d corespunzător etichetei N' (ce are stânga w_1 și în dreapta w_2) prin arborele asociat unei reguli $N' \rightarrow w$, obținem un arbore asociat N -derivării $N \xrightarrow{*}_G w_1N'w_2 \Rightarrow_G w_1ww_2$. De exemplu, înlocuind în arborele din Figura 8.19(a) frunza etichetată prin N_1 cu arborele asociat regulii $N_1 \rightarrow N_0N_1$, obținem arborele din Figura 8.19(b) ce corespunde derivării $N_0 \Rightarrow aN_1N_2 \Rightarrow aN_0N_1N_2$.

Un arbore asociat unei N -derivări pentru care cuvântul final este din T^* va fi numit *arbore de N -derivare* al gramaticii (simbolul N va fi omis atunci când el este axioma gramaticii).

Frontiera unui arbore de N -derivare este cuvântul format prin concatenarea simbolurilor (terminale, neterminale și λ) ce etichetează nodurile frunză ale arborelui, privite drept cuvinte și luate în ordine de la stânga la dreapta. De exemplu, frontiera

¹²Abordarea noastră va fi informală dar credem că suficient de clară. Pentru o abordare formală a arborilor local ordonați asociați regulilor și derivărilor indicăm [84].

arborelui din Figura 8.19(b) este $aN_0N_1N_2$. Este clar că limbajul generat de o gramatică independentă de context este exact mulțimea frontierelor arborilor de derivare ai gramaticii.

Dată o gramatică independentă de context G , considerăm următoarele $\Sigma(G)$ -algebrelor:

1. **$\Sigma(G)$ -algebra arborilor de derivare ai gramaticii G .** Această algebra, notată prin $\mathbf{R}(G)$, este definită astfel:

- pentru orice sort s_N , $R(G)_{s_N}$ este mulțimea tuturor arborilor de N -derivare ai gramaticii G ;
- pentru orice regulă $p : N \rightarrow u_0N_1u_1 \dots u_kN_ku_{k+1}$, operația $\sigma_p^{R(G)}$ de tip $(s_{N_1} \dots s_{N_k}, s_N)$ pe argumentele $d_1 \in R(G)_{s_{N_1}}, \dots, d_k \in R(G)_{s_{N_k}}$ produce arborele de N -derivare obținut din arborele asociat regulii p în care subarborele cu rădăcina N_i este înlocuit prin d_i , pentru orice i (a se vedea modul de construcție al arborelui din Figura 8.19(b));
- pentru orice regulă $p : N \rightarrow u$, operația $\sigma_p^{R(G)}$ de sort s_N este identificată cu arborele asociat regulii p ;

2. **$\Sigma(G)$ -algebra adâncimii (înălțimii) arborilor de derivare ai gramaticii G .** Această algebra, notată prin $\mathbf{H}(G)$, este definită astfel:

- pentru orice sort s_N , $H(G)_{s_N}$ este mulțimea \mathbf{N} ;
- pentru orice regulă $p : N \rightarrow u_0N_1u_1 \dots u_kN_ku_{k+1}$, operația $\sigma_p^{H(G)}$ de tip $(s_{N_1} \dots s_{N_k}, s_N)$ pe argumentele $n_1, \dots, n_k \in \mathbf{N}$ produce numărul $1 + \max\{n_1, \dots, n_k\} \in \mathbf{N}$;
- pentru orice regulă $p : N \rightarrow u$, operația $\sigma_p^{H(G)}$ de sort s_N este identificată prin $1 \in \mathbf{N}$.

Este ușor de văzut că orice term de bază de sort s_N al algebrei $\mathbf{T}_{\Sigma(G)}$ definește exact un arbore de N -derivare al gramaticii G . De exemplu, termul

$$t = \sigma_{p_1}(\sigma_{p_3}(\sigma_{p_2}(\sigma_{p_5}), \sigma_{p_4}), \sigma_{p_5})$$

de sort N_0 definește unic arborele de derivare din Figura 8.20, care este de fapt $\text{eval}_{\mathbf{R}(G)}(t)$. Ca urmare, $\mathbf{R}(G)$ acționează ca un domeniu semantic pentru $\mathbf{T}_{\Sigma(G)}$, furnizând o semantică prin arbori de derivare.

În cazul algebrei $\mathbf{R}(G)$ se poate arăta chiar mai mult, și anume că ea este algebra inițială în clasa $\Sigma(G)$ -algebrelor. Aceasta rezultă imediat observând că orice arbore de N -derivare al gramaticii G definește unic un term de sort s_N al algebrei $\mathbf{T}_{\Sigma(G)}$.

Teorema 8.9.1.3. Pentru orice gramatică independentă de context G , algebra $\mathbf{R}(G)$ a arborilor de derivare ai gramaticii G este algebra inițială în clasa tuturor algebrelor de signură $\Sigma(G)$.

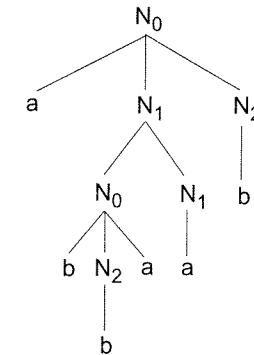


Figura 8.20: Arbore de derivare

$\Sigma(G)$ -algebra $\mathbf{H}(G)$ constituie un alt domeniu semantic pentru $\mathbf{T}_{\Sigma(G)}$. Unicul homomorfism $\text{eval}_{\mathbf{H}(G)}$ de la $\mathbf{T}_{\Sigma(G)}$ la $\mathbf{H}(G)$ asociază fiecărui term t de sort s_N adâncimea arborelui de N -derivare asociat lui t (arborii asociați regulilor gramaticii au adâncimea 1). De exemplu, aplicând acest homomorfism termului

$$t = \sigma_{p_1}(\sigma_{p_3}(\sigma_{p_2}(\sigma_{p_5}), \sigma_{p_4}), \sigma_{p_5})$$

obținem $\text{eval}_{\mathbf{H}(G)}(t) = 4$ care este adâncimea arborelui de derivare asociat lui t .

Așa cum se vede, există o bogătie de modalități de asociere de semantici structurilor generate de o gramatică independentă de context. Vom încheia secțiunea printr-un nou exemplu care ne va arăta importanța modului de alegere a unei gramatici independente de context pentru descrierea sintaxei unei mulțimi de expresii (programe).

Exemplul 8.9.1.3. Fie G gramatica de mai jos, împreună cu signura asociată acesteia.

$V = \{N_0, N_1\}$	$S = \{s_{N_0}, s_{N_1}\}$
$T = \{0, 1\}$	
$p_1 : N_0 \rightarrow N_1$	$\sigma_{p_1} : (s_{N_1}, s_{N_0})$
$p_2 : N_0 \rightarrow N_0N_1$	$\sigma_{p_2} : (s_{N_0}s_{N_1}, s_{N_0})$
$p_3 : N_1 \rightarrow 0$	$\sigma_{p_3} : (\lambda, s_{N_1})$
$p_4 : N_1 \rightarrow 1$	$\sigma_{p_4} : (\lambda, s_{N_1})$

Această gramatică generează toate secvențele binare nevide peste 0 și 1.

Algebra $\mathbf{T}_{\Sigma(G)}$, unde G este gramatica din Exemplul 8.9.1.3, este algebra tuturor secvențelor binare. Dacă considerăm $\Sigma(G)$ -algebra \mathbf{A} dată prin:

- $A_{s_{N_1}} = \{0, 1\} \subset \mathbf{N}$;
- $A_{s_{N_0}} = \mathbf{N}$;

- $\sigma_{p_3}^A$ este constanta $0 \in A_{s_{N_1}}$;
- $\sigma_{p_4}^A$ este constanta $1 \in A_{s_{N_1}}$
- $\sigma_{p_1}^A : A_{s_{N_1}} \rightarrow A_{s_{N_0}}$ este funcția incluziune;
- $\sigma_{p_2}^A : A_{s_{N_0}} \times A_{s_{N_1}} \rightarrow A_{s_{N_0}}$ este funcția dată prin

$$\sigma_{p_2}^A(n, c) = 2n + c,$$

pentru orice $(n, c) \in A_{s_{N_0}} \times A_{s_{N_1}}$,

atunci unicul homomorfism semantic de la $\mathbf{T}_{\Sigma(G)}$ la \mathbf{A} nu face altceva decât să asocieze fiecarei secvențe binare t numărul natural 0, dacă t este formată numai din 0, sau numărul natural cu reprezentarea binară $1t'$, dacă t este de forma $0 \cdots 01t'$.

Înlocuim acum regula p_2 în gramatica G prin

$$p'_2 : N_0 \rightarrow N_1 N_0,$$

și fie G' gramatica astfel obținută. Această gramatică generează exact aceleași secvențe binare nevide ca și gramatica G . Dacă însă dorim să stabilim o semantică similară pentru gramatica G' , constatăm că aceasta nu mai poate fi făcută doar printr-o simplă modificare a operației $\sigma_{p_2}^A$. Putem stabili o semantică similară dacă considerăm $\Sigma(G')$ -algebra \mathbf{A}' dată prin:

- $A_{s_{N_1}} = \{(0, 1), (1, 1)\} \subset \mathbf{N} \times \mathbf{N}$;
- $A_{s_{N_0}} = \{(n, l) | l \geq 1 + \lfloor \log n \rfloor\} \subseteq \mathbf{N} \times \mathbf{N}$;
- $\sigma_{p_3}^A$ este constanta $(0, 1) \in A_{s_{N_1}}$;
- $\sigma_{p_4}^A$ este constanta $(1, 1) \in A_{s_{N_1}}$
- $\sigma_{p_1}^A : A_{s_{N_1}} \rightarrow A_{s_{N_0}}$ este funcția incluziune;
- $\sigma_{p'_2}^A : A_{s_{N_1}} \times A_{s_{N_0}} \rightarrow A_{s_{N_0}}$ este funcția dată prin

$$\sigma_{p'_2}^A((c, 1), (n, l)) = (c2^l + n, l + 1),$$

pentru orice $((c, 1), (n, l)) \in A_{s_{N_0}} \times A_{s_{N_1}}$.

O pereche $(n, l) \in A_{s_{N_0}}$ ne spune că numărul n este gândit ca având o reprezentare binară pe l poziții. Aceasta se obține din reprezentarea binară a lui n adăugând la stânga zero-uri până când lungimea acestei reprezentări binare devine l . De exemplu, perechea $(1, 3)$ reprezintă secvența 001 care poate fi gândită ca reprezentarea binară a lui 1 pe 3 poziții. Acest mod de a privi lucrurile este fundamental în a stabili semantică gramaticii G' așa cum am spus mai sus că am dori-o. Cititorul este invitat să analizeze exemplul ce urmează pentru a realiza aceasta.

Fie termul $t = \sigma_{p'_2}(\sigma_{p_4}, \sigma_{p'_2}(\sigma_{p_3}, \sigma_{p'_2}(\sigma_{p_3}, \sigma_{p_1}(\sigma_{p_4}))))$ de sort s_{N_0} al algebrei inițiale $\mathbf{T}_{\Sigma(G')}$. El corespunde frontierei 1001 a unui arbore de derivare al gramaticii G' . Atunci:

$$\begin{aligned} eval_{\mathbf{A}'}(t) &= \sigma_{p'_2}^{A'}(\sigma_{p_4}^{A'}, \sigma_{p'_2}^{A'}(\sigma_{p_3}^{A'}, \sigma_{p'_2}^{A'}(\sigma_{p_3}^{A'}, \sigma_{p_1}^{A'}(\sigma_{p_4}^{A'})))) \\ &= \sigma_{p'_2}^{A'}((1, 1), \sigma_{p'_2}^{A'}((0, 1), \sigma_{p'_2}^{A'}((0, 1), \sigma_{p_1}^{A'}((1, 1))))) \\ &= \sigma_{p'_2}^{A'}((1, 1), \sigma_{p'_2}^{A'}((0, 1), \sigma_{p'_2}^{A'}((0, 1), (1, 1)))) \\ &= \sigma_{p'_2}^{A'}((1, 1), \sigma_{p'_2}^{A'}((0, 1), (1, 2))) \\ &= \sigma_{p'_2}^{A'}((1, 1), (1, 3)) \\ &= (9, 4). \end{aligned}$$

Că urmare, semantica secvenței 1001 este perechea $(9, 4)$ care ne dă atât numărul cu reprezentarea binară 1001 cât și lungimea reprezentării binare a acestuia.

Concluzia pe care o desprindem din acest exemplu este că alegera gramaticii ce urmează a descrie sintaxa expresiilor formale în care suntem interesați își lasă amprenta foarte clar asupra modului în care se poate asocia un domeniu semantic potrivit acestor expresii.

8.9.1.2. Semantica programelor structurate

În Secțiunea 7.3.1.4 s-a introdus o clasă de programe structurate peste o bază și s-a arătat cum se poate asocia acestora o semantică denotațională. În această secțiune vom arăta cum putem asocia acestora o semantică prin metoda semantică algebrei inițiale. Pentru o mai bună înțelegere a metodei vom specializa puțin sintaxa din Secțiunea 7.3.1.4 considerând o mulțime \mathcal{V} de variabile, o mulțime \mathcal{C} de simboluri constante, o mulțime $\{\text{succ}, \text{pred}, -, +, *\}$ de simboluri funcționale ne-constante, două constante propoziționale *true* și *false*, și o mulțime $\{\neg, \wedge, =, \leq\}$ de simboluri propoziționale ne-constante. Toate aceste mulțimi de simboluri sunt disjuncte două câte două.

Definim acum *programele structurate* peste această bază prin intermediul gramaticii infinite G ce are 4 simboluri neterminale, I (pentru identificatori), E (pentru expresii aritmetice), L (pentru expresii logice) și W (pentru programe), axioma W și regulile

$$\begin{aligned} I &::= x \\ E &::= I \mid c \mid (-E) \mid (E + E) \mid (E * E) \\ L &::= \text{true} \mid \text{false} \mid \neg L \mid (L \wedge L) \mid (E = E) \mid (E \leq E) \\ W &::= I := E \mid W; W \mid \text{if } L \text{ then } W \text{ else } W \text{ fi} \mid \\ &\quad \text{while } L \text{ do } W \text{ od} \end{aligned}$$

pentru orice $x \in \mathcal{V}$ și $c \in \mathcal{C}$ (simbolul ::= înlocuiește \rightarrow). Toate celelalte elemente ce intervin în definirea acestor reguli sunt simboluri terminale ($x, c, \text{succ}, (,)$ etc).

Acestei gramatici îi putem ataşa o signură $\Sigma(G)$ ca în secțiunea anterioară. Sorturile vor fi s_I , s_E , s_L și s_W , iar simbolurile funcționale vor fi notate, în ordinea regulilor gramaticii, prin

$$\sigma_x, \sigma_{i2e}, \sigma_c, \sigma_{neg}, \sigma_{add}, \sigma_{mult}, \sigma_{true}, \sigma_{false},$$

$$\sigma_{not}, \sigma_{and}, \sigma_{eq}, \sigma_{leq}, \sigma_{assign}, \sigma_{seq}, \sigma_{if}, \sigma_{while},$$

pentru orice $x \in \mathcal{V}$ și $c \in \mathcal{C}$ (credem că notația este suficient de sugestivă pentru a permite identificarea exactă a simbolului funcțional asociat regulii corespunzătoare a gramaticii).

Expresia

while ($y \leq x$) **do** $x := (x + (-y))$ **od**

este un program structurat generat de o gramatică aşa cum este cea de mai sus.

Algebra termilor de bază $\mathbf{T}_{\Sigma(G)}$ furnizează sintaxa abstractă a programelor structurate, motiv pentru care o vom numi *algebra programelor structurate*. Programul dat ca exemplu mai sus este sintaxa concretă a termului de bază

$$\sigma_{while}(\sigma_{leq}(\sigma_{i2e}(\sigma_y), \sigma_{i2e}(\sigma_x)), \sigma_{assign}(\sigma_x, \sigma_{add}(\sigma_{i2e}(\sigma_x), \sigma_{neg}(\sigma_{i2e}(\sigma_y))))).$$

A da o semantică pentru programele structurate (definite ca mai sus) revine la a fixa o $\Sigma(G)$ -algebră \mathbf{A} pe post de domeniu semantic. Atunci, unicul homomorfism $eval_{\mathbf{A}}$ de la $\mathbf{T}_{\Sigma(G)}$ la \mathbf{A} asociază fiecărui program t (în sintaxă abstractă) semantică acestuia. Vom prezenta în cele ce urmează un astfel de domeniu semantic considerând $\mathcal{C} = \mathbf{Z}$.

Fie Γ mulțimea tuturor funcțiilor de la \mathcal{V} la \mathbf{Z} . Elementele acestei mulțimi vor fi numite *stări* (a se vedea Secțiunea 7.3.1.5). În clasa tuturor $\Sigma(G)$ -algebrelor vom considera acum algebra \mathbf{A} definită după cum urmează:

- $A_{s_I} = \mathcal{V}$;
- $A_{s_E} = (\Gamma \rightarrow \mathbf{Z})$;
- $A_{s_L} = (\Gamma \rightarrow \{\text{true}, \text{false}\})$;
- $A_{s_W} = (\Gamma \rightsquigarrow \Gamma)$;
- $\sigma_x^A \in A_{s_I}$ este dată prin $\sigma_x^A = x$, pentru orice $x \in \mathcal{V}$;
- $\sigma_{i2e}^A : A_{s_I} \rightarrow A_{s_E}$ este dată prin $\sigma_{i2e}^A(x)(\gamma) = \gamma(x)$, pentru orice $x \in A_{s_I}$ și $\gamma \in \Gamma$;
- $\sigma_c^A \in A_{s_E}$ este dată prin $\sigma_c^A(\gamma) = c$, pentru orice $\gamma \in \Gamma$;
- $\sigma_{neg}^A : A_{s_E} \rightarrow A_{s_E}$ este dată prin $\sigma_{neg}^A(f)(\gamma) = -f(\gamma)$, pentru orice $f \in A_{s_E}$ și $\gamma \in \Gamma$;

- $\sigma_{add}^A : A_{s_E} \times A_{s_E} \rightarrow A_{s_E}$ este dată prin $\sigma_{add}^A(f_1, f_2)(\gamma) = f_1(\gamma) + f_2(\gamma)$, pentru orice $f_1, f_2 \in A_{s_E}$ și $\gamma \in \Gamma$;
- $\sigma_{mult}^A : A_{s_E} \times A_{s_E} \rightarrow A_{s_E}$ este dată prin $\sigma_{mult}^A(f_1, f_2)(\gamma) = f_1(\gamma)f_2(\gamma)$, pentru orice $f_1, f_2 \in A_{s_E}$ și $\gamma \in \Gamma$;
- $\sigma_{true}^A \in A_{s_L}$ este dată prin $\sigma_{true}^A(\gamma) = \text{true}$, pentru orice $\gamma \in \Gamma$;
- $\sigma_{false}^A \in A_{s_L}$ este dată prin $\sigma_{false}^A(\gamma) = \text{false}$, pentru orice $\gamma \in \Gamma$;
- $\sigma_{not}^A : A_{s_L} \rightarrow A_{s_L}$ este dată prin $\sigma_{not}^A(f)(\gamma) = \neg f(\gamma)$, pentru orice $f \in A_{s_L}$ și $\gamma \in \Gamma$;
- $\sigma_{and}^A : A_{s_L} \times A_{s_L} \rightarrow A_{s_L}$ este dată prin $\sigma_{and}^A(f_1, f_2)(\gamma) = f_1(\gamma) \wedge f_2(\gamma)$, unde \wedge semnifică conjuncția valorilor de adevăr, pentru orice $f_1, f_2 \in A_{s_L}$ și $\gamma \in \Gamma$;
- $\sigma_{eq}^A : A_{s_E} \times A_{s_E} \rightarrow A_{s_L}$ este dată prin

$$\sigma_{eq}^A(f_1, f_2)(\gamma) = \begin{cases} \text{true}, & \text{dacă } f_1(\gamma) = f_2(\gamma) \\ \text{false}, & \text{dacă } f_1(\gamma) \neq f_2(\gamma), \end{cases}$$

pentru orice $f_1, f_2 \in A_{s_E}$ și $\gamma \in \Gamma$;

- $\sigma_{leq}^A : A_{s_E} \times A_{s_E} \rightarrow A_{s_L}$ este dată prin

$$\sigma_{leq}^A(f_1, f_2)(\gamma) = \begin{cases} \text{true}, & \text{dacă } f_1(\gamma) \leq f_2(\gamma) \\ \text{false}, & \text{dacă } f_1(\gamma) > f_2(\gamma), \end{cases}$$

pentru orice $f_1, f_2 \in A_{s_E}$ și $\gamma \in \Gamma$;

- $\sigma_{assign}^A : A_{s_I} \times A_{s_E} \rightarrow A_{s_W}$ este dată prin $\sigma_{assign}^A(x, f)(\gamma) = \gamma[x/f(\gamma)]$, pentru orice $x \in A_{s_I}$, $f \in A_{s_E}$ și $\gamma \in \Gamma$;

- $\sigma_{seq}^A : A_{s_W} \times A_{s_W} \rightarrow A_{s_W}$ este dată prin $\sigma_{seq}^A(f_1, f_2)(\gamma) = f_2(f_1(\gamma))$, pentru orice $f_1, f_2 \in A_{s_W}$ și $\gamma \in \Gamma$;

- $\sigma_{if}^A : A_{s_L} \times A_{s_W} \times A_{s_W} \rightarrow A_{s_W}$ este dată prin

$$\sigma_{if}^A(g, f_1, f_2)(\gamma) = \begin{cases} f_1(\gamma), & \text{dacă } g(\gamma) = \text{true} \\ f_2(\gamma), & \text{dacă } g(\gamma) = \text{false}, \end{cases}$$

pentru orice $f_1, f_2 \in A_{s_W}$, $g \in A_{s_L}$ și $\gamma \in \Gamma$;

- $\sigma_{while}^A : A_{s_L} \times A_{s_W} \rightarrow A_{s_W}$ este dată prin

$$\sigma_{while}^A(g, f)(\gamma) = \begin{cases} f^n(\gamma), & \text{dacă } (\forall 0 \leq i \leq n-1)(g(f^i(\gamma)) = \text{true}) \\ \gamma, & \text{dacă } g(f^n(\gamma)) = \text{false} \end{cases}$$

pentru orice $f \in A_{s_W}$, $g \in A_{s_L}$ și $\gamma \in \Gamma$.

Algebra \mathbf{A} va fi numită *algebra transformărilor de stare*. Cum algebra programelor structurate este inițială, va exista un unic homomorfism $eval_{\mathbf{A}}$ de la ea la algebra transformărilor de stare. Ca urmare, algebra transformărilor de stare acționează ca un domeniu semantic pentru programele structurate. Semantica unui astfel de program nu este altceva decât o funcție care ne arată cum programul transformă o stare (inițială) într-o altă stare (finală). De exemplu, considerând programul t (în sintaxă abstractă) dat ca exemplu mai sus și o stare γ pentru care $\gamma(x), \gamma(y) \in \mathbf{N}$ și $\gamma(y) \neq 0$, are loc $eval_{\mathbf{A}}(t)(\gamma) = \gamma'$, unde $\gamma'(x) = \gamma(x) \text{ mod } \gamma(y)$.

Evident, orice altă algebră din clasa tuturor algebrelor de semnătură $\Sigma(G)$ va acționa ca domeniu semantic pentru programele structurate aşa cum au fost definite prin gramatica G . Ca urmare, o semantică pentru programele structurate revine la a fixa o $\Sigma(G)$ -algebră.

8.9.1.3. Traduceri de programe și compilare

Aparatul algebrelor universale multisortate oferă un formalism destul de adekvat și elegant pentru formularea unor probleme cu privire la traducerea programelor și corectitudinea compilatoarelor.

Prin *traducerea programelor* înțelegem procesul prin care un program scris într-un anumit limbaj de programare (de obicei într-un limbaj de programare de nivel înalt) este tradus/translatat într-un program echivalent scris în alt limbaj de programare (de obicei într-un limbaj de programare cât mai apropiat de limbajul mașinii, aşa cum ar fi limbajul de asamblare). De fapt, aceasta este activitatea specifică pe care o realizează un compilator.

Utilizând aparatul algebrelor universale multisortate, traducerea unei structuri într-o altă structură se face după următorul principiu. Sintaxa (abstractă a) structurii originale este specificată printr-o algebră inițială \mathbf{T}_{Σ} , iar sintaxa structurii țintă (scop), printr-o Σ -algebră \mathbf{C} . Unicul homomorfism $eval_{\mathbf{C}}$ de la algebra \mathbf{T}_{Σ} la \mathbf{C} este cel ce realizează traducerea. *Corectitudinea traducerii* înseamnă păstrarea semantică structurii inițiale. Dacă \mathbf{A} este un domeniu semantic pentru structura inițială, atunci corectitudinea traducerii se va reduce la existența unui homomorfism h de la \mathbf{C} la \mathbf{A} astfel încât $h \circ eval_{\mathbf{C}} = eval_{\mathbf{A}}$ (a se vedea diagrama din Figura 8.21(a)). Pentru a ușura construcția algebrei \mathbf{C} se preferă uneori construcția unei algebri mai “lărgi” \mathbf{F} de la care \mathbf{C} se obține ca fiind $eval_{\mathbf{F}}(\mathbf{T}_{\Sigma})$ (a se vedea diagrama din Figura 8.21(b)).

Vom exemplifica metodologia de mai sus considerând algebra programelor structurate aşa cum au fost definite în secțiunea anterioară, programe ce vor fi traduse în programe pentru *mașini cu stivă*. Nu vom urmări optimalitatea sau eficiența traducerii; scopul nostru este doar de a exemplifica traducerea.

O *mașină cu stivă* (MS) este un dispozitiv de calcul format din următoarele elemente (a se vedea Figura 8.22):

- o bandă semi-infinită la dreapta, numită *memoria mașinii*, ce este divizată în celule. Fiecare celulă poate reține un număr întreg arbitrar sau una din valorile

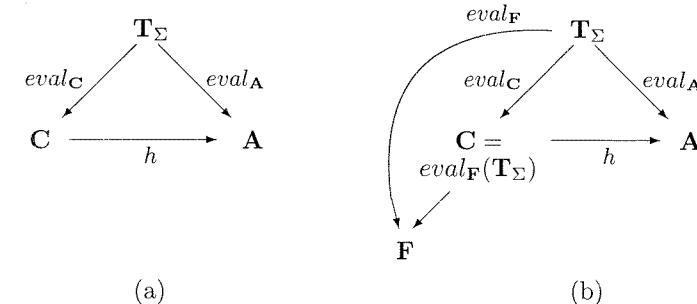


Figura 8.21: Corectitudinea traducerii

de adevăr *false* sau *true*, conținutul ei putând fi citit și/sau rescris.

Vom presupune că există o corespondență bijectivă între celulele acestei benzii și multimea \mathcal{V} de variabile (utilizată în definirea programelor structurate). Ca urmare a acestei corespondențe, celulele benzii vor fi unic identificate prin elemente $x \in \mathcal{V}$:

- o bandă semi-infinită la dreapta, numită *stiva mașinii*, ce este utilizată în regim de stivă (cu operațiile specifice de inserare în stivă și extragere a vârfului stivei). Banda este divizată în celule, fiecare celulă putând reține un număr întreg arbitrar sau una din valorile de adevăr *false* sau *true*;
- un *registru* R ce poate reține un număr întreg arbitrar sau una din valorile de adevăr *false* sau *true*. Conținutul acestuia poate fi citit și/sau rescris;
- o *unitate de control* ce este capabilă a executa un program (o secvență de instrucțiuni separate prin “;”);
- un *contor de program* PC ce indică în orice moment instrucțiunea ce urmează a fi executată. Dacă instrucțiunea ce se execută specifică un salt la altă instrucțiune, atunci contorul se actualizează corespunzător; altfel, el se actualizează la instrucțiunea imediat următoare.

Instrucțiunile ce intră în alcătuirea programelor unei MS pot fi de una din următoarele tipuri:

1. v , unde $v \in \mathbf{Z} \cup \{\text{true}, \text{false}\}$. Această instrucțiune pună valoarea v în stivă;
2. $get(x)$, unde $x \in \mathcal{V}$. Această instrucțiune citește conținutul celulei de memorie x și pună valoarea respectivă în stivă;
3. $put(x)$, unde $x \in \mathcal{V}$. Această instrucțiune extrage valoarea din vârful stivei și o pună în celula de memorie x ;
4. frR . Această instrucțiune pună conținutul registrului R în stivă;

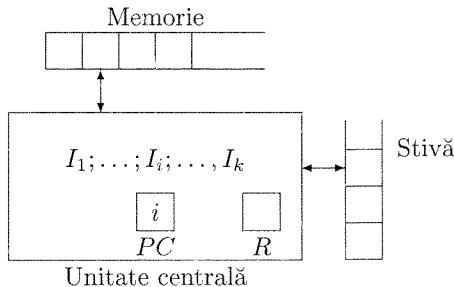


Figura 8.22: Mașină cu stivă

5. *toR*. Această instrucțiune extrage valoarea din vârful stivei și o pune în registrul *R*;
6. *add*. Această instrucțiune extrage ultimele două valori din vârful stivei, le adună și pune rezultatul în stivă;
7. *mult*. Această instrucțiune extrage ultimele două valori din vârful stivei, le înmulțește și pune rezultatul în stivă;
8. *neg*. Această instrucțiune extrage valoarea din vârful stivei, fie aceasta *v*, și pune în stivă *-v*;
9. *leq*. Această instrucțiune extrage ultimele două valori din vârful stivei și le compară. Dacă cea de a doua valoare, în ordinea în care este luată din stivă, este mai mică sau egală cu prima, atunci se pune *true* în stivă; altfel, se pune *false*;
10. *branch(n)*, unde $n \geq 1$. Această instrucțiune extrage valoarea din vârful stivei și dacă aceasta este *true* atunci schimbă conținutul contorului de program la *n*; altfel, conținutul contorului de program este incrementat cu o unitate.

Un *program pentru o mașină cu stivă* este o secvență de instrucțiuni

$$\alpha = I_1; I_2; \dots; I_k.$$

Lungimea programului α , notată prin $|\alpha|$, este numărul de instrucțiuni din program. Prin $(\alpha + i)$ vom nota programul obținut din programul α prin înlocuirea instrucțiunilor de forma *branch(m)* prin *branch(m+i)*.

Modul de execuție a unui program de către o mașină cu stivă este cât se poate de transparent. Inițial, contorul de program este fixat pe 1, indicând faptul că instrucțiunea ce urmează a fi executată este prima instrucțiune a programului. Execuția unei instrucțiuni diferită de *branch* actualizează contorul de program prin incrementarea acestuia cu 1. Există două cazuri în care mașina ce execută un program se poate opri:

- *oprire prin eroare de program*. Aceasta apare atunci când execuția unei instrucțiuni necesită un număr de operanzi pe care stiva nu îl conține, sau când tipurile operanzilor extrași din stivă nu corespund instrucțiunii;
- *oprire prin terminarea programului*. Aceasta apare când valoarea curentă a contorului de program a depășit lungimea programului.

Vom nota prin *ProgS* mulțimea tuturor programelor unei mașini cu stivă.

Revenim acum la gramatica G ce descrie sintaxa concretă a programelor structurate și considerăm $\Sigma(G)$ -algebra **F** dată prin:

- $F_{s_I} = \mathcal{V}$;
- $F_{s_E} = F_{s_L} = F_{s_W} = \text{ProgS}$;
- $\sigma_x^F \in F_{s_I}$ este dată prin

$$\sigma_x^F = x,$$
 pentru orice $x \in \mathcal{V}$;
- $\sigma_{i2e}^F : F_{s_I} \rightarrow F_{s_E}$ este dată prin

$$\sigma_{i2e}^F(x) = \text{get}(x),$$
 pentru orice $x \in F_{s_I}$;
- $\sigma_c^F \in F_{s_E}$ este dată prin

$$\sigma_c^F = c,$$
 pentru orice $c \in \mathbf{Z}$;
- $\sigma_{neg}^F : F_{s_E} \rightarrow F_{s_E}$ este dată prin

$$\sigma_{neg}^F(\alpha) = \alpha; neg,$$
 pentru orice $\alpha \in F_{s_E}$;
- $\sigma_{add}^F : F_{s_E} \times F_{s_E} \rightarrow F_{s_E}$ este dată prin

$$\sigma_{add}^F(\alpha, \beta) = \alpha; \beta; add,$$
 pentru orice $\alpha, \beta \in F_{s_E}$;
- $\sigma_{mult}^F : F_{s_E} \times F_{s_E} \rightarrow F_{s_E}$ este dată prin

$$\sigma_{mult}^F(\alpha, \beta) = \alpha; \beta; mult,$$
 pentru orice $\alpha, \beta \in F_{s_E}$;

- $\sigma_{true}^F \in F_{s_L}$ este dată prin

$$\sigma_{true}^F = true;$$

- $\sigma_{false}^F \in F_{s_L}$ este dată prin

$$\sigma_{false}^F = false;$$

- $\sigma_{not}^F : F_{s_L} \rightarrow F_{s_L}$ este dată prin

$$\begin{aligned}\sigma_{not}^F(\alpha) &= \alpha; branch(|\alpha| + 5); \\ &\quad true; true; branch(|\alpha| + 6); \\ &\quad false,\end{aligned}$$

pentru orice $\alpha \in F_{s_L}$;

- $\sigma_{and}^F : F_{s_L} \times F_{s_L} \rightarrow F_{s_L}$ este dată prin

$$\begin{aligned}\sigma_{and}^F(\alpha, \beta) &= \alpha; branch(|\alpha| + 5); \\ &\quad false; true; branch(|\alpha| + |\beta| + 5); \\ &\quad (\beta + |\alpha| + 4),\end{aligned}$$

pentru orice $\alpha, \beta \in F_{s_L}$ (ideea ce stă la baza construcției acestui program este următoarea: dacă valoarea de adevăr rezultată prin evaluarea lui α este *true*, atunci se trece la evaluarea lui β . Valoarea rezultată prin evaluarea lui β este cea finală);

- $\sigma_{eq}^F : F_{s_E} \times F_{s_E} \rightarrow F_{s_L}$ este dată prin

$$\begin{aligned}\sigma_{eq}^F(\alpha, \beta) &= \alpha; \beta; leq; branch(|\alpha| + |\beta| + 6); \\ &\quad false; true; branch(2|\alpha| + 2|\beta| + 7); \\ &\quad \beta; \alpha; leq,\end{aligned}$$

pentru orice $\alpha, \beta \in F_{s_E}$;

- $\sigma_{leq}^F : F_{s_E} \times F_{s_E} \rightarrow F_{s_L}$ este dată prin

$$\sigma_{leq}^F(\alpha, \beta) = \alpha; \beta; leq,$$

pentru orice $\alpha, \beta \in F_{s_E}$;

- $\sigma_{assign}^F : F_{s_I} \times F_{s_E} \rightarrow F_{s_W}$ este dată prin

$$\sigma_{assign}^F(x, \alpha) = \alpha; put(x),$$

pentru orice $x \in F_{s_I}$ și $\alpha \in F_{s_E}$;

- $\sigma_{seq}^F : F_{s_W} \times F_{s_W} \rightarrow F_{s_W}$ este dată prin

$$\sigma_{seq}^F(\alpha, \beta) = \alpha; (\beta + |\alpha| + 1),$$

pentru orice $\alpha, \beta \in F_{s_W}$;

- $\sigma_{if}^F : F_{s_L} \times F_{s_W} \times F_{s_W} \rightarrow F_{s_W}$ este dată prin

$$\begin{aligned}\sigma_{if}^F(\alpha, \beta_1, \beta_2) &= \alpha; branch(|\alpha| + |\beta_2| + 4); \\ &\quad (\beta_2 + |\alpha| + 1); \\ &\quad true; branch(|\alpha| + |\beta_1| + |\beta_2| + 4); \\ &\quad (\beta_1 + |\alpha| + |\beta_2| + 3),\end{aligned}$$

pentru orice $\alpha \in F_{s_L}$ și $\beta_1, \beta_2 \in F_{s_W}$;

- $\sigma_{while}^F : F_{s_L} \times F_{s_W} \rightarrow F_{s_W}$ este dată prin

$$\begin{aligned}\sigma_{while}^F(\alpha, \beta) &= \alpha; branch(|\alpha| + 4); \\ &\quad true; branch(|\alpha| + |\beta| + 6); \\ &\quad (\beta + |\alpha| + 3); true; branch(1),\end{aligned}$$

pentru orice $\alpha \in F_{s_L}$ și $\beta \in F_{s_W}$.

“Corectitudinea” definiției operațiilor de mai sus se bazează pe următoarele observații:

- programele (mașinii cu stivă) asociate expresiilor aritmetice și logice se termină întotdeauna;
- programele asociate expresiilor aritmetice nu conțin instrucțiunea *branch*;
- programele asociate expresiilor logice pot conține instrucțiunea *branch* dar, în acest caz, aceasta schimbă conținutul contorului de program la un număr cuprins între 1 și $|\alpha| + 1$, unde α este programul în cauză.

Unicul homomorfism $eval_F$ de la $T_{\Sigma(G)}$ la F asociază unui program structurat traducerea acestuia într-un program “echivalent” pentru mașini cu stivă. De exemplu, dacă considerăm programul structurat

$$\sigma_{while}(\sigma_{leq}(\sigma_{i2e}(\sigma_y), \sigma_{i2e}(\sigma_x)), \sigma_{assign}(\sigma_x, \sigma_{add}(\sigma_{i2e}(\sigma_x), \sigma_{neg}(\sigma_{i2e}(\sigma_y))))),$$

atunci programul asociat prin $eval_F$ este

$$get(y); get(x); leq; branch(7); true; branch(13);$$

$$get(x); get(y); neg; add; true; branch(1).$$

Vom nota prin **C** algebra $eval_F(T_{\Sigma(G)})$ și vom arăta că traducerea realizată de $eval_C$ este corectă relativ la algebra semantică **A** descrisă în secțiunea anterioară. Adică, vom arăta că există un homomorfism h de la **C** la **A** astfel încât $h \circ eval_C = eval_A$ (a se vedea diagrama din Figura 8.21(b)). Homomorfismul h va fi obținut prin restricția unei funcții parțiale $f : F \rightsquigarrow A$ la **C**. Pentru a defini această funcție avem nevoie de câteva elemente pregătitoare.

Prin *configurație* a unei mașini cu stivă vom înțelege un vector (γ, δ, r, c) , unde $\gamma : V \rightarrow \mathbf{Z} \cup \{\text{true}, \text{false}\}$ este o stare a memoriei mașinii, $\delta \in (\mathbf{Z} \cup \{\text{true}, \text{false}\})^*$ reprezintă conținutul stivei mașinii, $r \in \mathbf{Z} \cup \{\text{true}, \text{false}\}$ este conținutul registrului, iar $c \in \mathbf{N}$ este conținutul contorului de program. Prin $\text{top}(\delta)$ vom nota elementul din vârful stivei δ , dacă stiva este nevidă, și λ , altfel.

Dat un program α , vom nota

$$(\gamma, \delta, r, c) \xrightarrow{\alpha} (\gamma', \delta', r', c')$$

dacă mașina, pornind de la configurația (γ, δ, r, c) și programul α , se oprește prin terminarea programului în configurația $(\gamma', \delta', r', c')$.

Funcția f este definită atunci astfel:

- $f_{s_I} : F_{s_I} \rightarrow A_{s_I}$ este dată prin

$$f_{s_I}(x) = x,$$

pentru orice $x \in F_{s_I}$;

- $f_{s_E} : F_{s_E} \rightarrow A_{s_E}$ este dată prin

$$f_{s_E}(\alpha)(\gamma) = \begin{cases} z, & \text{dacă } (\gamma, \lambda, 0, 1) \xrightarrow{\alpha} (\gamma', \delta, r, c) \text{ și } \text{top}(\delta) = z \in \mathbf{Z} \\ \uparrow, & \text{altfel,} \end{cases}$$

pentru orice $\alpha \in F_{s_E}$ și $\gamma \in \Gamma$;

- $f_{s_L} : F_{s_L} \rightarrow A_{s_L}$ este dată prin

$$f_{s_L}(\alpha)(\gamma) = \begin{cases} b, & \text{dacă } (\gamma, \lambda, 0, 1) \xrightarrow{\alpha} (\gamma', \delta, r, c) \text{ și } \text{top}(\delta) = b \\ \uparrow, & \text{altfel,} \end{cases}$$

pentru orice $\alpha \in F_{s_L}$ și $\gamma \in \Gamma$;

- $f_{s_W} : F_{s_W} \rightarrow A_{s_W}$ este dată prin

$$f_{s_W}(\alpha)(\gamma) = \begin{cases} \gamma', & \text{dacă } (\gamma, \lambda, 0, 1) \xrightarrow{\alpha} (\gamma', \delta, r, c) \\ \uparrow, & \text{altfel,} \end{cases}$$

pentru orice $\alpha \in F_{s_W}$ și $\gamma \in \Gamma$.

Definim acum funcția $h : C \rightarrow A$ prin $h = f|_C$. Are loc:

Teorema 8.9.1.4. h este homomorfism de la **C** la **A** ce satisfacă $h \circ eval_C = eval_A$.

Demonstrație. Vom schița demonstrația acestei teoreme verificând câteva din proprietățile de homomorfism pe care trebuie să le satisfacă funcția h , restul rămânând în seama cititorului.

În primul rând facem următoarele remarcări:

1. programele structurate, așa cum au fost definite, au ca date de intrare valori întregi. Mașinile cu stivă pot avea în memorie la un moment dat și valori Booleene. Însă, dacă programul executat de o mașină cu stivă este din **C** (adică, el provine de la un program structurat), atunci programul în cauză, pornind numai cu numere întregi în memorie, nu va introduce niciodată valori Booleene în memorie;
2. dacă t este un term de sort $s \in \{s_E, s_L\}$ iar α este programul asociat acestuia ($eval_{C,s}(t) = \alpha$), atunci

$$(\gamma, \delta, r, 1) \xrightarrow{\alpha} (\gamma, \delta', r', |\alpha| + 1),$$

unde $\delta' = \delta v$, $eval_{A,s}(t)(\gamma) = v$ și $r' \in \mathbf{Z} \cup \{\text{true}, \text{false}\}$, pentru orice configurație $(\gamma, \delta, r, 1)$ cu $\gamma \in \Gamma$.

Vom arăta acum că are loc

$$(1) \quad h_{s_I}(\sigma_x^C) = \sigma_x^A(h_{s_I}(\emptyset)),$$

pentru orice $x \in C_{s_I}$.

Conform definiției funcțiilor σ_x^C și h , are loc $h_{s_I}(\sigma_x^C) = h_{s_I}(x) = x$. Pe de altă parte, $\sigma_x^A(h_{s_I}(\emptyset)) = \sigma_x^A(\emptyset) = x$, ceea ce arată că are loc (1).

Vom arăta că are loc

$$(2) \quad h_{s_E}(\sigma_{i2e}^C(x)) = \sigma_{i2e}^A(h_{s_I}(x)),$$

pentru orice $x \in C_{s_I}$.

Conform definiției, $\sigma_{i2e}^C(x) = get(x)$ și, ca urmare,

$$(\gamma, \lambda, 0, 1) \xrightarrow{get(x)} (\gamma, \gamma(x), 0, 2).$$

pentru orice $\gamma \in \Gamma$. Deci $h_{s_E}(\sigma_{i2e}^C(x))(\gamma) = \gamma(x)$. Cum

$$\sigma_{i2e}^A(h_{s_I}(x))(\gamma) = \sigma_{i2e}^A(x)(\gamma) = \gamma(x),$$

obținem (2).

Vom arăta că are loc

$$(3) \quad h_{s_L}(\sigma_{not}^C(\alpha)) = \sigma_{not}^A(h_{s_L}(\alpha)),$$

pentru orice $\alpha \in C_{s_L}$.

Conform celei de a doua remarcări de mai sus, programul α aplicat unei configurații $(\gamma, \lambda, 0, 1)$, unde $\gamma \in \Gamma$, va produce o configurație $(\gamma, b, r, |\alpha| + 1)$, unde b este o valoare Booleană. Să presupunem că $b = \text{true}$. Conform definiției programului $\sigma_{not}^C(\alpha)$, are loc:

$$(\gamma, \lambda, 0, 1) \rightarrow (\gamma, \text{true}, r, |\alpha| + 1) \rightarrow (\gamma, \lambda, r, |\alpha| + 5) \rightarrow (\gamma, \text{false}, r, |\alpha| + 6),$$

ceea ce arată că $h_{s_L}(\sigma_{not}^C(\alpha))(\gamma) = \text{false}$. Cum $h_{s_L}(\alpha)(\gamma) = \text{true}$, obținem (3). În mod similar se discută cazul $b = \text{false}$.

Vom arăta că are loc

$$(4) \quad h_{s_W}(\sigma_{assign}^C(x, \alpha)) = \sigma_{assign}^A(h_{s_I}(x), h_{s_E}(\alpha)),$$

pentru orice $x \in C_{s_I}$ și $\alpha \in C_{s_E}$.

Conform definiției, $\sigma_{assign}^C(x, \alpha) = \alpha; put(x)$. Atunci

$$(\gamma, \lambda, 0, 1) \xrightarrow{\alpha} (\gamma, \delta, r, |\alpha| + 1) \xrightarrow{put(x)} (\gamma', \delta', r, |\alpha| + 2),$$

unde $\delta = \delta'v$, $\gamma'(y) = \gamma(y)$, pentru orice $y \neq x$, și $\gamma'(x) = v$. Aceasta arată că $h_{s_W}(\sigma_{assign}^C(x, \alpha))(\gamma) = \gamma'$. Pe de altă parte,

$$\sigma_{assign}^A(h_{s_I}(x), h_{s_E}(\alpha))(\gamma) = \sigma_{assign}^A(x, h_{s_E}(\alpha))(\gamma) = \sigma_{assign}^A(x, v)(\gamma) = \gamma',$$

ceea ce stabilește (4).

Restul proprietăților de homomorfism pe care trebuie să le satisfacă h , precum și proprietatea $h \circ eval_C = eval_A$, se verifică în mod similar celor de mai sus. \square

Teorema 8.9.1.4 stabilește corectitudinea traducerii programelor structurate în programe pentru mașini cu stivă, relativ la semantica dată de A .

8.9.2. Specificarea algebrică a tipurilor abstracte de date

8.9.2.1. Introducere

Metodologia clasică de dezvoltare de soft constă în scrierea unui program pentru problema dată, compilarea acestuia și apoi execuția lui (Figura 8.23). Testarea programului pe o colecție de date critice de intrare este o necesitate. În timp, programul este întreținut și perfecționat.

Această metodologie are cel puțin două dezavantaje mari:

- bazată pe testare, ea ar putea confirma existența erorilor, dar nu și lipsa lor;
- rezultatele sunt comparate cu ceea ce ne aşteptăm după cum am înțeles problema.

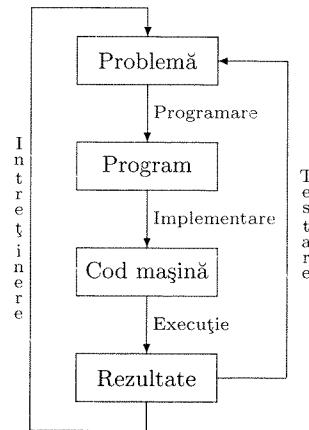


Figura 8.23: Procesul de realizare a unui produs software

O îmbunătățire a acestei metodologii trebuie să includă, în mod necesar, pași de specificare formală a problemei, aşa cum este prezentat în Figura 8.24. Această nouă variantă include doi pași suplimentari, unul de specificare informală și unul de specificare formală. În urma specificării formale se poate trece la o prototipizare rapidă care să ne spună dacă modelul produs este ceea ce dorim. Faza de implementare a specificării formale poate fi urmată de verificare. Având o specificare formală, este foarte potrivit a ne pune problema verificării, putând astfel elimina multe cazuri nedorite. Faza de verificare, aşa cum am văzut în Secțiunea 7.3.2, confirmă corectitudinea specificării relativ la anumite proprietăți. Ea poate fi înlocuită cu o fază de analiză care are un grad de automatizare mai mare. Însă, accentuăm că utilizarea verificării sau analizei se poate face în baza specificării formale.

În această secțiune vom fi interesați de specificarea tipurilor de date. O *specificare* a unei clase de obiecte înseamnă descrierea clasei respective prin intermediul unor proprietăți ale acesteia, proprietăți care se doresc a fi exploatațe. În principal, aceste proprietăți sunt date prin intermediul formulelor într-o anumită logică asociată clasei de obiecte ce urmează a fi specificată. În cazul tipurilor de date, vom utiliza ecuații.

Specificațiile pot fi clasificate în [130]:

1. *specificații atomice*. Aceste specificații sunt construite pornind de la bază, fără a avea ca suport alte specificații. În principal, o specificație atomică constă dintr-o semnatură Σ și o mulțime de formule Φ într-o logică asociată semnaturii. Semantica specificației este definită ca fiind clasa tuturor Σ -algebrelor ce sunt modele ale tuturor formulelor din Φ . Printre specificațiile atomice, cele mai utilizate sunt:

- (a) *specificații loose*. În cadrul acestor specificații nu se folosesc o logică

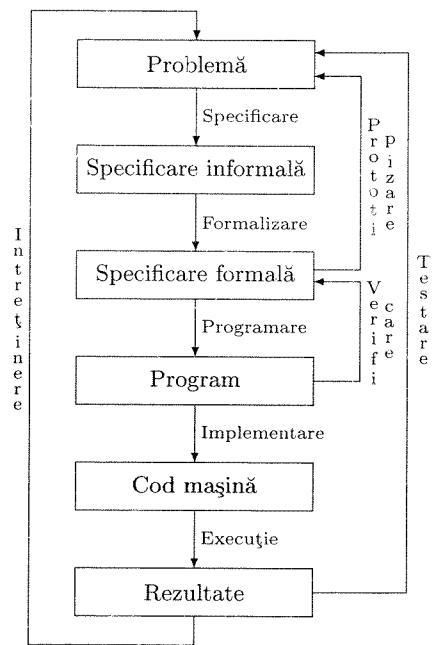


Figura 8.24: Realizarea unui produs software cu verificare

specifică. Sintaxa acestora este de forma $Spec = (\Sigma, \Phi)$, unde $\Phi \subseteq L(\Sigma)$ este o mulțime de formule de un anumit tip peste Σ , iar semantica este $\mathcal{M}(Spec) = Mod(\Phi)$;

- (b) *specificații inițiale*. Spre deosebire de specificațiile loose, aceste tipuri de specificații se bazează pe o logică specifică, și anume, logica ecuațională. Sintaxa acestora este de forma $Spec = (\Sigma, X, E)$, unde E este o mulțime de ecuații peste Σ și X , iar semantica $\mathcal{M}(Spec)$ este clasa tuturor algebrelor izomorfe cu $\mathbf{T}_\Sigma / \equiv_E$;
 - (c) *specificații constructive*. Acestea sunt definite prin intermediul logicii ecuaționale. Ele constau dintr-o signatură, o mulțime de ecuații și o mulțime de operații numite *constructori*. Constructorii ajută la definirea de noi obiecte, în manieră inductivă. Semantica acestor specificații este furnizată prin definirea în mod explicit a unei algebri;
2. *specificații compuse*. Acestea sunt specificații obținute prin compunerea de specificații atomice. În general, specificațiile compuse se realizează într-un anumit limbaj de specificare utilizând construcții specifice.

În secțiunile ce urmează ne vom referi numai la specificații inițiale. Pentru detalii

suplimentare atât asupra specificațiilor inițiale cât și a altor tipuri de specificații cititorul este îndrumat, de exemplu, către [130].

8.9.2.2. Specificații inițiale ale tipurilor abstracte de date

Informal, un tip de dată este o pereche formată dintr-o mulțime de domenii împreună cu o mulțime de operații pe aceste domenii. De exemplu, tipul *integer* este format dintr-o submulțime a mulțimii numerelor întregi, a cărei dimensiune depinde de implementare, împreună cu operații uzuale ca adunarea, scăderea, înmulțirea, împărțirea, operații logice etc.

O algebră universală multisortată este o colecție de mulțimi împreună cu o colecție de operații pe aceste mulțimi. Ca urmare, este cât se poate de natural să adoptăm algebrele universale multisortate ca modele matematice ale tipurilor de date. O astfel de abordare prezintă multe avantaje, printre care:

- precizie matematică;
- independență de implementare;
- posibilitate de a defini axiomatic operații;
- tehnici formale de verificare.

Pe lângă tipurile uzuale de date, multe limbaje de programare moderne permit utilizatorilor să definească noi tipuri de date pornind de la tipuri existente, cum ar fi *stack*, *list*, *queue* etc. De obicei, aceasta se face prin specificarea unor domenii (care pot fi definite inductiv, cum ar fi mulțimea numerelor naturale) și a unui număr de operații (definite în manieră recursivă, prin ecuații).

În tot ceea ce urmează vom presupune că Σ este o signatură iar X este o familie disjunctă de variabile, disjunctă și de Σ . În plus, vom presupune că pentru orice sort există oricât de multe variabile avem nevoie.

Definiția 8.9.2.1. Se numește *specificație inițială* orice sistem $Spec = (\Sigma, X, E)$, unde Σ și X sunt ca mai sus, iar E este o mulțime de ecuații peste Σ și X .

Vom simplifica terminologia de “specificație inițială” la cea de *specificație* deoarece numai astfel de specificații vom folosi.

O specificație $Spec = (\Sigma, X, E)$ are scopul de a

- fixa nume de domenii (prin intermediul sorturilor);
- furniza nume de operații (prin intermediul simbolurilor funcționale);
- stabili proprietățile operațiilor pe care dorim să le implementăm (prin intermediul ecuațiilor).

Informaticienii descriu specificațiile în manieră tabelară, așa cum este specificația **Bool** de mai jos.

```
Bool = sorts: bool
      opns: true : bool
            false : bool
            neg : bool → bool
            and : bool bool → bool
      eqns: neg(true) = false
            neg(false) = true
            and(true, true) = true
            and(true, false) = false
            and(false, true) = false
            and(false, false) = false
```

Această specificație constă dintr-un singur sort *bool* (specificat în rubrica *sorts*), două constante de sort *bool*, noteate *true* și *false*, o operație *neg* de tip (*bool, bool*) și o operație *and* de tip (*bool bool, bool*) (toate acestea specificate în rubrica *opns*), și 6 ecuații ce descriu proprietățile operațiilor (specificate în rubrica *eqns*).

Specificația **Nat** de mai jos este descrisă în manieră similară, dar conține și specificații de variabile în rubrica *vars*.

```
Nat = sorts: nat
      opns: zero : nat
            succ : nat → nat
            add : nat nat → nat
      vars: x : nat
            y : nat
      eqns: add(x, zero) = x
            add(x, succ(y)) = succ(add(x, y))
```

Specificația **NatBool** combină specificațiile **Bool** și **Nat** și mai adaugă câteva nume de operații împreună cu ecuațiile corespunzătoare.

```
NatBool = sorts: bool
      nat
      opns: true : bool
            false : bool
            and : bool bool → bool
            neg : bool → bool
            zero : nat
            succ : nat → nat
            add : nat nat → nat
            mult : nat nat → nat
            leq : nat nat → bool
      vars: x : nat
            y : nat
      eqns: neg(true) = false
            neg(false) = true
```

```
and(true, true) = true
and(true, false) = false
and(false, true) = false
and(false, false) = false
add(x, zero) = x
add(x, succ(y)) = succ(add(x, y))
mult(x, zero) = zero
mult(x, succ(y)) = add(mult(x, y), x)
leq(x, add(x, y)) = true
leq(add(x, succ(y)), x) = false
```

Pentru a avea flexibilitate, putem realiza specificații modulare și, apoi, să le combinăm aşa cum este arătat mai jos pentru specificația **NatBool**:

```
NatBool = Bool + Nat +
      opns: mult : nat nat → nat
            leq : nat nat → bool
      vars: x : nat
            y : nat
      eqns: mult(x, zero) = zero
            mult(x, succ(y)) = add(mult(x, y), x)
            leq(x, add(x, y)) = true
            leq(add(x, succ(y)), x) = false
```

Definiția 8.9.2.2. Fie $Spec = (\Sigma, X, E)$ o specificație. Se numește *Spec-algebră* orice algebră ce este model al mulțimii E de ecuații.

O *Spec-algebră* este o implementare concretă a tipului de dată specificat de *Spec*. Ca urmare, mulțimea tuturor *Spec-algebrelor* poate fi gândită ca fiind *semantica specificației Spec*. O astfel de semantică poartă denumirea de *semantică loose*.

Fie $Spec = (\Sigma, X, E)$ o specificație. Orice term t peste Σ și X are o anumită interpretare într-o *Spec-algebră* \mathbf{A} . Dacă termul este de bază, atunci interpretarea lui este $eval_{\mathbf{A}, s}(t)$, presupunând că t este de sort s ($eval_{\mathbf{A}}$ fiind unicul homomorfism de la \mathbf{T}_Σ la \mathbf{A}). Putem gândi t ca o reprezentare sintactică a elementului de dată $eval_{\mathbf{A}, s}(t)$. Însă, \mathbf{A} poate conține elemente care să nu aibă reprezentări sintactice prin intermediul specificației *Spec*. Aceasta se întâmplă exact atunci când $eval_{\mathbf{A}}(\mathbf{T}_\Sigma)$ este subalgebră proprie în \mathbf{A} . Adică, exact atunci când \mathbf{A} nu este algebră minimală (reamintim că \mathbf{T}_Σ este algebră minimală).

Ca o concluzie, dacă \mathbf{A} este minimală, atunci orice element de dată $a \in A_s$, unde s este un sort, admite o reprezentare sintactică în specificația *Spec*, reprezentare sub forma unui term de bază.

Doi termi de bază distincți, t_1 și t_2 , pot avea aceeași interpretare într-o *Spec-algebră* \mathbf{A} . De exemplu,

$$t_1 = add(zero, succ(zero))$$

și

$$t_2 = \text{succ}(\text{zero})$$

interpretații în \mathbf{A} pot conduce la același element de dată, $\text{eval}_{\mathbf{A}, \text{nat}}(\text{succ}(\text{zero}))$ (de exemplu, într-o algebră de numere naturale acest element este numărul natural 1). O pereche de termi (t_1, t_2) ca cei de mai sus poartă denumirea de *confuzie*. Ar fi de preferat să putem stabili sintactic dacă o pereche de termi (t_1, t_2) este sau nu o confuzie. Adică, am dori să avem îndeplinită următoarea proprietate

$$\mathbf{A} \models t_1 = t_2 \Leftrightarrow E \vdash t_1 = t_2,$$

pentru orice doi termi t_1 și t_2 de același sort și orice *Spec*-algebră \mathbf{A} . Evident, implicăția

$$E \vdash t_1 = t_2 \Rightarrow \mathbf{A} \models t_1 = t_2$$

este întotdeauna adevărată. Cealaltă implicăție poate să nu fie.

Definiția 8.9.2.3. Fie $\text{Spec} = (\Sigma, X, E)$ o specificație. Spunem că o Σ -algebră \mathbf{A} este *tipică pentru specificația Spec* dacă are loc:

$$\mathbf{A} \models t_1 = t_2 \Leftrightarrow E \vdash t_1 = t_2,$$

pentru orice doi termi de bază t_1 și t_2 de același sort.

Teorema 8.9.2.1. Fie $\text{Spec} = (\Sigma, X, E)$ o specificație. O Σ -algebră \mathbf{A} este minimală și tipică pentru specificația *Spec* dacă și numai dacă \mathbf{A} este algebră inițială în clasa $\text{Mod}(E)$.

Demonstrație. Să presupunem că \mathbf{A} este minimală și tipică. Vom arăta că \mathbf{A} este izomorfă cu $\mathbf{T}_{\Sigma/E}$. Cum $\mathbf{T}_{\Sigma/E}$ este inițială și clasa $\text{Mod}(E)$ este închisă la izomorfism, \mathbf{A} va fi în clasa $\text{Mod}(E)$ și, în plus, inițială în această clasă.

Știm că există un unic homomorfism h de la \mathbf{T}_{Σ} la \mathbf{A} . Cum \mathbf{A} este minimală, h trebuie să fie epimorfism. Conform primei teoreme de izomorfism, $\mathbf{T}_{\Sigma}/\ker(h) \cong \mathbf{A}$. Deci ceea ce ne rămâne de arătat este că $\mathbf{T}_{\Sigma}/\ker(h)$ și $\mathbf{T}_{\Sigma/E}$ sunt izomorfe. Fie

$$f : \mathbf{T}_{\Sigma}/\ker(h) \rightarrow \mathbf{T}_{\Sigma/E}$$

dată prin

$$f_s([t]_{\ker(h_s)}) = [t]_{=_{E,s}},$$

pentru orice sort s și term t de sort s .

Utilizând faptul că \mathbf{A} este tipică obținem imediat că f este bine definită (dacă t și t' sunt în relația $\ker(h_s)$, atunci proprietatea algebrei \mathbf{A} de a fi tipică conduce la faptul că t și t' sunt în relația $=_{E,s}$).

Cititorul este invitat să arate că f este homomorfism bijectiv, încheind astfel prima parte a demonstrației.

Reciproc, dacă \mathbf{A} este inițială în clasa $\text{Mod}(E)$, atunci \mathbf{A} este izomorfă cu $\mathbf{T}_{\Sigma/E}$ și, cum această algebră este minimală, \mathbf{A} va fi minimală.

Fie $t = t'$ o ecuație. Utilizând iarăși faptul că \mathbf{A} este inițială în clasa $\text{Mod}(E)$, obținem că $E \models t = t'$ dacă și numai dacă $\mathbf{A} \models t = t'$. Combinând acum cu teorema de corectitudine și completitudine a logicii ecuaționale deducem că \mathbf{A} este tipică. \square

Ca urmare a discuției purtate mai sus și a Teoremei 8.9.2.1, definim *semantica unei specificații inițiale* ca fiind clasa tuturor algebrelor inițiale din $\text{Mod}(E)$. Această clasă se mai notează prin $\text{ADT}(\text{Spec})$ și se mai numește *tipul abstract de dată inducător definit de specificația Spec*. Evident, $\text{ADT}(\text{Spec})$ este clasa tuturor algebrelor izomorfe cu $\mathbf{T}_{\Sigma/E}$.

Metodologia care este de preferat a fi urmată atunci când se dorește specificarea unui tip de dată este următoarea [74]:

1. *Specificare informală.* În cadrul acestui pas se stabilesc obiectivele urmărite, cu fixarea informală a domeniilor și operațiilor ce urmează a fi descrise;
2. *Specificare formală.* Se stabilește un model matematic pentru tipul de date ce se urmărește a fi specificat. De fapt, modelul matematic este de preferat să fie sub forma unei algebri \mathbf{A} ;
3. *Construcția unei specificații.* Se construiește efectiv specificația *Spec* dorită;
4. *Corectitudine.* Se încearcă să se verifice, pe cât posibil, dacă specificația *Spec* respectă modelul matematic \mathbf{A} (adică, dacă \mathbf{A} este o *Spec*-algebră). De preferat este ca \mathbf{A} să fie element al clasei $\text{ADT}(\text{Spec})$ (ceea ce se poate vedea încercând să se stabilească izomorfismul dintre $\mathbf{T}_{\Sigma/E}$ și \mathbf{A}).

Pasul de corectitudine depinde, evident, de modelul matematic obținut la pasul al doilea. Pentru specificații complexe, obținerea unui astfel de model matematic (sub formă unei algebri) poate constitui o problemă. Dar, odată ce ce s-a obținut un astfel de model, se poate recurge la verificarea corectitudinii specificării.

Așa cum se menționează în [74], lucrurile pot fi privite și invers. Adică, presupunând că specificația este corectă, se poate încerca verificarea modelului (prin stabilirea izomorfismului de la pasul 4).

Vom exemplifica în cele ce urmează metodologia de mai sus.

Exemplul 8.9.2.1. Dorim să specificăm un tip de date ale cărei elemente sunt stive peste un alfabet dat.

Specificare informală. Presupunem că M este o mulțime finită, iar tipul de date pe care dorim să îl definim este format din stive peste M , cu operațiile uzuale *Push* și *Pop*. Facem următoarele remarcări:

- o stivă va fi modelată ca un cuvânt peste M ;
- stiva vidă va fi modelată prin cuvântul vid;
- vârful stivei va fi la dreapta cuvântului;

- operația *Push* va adăuga la o stivă un nou element din M (adăugarea se va face la dreapta și va fi simulață prin concatenare de cuvinte);
- operația *Pop* va elimina elementul din vârful stivei. Dacă stiva este vidă, atunci rezultatul va fi tot stiva vidă.

Specificare formală. Fie $M = \{a_1, \dots, a_n\}$ o mulțime cu n elemente, unde $n \geq 1$. M va fi mulțimea peste care se construiesc stivele. Mulțimea tuturor stivelor va fi monoidul liber M^* generat de M . Stiva vidă va fi λ .

Operațiile *Push* și *Pop* sunt definite astfel:

- $\text{Push} : M^* \times M \rightarrow M^*$ este dată prin

$$\text{Push}(w, a) = wa,$$

pentru orice $w \in M^*$ și $a \in M$;

- $\text{Pop} : M^* \rightarrow M^*$ este dată prin

$$\text{Pop}(w) = \begin{cases} w', & \text{dacă } (\exists a \in M)(w = w'a) \\ \lambda, & \text{altfel,} \end{cases}$$

pentru orice $w \in M^*$.

Am obținut astfel o Σ -algebră \mathbf{A} ce poate fi descrisă astfel:

- $S = \{\text{alph}, \text{stack}\}$ este o mulțime cu două sorturi;
- Σ este o signatură dată prin:
 - $\Sigma_{\text{alph}} = \{e_1, \dots, e_n\}$;
 - $\Sigma_{\text{stack}} = \{\text{estack}\}$;
 - $\Sigma_{\text{stack alph, stack}} = \{\text{push}\}$;
 - $\Sigma_{\text{stack, stack}} = \{\text{pop}\}$;
 - $\Sigma_{w,s} = \emptyset$, în toate celelalte cazuri;
- $A_{\text{alph}} = M$ și $A_{\text{stack}} = M^*$;
- $e_i^A = a_i$ pentru orice $1 \leq i \leq n$, $\text{estack}^A = \lambda$, $\text{push}^A = \text{Push}$ și $\text{pop}^A = \text{Pop}$.

Construcția specificației. Considerăm specificația **Stack** dată ca mai jos.

```
Stack = sorts: alph
          stack
          opns:   e1      : alph
                  ...
                  en      : alph
```

```
estack : stack
push   : stack alph → stack
pop    : stack → stack
vars:   x      : stack
        y      : alph
eqns:  pop(estack) = estack
       pop(push(x,y)) = x
```

Corectitudine. Vom arăta în cele ce urmează că algebra \mathbf{A} este în $ADT(\mathbf{Stack})$, arătând că este izomorfă cu $\mathbf{T}_{\Sigma}/=_E$.

În primul rând observăm că \mathbf{A} este model al mulțimii E de ecuații a specificației **Stack**. În adevăr, dacă γ este o asignare în \mathbf{A} , atunci:

$$\begin{aligned} \bar{\gamma}_{\text{stack}}(\text{pop}(\text{estack})) &= \text{pop}^A(\text{estack}^A) \\ &= \text{Pop}(\lambda) \\ &= \lambda \\ &= \text{estack}^A \\ &= \bar{\gamma}_{\text{stack}}(\text{estack}) \end{aligned}$$

și

$$\begin{aligned} \bar{\gamma}_{\text{stack}}(\text{pop}(\text{push}(x, y))) &= \text{pop}^A(\text{push}^A(\gamma_{\text{stack}}(x), \gamma_{\text{alph}}(y))) \\ &= \text{Pop}(\text{Push}(\gamma_{\text{stack}}(x), \gamma_{\text{alph}}(y))) \\ &= \text{Pop}(\gamma_{\text{stack}}(x)\gamma_{\text{alph}}(y)) \\ &= \gamma_{\text{stack}}(x) \\ &= \bar{\gamma}_{\text{stack}}(x). \end{aligned}$$

Definim funcția $f : T_{\Sigma}/=_E \rightarrow A$ astfel încât diagrama din Figura 8.25 să fie comutativă, adică

$$f_s([t]_{=_E, s}) = \text{eval}_{\mathbf{A}, s}(t),$$

pentru orice sort s și term t de sort s ($f_{=_E}$ este epimorfismul natural induș de $=_E$).

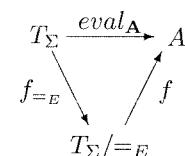


Figura 8.25: Izomorfismul dintre $\mathbf{T}_{\Sigma}/=_E$ și \mathbf{A}

Vom arăta că f este izomorfism parcurgând următorii pași:

- f este bine definită. Dacă t și t' sunt termi de sort s și $t =_{E,s} t'$, atunci

$$\begin{aligned} t =_{E,s} t' &\Leftrightarrow E \vdash t = t' \\ &\Leftrightarrow E \models t = t' \\ &\Leftrightarrow \text{Mod}(E) \models t = t' \\ &\Rightarrow \mathbf{A} \models t = t' \\ &\Rightarrow \text{eval}_{\mathbf{A},s}(t) = \text{eval}_{\mathbf{A},s}(t'), \end{aligned}$$

ceea ce arată că $f_s([t] =_{E,s}) = f_s([t'] =_{E,s})$;

- f este homomorfism:

- fie $1 \leq i \leq n$. Atunci:

$$\begin{aligned} f_{\text{alph}}(e_i^{T_\Sigma / =_E}) &= f_{\text{alph}}([e_i^{T_\Sigma}] =_{E,\text{alph}}) \\ &= f_{\text{alph}}([e_i] =_{E,\text{alph}}) \\ &= \text{eval}_{\mathbf{A},\text{alph}}(e_i) \\ &= e_i^A; \end{aligned}$$

- pentru estack are loc:

$$\begin{aligned} f_{\text{stack}}(\text{estack}^{T_\Sigma / =_E}) &= f_{\text{stack}}([\text{estack}^{T_\Sigma}] =_{E,\text{stack}}) \\ &= f_{\text{stack}}([\text{estack}] =_{E,\text{stack}}) \\ &= \text{eval}_{\mathbf{A},\text{stack}}(\text{estack}) \\ &= \text{estack}^A; \end{aligned}$$

- dacă t_1 este term de sort stack iar t_2 este term de sort alph , atunci:

$$\begin{aligned} f_{\text{stack}}(\text{push}^{T_\Sigma / =_E}([t_1] =_{E,\text{stack}}, [t_2] =_{E,\text{alph}})) &= \\ &= f_{\text{stack}}([\text{push}^{T_\Sigma}(t_1, t_2)] =_{E,\text{stack}}) \\ &= f_{\text{stack}}([\text{push}(t_1, t_2)] =_{E,\text{stack}}) \\ &= \text{eval}_{\mathbf{A},\text{stack}}(\text{push}(t_1, t_2)) \\ &= \text{push}^A(\text{eval}_{\mathbf{A},\text{stack}}(t_1), \text{eval}_{\mathbf{A},\text{alph}}(t_2)) \\ &= \text{push}^A(f_{\text{stack}}([t_1] =_{E,\text{stack}}), f_{\text{alpha}}([t_2] =_{E,\text{alph}})); \end{aligned}$$

- dacă t este term de sort stack , atunci:

$$\begin{aligned} f_{\text{stack}}(\text{pop}^{T_\Sigma / =_E}([t] =_{E,\text{stack}})) &= \\ &= f_{\text{stack}}([\text{pop}^{T_\Sigma}(t)] =_{E,\text{stack}}) \\ &= f_{\text{stack}}([\text{pop}(t)] =_{E,\text{stack}}) \\ &= \text{eval}_{\mathbf{A},\text{stack}}(\text{pop}(t)) \\ &= \text{pop}^A(\text{eval}_{\mathbf{A},\text{stack}}(t)) \\ &= \text{pop}^A(f_{\text{stack}}([t] =_{E,\text{stack}})); \end{aligned}$$

- f este injectivă. Studiul injectivității acestei funcții este ceva mai delicat. Ideea de bază este de a defini niște reprezentanți de clasă "canonici" și de a lucra numai cu aceștia în stabilirea injectivității. Definim mulțimile

$$C_{\text{alph}} = \{e_1, \dots, e_n\}$$

și

$$C_{\text{stack}} = \{\text{push}(\dots \text{push}(\text{estack}, \alpha_1), \dots, \alpha_m) | (\forall i)(\alpha_i \in C_{\text{alph}})\} \cup \{\text{estack}\}.$$

Aceste mulțimi au următoarele proprietăți importante:

- (1) pentru orice sort s și $t \in T_{\Sigma,s}$ există $c \in C_s$ astfel încât $t =_{E,s} c$;
- (2) pentru orice sort s și $c_1, c_2 \in C_s$, dacă $\text{eval}_{\mathbf{A},s}(c_1) = \text{eval}_{\mathbf{A},s}(c_2)$, atunci $c_1 = c_2$.

Prima proprietate se obține cu ușurință prin inducție structurală, iar a doua de la definiția funcției eval_A și a modului de interpretare a simbolurilor funcționale în algebra \mathbf{A} .

Să stabilim acum injectivitatea funcției f . Fie s un sort, iar t și t' termi de sort s . Atunci, există $c, c' \in C_s$ astfel încât $t =_{E,s} c$ și $t' =_{E,s} c'$. Dacă presupunem că are loc $f_s([t] =_{E,s}) = f_s([t'] =_{E,s})$, atunci obținem $\text{eval}_{\mathbf{A},s}(c) = \text{eval}_{\mathbf{A},s}(c')$ de la care urmează $c = c'$ și, deci, $t =_{E,s} t'$. Aceasta însă stabilește injectivitatea funcției f ;

- f este surjectivă. Făcând apel la construcția de la pasul anterior, este ușor de verificat că, pentru orice sort s , f_s aplicată claselor de echivalentă cu reprezentanți din C_s acoperă A_s .

Toate acestea ne arată că $\mathbf{A} \in \text{ADT}(\text{Stack})$.

Demonstrația de corectitudine pentru specificații complexe poate fi foarte dificilă. Ca urmare, în astfel de cazuri poate fi suficient să arăta doar că modelul formal (algebra de la pasul al doilea) satisface specificația Spec în cauză (adică, este o Spec -algebră).

Operația pop din Exemplul 8.9.2.1, aplicată stivei vide sau oricărei stive cu exact un element returnează stiva vidă. Într-o abordare mai realistă, operația pop ar trebui ori să fie definită numai pentru stive nevide ori, dacă este definită și pentru stiva vidă, atunci să returneze în acest caz un element special, de exemplu "*error*", care să semnaleze faptul că stiva este vidă.

Pentru tipul de dată din Exemplul 8.9.2.1 nu s-a definit o operație *top* care, aplicată unei stive, să returneze elementul din vârful stivei. Definirea unei astfel de operații ar trebui făcută ori numai pentru stive nevide ori, dacă se consideră și stiva vidă, atunci operația ar trebui definită similar operației pop .

Algebrele universale multisortate, în varianta în care au fost introduse, nu pot trata în mod corespunzător astfel de cazuri de definire parțială a unor operații. Din acest motiv, în literatura de specialitate au fost propuse diverse extensii ale algebrelor multisortate, extensii asupra căror vom discuta sumar în secțiunea următoare.

8.9.2.3. Tratarea exceptiilor

Tratarea exceptiilor, adică a unor cazuri speciale ce apar în definirea unor operații cu tipuri de date, s-a dovedit o adevărată piatră de încercare pentru teoria specificării algebrice a tipurilor de date. Acest lucru a fost evidențiat de grupul ADJ încă de la început (a se vedea, de exemplu, [73, 74]).

Vom începe cu câteva exemple care să arate cititorului dificultățile ce pot apărea în tratarea exceptiilor (a se vedea și [7]). Să presupunem că am dori să specificăm un tip de dată de numere naturale cu operații uzuale, cum ar fi *pred* (precedesor), *mult* (înmulțire) etc. O definiție naturală pentru *pred* ar fi

$$\text{pred}(\text{succ}(x)) = x,$$

unde x este o variabilă de tip *nat* (sortul corespunzător mulțimii numerelor naturale), iar *succ* desemnează operația successor. În acest mod, operația *pred* nu este definită pentru constanta *zero*. O idee naturală de a defini această operație și pentru *zero* ar consta în considerarea unui element *error* de sort *nat* și introducerea axiomei

$$\text{pred}(\text{zero}) = \text{error}.$$

Într-o astfel de abordare ar trebui să răspundem întrebării: cine este *succ(error)*? Sau, cine este *mult(error, zero)*? Răspunsul natural la această întrebare ar fi *error*. Adică, *succ(error) = error*, *mult(error, zero) = error* etc. Pe de altă parte, una din axiomele ce definește operația de înmulțire trebuie să fie de forma

$$\text{mult}(x, \text{zero}) = \text{zero}.$$

Aceasta vine însă în contradicție cu *mult(error, zero) = error* deoarece conduce la *error = zero*. O soluție pentru a evita această inconsistенță ar fi să definim *mult(x, zero) = zero* doar pentru $x \neq \text{error}$. Ca urmare, am avea nevoie de ecuații condiționate de tipul

$$x \neq \text{error} \Rightarrow \text{mult}(x, \text{zero}) = \text{zero}.$$

Însă, cu astfel de ecuații nu mai este asigurată, în general, existența algebrei inițiale, ceea ce constituie o deficiență majoră¹³.

Printre abordările majore de tratare a exceptiilor în cadrul specificăriilor algebrice, propuse de-a lungul timpului, menționăm: abordarea prin predicate *ok* [74], abordarea prin algebre cu sorturi ordonate [73, 75], abordarea prin algebre cu apartenență [147] și abordarea prin algebre cu etichete [7]. Vom trece în revistă sumar primele 3 abordări.

¹³Ecuăția condiționată “ $x \neq \text{error} \Rightarrow \text{mult}(x, \text{zero}) = \text{zero}$ ” are ca precondiție atomul negativ $\neg(x = \text{error})$. Pentru specificații ce conțin ecuații cu atomi negativi în precondiții nu mai este asigurată, în general, existența algebelor inițiale [224].

Predicate *ok*. În [74] (a se vedea Exemplul 7E), autorii au introdus *predicate ok* pentru a putea aborda situații de tipul celor de mai sus. Un predicat *ok* spune când o structură este bine formată (dintr-un anumit punct de vedere) și, deci, i se poate aplica o anumită operație. Astfel, dacă definim predicatul *ok* prin

$$\begin{aligned} \text{ok}(\text{error}) &= \text{false} \\ \text{ok}(\text{zero}) &= \text{true} \\ \text{ok}(\text{succ}(x)) &= \text{ok}(x) \\ \text{ok}(\text{pred}(\text{zero})) &= \text{false} \\ \text{ok}(\text{pred}(\text{succ}(x))) &= \text{ok}(x) \\ \text{ok}(\text{mult}(x, y)) &= \text{ok}(x) \text{ and } \text{ok}(y), \end{aligned}$$

atunci am putea defini operația *mult* prin

$$\text{ok}(\text{mult}(x, \text{zero})) = \text{true} \Rightarrow \text{mult}(x, \text{zero}) = \text{zero}.$$

Chiar dacă acestă abordare ar părea rezonabilă, ea conduce la specificări foarte complexe aşa cum este menționat și în [74] (pagina 118)

“... the resulting total specification ... is unbelievable complicated. It is also unnecessarily complicated.”

În plus, recuperarea exceptiilor poate conduce la inconsistențe. De exemplu, dacă am recupera *succ(pred(zero))* prin *zero*, atunci aceasta ar conduce la inconsistența *false = true* deoarece

$$\text{false} = \text{ok}(\text{succ}(\text{pred}(\text{zero}))) = \text{ok}(\text{zero}) = \text{true}.$$

Algebre cu sorturi ordonate. În 1978, Joseph Goguen a propus o abordare a modului de tratare a exceptiilor prin *algebre cu sorturi ordonate* [73, 75], abordare ce s-a dovedit ulterior ca fiind una din cele mai potrivite. Ideea de bază constă în a defini subdomenii ale unor domenii pe care anumite funcții sunt partiale și a defini aceste funcții parțiale ca funcții totale pe respectivele subdomenii. De exemplu, în cazul simbolului funcțional *pred*, putem defini un sort *posnat* care să definească multimea tuturor numerelor naturale strict pozitive. Acest sort este tratat ca un *subsort* al sortului *nat* în sensul că domeniul definit de *posnat* este subdomeniul al domeniului definit de *nat*. Dacă mai considerăm un sort *errnat* care să conțină doar *error* și un sort *excnat* care să conțină doar *zero*, atunci putem defini *pred* prin supraîncărcare prin

$$\begin{aligned} \text{pred} &: \text{posnat} \rightarrow \text{nat} \\ \text{pred} &: \text{excnat} \rightarrow \text{errnat} \\ \text{pred} &: \text{errnat} \rightarrow \text{errnat} \end{aligned}$$

împreună cu ecuațiile corespunzătoare

$$\begin{aligned} \text{pred}(\text{succ}(x)) &= x \\ \text{pred}(\text{zero}) &= \text{error} \\ \text{pred}(\text{error}) &= \text{error} \end{aligned}$$

(prima ecuație este pentru simbolul funcțional *pred* de tip (*posnat, nat*), a doua pentru cel de tip (*excnat, errnat*), iar a treia pentru cel de tip (*errnat, errnat*)).

Vom prezenta în cele ce urmează, succint, elementele de bază asupra algebrelor cu sorturi ordonate. Primul lucru important într-o astfel de abordare este de a considera o relație de ordine parțială \leq pe mulțimea S a sorturilor. Astfel, dacă $s_1 \leq s_2$, atunci vom spune că s_1 este *subsort* al lui s_2 . Relația de ordine parțială \leq se extinde la S^* prin

$$s_1 \cdots s_n \leq s'_1 \cdots s'_m \Leftrightarrow n = m \wedge (\forall 1 \leq i \leq n)(s_i \leq s'_i)$$

și la $S^* \times S$ prin

$$(w_1, s_1) \leq (w_2, s_2) \Leftrightarrow w_1 \leq w_2 \wedge s_1 \leq s_2.$$

Perechea (S, \leq) este numită *dirijată prin componente conexe* dacă pentru orice componentă conexă $S' \subseteq S$ și orice $s_1, s_2 \in S'$, există $s \in S'$ astfel încât $s_1 \leq s$ și $s_2 \leq s$.

O *signatură S-sortată ordonată* este o pereche (Σ, \leq) formată dintr-o *signatură S-sortată* Σ și o relație de ordine parțială \leq pe S astfel încât are loc următoarea proprietate de monotonie:

$$(\forall \sigma \in \Sigma)(\sigma \in \Sigma_{w_1, s_1} \cap \Sigma_{w_2, s_2} \wedge w_1 \leq w_2 \Rightarrow s_1 \leq s_2).$$

O *signatură ordonată* (Σ, \leq) este numită *regulată* dacă pentru orice simbol funcțional σ și orice $w_0 \in S^*$, dacă σ are tipul (w_1, s_1) și $w_0 \leq w_1$, atunci există un cel mai mic tip (w, s) astfel încât $w_0 \leq w$ și $\sigma \in \Sigma_{w, s}$.

Fie (Σ, \leq) o *signatură ordonată*. O (Σ, \leq) -algebră este o Σ -algebră \mathbf{A} ce satisfacă în plus cerințele:

- pentru orice $s, s' \in S$, dacă $s \leq s'$, atunci $A_s \subseteq A_{s'}$;
- pentru orice $\sigma \in \Sigma$, dacă σ are atât tipul (w_1, s_1) cât și tipul (w_2, s_2) și $w_1 \leq w_2$, atunci $\sigma_{w_2, s_2}^A|_{A_{w_1}} = \sigma_{w_1, s_1}^A$ (prin $\sigma_{w, s}^A$ s-a notat operația de tip (w, s) indușă de $\sigma \in \Sigma_{w, s}$ în algebra A).

Un *homomorfism* de la o (Σ, \leq) -algebră \mathbf{A} la o (Σ, \leq) -algebră \mathbf{B} trebuie să satisfacă, pe lângă cerința uzuală de a fi homomorfism de Σ -algebrelor, și proprietatea $h_{s'}|_{A_s} = h_s$, pentru orice două sorturi s și s' cu $s \leq s'$.

Fie X o familie disjunctă de variabile, disjunctă și de (Σ, \leq) (aceste cerințe asupra familiei X de variabile vor fi considerate implicit impuse de-a lungul întregii secțiuni). *Termii peste* (Σ, \leq) și X se definesc în mod uzual dar cu următoarea cerință suplimentară:

- dacă $t = \sigma(t_1, \dots, t_n)$ este term de sort s și $s \leq s'$, atunci t este term și de sort s' .

Familia termilor peste (Σ, \leq) și X poate fi organizată ca o (Σ, \leq) -algebră, notată $\mathbf{T}_{(\Sigma, \leq)}(X)$. În cazul în care $X = \emptyset$ vom nota această algebră, similar cazului clasic, prin $\mathbf{T}_{(\Sigma, \leq)}$.

Teorema 8.9.2.2. ([75]) Dacă (Σ, \leq) este o signatură ordonată regulată, atunci $\mathbf{T}_{(\Sigma, \leq)}$ este algebră inițială în clasa tuturor (Σ, \leq) -algebrelor.

Regulile de deducție ale logicii ecuaționale pentru cazul signaturilor standard pot fi ușor rescrise și pentru signaturi ordonate. Vom face aceasta generalizând totodată și conceptul de ecuație la cel de *ecuație condiționată*. O astfel de ecuație este de forma “ $t = t'$ if C ”, unde $t = t'$ este o ecuație iar C este o mulțime finită (posibil vidă) de ecuații. Spunem că o astfel de ecuație este *validă* într-o (Σ, \leq) -algebră \mathbf{A} dacă, pentru orice asignare γ de la X în \mathbf{A} ce satisfacă $\bar{\gamma}_{s'}(u) = \bar{\gamma}_{s'}(v)$, pentru orice sort s' și ecuație $u = v$ de sort s' din C , are loc $\bar{\gamma}_s(t) = \bar{\gamma}_s(t')$, unde s este sortul ecuației $t = t'$.

Acum, regulile de deducție pentru cazul signaturilor ordonate sunt:

1. (reflexivitate)

$$\frac{}{t = t}$$

pentru orice sort s și $t \in T_{(\Sigma, \leq)}(X)_s$;

2. (simetrie)

$$\frac{t = t'}{t' = t}$$

pentru orice sort s și $t, t' \in T_{(\Sigma, \leq)}(X)_s$;

3. (tranzitivitate)

$$\frac{t = t', t' = t''}{t = t''}$$

pentru orice sort s și $t, t', t'' \in T_{(\Sigma, \leq)}(X)_s$;

4. (congruență)

$$\frac{(\forall s')(\forall x \in X_{s'})(f_{s'}(x) = g_{s'}(x))}{f_s(t) = g_s(t)}$$

pentru orice substituții f și g , $s \in S$ și $t \in T_{(\Sigma, \leq)}(X)_s$;

5. (substituție)

$$\frac{t = t' \text{ if } C, (\forall s')(\forall u = v \in C \text{ de sort } s')(f_{s'}(u) = f_{s'}(v))}{f_s(t) = f_s(t')}$$

pentru orice ecuație condițională $t = t' \text{ if } C$ și orice substituție f , unde s este sortul ecuației $t = t'$.

Dată o mulțime E de ecuații condiționate și o ecuație $t = t'$ peste (Σ, \leq) și X , putem introduce concepțele de *consecință sintactică* ($E \vdash t = t'$) și *consecință semantică* ($E \models t = t'$) ca și în cazul clasic. Atunci, obținem următorul rezultat important [75]:

Teorema 8.9.2.3. (Teorema de corectitudine și completitudine)

Fie (Σ, \leq) o signură sortată și X o familie disjunctă de variabile. Dacă (S, \leq) este dirijată prin componente conexe, iar (Σ, \leq) este regulată, atunci, pentru orice multime E de ecuații condiționate și orice ecuație $t = t'$ peste (Σ, \leq) și X , are loc

$$E \models t = t' \Leftrightarrow E \vdash t = t'.$$

Este important de remarcat că Teorema 8.9.2.3 este o extensie a Teoremei 8.8.2.1 atât prin aceea că s-a considerat o ordine parțială pe sorturi cât și prin aceea că s-au considerat ecuații condiționate. Astfel, renunțându-se la ordinea parțială pe sorturi se obține o extensie a Teoremei 8.8.2.1 prin ecuații condiționate, ceea ce poate fi de mare ajutor atunci când se dorește să se realizeze specificări ale tipurilor de date prin intermediul algebrelor multisortate (fără ordine pe sorturi).

Cu ajutorul algebrelor cu sorturi ordonate putem furniza specificații elegante tipurilor de date în care intervin operații parțial definite. De exemplu, următoarea specificație pentru un tip de dată de stive peste numere naturale include operațiile parțiale *pop* și *top* (în cadrul acestei specificații, *nestack* este un sort pentru stive nevide, subsort al sortului *stack*):

```
StackNat = sorts: nat
                  stack
                  nestack
      subsorts: nestack < stack
      opns: zero : nat
             succ : nat → nat
             estack : stack
             push : stack nat → nestack
             pop : nestack → stack
             top : nestack → nat
      vars: x : stack
             y : nat
      eqns: pop(push(x, y)) = x
             top(push(x, y)) = y
```

Sunt algebrele cu sorturi ordonate extensii proprii ale algebrelor multisortate standard? Răspunsul, din punct de vedere al deducției ecuaționale, este negativ.

Dată o pereche (S, \leq) dirijată prin componente conexe și o signură (Σ, \leq) ordonată și regulată, definim o signură Σ' ce conține toate simbolurile funcționale din Σ și, în plus, conține simbolurile funcționale $c_{s,s'}$, pentru orice sorturi s și s' cu $s \leq s'$. Considerăm, de asemenea, și mulțimea J formată din următoarele ecuații:

- $c_{s,s}(x) = x$, pentru orice sort s ;
- $x = y$ if $c_{s,s'}(x) = c_{s,s'}(y)$, pentru orice $s, s' \in S$ cu $s \leq s'$;

- $c_{s',s''}(c_{s,s'}(x)) = c_{s,s''}(x)$, pentru orice $s, s', s'' \in S$ cu $s \leq s' \leq s''$;
- $c_{s,s'}(\sigma_{s_1 \dots s_n, s}(x_1, \dots, x_n)) = \sigma_{s'_1 \dots s'_n, s'}(c_{s_1, s'_1}(x_1), \dots, c_{s_n, s'_n}(x_n))$, pentru orice σ de tip $(s_1 \dots s_n, s)$ și $(s'_1 \dots s'_n, s')$ cu $s_1 \leq s'_1, \dots, s_n \leq s'_n$.

Observăm că aceasta este o mulțime de ecuații condiționate.

Acum, pentru orice (Σ, \leq) -algebră \mathbf{A} se poate defini o Σ' -algebră \mathbf{A}' considerând $A'_s = A_s$, pentru orice sort s , $\sigma^{A'} = \sigma^A$, pentru orice $\sigma \in \Sigma$, și $c_{s,s'}(a) = a$, pentru orice $s \leq s'$ și $a \in A_s$. Algebra \mathbf{A}' satisfacă toate ecuațiile din J , conform construcției. Ca urmare, orice ecuație (condiționată) validă în \mathbf{A} va fi validă și în \mathbf{A}' , și reciproc (a se vedea Teorema 4.4 în [75]).

Legatura dintre algebre cu sorturi ordonate și algebre standard poate fi făcută și mai precisă apelând la teoria categoriilor. Pentru detalii, cititorul este îndrumat către [75].

Algebre cu apartenență. În 1998, José Meseguer a propus *algebrele cu apartenență* [147] ca un nou formalism pentru o mai bună abordare a exceptiilor și erorilor. Ideea de bază constă în definirea unor *suprasorturi* din ale căror domenii să se poată extrage domeniile pe care să fie definite corect operațiile. De exemplu, mulțimea tuturor secvențelor finite de noduri într-un graf reprezintă un domeniu indus de un suprasort ar sortului drumurilor în graf. Exceptiile și erorile vor fi obiecte ale domeniilor definite de suprasorturi, dar nu obiecte ale domeniilor definite de sorturi.

Fie S o mulțime de sorturi și K o mulțime de *suprasorturi*¹⁴ disjunctă de S . O signură (K, S) -sortată este un cuplu (Σ, π) format dintr-o signură K -sortată ce verifică proprietatea $\Sigma_{w,k} \cap \Sigma_{w,k'} = \emptyset$, pentru orice $(w, k), (w, k') \in K^* \times K$ cu $k \neq k'$, și o funcție $\pi : S \rightarrow K$. Intuitiv, funcția π arată care este suprasortul din care se vor extrage sorturile.

O (Σ, π) -algebră, unde (Σ, π) este o signură (K, S) -sortată, este un sistem $\mathbf{A} = (\bar{A}, \Sigma^{\bar{A}}, A)$ format dintr-o Σ -algebră $(\bar{A}, \Sigma^{\bar{A}})$ și o familie S -indexată A de mulțimi astfel încât $A_s \subseteq \bar{A}_{\pi(s)}$, pentru orice $s \in S$.

Considerăm fixată o signură (K, S) -sortată (Σ, π) și o familie K -indexată și disjunctă de variabile X , disjunctă și de Σ . Formulele atomice peste (Σ, π) și X sunt

- ecuații $t = t'$, unde $t, t' \in T_{\Sigma}(X)_k$ sunt doi termi de același suprasort k ,
- sau formule de apartenență $t : s$, unde $s \in S$ și t este un term de sort $\pi(s)$.

Formulele peste (Σ, π) și X sunt construcții de forma

$$\alpha \text{ if } C,$$

unde α este o formulă atomică, iar C este o mulțime finită (posibil vidă) de formule atomice peste (Σ, π) și X . Ceea ce se găsește în stânga lui “if” va fi numit *antetul formulei*, iar ceea ce este în dreapta, *precondiția formulei*.

Vom simplifica adesea terminologia de “formulă (atomică) peste (Σ, π) și X ” la cea de “formulă (atomică)”.

Evident, orice formulă este de una din următoarele două tipuri:

¹⁴Terminologia în limba engleză utilizată în [147] este cea de “kind”.

- (ecuație condiționată)

$$t = t' \text{ if } u_1 = v_1, \dots, u_n = v_n, t_1 : s_1, \dots, t_m : s_m$$

- (apartenență condiționată)

$$t : s \text{ if } u_1 = v_1, \dots, u_n = v_n, t_1 : s_1, \dots, t_m : s_m$$

Mulțimea tuturor acestor formule formează ceea ce se numește *logica ecuațională cu apartenență peste* (Σ, π) și X . Interpretarea formulelor acestei logici în (Σ, π) -algebrelor $\mathbf{A} = (\bar{A}, \Sigma^{\bar{A}}, A)$ se face astfel:

- formula atomică $t = t'$ de suprasort $k \in K$ este *validă* în \mathbf{A} dacă este validă în Σ -algebra $(\bar{A}, \Sigma^{\bar{A}})$;
- formula atomică $t : s$ este *validă* în \mathbf{A} dacă $\bar{\gamma}_{\pi(s)}(t) \in A_s$, pentru orice asignare γ de la X în \bar{A} ;
- formula $\alpha \text{ if } C$ este *validă* în \mathbf{A} dacă antetul ei este validă în \mathbf{A} de îndată ce fiecare formulă atomică din C este validă în \mathbf{A} .

Regulile de deducție ale logicii ecuaționale cu apartenență îmbracă următoarea formă:

1. (reflexivitate)

$$\overline{t = t}$$

pentru orice suprasort k și $t \in T_{\Sigma}(X)_k$;

2. (simetrie)

$$\frac{t = t'}{t' = t}$$

pentru orice suprasort k și $t, t' \in T_{\Sigma}(X)_k$;

3. (tranzitivitate)

$$\frac{t = t', t' = t''}{t = t''}$$

pentru orice suprasort k și $t, t', t'' \in T_{\Sigma}(X)_k$;

4. (congruență)

$$\frac{(\forall k')(\forall x \in X_{k'})(f_{k'}(x) = g_{k'}(x))}{f_k(t) = g_k(t)}$$

pentru orice substituții f și g , $k \in K$ și $t \in T_{\Sigma}(X)_k$;

5. (substituție)

$$\frac{\alpha \text{ if } C, (\forall \beta \in C)(f(\beta))}{f(\alpha)}$$

pentru orice ecuație condițională $\alpha \text{ if } C$ și orice substituție f , unde prin “ $f(w)$ ” înțelegem “ $f_k(t) = f_k(t')$ ”, dacă w este o ecuație $t = t'$ de suprasort k , și “ $f_{\pi(s)}(t) : s$ ”, dacă w este de forma $t : s$;

6. (transfer apartenență)

$$\frac{t : s, t = t'}{t' : s}$$

pentru orice suprasort k , sort s și $t, t' \in T_{\Sigma}(X)_k$.

Dată o mulțime E de formule și o ecuație $t = t'$ peste (Σ, π) și X , putem introduce conceptele de *consecință sintactică* ($E \vdash t = t'$) și *consecință semantică* ($E \models t = t'$) ca și în cazul clasic. Atunci, obținem următorul rezultat important [75, 17]:

Teorema 8.9.2.4. (Teorema de corectitudine și completitudine)

Fie E o mulțime de formule și $t = t'$ o ecuație, ambele peste (Σ, π) și X . Atunci,

$$E \models t = t' \Leftrightarrow E \vdash t = t'.$$

Logica cu apartenență este destul de expresivă. Astfel, se pot exprima

- declarații de subsorturi. O declarație de tipul $s \leq s'$, specifică logicii cu sorturi ordonate, poate fi realizată prin formula “ $x : s' \text{ if } x : s$ ”;
- declarații pe intersecție de sorturi, cum ar fi “ $x : s \text{ if } x : s', x : s''$ ”;
- declarații de operații definite pe sorturi (și nu pe suprasorturi), cum ar fi “ $f(x_1, \dots, x_n) : s \text{ if } x_1 : s_1, \dots, x_n : s_n$ ”.

Să presupunem că dorim să realizăm o specificație ecuațională pentru drumuri într-un graf împreună cu operația de compunere a drumurilor. Evident, această operație este parțial definită. Vom realiza această specificație prin intermediul algebrelor cu apartenență, definind un suprasort *path?* al tuturor secvențelor posibile de noduri ale grafului.

Path	= kinds:	<i>path?</i>
	sorts:	<i>node</i>
		<i>edge</i>
		<i>path</i>
opns:	<i>sr</i>	: <i>path? → path?</i>
	<i>tg</i>	: <i>path? → path?</i>
	<i>conc</i>	: <i>path? path? → path?</i>
vars:	<i>x, y, z</i>	: <i>path?</i>
eqns:	<i>x</i>	: <i>path if x : node</i>

```

 $x : path \text{ if } x : edge$ 
 $sr(x) : node \text{ if } x : path$ 
 $sr(x) = x \text{ if } x : node$ 
 $sr(concat(x, y)) = sr(x) \text{ if } x : edge, y : path, tg(x) = sr(y)$ 
 $tg(x) : node \text{ if } x : path$ 
 $tg(x) = x \text{ if } x : node$ 
 $tg(concat(y, x)) = tg(x) \text{ if } x : edge, y : path, tg(y) = sr(x)$ 
 $concat(x, y) : path \text{ if } x : edge, y : path, tg(x) = sr(y)$ 
 $concat(x, y) = y \text{ if } x : node, y : path, x = sr(y)$ 
 $concat(y, x) = y \text{ if } x : node, y : path, x = tg(y)$ 
 $concat(x, concat(y, z)) = concat(concat(x, y), z)$ 

```

Primele două axiome spun că *node* și *edge* sunt subsorturi ale sortului *path*. Următoarele 3 axiome definesc operația *sr* (ce furnizează nodul inițial al unui drum). Astfel, prima axiomă (din cele 3) specifică că această operație este de tip (*path, node*), iar ultimele două axiome o definesc efectiv. Următoarele 3 axiome definesc operația *tg* (ce furnizează nodul final al unui drum), iar ultimele 4 axiome definesc operația de concatenare a drumurilor, specificând și asociativitatea acesteia.

Bibliografie

- [1] Agrawal, M., Kayal, N., Saxena, N. *PRIMES is in P*, Annals of Mathematics 160(2), 2004, 781-793.
- [2] Armbrust, M., Schmidt, J. *Zum Cayleyschen Darstellungssatz*, Mathematische Annalen 154, 1964, 70-72.
- [3] Ash, R.B. *Information Theory*, Wiley, 1965.
- [4] Bachman, P. *Die analytische Zahlentheorie*, Teubner, Leipzig, 1894.
- [5] Bernays, P. *A System of Axiomatic Set Theory II*, Journal of Symbolic Logic 6, 1941, 1-17.
- [6] Bernays, P. *A System of Axiomatic Set Theory VII*, Journal of Symbolic Logic 19, 1954, 81-96.
- [7] Bernot, G., Gall, P.L., Aiguier, M. *Label Algebras and Exception Handling*, Science of Computer Programming 23, 2-3, 1994, 227-286.
- [8] Berstel, J., Perrin, D. *Theory of Codes*, Academic Press, 1985.
- [9] Birkhoff, G. *On the Combination of Subalgebras*, Proceedings of the Cambridge Philosophical Society 29, 1933, 441-464.
- [10] Birkhoff, G. *On the Structure of Abstract Algebras*, Proceedings of the Cambridge Philosophical Society 31, 1935, 433-454.
- [11] Birkhoff, G. *On Groups of Automorphisms*, Rev. Un. Math. Argentina 11, 1946, 155-157 (în spaniolă).
- [12] Birkhoff, G., Frink, O. *Representations of Lattices by Sets*, Transactions of the American Mathematical Society 64, 1948, 299-316.
- [13] Birkhoff, G. *Lattice Theory*, Colloquium Publications, vol. 25 of the American Mathematical Society, 1995 (a 8-a ediție).
- [14] Boole, G. *Mathematical Analysis of Logic, Being an Essay Toward a Calculus of Deductive Reasoning*, Macmillan, Barclay and Macmillan, London, 1847.

- [15] Boole, G. *An investigation into the Laws of Thought, on Which are Founded the Mathematical Theories of Logic and Probabilities*, Walton and Maberley, London, 1854.
- [16] Borgers, A. *Development of the Notion of Set and of the Axioms of Sets*, Synthese 7, 1949, 374-390.
- [17] Bouhoula, A., Jouannaud, J.-P., Meseguer, J. *Specification and Proof in Membership Equational Logic*, Theoretical Computer Science 236, 2000, 35-132.
- [18] Bourbaki, N. *Théorie des ensembles*, Actualites Scientifiques et Industrielles 846, Herman et Cie, Paris, 1939.
- [19] Bourbaki, N. *Théorie des ensembles*, Ch. 1-2, Paris, 1954 (a două ediție, 1960); Ch. 3, Paris, 1956 (a două ediție, 1963).
- [20] Bourbaki, N. *General Topology*. Addison-Wesley, Reading, Mass., 1968.
- [21] Burmeister, P. *A model theoretic approach to partial algebras*, Akademie Verlag, Berlin, 1986.
- [22] Burris, S., Sankappanavar, H.P. *A Course in Universal Algebra*, Springer-Verlag, 1981.
- [23] Cadiou, J.M., Lévy, J.J. *Mechanizable Proofs About Parallel Processes*, Proceedings of the 14th Annual IEEE Symposium on Switching and Automata Theory, 1973, 34-48.
- [24] Cantor, G. *Ein Beitrag zur Mannigfaltigkeitslehre*, Journal für Mathematik 84, 1878, 242-258 (de asemenea în [28], 119-138).
- [25] Cantor, G. *Über unendliche, lineare Punktmanigfaltigkeiten (V)*, Mathematische Annalen 21, 1883, 545-591.
- [26] Cantor, G. *Beiträge zur Begründung der transfiniten Mengenlehre I*, Mathematische Annalen 46, 1895, 418-512.
- [27] Cantor, G. *Beiträge zur Begründung der transfiniten Mengenlehre (II)*, Mathematische Annalen 49, 1897, 207-246.
- [28] Cantor, G. *Gesammelte Abhandlungen mathematischen und philosophischen Inhalts*, E. Zermelo (ed.), Berlin, 1932.
- [29] Chang, C.C. *Some General Theorems on Direct Products and Their Applications in the Theory of Models*, Nederl. Akad. Wetensch. Proc. ser. A 57, 1954, 592-598.
- [30] Church, A. *A Set of Postulates for the Foundation of Logic* Annals of Mathematics 34(4), 1933, 839-864.

- [31] Church, A. *An Unsolvable Problem of Elementary Number Theory*, American Journal of Mathematics 58, 1936, 345-363.
- [32] Clifford, A.H., Preston, G.B.. *The Algebraic Theory of Semigroups*, Mathematical Surveys 7, vol. 1, American Mathematical Society, Providence, 1961.
- [33] Clifford, A.H., Preston, G.B. *The Algebraic Theory of Semigroups*, Mathematical Surveys 7, vol. 2, American Mathematical Society, Providence, 1967.
- [34] Cocks, C. *An Identity Based Encryption Scheme Based on Quadratic Residues Archived*, Proceedings of the 8th IMA International Conference on Cryptography and Coding, 2001, 360-363.
- [35] Cohen, P. *The Independence of the Continuum Hypothesis 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).
- [36] Cohn, P.M. *Universal Algebra*, a două ediție, Reidel Publishing Company, 1981.
- [37] Cohn, P.M. *Classic Algebra*, John Wiley & Sons, 2000.
- [38] Csiszár, I., Körner, J. *Information Theory: Coding Theorems for Discrete Memoryless Systems*, Academic Press, 1981.
- [39] Daemen, J., Rijmen, V. *The Design of Rijndael*, Springer-Verlag, 2002.
- [40] Davida, G. *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.
- [41] Davis, A.C. *A Characterization of Complete Lattices*, Pacific Journal of Mathematics 5, 1955, 311-319.
- [42] Dedekind, R. *Was sind und was sollen die Zahlen?*, Braunschweig, 1888 (a 6-a ediție, Braunschweig, 1930).
- [43] Dedekind, R. *Über die von drei Moduln erzeugte Dualgruppe*, Mathematische Annalen 53, 1900, 371-403.
- [44] Dedekind, R. *Gesammelte mathematische Werke I, II, III*, Volume editate de R. Fricke, E. Noether și O. Ore, Braunschweig, 1930-1932.
- [45] Devlin, K. *The Joy of Sets. Fundamentals of Contemporary Set Theory*, Springer-Verlag, a două ediție, 1993.
- [46] Diffie, W., Hellman, M.E. *Multiuser Cryptographic Techniques*, Proceedings of AFIPS National Computer Conference, 1976, 109-112.

- [47] Diffie, W., Hellman, M.E. *New Directions in Cryptography*, IEEE Transactions on Information Theory 6, 1976, 644-654.
- [48] Dubreil, P. *Contribution a la theorie de demi-groupes*, Mem. Acad. Sci. France 2(63), 1941.
- [49] Dubreil-Jacotin, P. *Sur l'immersion d'un semi-groupe dans un groupe*, C.R. Acad. Sci. Paris 225, 1947, 787-788.
- [50] Electronic Frontier Foundation. *Cracking DES. Secrets of Encryption Research, Wiretap Politics & Chip Design*, O'Reilly, 1998.
- [51] ElGamal, T. *A Public Key Cryptosystem and a Digital Signature Based on Discrete Logarithms*, IEEE Transactions on Information Theory 31, 1985, 469-472.
- [52] Federal Register. *Proposed Federal Information Processing Standard for Digital Signature Standard (DSS)*, Federal Register 169, 1991, 42980-42982.
- [53] Fejer, P.A., Simovici,D.A. *Foundations of Computer Science. Volume I: Sets, Relations and Induction*, Springer-Verlag, 1991.
- [54] Federal Information Processing Standard Publication 186-2. *Digital Signature Standard (DSS)*, National Institute of Standards and Technology (NIST), 2000.
- [55] Federal Information Processing Standard Publication 197. *Advanced Encryption Standard*, National Institute of Standards and Technology (NIST), 2001.
- [56] Fraenkel, A. , Journal für die Reine und Angewandte Mathematik (A. L. Crelle), vol. 145, 1914.
- [57] Fraenkel, A. *Zu den Grundlagen der Cantor-Zermeloschen Mengenlehre*, Mathematische Annalen 86, 1922, 230-237.
- [58] Fraenkel, A. *Abstract Set Theory*, a doua ediție, North-Holland, 1961.
- [59] Fraenkel, A., Bar-Hillel, Y. *Foundations of Set Theory*, North-Holland, 1958.
- [60] Fraenkel, A., Bar-Hillel, Y., Levy, A. *Foundations of Set Theory*, a doua ediție, North-Holland, 1984.
- [61] Frege, G. *Die Grundlagen der Aritmetik. Eine logisch-mathematische Untersuchung über den Begriff der Zahl*, Breslau, 1884.
- [62] Frege, G. *Grundgesetze der Aritmetik*, Vol. I, Jena, 1893; Vol. II, Jena, 1903.
- [63] Fuchs, L. *On Subdirect Unions*, Acta Math. Sci. Hungar. 3, 1952, 103-120.
- [64] Gauss, C.F. *Neuer Beweis eines arithmetischen Satzes*, Commentationes Societatis Regiae Scientiarum Gottingensis, vol. XVI, Gottingae, 1808.

- [65] Gauss, C.F., Maser, H. *Untersuchungen Über Höhere Aritmetik*, Chelsea Publishing Company, New York, 1965 (conține traducerea din latină în germană a operei complete a lui Gauss asupra teoriei numerelor).
- [66] Gauss, C.F. *Disquisitiones Arithmeticae*, 1801 (revised English translation by W.C. Waterhouse, Springer-Verlag, 1986).
- [67] Geach, P., Black, M. *Translations from the Philosophical Writings of Gottlob Frege*, Blackwell, Oxford, 1952.
- [68] Gierz, G., Hofmann, K.H., Keimel, K., Lawson, J.D., Mislove, M., Scott, D.S. *Continuous Lattices and Domains*, Encyclopedia of Mathematics and Its Applications, vol. 93, 2003.
- [69] Gilbert, E.N., Moore, E.F. *Variable length binary encodings*, Bell System Tech. J. 38, 1959, 933-967.
- [70] Gödel, K. *The Consistency of the Axiom of Choice and the Generalized Continuum Hypothesis*, Proceedings of the National Academy of Sciences
- [71] Gödel, K. *Consistency Proof for the Generalized Continuum Hypothesis*, Proceedings of the National Academy of Sciences (USA) 25, 1938, 220-224.
- [72] Goguen, J.A., Thatcher, J.W., Wagner, E.G., Wright, J.B. *Initial Algebra Semantics and Continuous Algebras*, Journal of the Association for Computing Machinery, 24(1), 1977, 68-95.
- [73] Goguen, J.A. *Order Sorted Algebra: Exceptions and Error Sorts, Coercion and Overloading Operators*, UCLA, Semantics Theory of Computation Report no. 14, 1978.
- [74] Goguen, J.A., Thatcher, J.W., Wagner, E.G., Wright, J.B. *Initial Algebra Approach to the Specification, Correctness, and Implementation of Abstract Data Types*, Current Trends in Programming Methodology (R.T. Yeh, ed.), vol. IV, Prentice-Hall, 1978, 80-149.
- [75] Goguen, J.A., Meseguer, J. *Order-sorted Algebra I: Equational Deduction for Multiple Inheritance, Overloading, Exceptions, and Partial Operations*, Theoretical Computer Science 105(2), 1992, 217-0-273.
- [76] Grätzer, G., Schmidt, E.T. *Characterizations of Congruence Lattices of Abstract Algebras*, Acta Sci. Math. (Szeged) 24, 1963, 34-59.
- [77] Grätzer, G. *Universal Algebra*, Springer Verlag, 1979 (a doua ediție).
- [78] Green, J.A. *On the Structure of Semigroups*, Ann. Math. 54, 1951, 163-172.
- [79] Hall, F.M. *An Introduction to Abstract Algebra*, Cambridge University Press, 1969.

- [80] Halmos, P.R. *Naive Set Theory*, Springer-Verlag, 1974.
- [81] Hamilton, W.R. *On Quaternions or on a New System of Imaginaries in Algebra*, Phil. Mag. 3rd Ser., 1844, 10-13.
- [82] Hamming, R.W. *Coding and Information Theory*, Prentice-Hall, 1986.
- [83] Hardy, G.H., Wright, E.M. *An Introduction to the Theory of Numbers*, Clarendon Press, Oxford, a 5-a ediție, 1990.
- [84] Harrison, M.A. *Introduction to Formal Language Theory*, Addison-Wesley, 1978.
- [85] Hartogs, F. *Über das Problem der Wohlordnung*, Mathematische Annalen 76, 1915, 438-443.
- [86] Hausdorff, F. *Grundzüge der Mengenlehre*, Leipzig, 1914.
- [87] Hellman, M.E. *The Mathematic sof Public-Key Cryptography*, Scientific American 241, 1979, 146-157.
- [88] Higgins, P.M. *Algebras with a Scheme of Operators*, Mathematische Nachrichten 27, 1963, 115-132.
- [89] Higgins, P.M. *Techniques of Semigroup Theory*, Oxford University Press, 1992.
- [90] Hill, R. *A First Course in Coding Theory*, Clarendon Press, 1993.
- [91] Hilbert, D. *Die Theorie der algebraischen Zahlkörper*, Jahresbericht der Deutschen Mathematiker Vereinigung, Vol. 4, 1897.
- [92] Howie, J.M. *An Introduction to Semigroup Theory*, Academic Press, 1976.
- [93] Hrbacek, K., Jech, T. *Introduction to Set Theory*, Marcel Dekker, 1978.
- [94] Hua, L.K. *Introduction to Number Theory*, Springer-Verlag, Berlin, 1982.
- [95] Huffman, D.A. *A Method for the Construction of Minimum Redundancy Codes*, Proceedings of IRE 40, 1952, 1098-1101.
- [96] Hungerford, Th.W. *Algebra*, a 5-a ediție, Springer-Verlag, 1989.
- [97] Isbell, J.R. *Subobjects, Adequacy, Completeness and Categories of Algebras*, Rozprawy Mat. 36, 1964, 33 pag.
- [98] Jacobi, C.G.J. *Über die Kreisteilung und ihre Anwendung auf die Zahlentheorie*, Bericht Ak. Wiss. Berlin, 1837, 127-136.
- [99] Jacobsthal, E. *Über den Aufbau der transfiniten Arithmetik*, Mathematische Annalen 66, 1909, 145-194.

- [100] Janssen, T.M.V. *Algebraic Translations, Correctness and Algebraic Compiler Construction*, theoretical Computer Science 199, 1998, 25-56.
- [101] Jech, T.J. *About the axiom of choice*, în J. Barwise (ed.): *Handbook of Mathematical Logic (Part B)*, Amsterdam, North-Holland, 1977.
- [102] Jech, T.J. *Set Theory*, Springer-Verlag, 1978 (a două ediție, 1997).
- [103] Kahn, D. *The Codebreakers: The Story of Secret Writing*, Macmillan Publishing Co., 1967.
- [104] Kleinjung, Th. et al. *Factorization of a 768-bit RSA modulus*, Cryptology ePrint Archive, Technical Report 006, 2010.
- [105] Knaster, B. *Un théorème sur les fonctions d'ensembles*, Ann. Soc. Polon. Math. 6, 1928, 133-134.
- [106] Kraft, L.G. *A Device for Quantizing, Grouping, and Coding Amplitude Modulated Pulses*, M.S. Thesis, Electrical Engineering Department, Massachusetts Institute of Technology, 1949.
- [107] Kranakis. *Primality and Cryptography*, Wiley-Teubner, Series on Applicable Theory in Computer Science, 1986.
- [108] Kunen, K. *Set Theory. An Introduction to Independence Proofs*, North Holland, 1980.
- [109] Kuratowski, K. *Sur la notion de l'ordre dans la théorie des ensembles*, Fundamenta Mathematicae 2, 1921, 161-171.
- [110] Kuratowski, K. *Une méthode d'élimination des nombres transfinis des raisonnements mathématiques*, Fundamenta Mathematicae 3, 1922, 76-108.
- [111] Kuratowski, K., Mostowski, A. *Set Theory*, North-Holland, 1968.
- [112] Lagrange, J.L. *Démonstration d'un théorème nouveau concernant les nombres premiers*, Nouveaux Mémoires de l'Académie Royale des Sciences et Belles-Lettres (Berlin), vol. 2, 1771, 125-137.
- [113] Lallement, G. *Semigroups and Combinatorial Applications*, John Wiley & Sons, 1979.
- [114] Landau, E. *Handbuch der Lehre von der Verteilung der Primzahlen*, Teubner, Leipzig, 1909.
- [115] Landau, E. *Vorlesungen über Zahlentheorie*, Hirzel, Leipzig, 1927.
- [116] Lang, S. *Linear Algebra*, Springer-Verlag, 1987.
- [117] Legendre, A.M. *Essai sur la théorie des nombres*, Paris, Duprat, 1798.

- [118] Lemmermeyer, F. *Reciprocity Laws: from Euler to Eisenstein*. Springer-Verlag, Berlin, 2000.
- [119] Lenstra, A.K. *Memo on RSA Signature Generation in the Presence of Faults*, personal communication, 1996.
- [120] Lewis, C.H., Rosen, B.K. *Recursively Defined Data Types (I)*, Proceedings of the ACM Symposium on Principles of Programming Languages, 1973, 125-138.
- [121] Leśniewski, S. *Grundzüge eines neuen Systems der Grundlagen der Mathematik*, Fundamenta Mathematicae 14, 1929, 1-81.
- [122] Levi, F.W. *On Semigroups*, Bull. Calcutta Math. Soc. 36, 1944, 141-146.
- [123] Levy, A. *The Independence of Certain Consequences of the Axiom of Choice*, Fundamenta Mathematicae 54, 1964, 135-157.
- [124] Levy, A. *Basic Set Theory*. Springer-Verlag, 1979.
- [125] Levi, B. *Intorno alla teoria degli aggregati*, Royale Istituto Lombardo di Science e Lettere, Rendiconti 2, 1902, 863-868.
- [126] Lidl, R., Niederreiter, H. *Introduction to Finite Fields and Their Applications*, Cambridge University Press, 1986.
- [127] Van Lindt. *Introduction to Coding Theory*, Springer-Verlag, 1982.
- [128] Ljapin, E.S. *Semigroups*, Nauka, Moscow, 1960 (în rusește).
- [129] Loeckx, J., Sieber, K. *The Foundations of Program Verification*, John Wiley and Sons, 1984 (a două ediție, 1987).
- [130] Loeckx, J., Ehrich, H.-D., Wolf, M. *Specification of Abstract Data Types*, Wiley & Teubner, 1996.
- [131] MacWilliams, F.J., Sloane, N.J.A. *The Theory of Error-Correcting Codes*, North-Holland, 1977.
- [132] Manna, Z. *Mathematical Theory of Computation*, McGraw-Hill, 1974.
- [133] Merkle, R.C. *Secure Communication Over Insecure Channels*, Communications of the ACM 4, 1978, 294-299.
- [134] Merkle, R.C., Hellman, M. *Hiding Information and Signatures in Trapdoor Knapsacks*, IEEE Transactions on Information Theory 5, 1978, 525-530.
- [135] Merkle, R.C. *Secrecy, Authentication, and Public Key Systems*, Ph.D. dissertation, Stanford University, 1979.
- [136] Markowski, G. *Categories of Chain-Complete Posets*, IBM Technical Report RC 5100, T.J. Watson Research Center, Yorktown Heights, 1974.

- [137] Markowski, G. *Chain-Complete Posets and Directed Sets with Applications*, Algebra Universalis 6, 1976, 53-68.
- [138] Markowski, G. *Bases for Chain-Complete Posets*, IBM Journal of Research Development, 1976, 138-147.
- [139] McCarthy, J. *A Basis for a Mathematical Theory of Computation*, in Computer Programming and Formal Systems (P. Braffort and D. Hirschberg, eds.), North-Holland, 1963, 33-70.
- [140] McEliece, R.J. *The Theory of Information and Coding*, Cambridge University Press, 2002.
- [141] McWilliams, F., Sloane, J. *The Theory of Error Correcting Codes*, North-Holland, 1977.
- [142] Meinke, K., Tucker, J.V. *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.
- [143] McEliece, R. *The Theory of Information and Coding*, Addison-Wesley, 1977.
- [144] McMillan, B. *Two Inequalities Implied by Unique Decipherability*, IRE Transactions on Information Theory IT-2, 1956, 114-116.
- [145] McShane, E.J. *Partial Orderings and Moore-Smith Limits*, American Mathematical Monthly 59, 1952, 1-11.
- [146] McShane, E.J. *Order-Preserving Maps and Integration Processes*, Annals of Mathematical Studies 31, Princeton, 1953.
- [147] Meseguer, J. *Membership algebra as a Logical Framework for Equational Specification*, Proc. of the 12th International Workshop on Recent Trends in Algebraic Development Techniques WADT'97, Lecture Notes in Computer Science 1376, 1998, 18-61.
- [148] Mirimanoff, D. *Les antinomies de Russell et de Burali-Forti et le probleme fondamental de la theorie des ensembles*, L'Enseignement Mathématique 19, 1917, 37-52.
- [149] Mitchell, J.C. *Foundations of Programming Languages*, The MIT Press, 1996.
- [150] Moore, E.H., Smith, H.L. *A General Theory of Limits*, American Journal of Mathematics 44, 1922, 102-121.
- [151] Morris, F.L. *Advice on Structuring Compilers and Proving Them correct*, in Proc. of the ACM Symposium on Principles of Programming Languages, Boston, 1973, 144-152.

- [152] Mosses P. *A Constructive Approach to Compiler Correctness*, in Proc. of the Workshop "Semantics-directed Compiler Generation", LNCS 94, 1980, 189-210.
- [153] Mostowski, A. *Über die Unabhängigkeit des Wohlordnungssatzes vom Ordnungsprinzip*, Fundamenta Mathematicae 32, 1939, 201-252.
- [154] Naur, P., (ed.). *Revised Report on the Algorithmic Language ALGOL 60*, Communications of the ACM, Vol. 3(5), 1960, 299-314.
- [155] Neumann, J., von. *Zur Einführung der transfiniten Zahlen*, Acta Litterarum ac Scientiarum Regiae Universitatis Hungaricae Francisco-Josephinae, Sectio Scientiarum Mathematicarum 1, 1923, 199-208.
- [156] Neumann, J., von. *Eine Axiomatisierung der Mengenlehre*, Journal für Mathematik 154, 1925, 219-240 (corrections in Journal für Mathematik 155, 1926, 128).
- [157] Neumann, J., von. *On Regular Rings*, Proceedings of the National Academy of Sciences of the United States of America 22, 1936, 707-713.
- [158] Neumann, J., von. *Die Axiomatisierung der Mengenlehre*, Mathematische Zeitschrift 27, 1928, 669-752.
- [159] Nielson, H.R., Nielson, F., Hankin, Ch. *Principles of Program Analysis*, Springer-Verlag, 1998.
- [160] Nielson, H.R., Nielson, F. *Semantics with Applications: A Formal Introduction*, Wiley Professional Computing, 1992 (ediție revizuită în iulie 1999, disponibilă on-line din pagina autorilor).
- [161] NIST 185. *Digital Signature Standard*, National Institute of Standards and Technology, Federal Information Processing Standards Publication 185, U.S. Department of Commerce, 1994.
- [162] Papadimitriou, C.H. *Computational Complexity*, Addison-Wesley, 1994.
- [163] Peano, G. *Démonstration de l'intégrabilité des équations différentielles ordinaires*, Mathematische Annalen 37, 1890, 182-228.
- [164] Peano, G. *Formulaire de Mathématiques*, Torino, 1895 (a 5-a ediție sub denumirea *Formulario Mathematico*, Torino, 1905-1908).
- [165] Polak, W. *Compiler Specification and Verification*, LNCS 124, 1881.
- [166] Precupanu, A. *Bazele analizei matematice*, Editura Polirom, Iași, 1998.
- [167] Petrich, M. *Introduction to Semigroups*, Merill, Columbus, Ohio, 1973.
- [168] Post, E. *A Variant of a Recursively Unsolvable Problem*, Bulletin of the American Mathematical Society 52, 1946, 264-268.

- [169] Quine, W.V. *Mathematical Logic*, New York, 1940.
- [170] Rees, D. *On Semi-groups*, Proc. Cambridge Phil. Soc. 36, 1940, 387-400.
- [171] Reichel, H. *Initial Computability, Algebraic Specifications, and Partial Algebras*, Oxford University Press, 1987.
- [172] Rieger, L. *A Contribution to Gödel's Axiomatic Set Theory I*, Czechoslovak Mathematical Journal 7 (82), 1957, 323-357.
- [173] Rivest, R.L., Shamir, A., Adleman, L.M. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, Communications of the ACM 2, 1978, 120-126.
- [174] Roman, S. *Coding and Information Theory*, Springer-Verlag, 1992.
- [175] Rosen, B.K. *Program Equivalence and Context-Free Grammars*, in Proceedings of the 13th Annual IEEE Symposium on Switching and Automata Theory, 1972, 7-18 (revised version as Research Report RC-4822, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 1974).
- [176] Rosen, K.H. *Elementary Number Theory and Its Applications*, Addison Wesley Longman (a 4-a ediție), 2000.
- [177] Rosser, J.B. *The n-th Prime is Greater Than $n \ln n$* , Proceedings of London Mathematical Society 2, 1939, 21-44.
- [178] Rosser, J.B., Schoenfeld, L. *Approximate Formulas for Some Functions of Prime Numbers*, Illinois Journal of Mathematics 6, 1962, 64-89.
- [179] Russell, B. *The Principles of Mathematics*, London, 1903 (a doua ediție, London, 1937).
- [180] Salomaa, A. *Jewels of Formal Language Theory*, Computer Science Press, 1981.
- [181] Salomon, D. *Data Compression. The Complete Reference*, Springer-Verlag (a treia ediție), 1998.
- [182] Sardinas, A.A., Patterson, P.W. *A Necessary and Sufficient Condition for the Unique Decomposition of Coded Messages*, IRE Internat. Conv. Rec. 8, 1953, 104-108.
- [183] Sayood, K. *Introduction to Data Compression*, Morgan Kaufmann Publishers (a doua ediție), 2000.
- [184] Schneier, B. *Applied Cryptography*, John Wiley & Sons, 1996.
- [185] Schönfinkel, M. *Über die Bausteine der mathematischen Logik*, Mathematische Annalen, vol. 92, 1924.

- [186] Schröder, E. *Vorlesungen über die Algebra und Logik*, 1890 (1891, 1895, 1905).
- [187] Schützenberger, M.P. *Une théorie algébrique du codage*, Séminaire Dubreil-Pisot, Exposé no. 15, 1955-1956).
- [188] Shanks, D. *Class Number, a Theory of Factorization, and Genera*, Symposium of Pure Mathematics, 1972.
- [189] Shannon, C.E. *A Mathematical theory of Communication*, Bell Syst. Tech. J. 27, no. 3, 1948, 379-423 și 623-656.
- [190] Shannon, C.E. *Communication Theory of Secrecy Systems*, Bell Syst. Tech. J. 28, no. 4, 1949, 656-715.
- [191] Shapiro, H.N. *Introduction to the Theory of Numbers*, John Wiley & Sons, 1983.
- [192] Sierpiński, W. *Elementary Theory of Numbers*, Państwowe Wydawnictwo Naukowe, 1964
- [193] Skolem, T. *Einige Bemerkungen zur axiomatischen Begründung der Mengenlehre*, Wiss. Vorträge gehalten auf dem 5 Kongress der scandinav. Mathematiker in Helsingfors, 1922, 217-232.
- [194] Slepian, D. *Some Further Theory on Group Codes*, Bell System Tech. Journal 39, 1960, 1219-1252.
- [195] Sloane, N.J.A. *Recent Bounds for Codes, Sphere Packing and Related Problems Obtained by Linear Programming and Other Methods*, Contemporary Mathematics 9, 1982, 153-85.
- [196] Ștefănescu, M. *Teoria lui Galois*, Editura Ex Ponto, Constanța, 2002.
- [197] Steinitz, E. *Bedingt konvergente Reihen und konvexe Systeme*, J. Reine Angew. Math., 143, 1913, 128-175.
- [198] Stoy, J.E. *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory*, The MIT Press, 1977.
- [199] Suppes, P. *Axiomatic Set Theory*, Dover, New York, 1972.
- [200] Suschkevitsch, A.K. *Über die endlichen Gruppen ohne das Gesetz der eindeutigen Umkehrbarkeit*, Mathematische Annalen 99, 1928, 30-50.
- [201] Suschkevitsch, A.K. *Theory of Generalized Groups*, Kharkov, 1937 (în rusește).
- [202] Tarski, A. *Sur quelques théorèmes qui équivalent à l'axiome du choix*, Fundamenta Mathematicae 5, 1924, 147-154.
- [203] Tarski, A. *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.

- [204] Thantcher, J.W., Wagner, E.G., Wright, J.B. *More on Advice on Structuring Compilers and Proving Them correct*, Theoretical Computer Science 15(3), 1981, 223-249.
- [205] Thue, A. *Über unendliche Zeichenreihen*, Videnskapsselskapets Skrifter, I. Mat.-naturv. Klasse, Kristiania, 1906, 1-22.
- [206] Thue, A. *Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen*, Videnskapsselskapets Skrifter, I. Mat.-naturv. Klasse, Kristiania, 1912, 1-67.
- [207] Tiplea, F.L. *Introducere în teoria multimilor*, Editura Universității "Al. I. Cuza", Iași, 1998.
- [208] Tiplea, F.L., Mäkinen, E., Enea, C. *SE-Systems, Timing Mechanisms, and Time-Varying Codes*, International Journal of Computer Mathematics 79(9), 2002.
- [209] Tiplea, F.L., Mäkinen, E., Trincă, D., Enea, C. *Characterization Results for Time-Varying Code*, Fundamenta Informaticae 52, 2003, 1-13.
- [210] Tiplea, F.L., Iftene, S., Hrițcu, C., Goriac, I., Gordan, R., Erbiceanu, E. *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.
- [211] Tiplea, F.L., Enea, C. *Abstractions of Data Types*, Acta Informatica 42 (8-9), 2006, 639-671.
- [212] Tiplea, F.L., Iftene, S., Teșeleanu, G., Nica, A.-M. *On the distribution of quadratic residues and non-residues modulo composite integers and applications to cryptography*, Appl. Math. Comput., vol. 372, 2020.
- [213] Tiplea, F.L. *Criptografie* (în pregătire).
- [214] Tukey, J.W. *Convergence and Uniformity in Topology*, Annales of Mathematical Studies 2, Princeton, 1940.
- [215] Vaught, R.L. *Set Theory. An Introduction*, Birkhäuser (a două ediție), 1995.
- [216] Vuillemin, J. *Correct and Optimal Implementations of Recursion in a Simple Programming Language*, Proceedings of the 5th Annual ACM Symposium on Theory of Computing, 1973, 224-239.
- [217] Wagner, E.G. *Universal Algebra for Computer Scientists*, curs disponibil în pagina autorului la adresa <http://www.ii.uib.no/~wagner/>.
- [218] Waring, Ed. *Meditationes Algebraicae*, Cambridge, England, 1770 (în latină).
- [219] Weyl, H. *Raum, Zeit und Materie*, Berlin, 1923.

- [220] Whitehead, A.N. *A Treatise on Universal Algebra*, Cambridge University Press, 1898.
- [221] Whitehead, A.N., Russell, B. *Principia Mathematica I,II,III*, Cambridge, 1910, 1912, 1913.
- [222] Wiener, N. *A Simplification of the Logic of Relations*, Proceedings of the Cambridge Philosophical Society 17, 1914, 387-390.
- [223] Wiener, M. *Cryptanalysis of short RSA secret exponent*, IEEE Transactions on Information Theory 36, 1990, 553-558.
- [224] Wirsing, M., Broy, M. *Abstract Data Types as Lattices of Finitely Generated Models*, Proc. of the 9th International Symposium on Mathematical Foundations of Computer Science, Rydzyna (Poland), Lecture Notes in Computer Science 88, Springer-Verlag, 1980.
- [225] Wussing, H. *Genesis of the Abstract Group Concept*, MIT Press, 1984.
- [226] Wolk, F.S. *Dedekind Completeness and a Fixed-Point Theorem*, Canadian Journal of Mathematics, vol. IX, no. 3, 1957, 400-405.
- [227] Zermelo, E. *Beweis, das jede Menge wohlgeordnet werden kann*, Mathematische Annalen 59, 1904, 514-516.
- [228] Zermelo, E. *Untersuchung über die Grundlagen der Mengenlehre (I)*, Mathematische Annalen 65, 1908, 261-281.
- [229] Zermelo, E. *Über Grenzzahlen und Mengenbereiche*, Fundamenta Mathematicae 16, 1930, 29-47.
- [230] Zermelo, E. *Grundlagen einer allgemeinen Theorie der mathematischen Satzsysteme*, Fundamenta Mathematicae 25, 1935, 136-146.
- [231] Zorn, M. *A Remark on Method in Transfinite Algebra*, Bulletin of the American Mathematical Society 41, 1935, 667-670.
- [232] Yoneda, N. *On the Homology Theory of Modules*, Journal of the Faculty of Sciences of Tokyo I-7, 1954, 193-227.

Lista figurilor

1.1	Reprezentarea operațiilor Booleene cu multimi prin diagrame Venn	33
1.2	Reprezentări grafice ale aceleiași relații binare	51
1.3	Reprezentare simplificată a unei relații reflexive și tranzitive	57
1.4	ρ este mai fină decât θ	61
1.5	Reprezentare schematică a unei funcții	64
1.6	Reprezentare grafică a compunerii de funcții	66
1.7	Descompunerea din Teorema 1.2.3.1	68
1.8	Descompunerea din Teorema 1.2.3.2	69
1.9	Descompunerea din Corolarul 1.2.3.2	70
1.10	Afirmații echivalente Axiomei alegerii	82
1.11	Structuri relaționale	95
1.12	Diagrame Hasse	97
1.13	Intersecție de mpo	105
1.14	Reuniune de mpo	105
1.15	Sumă de mpo	107
1.16	Produs cartezian de mpo	107
1.17	Inf-semilatici complete, sup-semilatici complete și latici complete	112
1.18	Laticile M_3 și N_5	118
1.19	a) Sublaticea $\{v \wedge w, v, u \vee v, \alpha, \beta\}$; b) Sublaticea $\{s, r, x, y, z\}$	120
1.20	Exemplu de aplicare a Teoremei 1.4.3.4	122
1.21	Sublaticea $\{\emptyset, a, b, c, l\}$	123
1.22	Exemplificare a proprietății de modularitate a laticilor	123
3.1	Relațiile lui Green	208
3.2	Teorema lui Levi	212
3.3	a) Ordine lexicografică; b) Ordine lexicografică pe cuvinte de aceeași lungime	215
3.4	Proprietatea din Teorema 3.4.1.1	225
3.5	Proprietatea din Teorema 3.4.1.2	226
3.6	Proprietatea de universalitate pentru (S_1, \cdot) și (S_2, \circ)	227
3.7	Proprietatea de echidivizibilitate	228
3.8	u și v nu au tăieturi comune	234
3.9	Delimitare unică a primul cuvânt cod	242

3.10 Cuvinte cod distințe dar comparabile prin relația \leq_{pref}	242
3.11 Separarea de noi cuvinte cod din x	243
3.12 Completarea lui x până la un cuvânt cod	243
3.13 Reducerea unei surse de informație	258
3.14 Construcția unui cod Huffman	258
3.15 O altă reducere a aceleiași surse de informație	259
3.16 Reprezentare arborescentă a codurilor Huffman	260
3.17 Construcția codului h_{ue}	261
3.18 Reprezentare arborescentă a codului h_{ue}	261
3.19 Funcția $H(p, 1 - p)$	265

4.1 Interacțiunea dintre utilizatorii legali și ilegali ai unui criptosistem	295
4.2 Corespondența literă-cifră	297
4.3 Schema de semnare cu interpunerea unei terțe părți	305

5.1 Relații între clase de inele	320
5.2 Numărul de runde în Rijndael	345
5.3 Transformarea ShiftRows	345
5.4 Transformarea <i>SubBytes</i> aplicată lui $z_{1,2}$	346
5.5 Transformarea <i>MixColumns</i> aplicată coloanei a treia	347

6.1 Transmiterea informației prin canale cu zgomot	370
6.2 Canal BSC	371

7.1 Relații între clase de mpo	401
--	-----

8.1 Homomorfism de signaturi disjuncte	470
8.2 Compunere de homomorfisme de signaturi disjuncte	471
8.3 Compatibilitatea congruențelor cu operațiile algebrei	478
8.4 Construcția mulțimii B_s^ρ	481
8.5 Reprezentare grafică a Teoremei de corespondență	483
8.6 Proprietatea de homomorfism	487
8.7 Descompunerea homomorfismelor	490
8.8 Reprezentare grafică a primei teoreme de izomorfism	495
8.9 Reprezentare grafică a celei de a doua teoreme de izomorfism	495
8.10 Reprezentare grafică a celei de a treia teoreme de izomorfism	496
8.11 Proprietatea de universalitate a produsului de algebrelle	498
8.12 Proprietatea de proiectivitate a algebrelor de termi	510
8.13 Proprietatea din Definiția 8.7.2.1(1)	512
8.14 Ilustrare grafică a demonstrației Propoziției 8.7.2.1	513
8.15 Proprietatea de universalitate pentru \mathbf{A}_1 și \mathbf{A}_2	514
8.16 Demonstrația Propoziției 8.7.2.3	514
8.17 Demonstrația Lemei 8.7.3.1	517
8.18 Sintaxă concretă, sintaxă abstractă, semantică	536

8.19 a) Arborele asociat regulii $N_0 \rightarrow aN_1N_2$; b) Arbore asociat derivării $N_0 \Rightarrow_G aN_1N_2 \Rightarrow_G aN_0N_1N_2$	537
8.20 Arbore de derivare	539
8.21 Corectitudinea traducerii	545
8.22 Mașină cu stivă	546
8.23 Procesul de realizare a unui produs software	553
8.24 Realizarea unui produs software cu verificare	554
8.25 Izomorfismul dintre $\mathbf{T}_{\Sigma}/=_E$ și \mathbf{A}	561

Index

- λ-notație, 428
- λ-term, 428
- închidere
 - a unei clase de algebrelă la homomorfisme, 515
 - a unei clase de algebrelă la izomorfisme, 515
 - a unei clase de algebrelă la produse directe, 515
 - a unei clase de algebrelă la subalgebrelă, 515
 - a unei mulțimi, 85
 - la echivalență, 87
 - reflexivă, 87
 - reflexivă și tranzitivă, 88
 - reflexivă și tranzitivă a unei relații, 57
 - simetrică, 87
 - tranzitivă, 87
 - tranzitivă a unei relații, 57
- înfășurătoare de translație, 217
- înmulțire
 - cu scalari, 352
- A doua teoremă de izomorfism pentru algebrelă, 494
- A treia teoremă de izomorfism pentru algebrelă, 495
- adunare
 - a vectorilor, 352
- afirmație duală, 100
- alfabet, 210
 - al unei surse de informație, 253
- ordonat, 213
- algebră, 125
 - a adâncimii arborilor de derivare, 538
 - a arborilor de derivare, 538
 - a cuvintelor, 535
 - a derivărilor extrem stângi, 535
 - a programelor structurate, 542
 - a termilor, 507
 - a termilor de bază, 507
 - a transformărilor de stare, 544
- Booleană, 134
 - Booleană a mulțimii părților, 134
 - Booleană a mulțimii valorilor de aderăvară, 134
- cât/factor, 133, 479
- ciclică, 128
- cu apartenență, 569
- cu sorturi ordonate, 565
- de ordin ∞ , 129
- de ordin finit, 129
- decompozabilă, 500
- finală, 512
- finit generată, 128
- finită, 125
- generată, 476
- generată de, 128
- idecompozabilă, 500
- inițială, 511
- liberă, 511
- minimală, 477
- multisortată, 472
- propriu decompozabilă, 502
- propriu idecompozabilă, 502
- simplă, 483
- tipică pentru o specificație, 558

trivială, 125, 473
 unisortată, 472
 universală, 125, 471
 vidă, 473
 algebre
 de același tip, 126
 algoritm
 de determinare a unei soluții pentru ecuații Diophantine liniare, 170
 de determinare al inversului multiplicativ modular, 175
 de semnare, 306
 de verificare a unei semnături, 306
 algoritmul
 baby-step giant-step, 291
 extins al lui Euclid, 169
 lui Euclid, 167, 168
 Sardinas-Patterson, 246
 analiză
 a dependențelor funcționale, 450
 a programelor, 449, 450
 a propagării constantelor, 450
 de semn, 450
 dinamică, 450
 statică, 450
 antilanț, 97
 apartine, 16
 aplicație
 liniară, 361
 arbore
 de derivare, 537
 local ordonat, 537
 aritate, 71, 467
 asignare, 431, 508
 atac
 adaptiv de plaintext ales, 296
 brute-force, 297
 de criptotext, 296
 de criptotext ales, 296
 de exponent secret mic, 303
 de plaintext ales, 296
 de plaintext cunoscut, 296
 prin căutare exhaustivă a cheilor (EKS), 297
 prin impersonificare, 307
 atacul
 lui Davida, 302
 lui Lenstra, 302
 lui Wiener, 303
 atom
 al unei algebrelor Booleene, 136
 atribuire, 431, 508
 automorfism
 de algebrelor, 131, 487
 de spații vectoriale, 361
 de structuri, 95
 axiomă
 a unei gramatici, 533
 axioma
 împerecherii, 24
 înlăcurii, 30, 146
 abstractiei, 16
 alegerii, 30, 78
 de existență a mulțimii vide, 22
 extensionalității, 20
 infinitului, 30, 35
 părților, 26
 regularității, 29
 reuniunii, 25
 separării, 22
 bază, 330
 a inducției, 39
 a unui spațiu vectorial, 357
 de numerație, 200
 duală, 368
 finită a unui spațiu vectorial, 357
 infinită a unui spațiu vectorial, 357
 ortogonală a unui spațiu vectorial, 366
 pentru λ-notație, 428
 pentru programe recursive, 435
 pentru programe structurate, 441
 bijecție, 28
 bit, 200
 de paritate, 372
 de verificare a parității, 372
 bloc al unei partiții, 30

c-funcție, 146
 normală, 150
 c-relație, 146
 limitată la dreapta, 146
 limitată la stânga, 146
 cât
 al împărțirii, 158
 canal
 asimetric, 371
 binar, 370
 BSC, 371
 cu zgomot, 370
 fără zgomot, 370
 simetric, 371
 canal de transmisie (comunicație), 369
 caracter
 criptotext, 293
 plaintext, 293
 caracteristică
 a unui inel, 327
 zero, 327
 cardinal, 154
 finit, 156
 infinit, 156
 cel mai
 mare divizor comun, 164
 mare element, 98
 mare minorant, 98
 mic element, 38, 98
 mic majorant, 97
 mic multiplu comun, 165
 cheie
 de criptare, 292, 293
 de semnare, 306
 cifră
 a bazei de numerație, 200
 binară, 200
 cea mai puțin semnificativă, 200
 cea mai semnificativă, 200
 clasă, 19
 a tuturor pridinalilor, 143
 axiomatizabilă, 520
 bine ordonată, 146
 de algebrelor, 511
 de echivalentă modulo, 173
 de echivalentă, 58
 de funcții compatibile, 146
 de resturi modulo, 173
 ecuațională, 520
 netrivială de algebrelor, 511
 proprie bine ordonată, 146
 proprie, 20
 universală, 23
 cod, 237
 binar, 238, 370
 biprefix, 238
 bloc, 238, 370
 corector de erori, 378
 de lungime variabilă, 239
 de tip (n, m, d) , 379
 detector de erori, 377
 dual, 389
 finit, 238
 Huffman, 254
 Huffman adaptiv, 259
 instantaneu, 238
 liniar, 382
 maximal, 241
 perfect, 381
 prefix, 238
 sufix, 238
 codificare
 a unei mulțimi, 240
 a unei surse de informație, 254
 codomeniu
 al unei funcții, 28
 al unei relații, 27
 al unei secvențe, 43
 codul ISBN, 384
 coeficient
 al unui polinom, 332
 combinație liniară, 330, 355
 complement
 al unui element, 134
 al unui spațiu vectorial, 364
 modulo 2 al unui bit, 370
 complementara unei mulțimi, 24
 completitudine, 393

compunere
 de funcții, 65
 concatenare
 a cuvintelor, 211
 configurație
 a unei mașini cu stivă, 550
 confuzie, 558
 congruență
 într-o algebră, 132, 478
 într-un grup, 270
 factor, 502
 principală, 485
 total invariantă, 526
 consecință
 semantică, 520
 sintactică, 523
 constantă, 71
 a unei algebrelor, 471
 construcție inductivă, 91, 466
 contor de program
 al unei mașini cu stivă, 545
 contradicție, 15
 convergentă a unei fracții continue finite
 simple, 171
 coordonată, 358
 corectitudine
 a programelor, 449
 a traducerii, 544
 parțială, 449
 totală, 449
 corespondență
 literă-cifră, 297
 corp, 318
 algebraic peste, 337
 de scalari al unui spațiu vectorial, 352
 necomutativ, 318
 criptanaliză, 295
 criptologie, 295
 criotosistem, 293
 asimetric, 294
 bloc, 294
 cu chei sir, 294
 cu chei publice, 294
 simetric, 294

criptosistemul
 Rijndael, 343
 RSA, 299
 criptotext, 293
 criteriu
 lui Euler, 190
 lui Gauss, 192
 lui Schützenberger, 234
 lui Schützenberger pentru
 coduri, 242
 cuvânt
 cod, 237
 de control, 533
 de control la stânga, 533
 peste un alfabet, 211
 vid (nul), 211
 cuvinte egale, 211
 deducție
 a unei ecuații, 523
 demonstrație
 prin dublă inclusiune, 21
 prin inducție matematică, 39
 deplasare, 297
 derivare, 533
 extrem stângă, 533
 terminală, 533
 derivată
 a unui polinom, 336
 descompunere
 în factori primi, 161
 a unei algebrelor, 501
 a unui număr, 161
 impropriu, 501
 propriu, 501
 diagonală a, 49
 diagramă
 Hasse, 97
 Venn, 33
 diferență
 de mulțimi, 23
 simetrică de mulțimi, 33
 dimensiune
 a unui corp peste un subcorp, 331

dimensiune a unui spațiu vectorial, 360
 distanță
 a unui cod, 377
 Hamming, 375
 distribuție
 a numerelor prime, 162
 Bernoulli, 248
 de lungime a unui cod, 238
 de probabilitate, 248
 de probabilitate pozitivă, 248
 uniformă, 249
 divide, 158, 334
 divizor, 158, 334
 al lui zero, 318
 propriu, 158
 domeniu
 (de definiție) al unei funcții, 28
 al unei relații, 27
 al unei secvențe, 43
 de integritate, 319
 semantic, 532
 echivalentă asimptotică, 163
 echivalentele lui Green, 208
 ecuație
 condițională, 567
 congruențială, 178
 congruențială liniară, 178
 de bază, 519
 de sort s , 519
 recursivă, 435, 436
 validă într-o algebră, 520
 element, 15
 al unei mulțimi, 15, 18
 algebraic peste, 337
 cu simplificare, 206
 cu simplificare la dreapta, 206
 cu simplificare la stânga, 206
 idempotent, 206
 inversabil (la stânga, la dreapta)
 într-un monoid, 205
 inversabil într-un semigrup, 205
 maximal, 38, 97
 minimal, 98
 regulat într-un semigrup, 222
 zero al unui inel, 315
 zero al unui semigrup, 205
 zero la dreapta al unui semigrup, 205
 zero la stânga al unui semigrup, 205
 elemente
 comparabile, 38, 96
 incomparabile, 96
 liniar independente, 330
 emițător (codor, codificator), 369
 endomorfism
 de algebrelor, 131, 487
 de spații vectoriale, 361
 de structuri, 94
 epimorfism
 de algebrelor, 131, 486
 de spații vectoriale, 361
 natural induc, 493
 evaluare, 508
 extensie, 336
 algebraică, 337
 finită, 337
 prin, 337
 simplă, 337
 factor
 drept al unui cuvânt, 213
 stâng al unui cuvânt, 213
 stâng al unui element într-un mono-
 id, 231
 familie
 închisă, 465
 a variabilelor unui term, 509
 cât/factor, 464
 de mpo, 104
 de mulțimi, 23
 de mulțimi disjuncte, 23
 de reprezentanți, 77
 disjunctă de mpo, 104
 indexată, 73
 indexată de mulțimi, 73
 indexată disjunctă, 73
 indexată nevidă, 73
 fiabilitate

(acuratețe) a unui canal, 371
 a principiului CMLD, 376
 forma standard a unei matrice generatoare, 386
 formulă, 18
 atomică, 18
 propozițională, 89
 fracție continuă
 finită, 170
 finită simplă, 171
 frontiera
 unui arbore, 537
 funcție, 28, 62
 1 – 1, 28
 S-sortată, 465
 λ -definibilă, 428
 a lui Euler, 175
 a unei algebrelor, 471
 bijectivă, 28
 Booleană, 63
 caracteristică, 63
 ce păstrează infimumul, 99
 ce păstrează supremumul, 99
 constantă, 63
 continuă, 409
 continuă în sens strict, 409
 continuă strictă, 409
 Curry, 413
 de alegere, 79
 de criptare, 293
 de decriptare, 293
 de evaluare, 508
 de interpretare, 508
 de la ... la ..., 28, 62
 de p-interpretare a programelor, 452
 de p-interpretare a termilor și expresiilor logice, 452
 de punct fix, 424
 de tip a unei signaturi, 468
 definită pe ... cu valori în ..., 28, 62
 extinsă naturală, 108
 identică, 63
 inclusiune, 63
 injectivă, 28

liniară, 361
 lungime, 232
 monotonă, 95
 monotonă strictă, 95
 parțială definită pe ... cu valori în ..., 62
 parțială, 62
 parțială bijectivă, 66
 parțială de la ... la ..., 62
 parțială injectivă, 65
 parțială surjectivă, 65
 pe, 28
 proiecție, 64, 77
 recursivă, 428
 semantică a unui λ -term, 431
 semantică a unui program recursiv, 437
 semantică a unui program structurat, 444
 strict parțială, 62
 surjectivă, 28
 total nedefinită, 63
 totală, 62
 vidă, 28
 funcții
 compatibile, 44
 generator, 128
 generator de chei, 294
 grad
 al polinomului nul, 333
 al unui element, 338
 al unui polinom, 333
 modulo al unui polinom, 178
 gramatică
 independentă de context, 532
 grup, 125, 269
 al automorfismelor unei
 algebrelor, 491
 al permutărilor, 272
 al unităților unui inel, 317
 al unităților unui monoid, 206
 al unui element idempotent, 206
 cât (factor), 278

ciclic, 128, 270, 281
 comutativ (abelian), 126, 270
 generat, 270
 simetric, 272
 grupul ADJ, 462
 homomorfism
 de algebrelor, 130, 486
 de grupuri, 270
 de inele, 322
 de signaturi, 470
 de spații vectoriale, 361
 de structuri, 94
 lungime, 232
 semantic, 532
 ideal
 (stâng, drept) generat de, într-un semigrup, 207
 într-un inel, 323
 într-un semigrup, 207
 drept (la dreapta) într-un semigrup, 207
 drept într-un inel, 323
 maximal, 326
 nul (trivial) într-un inel, 323
 principal (stâng, drept) generat de, într-un semigrup, 207
 principal într-un inel, 323
 propriu într-un inel, 323
 stâng (la stânga) într-un semigrup, 207
 stâng într-un inel, 323
 identitate pe, 49
 imagine
 a unei mulțimi printr-o funcție, 63
 a unei mulțimi printr-o relație, 53
 homomorfă a unei algebrelor, 488
 homomorfă inversă a unei algebrelor, 488
 inversă a unei mulțimi printr-o funcție, 70
 inversă a unei mulțimi printr-o relație, 53
 impersonificare, 307
 index
 al unui element într-un grup, 290
 al unui element într-un semigrup, 220
 indicator de cod, 249
 inegalitatea
 Kraft-McMillan, 253
 lui Gibbs, 263
 lui Hamming, 381
 lui Kraft, 253
 inel, 126, 315
 Boolean, 320
 cât, 325
 comutativ, 126, 315
 comutativ cu unitate, 126
 cu diviziune, 318
 cu unitate, 126, 317
 nenul, 317
 nul, 317
 inf-semilatice, 110
 completă, 111
 infimum, 98
 injecție, 28
 interpretare, 508
 a unei baze pentru λ -notație, 430
 a unei baze pentru programe structurale, 443
 inițială, 431
 intersecție
 a unei clase, 23
 a unei familii de mulțimi, 23
 a unei familii indexate de mulțimi, 73
 a unei subfamilii de algebrelor, 475
 de mpo, 105
 interval, 94
 invers
 al unui element, 126
 al unui element într-un grup, 270
 al unui element într-un monoid, 205
 al unui element într-un semigrup, 205
 la dreapta într-un grup, 273
 la dreapta al unei funcții, 67
 la dreapta al unui element într-un monoid, 205
 la stânga într-un grup, 273

la stânga al unei funcții, 67
 la stânga al unui element într-un monoid,
 205
 inversă
 a unei funcții, 65
 a unei relații, 53
 ipoteză inductivă, 39
 izomorfism
 de algebrelor, 131, 486
 de spații vectoriale, 361
 de structuri, 95
 lanț, 83
 al unei structuri, 94
 bine ordonat, 394
 latice, 110
 complementată, 138
 completă, 111
 completă a relațiilor de echivalență,
 112
 completă a submulțimilor unei mulțimi, 112
 diamant, 117
 distributivă, 117
 modulară, 124
 pentagon, 117
 vidă, 110
 legea reciprocității pătratice, 193
 Lema
 minsup–majinf, 102
 lema
 de schimb a lui Steinitz, 359
 lui Zorn, 153
 lider
 al unei clase de echivalență, 388
 limbaj, 214
 generat de o gramatică, 533
 independent la concatenare, 215
 limbaj formal, 18
 literă
 a unui alfabet, 210
 logaritm discret
 al unui element într-un grup, 290
 lungime

a unui cod, 238, 370
 a unui cuvânt, 211
 medie a unui homomorfism, 254
 lungime a reprezentării, 200
 măsură
 a unui limbaj, 249
 mașină
 cu stivă, 544
 mai
 mic decât, 38, 143
 mic sau egal cu, 38
 majorant, 97
 strict, 98
 margine
 superioară strictă, 98
 matrice
 de verificare a parității, 389
 mediană, 116
 memorie
 a unei mașini cu stivă, 544
 minorant, 98
 strict, 98
 mod de operare, 294
 cu cheie fixă, 294
 cu cheie variabilă, 294
 model
 al unei ecuații, 520
 modul al unui număr, 157
 monoid, 125, 203
 al endomorfismelor unei algebrelor, 491
 al transformărilor, 215
 ciclic, 128, 218
 conutativ (abelian), 126
 de transformări, 215
 echidivizibil, 228
 liber generat, 224
 regulat, 222
 monomorfism
 de algebrelor, 131, 486
 de spații vectoriale, 361
 morfism
 de signaturi, 470
 de structuri, 94

strict, 95
 mpo
 γ -completă prin submulțimi dirijate,
 399
 completă, 400
 completă în sens Dedekind, 399
 completă pointată, 401
 completă prin lanțuri, 400
 completă prin submulțimi, 394
 completă prin submulțimi dirijate, 397
 completă prin submulțimi filtrate, 397
 D-completă, 399
 duală, 100
 inf-completă prin submulțimi, 393
 plată, 104
 slab completă, 401
 sup-completă prin submulțimi, 393
 mulțime, 15, 18
 S-sortată, 463
 închisă, 85
 închisă într-o algebră, 127, 474
 închisă într-un spațiu vectorial, 354
 a numerelor naturale, 36
 a predecesorilor, 94
 a succesorilor, 94
 bine ordonată, 139
 biprefix, 238
 bloc, 238
 cât/factor, 58
 de axiome a unei clase, 520
 de egalitate a două homomorfisme,
 488
 de funcții compatibile, 44
 de generatori ai unei algebrelor, 128
 de generatori liberi ai unui semigrup
 (monoid), 224
 de indecsi, 73
 de reprezentanți, 30
 de sorturi, 463
 definită inductiv, 90
 dirijată, 83
 filtrată, 83
 finită, 43, 142
 inductivă, 35
 infinită, 43, 142
 liber inductiv definită, 92
 liniar ordonată, 83
 mărginită, 98
 mărginită inferior, 98
 mărginită inferior strict, 98
 mărginită superior, 97
 mărginită superior strict, 98
 majorată, 97
 majorată strict, 98
 maximală de vectori liniar independenți,
 358
 minorată, 98
 minorată strict, 98
 nenumărabilă, 142
 numărabilă, 142
 parțial ordonată, 83
 parțial ordonată strict, 83
 preordonată, 83
 prefix, 238
 quasi-ordonată, 83
 sortată liber inductiv definită, 466
 sufix, 238
 suport a unei algebrelor, 125, 471
 suport a unui sistem relațional, 93
 total ordonată, 83
 total ordonată strict, 83
 tranzitivă, 35
 mulțimi
 disjuncte, 23
 echipotente, 28
 multiplu, 158, 334
 întreg al unui element într-un grup,
 271
 ne-reziduu pătratic, 188
 necunoscută
 (variabilă) a unui polinom, 332
 notație
 exponent, 72
 infix, 72
 nucleu al unei funcții, 68, 493
 număr
 întreg, 49

cardinal, 154
 complex, 49
 compozit, 159
 compus, 159
 natural, 36
 ordinal, 142
 prim, 159
 rațional, 49
 real, 49
 numere
 congruente modulo, 172
 coprime, 159
 prime între ele, 159
 relativ prime, 159
 obiect individual, 18
 operație, 71
 n-ară, 71
 (partială) n-ară, 71
 a unei algebrelor, 471
 asociativă, 72
 Booleană, 33
 comutativă, 72
 constantă a unei algebrelor, 471
 distributivă față de, 72
 distributivă la dreapta față de, 72
 distributivă la stânga față de, 72
 idempotentă, 72
 nulară, 71
 particulară, 71
 opus
 al unui element într-un grup, 271
 al unui polinom, 333
 ordin
 al lui \mathcal{O} , 196
 al lui Ω , 196
 al lui Θ , 196
 al lui o , 196
 al unei algebrelor, 129
 al unui element, 129
 al unui element într-un grup, 270, 282
 al unui grup, 270
 de mărime, 196
 modulo al unui element, 285

ordinal, 142
 finit, 144
 infinit, 144
 initial, 155
 limită, 144
 succesor, 144
 p-stare, 451
 impropriu, 452
 propriu, 452
 paradox, 15
 al lui Russell, 16
 parametru
 de securitate, 307
 partiție a unei mulțimi, 30
 pas inductiv, 39
 pereche ordonată, 24
 perioadă
 a unui element într-un semigrup, 220
 permutare, 66
 plaintext, 293
 polinoame
 congruente, 334
 egale, 332
 polinom, 332
 constant, 333
 minimal al unui element, 338
 monic, 333
 unitate, 333
 zero (nul), 332
 pondere Hamming, 375
 precede, 96
 imediat, 96
 precizie
 multiplă, 200
 simplă, 200
 predecesor
 imediat, 97, 98
 predicat, 64
 S-sortat, 465
 admisibil, 425
 de apartenență, 18
 de egalitate, 18
 de egalitate slabă, 108

de egalitate tare, 108
 ok, 565
 prefix
 al unui cuvânt, 213
 Prima teoremă de izomorfism
 pentru algebrelor, 494
 Principiu
 inducției structurale, 86
 principiu
 bunei ordonări, 152
 de maximalitate al lui Hausdorff, 152
 dualității pentru mpo, 100
 inducție structurală pentru termi, 507
 inducție de punct fix, 426
 inducție finită, 41
 inducție matematică, 38
 inducție structurală pentru mulțimi
 sortate, 466
 inducției transfinite (I), 148
 inducției transfinite (II), 148
 recursie algebrică finită, 508
 substituției, 336
 problema
 axiomatizării, 522
 deducției, 522
 fundamentală în teoria codurilor bloc,
 379
 logaritmului discret, 290
 procedeu de ortogonalizare Gram-Schmidt,
 367
 produs
 (cartezian) de algebrelor, 497
 (direct) al unei familii indexate de
 mulțimi, 73
 cartezian (direct), 26
 cartezian de mpo, 107
 de funcții, 65
 de polinoame, 332
 de relații, 51
 direct de algebrelor, 499
 direct de mpo, 109
 scalar, 366
 scalar nedegenerat, 366
 subdirect de algebrelor, 504

produs neambiguu, 215
 program
 a unei mașini cu stivă, 546
 recursiv, 428, 434, 436
 structurat, 428, 441, 541
 proprietate
 de proiectivitate a algebrelor de termi,
 510
 de universalitate a algebrelor cât, 493
 de universalitate a algebrelor libere, 512
 de universalitate a produsului carte-
 zian de algebrelor, 498
 de universalitate a produsului direct
 de algebrelor, 499
 de universalitate a unui semigrup, 226
 funcțională, 146
 punct fix, 416
 putere
 întreagă a unui element într-un grup,
 270
 quasi-ordine, 83
 rădăcină
 a unui polinom, 336
 de multiplicitate k a unui polinom,
 336
 modulo a unui polinom, 178
 multiplă a unui polinom, 336
 simplă a unui polinom, 336
 rădăcină primitivă, 285
 rafinare, 59
 rată
 a informației, 372
 receptor (decoder, decodificator), 369
 reducere
 modulară, 174
 modulo, 174
 redusă a unei surse de informație, 254
 registru
 al unei mașini cu stivă, 545
 regulă
 a unei gramatici, 533
 de criptare, 293

de descriptare, 293
 de deducție, 522
 de semnare, 306
 de verificare a unei semnături, 306
 reguli
 de calcul a simbolului Jacobi, 194
 de calcul a simbolului Legendre, 193
 relație, 27
 S-sortată, 464
 S-sortată de echivalentă, 464
 n-ară, 57
 antisimetrică, 55
 asimetrică, 55
 binară, 27, 49
 completă, 50
 conexă, 55
 cu tip, 465
 de apartenență, 50
 de congruență modulo, 172
 de derivare, 533
 de divizibilitate, 158
 de echivalentă, 58
 de egalitate, 49
 de inclusiune, 50
 de inclusiune strictă, 50
 de la ... la ..., 27
 de ordine, 83
 de ordine direct lexicografică, 214
 de ordine invers lexicografică, 214
 de ordine parțială, 83
 de ordine parțială strictă, 83
 de ordine totală, 83
 de ordine totală strictă, 38, 83
 de pre-ordine, 83
 dirijată, 55
 filtrată, 56
 ireflexivă, 55
 mai fină, 61
 pe ..., 27
 reflexivă, 55
 simetrică, 55
 ternară, 57
 tranzitivă, 55
 vidă, 27, 49
 relațiile lui Green, 208
 reprezentare, 215
 într-o bază, 200
 fidelă, 215
 subdirectă de algebrelor, 504
 rest
 al împărțirii, 158
 restricție
 a unei relații, 50
 reunire
 a unei familii de mulțimi, 25
 a unei familii indexate de mulțimi, 73
 de mpo, 105
 disjunctă de mpo, 110
 reziduu pătratic, 188
 scădere
 în grup, 271
 scădere
 vectorilor, 352
 scalar, 352
 schemă
 de semnare digitală, 305
 secvență, 43
 binară, 200
 cod, 237
 finită (de lungime ...), 43
 infinită, 43
 transfinită, 149
 vidă, 43
 segment, 94
 inițial, 140
 semantică, 532
 a unei specificații inițiale, 557, 559
 de algebră inițială, 532
 denotațională a unui program recursiv, 438
 denotațională a unui program structurat, 446
 loose, 557
 semigrup, 125, 203
 cyclic, 128, 218
 comutativ (abelian), 126

cu simplificare (la stânga, la dreapta), sintaxă
 206
 cu unitate, 125
 cu zero (la stânga, la dreapta), 205
 de transformări, 215
 invers, 222
 liber generat, 224
 periodic, 220
 regulat, 222
 slab reductiv, 218
 semnătură, 304
 digitală, 305
 semnătura
 DSS, 311
 ElGamal, 307
 serie
 convergentă, 247
 cu termeni pozitivi, 247
 de numere reale, 247
 divergentă, 247
 sferă, 381
 shiftare, 297
 signatură, 467
 S-sortată, 467
 cu sorturi ordonate, 566
 disjunctă, 468
 finită, 469
 multisortată, 467
 ordonată, 566
 ordonată regulată, 566
 unisortată, 467
 simbol
 al unui alfabet, 210
 constantă, 467
 criptotext, 293
 de funcție, 467
 de operație, 467
 funcțional, 467
 Jacobi, 194
 Legendre, 191
 neterminat, 533
 plaintext, 293
 terminal, 533
 sindrom al unui vector, 389

abstractă, 536
 concretă, 536
 sistem, 30
 de criptare, 292, 293
 relațional, 93
 sistemul axiomatic Zermelo-Fraenkel cu Axioma alegerii, 17
 soluție
 modulo a unui polinom, 178
 sort, 463, 467
 spațiu
 al cheilor de criptare, 293
 vectorial, 351
 vectorial n -dimensional, 360
 vectorial dual, 367
 vectorial finit dimensional, 360
 vectorial infinit dimensional, 360
 vectorial ortogonal, 366
 vectorial trivial, 352
 spațiu vectorial
 cât, 366
 Spec-algebră, 557
 specificație, 553
 atomică, 553
 compusă, 554
 constructivă, 554
 inițială, 554, 555
 loose, 553
 stare, 542
 a unui program structurat, 443
 stivă
 a unei mașini cu stivă, 545
 structură
 relațională, 93
 rigidă, 140
 structuri
 izomorfe, 96
 sub-mpo, 94
 completă, 407
 subalgebră, 474
 a unei algebrelor, 127
 generată, 476
 generată de, 128

proprie, 474
 subcorp
 prim al unui corp, 329
 subcuvânt
 al unui cuvânt, 213
 subgrup, 270
 generat, 270
 normal, 277
 submultime, 21
 dirijată a unei structuri, 94
 filtrată a unei structuri, 94
 partial ordonată, 94
 partial ordonată completă, 407
 proprie, 21
 subsignatură, 469
 subsort, 566
 subspațiu vectorial, 355
 generat, 355
 substituție, 510
 substructură, 93
 succede, 96
 imediat, 97
 succesoare a unei multimi, 35
 succesor
 imediat, 38, 97, 98
 sufix
 al unui cuvânt, 213
 sumă
 a unei serii, 247
 de fuziune de mpo, 405
 de mpo, 106
 de polinoame, 332
 de spații vectoriale, 363
 directă de spații vectoriale, 363
 ordonată de mpo, 106
 ordonată disjunctă de mpo, 110
 separată de mpo, 405
 sup-semilatice, 110
 completă, 111
 suport
 al unei algebrelor, 125, 471
 supraîncarcarea operatorilor, 467
 supremum, 98, 144
 surjectie, 28

naturală, 69
 sursă
 de informație, 253
 de informație finită, 254
 de informație redusă, 254
 tabel Cayley, 71
 tabel Slepian, 388
 tehnică
 de atac, 295
 de criptanaliză, 295
 teorema
 împărțirii cu rest, 157
 împărțirii cu rest pentru polinoame, 334
 chineză a resturilor, 182
 coincidentei, 509
 comparației pentru ordini bune, 141
 constantelor, 522
 de coincidență, 434
 de continuitate și monotonie, 409
 de continuitate a funcției semantice
 a λ -termilor, 432
 de corectitudine și completitudine, 571
 de corectitudinea și completitudinea
 logicii ecuaționale, 528
 de corespondență, 482
 de existență și unicitate a corpurilor
 finite, 342
 de existență a algebrelor inițiale, 519
 de existență a algebrelor libere, 518
 de existență și unicitate a corpurilor
 de descompunere, 341
 de homomorfism, 133
 de izomorfism pentru algebrelor (prima), 133
 de izomorfism pentru grupuri (a doua), 281
 de izomorfism pentru grupuri (a treia), 281
 de izomorfism pentru grupuri (prima), 278
 de izomorfism pentru inele (a doua), 327

de izomorfism pentru inele (a treia), 327
 de izomorfism pentru inele (prima), 326
 Dubreil-Jacotin, 231
 fundamentală a aritmeticii, 161
 lui Euler, 177
 lui Fermat, 177
 lui Gauss, 288
 lui Lagrange, 180
 lui Lagrange pentru grupuri, 277
 lui Levi, 212, 230
 lui Park, 425
 lui Shannon pentru canale fără zgomot, 266
 lui Wilson, 190
 numerelor prime, 162
 recursiei, 44
 recursiei – varianta parametrică, 46
 recursiei transfinite, 149
 Sardinas-Patterson, 244
 substituției, 510
 varietății, 530
 term, 506
 de bază, 506
 terminare
 a programelor, 449
 testare
 a programelor, 449
 text
 cifrat, 293
 clar, 293
 criptat, 293
 inițial, 293
 sursă, 293
 tip, 428, 467
 abstract de dată, 559
 al codomeniului, 467
 al domeniului, 467
 de bază, 428
 de ordine, 147
 traducere
 a unui program, 544
 transformări

legate, 217
 transformare pe o mulțime, 215
 translație
 la dreapta, 216
 la stânga, 216
 unitate
 a unui grup, 126, 270
 a unui inel, 317
 a unui monoid, 125, 204
 la dreapta într-un grup, 273
 la stânga într-un grup, 273
 unitate de control
 a unei mașini cu stivă, 545
 univers de discurs, 18
 valoare absolută a unui număr, 157
 variabilă
 de ieșire, 451
 de intrare, 451
 liberă, 454
 liberă într-un λ -term, 433
 locală, 451
 varietate, 530
 vector, 352
 de verificare a parității, 389
 zero (nul), 352
 vectori
 liniar dependenți, 356
 liniar independenți, 356
 ortogonali (perpendiculari), 366
 verificare
 a programelor, 449
 zero
 la dreapta al unui semigrup, 205
 la stânga al unui semigrup, 205
 zero al unui semigrup, 205
 zgomot, 369