# GENEZA CONCEPTULUI DE GÂNDIRE ALGORITMICĂ ȘI NATURA MATEMATICII

DOI: 10.5281/zenodo.4269379
CZU: 510.62+512.54+519.76

Academician **Mitrofan CIOBANU**
E-mail: mmchoban@gmail.com
Doctorandă **Violeta POPOVICI-BUJOR**
E-mail: violetapopovicibujor@gmail.com
Universitatea de Stat din Tiraspol

**GENESIS OF THE CONCEPT OF ALGORITHMIC THINKING AND THE NATURE OF MATHEMATICS**

**Summary**. The article investigates the genesis of the algorithm concept. It is mentioned that the concept of algorithm and algorithmic thinking appeared much earlier than the word *algorithm*. The linguistic and methodological bases of mathematics and, in particular, of algorithmization, the role of the algorithm concept in the educational process and other human activities are analyzed. Algorithmics are a fundamental requirement in solving many types of problems and, in particular, in computer science. Algorithmic thinking means the presence of skills and competencies to apply algorithms to solve problems and to be aware of sequences of algorithms in the logical demonstration of statements. The content of the article is developed in the aspects of the educational process.

***Keywords:*** algorithm, algorithmic thinking, semantics, semiotics, educational process.

**Rezumat.** Articolul abordează geneza conceptului de algoritm. Se menționează că atât conceptul de algoritm, cât și cel de gândire algoritmică au apărut cu mult înainte de cuvântul *algoritm*. Sunt analizate bazele lingvistice și metodologice ale matematicii și, în particular, ale algoritmizării, rolul conceptului de algoritm în procesul educațional și în alte activități umane. Algoritmizarea este o cerință fundamentală în rezolvarea multor tipuri de probleme, în special, în informatică. Gândirea algoritmică înseamnă prezența abilităților și competențelor de a aplica algoritmi la rezolvarea problemelor și de a conștientiza secvențe de algoritmi în demonstrarea logică a afirmațiilor. Conținutul articolului este dezvoltat sub aspectul procesului educațional.

***Cuvinte-cheie:*** algoritm, gândire algoritmică, semantică, semiotică, proces educațional.

*Tot ceea ce suntem este rezultatul a ce am gândit. Mintea este totul. Devenim ceea ce gândim.*
**Buddha**

## INTRODUCERE

Gândirea se definește drept un *proces psihic cognitiv care reflectă în mod abstract și general esența lucrurilor și a relațiilor dintre ele, utilizând limba sau alt sistem de semne ca instrument, și are drept produse noțiuni, judecăți, raționamente* [1]. În procesul de gândire se formează idei despre lucruri, anumite judecăți și raționamente, abilități și competențe, reflecții asupra realității. Există șase operații fundamentale ale gândirii: analiza, sinteza, comparația, abstractizarea, generalizarea, concretizarea. Funcția de bază a gândirii este rezolvarea problemelor care asigură integrarea persoanei în mediu. Există două strategii majore pentru rezolvarea de probleme: strategiile algoritmice și strategiile euristice [2]. Strategiile algoritmice se pretează mai ales rezolvării problemelor bine definite.

Un algoritm, în sens general, este o secvență finită și ordonată de reguli sau proceduri care trebuie respectate în procesul de rezolvare a unor probleme. Există diverse definiții ale conceptului de algoritm. În activitatea umană cu algoritmi în sens general ne întâlnim la orice pas: la confecționarea diferitor obiecte; la realizarea, pe baza unui proiect, printr-un ansamblu de operații de prelucrare și asamblare, a unui sistem tehnic complex; la pregătirea diferitor bucate etc. [3; 4; 5].

În logica matematică și informatică conceptul de algoritm se utilizează într-un sens mai strict și mai restrâns. În celelalte domenii ale matematicii acest termen se folosește în sens general.

A forma o gândire algoritmică înseamnă a dezvolta abilități și competențe de a aplica algoritmi la rezolvarea problemelor, a conștientiza demonstrațiile algoritmice ale unor teoreme și așa operații mentale ca descompunerea (fapt ce permite a vedea părțile componente ale problemei), abstractizarea, generalizarea (identificarea unor probleme similare), conceptualizarea (capacitatea de a avea o mentalitate inovatoare care generează idei și de a forma abstracții). Gândirea algoritmică este importantă din următoarele motive:

- este o abordare structurată pentru a rezolva problemele;
- rezolvarea algoritmică reprezintă constructiv componentele problemei;
- procesul de rezolvare algoritmică utilizează diverse idei și niveluri de abstractizare;
- algoritmul rezolvă simultan o clasă de probleme și constituie o generalizare a rezolvării unei probleme concrete;
- activitatea în multe domenii cere un mod algoritmic de abordare a realității;
- rezolvarea problemelor în activitatea profesională presupune folosirea cunoștințelor dobândite în cadrul studiilor și în experiența de viață;
- cu resurse computaționale se pot rezolva numai problemele ce admit o abordare algoritmică și, prin urmare, gândirea algoritmică este mult mai generală decât cea computațională;
- orice algoritm construit poate fi considerat un pas în crearea altor algoritmi noi;
- învață a lua decizii operative în diverse situații (omul mereu se află la răscruci de drumuri);
- dezvoltă o gândire logică, eficientă și organizată.

Interesul și preocuparea pentru formarea gândirii algoritmice s-a manifestat în toate timpurile. Din antichitate și până astăzi au fost emise diverse ipoteze cu referire la conceptul de gândire algoritmică.

## GENEZA NOȚIUNII DE ALGORITM

Geneza noțiunii de algoritm interferează cu nașterea și evoluția matematicii și limbajelor, cu necesitățile aplicative ale matematicii.

Cuvântul *algoritm* are la origine numele în varianta latină a matematicianului persan al-Khwarizmi (anii 790–840), numele complet Abu Abdullah Muhammad bin Musa al-Khwarizmi. Cele mai importante proprietăți ale unui algoritm sunt: *corectitudinea* (furnizarea unei soluții corecte a problemei date); *caracterul determinist* (executarea repetată a algoritmului duce întotdeauna la aceleași rezultate); *generalitatea* (capacitatea de a rezolva o anumită clasă de probleme). Pentru algoritmii din informatică sunt importante și următoarele proprietăți: *claritatea* (pașii care trebuie parcurși sunt descriși cu exactitate și fără ambiguități); *verificabilitatea* (fiecare pas să poată fi verificat într-un timp rezonabil); *optimalitatea* și *finitudinea* (se termină după un număr minim de pași) [3; 6; 7; 8; 9].

Definiții formale și echivalente ale conceptului de algoritm de calcul au fost date în anii '30–'55 ai secolului al XX-lea în lucrările lui Kurt Friedrich Gödel (1906–1978), Jacques Herbrand (1908–1931), Stephen Cole Kleene (1909–1994), Emil Leon Post (1897–1954), Alonzo Church (1903–1995), Alan Mathison Turing (1912–1954), Norbert Wiener (1894–1964), Andrei A. Markov, junior (1903–1979), Andrei N. Kolmogorov (1903–1987), Alexey A. Lyapunov (1911–1973) [7; 8; 10; 11]. Este destul de explicită formula *Algoritm = logica + control* din lucrarea [11].

Conceptul de algoritm a apărut în legătură cu rezolvarea ecuațiilor pătrate. Marele savant persan al-Khwarizmi a propus o metodă generală de rezolvare a ecuațiilor pătrate. De exemplu, pentru a obține soluțiile ecuației pătrate $5x^2 - 35x + 60 = 0$, al-Khwarizmi propune utilizarea succesivă a următoarelor calcule:

Pasul 1. Calculăm pătratul coeficientului termenului liniar: $35 \cdot 35 = 1225$;

Pasul 2. Calculăm produsul coeficientului termenului pătratic la termenul liber:
$5 \cdot 60 = 300$;

Pasul 3. Calculăm produsul numărului 4 cu numărul obținut la Pasul 2:
$4 \cdot 300 = 1200$;

Pasul 4. Din numărul obținut la pasul întâi scădem numărul de la Pasul 3:
$1225 - 1200 = 25$;

Pasul 5. Calculăm rădăcina pătrată a numărului calculat la Pasul 4:
$\sqrt{25} = 5$;

Pasul 6. Din opusul coeficientului termenului liniar scădem numărul calculat la Pasul 4:
$35 - 5 = 30$;

Pasul 7. La opusul coeficientului termenului liniar adăugăm numărul calculat la Pasul 4:
$35 + 5 = 40$;

Pasul 8. Calculăm produsul numărului 2 la coeficientul termenului pătratic:
$2 \cdot 5 = 10$;

Pasul 9. Calculăm prima rădăcină $x_1 = 30 : 10 = 3$, care este raportul dintre numerele obținute la Pașii 6 și 8;

Pasul 10. Calculăm a doua rădăcină $x_2 = 40 : 10 = 4$, care este raportul dintre numerele obținute la Pașii 7 și 8.

În secolul al XII-lea, cartea lui al-Khwarizmi a fost tradusă în limba latină cu titlul *Algoritmi de numero Indorum*. În ea se expunea sistemul pozițional zecimal de numerație, algoritmii de adunare și înmulțire a numerelor naturale etc.

Scrierea ecuației pătrate la forma generală

$ax^2 + bx + c = 0$ (1)

a apărut cu mult mai târziu. Calculul reprezentat de pașii 1-10 realizează calculul rădăcinilor conform formulelor:

$$x_1 = \frac{-b-\sqrt{b^2-4ac}}{2a}, \quad x_2 = \frac{-b+\sqrt{b^2-4ac}}{2a}. \quad (2)$$

Aceste formule nu puteau fi scrise pe atunci din cauza lipsei unui limbaj matematic adecvat, dar matematicienii intuitiv le cunoșteau.

În antichitate, în Egipt, Babilon și Grecia, ecuațiile (1) se rezolvau grafic cu ajutorul riglei, compasului și al altor instrumente [12]. Matematicienii din Grecia Antică scriau ecuația în felul următor:

$x^2 - px + q^2 = 0,$ (1.1)

$x^2 - px - q^2 = 0,$ (1.2)

$x^2 + px + q^2 = 0,$ (1.3)

$x^2 + px - q^2 = 0,$ (1.4)

unde $p > 0$, $q > 0$. Fiecare dintre aceste patru ecuații se rezolva geometric după un anumit algoritm.

**Cazul I.** Ecuația (1.1): $x^2 - px + q^2 = 0$.

Soluțiile acestei ecuații sunt:

$$\left[\begin{array}{l} x_1 = \frac{p}{2} + \sqrt{\frac{p^2}{4} - q^2}, \\ x_2 = \frac{p}{2} - \sqrt{\frac{p^2}{4} - q^2}. \end{array}\right.$$

Se calculează numai soluțiile reale. Dacă $p < 2q$, atunci ecuația nu are soluții reale. Dacă $p = 2q$, atunci ecuația (2) este echivalentă cu ecuația $(x - q)^2 = 0$ și are două soluții egale $x_1 = x_2 = q$. Fie $p > 2q$. Efectuăm următoarele calcule grafice:

1. $y_1 = \frac{p}{2}$ .

2. Construim triunghiul dreptunghi $A0C$ cu ipotenuza $A0 = \frac{p}{2} = y_1$ și cateta $AC = q$ (figura 1).

3. Construim cercul $\omega$ cu centrul $0$ și raza $r = 0C$ .

4. Construim punctele $D_1$ și $D_2$ de intersecție a dreptei $0A$ cu cercul $\omega$ unde $D_2$ este situat între $0$ și $A$.

5. $x_1 =$ AD$_1$, $x_2 =$AD$_2$.

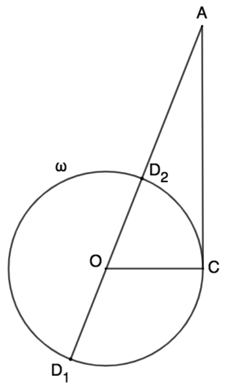


**Figura 1.** Construirea soluției – parte a algoritmului.

**Demonstrație:** Conform Teoremei lui Pitagora, $OC = \sqrt{\frac{p^2}{4} - q}$. Deci, $AD_1 = \frac{p}{2} + \sqrt{\frac{p^2}{4} - q^2} = x_1$ și

$AD_2 = \frac{p}{2} - \sqrt{\frac{p^2}{4} - q^2} = x_2$ .

**Cazul II.** Ecuația (1.2) $x^2 - px - q^2 = 0$:

Soluțiile ei sunt:

$$\left[\begin{array}{l} x_1 = \frac{p}{2} - \sqrt{\frac{p^2}{4} + q^2} \\ x_1 = \frac{p}{2} + \sqrt{\frac{p^2}{4} + q^2} \end{array}\right.$$

Efectuăm următoarele calcule grafice similare cu cazul precedent:

1. $y_1 = \frac{p}{2}$

2. Construim triunghiul dreptunghi $A0C$ cu ipotenuza $A0 = \frac{p}{2} = y_1$ și cateta $AC = q$ (figura 1).

3. Construim cercul $\omega$ cu centrul $0$ și raza r = $0C$ .

4. Construim punctele $D_1$ și $D_2$ de intersecție a dreptei $0A$ cu cercul $\omega$ unde $D_2$ este situat între $0$ și $A$.

5. $|x_1| =$ AD$_1$, $x_2 =$AD$_2$.

**Demonstrație:** Conform Teoremei lui Pitagora

$OA = \sqrt{\frac{p^2}{4} + q^2}$ . Deci, $AD_1 = \left|\frac{p}{2} - \sqrt{\frac{p^2}{4} + q^2}\right| = -x_1$ și

$AD_2 = \frac{p}{2} + \sqrt{\frac{p^2}{4} + q^2} = x_2$ .

**Cazul III.** Ecuația (1.3): $x^2 + px + q^2 = 0$. Efectuând substituția $y = -x$, obținem ecuația $y^2 - py + q^2 = 0$ și reducem la cazul I.

**Cazul IV.** Ecuația (1.4): $x^2 + px - q^2 = 0$. Efectuând substituția $y = -x$, obținem ecuația $y^2 - py - q^2 = 0$ și o reducem la cazul II.

Calculul reprezentat de o formulă pentru fiecare caz concret poate fi caracterizat drept realizarea unui algoritm. Prin urmare, fiecare formulă de calcul poate genera un algoritm. În perioada de până la crearea limbajului matematic se cunoștea algoritmul de calcul, dar nu era scrisă explicit formula de calcul pe care, probabil, matematicienii o sesizau intuitiv. Așa algoritmi de calcul se regăsesc în papirusurile străvechi: Papirusul Rhind din British Museum și Museum în New York City, care este un document din Egiptul Antic scris de scribul Ahmes în jurul anului 1650 î. Hr., copiat de pe alt document mai vechi cu cel puțin două secole și care conține 85 de probleme de matematică; Papirusul din Berlin, scris în perioada 1650–1990 î.Hr.;

Papirusul Matematic din Moscova, scris aproximativ în 1950 î.Hr. Aceste documente conțin rezolvări algoritmice care corespund formulelor de calcul al ariilor triunghiurilor, volumelor cubului și trunchiului de piramidă etc.

Algoritmul lui Euclid, o metodă eficientă de calcul al celui mai mare divizor comun și ciurul lui Eratostene, un algoritm simplu de descriere a tuturor numerelor prime până la un întreg specificat, a fost descoperit încă în Grecia Antică. Algoritmul lui Euclid are multe aplicații teoretice și practice: în criptografie, algebra abstractă, la rezolvarea ecuațiilor diofantice etc.

După cum observăm, conceptul de algoritm a apărut cu mii de ani înainte de definirea noțiunii de algoritm.

Menționăm că din punct de vedere al logicii matematice noțiunea de formulă calculabilă este complicată. E mai simplu de vorbit despre formule numerice calculabile. Formula sau expresia $f(x_1, x_2, ..., x_n)$ se numește numeric calculabilă în cazul în care pentru orice numere date $x_1, x_2, ..., x_n$ putem afla dacă expresia $f(x_1, x_2, ..., x_n)$ are sens, iar pentru orice numere $x_1, x_2, ..., x_n$ cu sens, există un unic algoritm de calcul al valorii numerice $x = f(x_1, x_2, ..., x_n)$.

Vom considera că:

1. expresiile $x_1 + x_2$, $x_1 - x_2$, $x_1 \cdot x_2$, $x_1 : x_2$, $ax$, $\sqrt{x}$, $\sqrt[3]{x}$,, $\sqrt[n]{x}$..., sunt formule calculabile;

2. dacă $f(x_1, x_2, ..., x_n)$, $g(x_1, x_2, ..., x_{m1})$, ... $g_n(x_{1+m_{n-1}}, ..., x_n)$,

unde $1 \leq m_1 < m_2 < ... < m_{n-1} < m_n = m$ sunt formule calculabile, atunci și compoziția lor $f(g_1(x_1, ... , x_{m_1}), ... g_n(x_{1+m_{n-1}}, ..., x_m)$ este o formulă calculabilă.

Observăm că în mulțimea numerelor pozitive, formula $x_1 - x_2$ are sens numai pentru $x_1 \geq x_2$, iar în mulțimea numerelor reale formula $\sqrt{x}$ are sens numai pentru $x \geq 0$.

Mulți algoritmi de calcul reprezintă formule de calcul, dar noțiunea de algoritm de calcul este mai generală. De exemplu, algoritmii de calcul cu ajutorul diferitor abace, apărute în civilizațiile antice cu cel puțin 3 000 ani î.Hr., realizează operațiile de adunare sau scădere, dar acțiunile efectuate nu pot fi descrise prin formule.

## BAZELE LINGVISTICE ALE GÂNDIRII ALGORITMICE

Majoritatea cuvintelor unei limbi naturale se referă la noțiuni, obiecte și locuri în spațiu-timp. Cuvintele semnifică obiectele prin intermediul conceptelor. După cum menționează E. Coșeriu, un concept ca atare nu poate să se actualizeze, nici „să se identifice" cu o reprezentare, căci asta ar echivala cu transformarea lui într-un „obiect", adică în altceva decât este conceptul însăși" [13, p. 301-302].

Cu alte cuvinte, un concept este totdeauna „virtual". Spinoza spunea: „Adevărata definiție a oricărui lucru nu implică nimic și nu exprimă nimic decât natura lucrului definit". Pentru Aristotel [14], numele sunt simboluri ale conceptelor, iar conceptele sunt reprezentări ale lucrurilor, acest proces desfășurându-se independent de limbaj, astfel încât formarea conceptelor ca reprezentări ale formei lucrurilor supuse examinării are loc înainte și independent de achiziția limbajului. Lucrurile și conceptele sunt aceleași pentru toţi. Ceea ce e diferit este simbolul, adică numele. Sensul cuvintelor și evoluția sensurilor cuvintelor se studiază de semantică, o ramură a lingvisticii. Din punct de vedere metodologic relațiile dintre semantică și sintaxă sunt echivalente cu relațiile dintre fond și formă.

Noţiunile empirice sunt noţiunile care au apărut în mod spontan în procesul real al vieţii. Noţiunile empirice din limbajele naturale au, de regulă, un conținut vag. Aceste noțiuni au fost abordate în diferite moduri în diferite timpuri. După cum afirma Gottfried Wilhelm Freiherr von Leibniz (1646–1716), „Dacă s-ar da definiţii edificatoare, disputele ar înceta curând". Asemănător gândea Voltaire (1694–1778): „Dacă vrei să discuţi cu mine, defineşte-ţi termenii" și André Maurois (1885–1967): „Se poate dovedi orice dacă cuvintele de care te servești nu sunt clar definite". În limbile naturale (vorbite) există diverse îmbinări de cuvinte (expresii idiomatice, proverbe, zicale, maxime, aforisme) în care fiecare cuvânt își pierde parțial sau total conținutul individual, iar îmbinarea prezintă un conținut nou. De exemplu: *a spăla putina*, *a căuta acul în carul cu fân*, *graba strică treaba*, *a bate apa-n piuă*, *a vorbi pe șleau*. Aceste îmbinări creează probleme mari traducătorilor.

Noţiunile ştiinţifice au apărut pe parcursul dezvoltării teoriilor ştiinţifice. Cuvintele *drum*, *casă*, *automobil* reprezintă noţiuni empirice, iar cuvintele *triunghi*, *linie dreaptă*, *vocală* reprezintă noţiuni teoretice. E. Coşeriu introduce conceptul de *funcţie semnificativă* care constă în „stabilirea unei conexiuni funcţionale între un semnificat şi un semnificant" [15, p. 55; 16]. Există diferite limbaje și diferite sisteme de semne. Potrivit uneia dintre principalele ipoteze ale semioticii, între diferite „sisteme de semne" există o „asemănare structurală" [17]. Acest principiu stă la baza inteligenței artificiale și, în particular, al traducerilor automate care au înregistrat succese fantastice în ultimii ani. Sistemele compuse și grandioase se bazează pe mai multe limbaje care servesc diferite niveluri ale proceselor informaționale. Studiul interacțiunii limbajelor este o problemă complicată și importantă a multilingvismului. De exemplu, programatorul trebuie să țină cont de interacțiunile dintre limbajul natural, limbajul de programare și limbajul calculatorului (software-ul calculatorului).

Din punct de vedere abstract, există un *limbaj universal* care conține *izomorfic* orice limbaj cunoscut pe moment. Un limbaj universal ar conține o totalitate consistentă de sinonime și îmbinări de cuvinte cu anumite semnificații.

Noţiunile empirice și ştiinţifice, dinamica lor și relaţiile dintre ele au fost studiate și de Lev Vîgotsky [18]. În opinia lui, noţiunile se nasc, se dezvoltă trecând anumite stadii, ajung la maturitate și în dezvoltarea lor mereu îşi schimbă formele. De exemplu, ideile intuitive de dependenţă funcţională și de continuitate au apărut în cele mai îndepărtate timpuri. Se cunoştea că aria pătratului depinde de mărimea laturii, aria cercului depinde de mărimea razei, aria triunghiului depinde de bază și înălțimea către bază etc. Noțiunea de continuitate este asociată cu structura numerelor reale și ordinea punctelor pe dreaptă:

- pentru orice două numere $a$ și $b$, unde $a < b$, există un număr $c$ astfel încât $a < c < b$;
- pentru orice două puncte diferite $A$ și $B$ există un punct $C$, astfel încât $C$ este situat între $A$ și $B$;
- pe orice semidreaptă $l$ cu originea $A$ și orice număr $b$ există un punct $B$, astfel încât distanța dintre punctele $A$ și $B$ este egală cu $b$.

Aceste idei și metoda coordonatelor, care a apărut în *Geometria* lui René Descartes din 1637 și în unele lucrări nepublicate ale lui Pierre de Fermat din anii 1635–1645, au stat la baza apariției conceptului de funcție.

Gottfried Wilhelm Freiherr von Leibniz și Johann Bernoulli introduc noțiunea de funcție care practic coincidea cu conceptul de reprezentare analitică. Ulterior, definiția funcției a fost dată de Leonhard Euler în 1751 și deja într-o formă aproape modernă în lucrările lui Sylvestre François Lacroix în 1806, ale lui Nikolai Lobacevski în 1834, ale lui Johann Peter Gustav Lejeune Dirichlet în 1837 pentru funcții numerice, fiind extinsă ulterior pentru funcții vectoriale. În 1879 Friedrich Ludwig Gottlob Frege introduce funcțiile logice, iar odată cu dezvoltarea teoriei mulțimilor de Georg Cantor apare conceptul contemporan de funcție (aplicație) în lucrările lui Julius Wilhelm Richard Dedekind în 1887 și Giuseppe Peano în 1911. Continuitatea funcției avea inițial un caracter intuitiv și conceptul (ε,δ)-continuitate apare pentru funcții numerice în lucrările lui Bernard Bolzano in 1817 și Augustin-Louis Cauchy în 1921. Bernard Bolzano în 1830 introduce conceptul de continuitate uniformă, dar lucrările lui au fost găsite și publicate abia în 1930. Édouard Goursat (1902–1913) și Camille Jordan (1909) introduc continuitatea în punct și continuitatea unilaterală pentru funcții de variabilă reală. Continuitatea uniformă a funcțiilor reale apare în 1872 în lucrarea lui Eduard Heine care a folosit unele idei din cursul de prelegeri ale lui Peter Gustav Lejeune Dirichlet din 1854. Maurice René Fréchet în 1906 a introdus conceptul de spațiu metric și a extins noțiunea de continuitate pentru aplicații ale spațiilor metrice. Conceptul de continuitate pentru aplicațiile spațiilor topologice a fost extins de Felix Hausdorff în 1914, iar André Weil în 1937 definește spațiile uniforme, continuitatea și continuitatea uniformă pentru aceste spații. În prezent, termenul *continuitate* este folosit în diverse sensuri pentru diferite structuri matematice.

Nivelul unui limbaj artificial depinde de „apropierea” lui de un limbaj natural. În multe cazuri doi oameni pot să aibă reprezentări diferite ale aceluiași cuvânt dintr-un limbaj natural. În cazul limbajelor matematice și, în particular, în cazul limbajelor de programare noțiunile științifice sau sunt primare, sau se definesc. Noțiunile primare se descriu cu ajutorul unui șir finit de afirmații, care în matematică se numesc axiome sau postulate. Aceste axiome nu se schimbă de-a lungul timpului. În limbajele artificiale numărul noțiunilor primare rămâne constant și celelalte noțiuni se definesc, iar afirmațiile noi se demonstrează.

Abordarea sintactico-semantică a definirii noțiunilor permite rezolvarea multor probleme și demonstrarea unor teoreme sub forma unui sistem de procese algoritmice. Această abordare permite stabilirea unor relații constructive între limbă și gândire, între limbă și realitate. În particular, universul algoritmic este o structură lingvistico-logică în care fiecare cuvânt își păstrează totalmente conținutul individual.

H. Gardner [19] afirmă că orice persoană dispune de anumite tipuri de inteligență și, prin urmare, procesul educațional trebuie să ofere posibilități ca fiecare om să atingă obiectivele profesionale și vocaționale care sunt adecvate pentru spectrul lui special de inteligențe. Observăm că inteligența matematică-logică este în concordanță cu inteligența vizual-spațială și inteligența verbal-lingvistică după Gardner. Prin urmare, educația algoritmică se ajustează nu numai la inteligența matematică-logică, ci și la celelalte forme de inteligență [20; 21]. Diverse probleme asociate cu relațiile dintre matematică, informatică și lingvistică au fost examinate în [3; 17; 18; 22; 23; 24; 25; 26; 27].

## METODA ALGORITMICĂ DE REZOLVARE A PROBLEMELOR

Elaborarea unui algoritm presupune câteva etape. În primul rând, se formulează clasa de probleme Π. Această clasă de probleme este omogenă: două probleme din clasă se deosebesc prin structura datelor inițiale. Prin urmare, putem considera că problemele din clasa Π au forma *P*, unde $P = \{p_1, p_2, \ldots, p_n\}$ sunt simbolurile datelor inițiale. Este cunoscută și o mulțime *S*, care prezintă soluțiile posibile pentru fiecare problemă din clasa respectivă.

Soluțiile se află în anumite relații cu datele inițiale. Parametrul simbolic $p_i$ primește valori dintr-o mulțime de date inițiale $P_i$. Pentru datele inițiale concrete *P* soluția poate fi un element din mulțimea *S* care se află în relațiile stabilite cu datele *P*.

Se cunoaște că pentru fiecare caz concret *R* din *S* există cel puțin o problemă concretă *P* pentru care *R* este soluția ei. Pentru rezolvarea problemelor din clasa Π este dată o mulțime de proceduri admisibile. Fiecare procedură are forma (*I*, *R*), unde *I* reprezintă simbolurile unei mulțimi de date concrete inițiale și *R* reprezintă simbolurile unei alte mulțimi de date finale. Mulțimile *I* și *R* satisfac următoarele condiții:

1. Elementele din *R* se află în anumite relații cu cele din *I*.

2. Numărul de elemente din *I* și *R* pot fi diferite.

3. In *I* și *R* numărul de elemente nu este fixat.

4. Valorile concrete ale simbolurilor din mulțimea *I* determină într-un mod stabilit anumite valori pentru *R* sau $R = \varnothing$. Deci pentru un caz concret *I* pot fi obținute mai multe mulțimi concrete *R*.

5. Pereche de forma $(I, \varnothing)$ reprezintă soluția vidă.

A rezolva algoritmic problemele *P* din clasa Π înseamnă:

- a determina condițiile pentru care problemele au soluții.
- a determina soluțiile problemei pentru fiecare caz.
- a elabora un algoritm care permite construirea soluțiilor.

Algoritmul de rezolvare reprezintă un șir finit de forma $A = \{I_j: j = 0, 1, \ldots, m\}$ cu proprietățile:

$P = I_0$ și $I_m = E$ este un element din *S*;

$(I_{j-1}, I_j) \in B$ pentru orice $j = 1, 2, \ldots, m$.

Pentru diferite probleme din clasa de probleme date Π numărul de pași realizați poate fi diferit. În procesul calculelor, s-a observat că unele secvențe de operații sau instrucțiuni trebuie executate de mai multe ori.

Schema generală de rezolvare algoritmică a problemelor Π are forma:

FORMULAREA CLASEI DE PROBLEME

⇓

ANALIZA DATELOR INIȚIALE ȘI FINALE

⇓

PROCESUL DE CĂUTARE A ALGORITMULUI

⇓

DESCRIEREA ALGORITMULUI

⇓

CERCETAREA ȘI VERIFICAREA ALGORITMULUI

Această schemă este similară cu metoda generală de rezolvare a problemelor de construcții geometrice cunoscută în Grecia Antică. Ea poate fi utilizată la demonstrarea unor teoreme și la rezolvarea altor tipuri de probleme. Potrivit acestei metodologii, la fiecare etapă se întreprind următoarele acțiuni:

**0. Formularea clasei de probleme.**

La această etapă se face cunoștință cu sensul problemelor din clasa Π, se studiază anumite cazuri particulare.

**1. Analiza datelor inițiale și finale.**

Se indică cum fiecare caz concret *R* din *S* produce o unică problemă *P* din Π pentru care *R* este o soluție.

**2. Procesul de elaborare a algoritmului.**

La această etapă se presupune că un simbol *R* al soluțiilor *S* este o soluție a problemei $P = \{p_1, p_2, \ldots, p_n\}$. Se caută anumite relații suplimentare care pot să contribuie la elaborarea algoritmului ce rezolvă clasa de probleme. La această etapă este indicată o reamintire a unor teoreme, formule și a anumitor tipuri de probleme anterioare rezolvate ce ne pot ajuta în problema curentă. Construind șiruri de perechi de forma $\{I_j: j = 0, 1, \ldots, k\}$ cu proprietățile $P = I_0$ și $(I_{j-1}, I_j)$ din *B*, problema inițială poate fi redusă la o altă problemă. Unele dintre datele $I_j$ pot fi obținute și prin anumite „transformări admisibile” ale datelor $I_j$. Această etapă se finalizează în cazul când $R_k = E$ pentru un anumit *k* și notăm $k = m$. Cu aceasta algoritmul căutat este stabilit.

**3. Descrierea algoritmului.**

Se descrie detaliat algoritmul elaborat la etapa precedentă: $A = \{I_j: j = 0, 1, \ldots, m\}$.

**4. Cercetarea și verificarea algoritmului.**

La această etapă pentru fiecare $j < m$ se determină condițiile $C_j$ pentru care $R_j \neq \varnothing$. Notăm $C = \{C_j: j = 0, 1, \ldots, m\text{-}1\}$. În condițiile *C*, problemele din Π au soluții. Dacă pentru careva $I_j$ vom avea mai multe realizări $I_j$ cu condițiile $C$, atunci putem beneficia de mai multe soluții care se obțin de diferite realizări ale algoritmului. Dacă există soluții care nu se construiesc cu algoritmul elaborat, atunci pentru acestea se caută alt algoritm, dacă un așa algoritm există.

În procesul elaborării algoritmilor apare necesitatea ca unele secvențe de pași să fie executate de mai multe ori. Repetarea acestor operații poartă numele de ciclu. Pentru fiecare problemă concretă numărul de pași în fiecare ciclu este finit și bine determinat. De exemplu, în algoritmul lui Euclid numărul de pași este diferit pentru diferite cazuri particulare, iar la fiecare pas se realizează teorema împărțirii cu rest pentru două numere naturale cunoscute. În didactica matematicii sunt cunoscute diverse abordări ce țin de formarea abilităților de rezolvare a problemelor prin descoperirea treptată a algoritmilor corespunzători și a metodelor generale de rezolvare. George Polya [28] descrie următoarele principii de rezolvare a problemei: înțelegerea problemei; elaborarea unui plan de rezolvare; realizarea planului; revizuire și extindere. Uneori, principiul înțelegerii problemei este neglijat. Exista diferite metode de rezolvare a unei probleme și în acest caz urmează de ales metoda optimă (mai favorabilă). La această etapă apare necesitatea de a elabora modele și algoritmi de rezolvare a problemei. În final, este necesară analiza algoritmului elaborat, care poate fi utilizat și pentru rezolvarea altor probleme.

Savanții din Grecia Antică considerau analiza problemei o etapă principală în procesul de elaborare a unui algoritm sau plan de rezolvare. Profesorul trebuie să ghideze elevii sau studenții în procesul de elaborare a algoritmului de rezolvare a problemei. Cu acest scop:

1. La baza elaborării algoritmului trebuie să fie acele teoreme și formule care au o legătură directă cu conținutul problemei.

2. În unele cazuri este necesar de revăzut rezolvările unor probleme similare analizate anterior.

3. Dacă folosim un model concret, este necesar ca acest model să prezinte unul dintre cele mai generale cazuri.

4. Algoritmul elaborat trebuie să fie complet și să conțină toți pașii necesari pentru a asigura trecerea de la un pas al rezolvării la următorul pas.

5. Algoritmul elaborat nu trebuie să conțină contradicții. Descrierea fiecărui pas urmează să fie o consecință logică a descrierii pașilor anteriori și o indicație pentru următorul pas.

6. Se recomandă ca numărul pașilor din algoritm să nu fie mare (în funcție de etapa de studii).

7. Algoritmul elaborat trebuie să ofere soluții pentru orice caz particular care are soluții.

Prin urmare, din punct de vedere metodologic:

- trebuie să formulăm problemele în mod explicit și adecvat limbajului matematic;
- trebuie să cunoaștem *limitele* gândirii logice ale elevilor sau studenților;
- trebuie să cunoaștem *limitele* gândirii algoritmice;
- în cazul rezolvării problemelor la calculator, trebuie să cunoaștem performanțele calculatoarelor.

Diverși algoritmi se construiesc în [29] la rezolvarea problemelor și demonstrarea teoremelor din cursul preuniversitar de geometrie.

## DESPRE UNELE MARI DESCOPERIRI ÎN MATEMATICĂ

Este important faptul că elaborarea unor algoritmi a impulsionat crearea unui limbaj matematic corespunzător. Nu orice clasă de probleme se rezolvă algoritmic. Din punct de vedere al informaticii, problemele cu un număr infinit de soluții nu pot fi rezolvate algoritmic. De regulă, rezolvările nealgoritmice conțin secvențe algoritmice. Unele rezolvări pot fi considerate algoritmice sub aspect general, dar nu sunt algoritmice sub aspect computațional care cere strict un număr finit de pași.

**Exemplul 1.** Sunt bine cunoscuți algoritmii pentru operațiile de adunare, scădere, înmulțire și împărțire a numerelor naturale. Acești algoritmi folosesc un anumit sistem de numerație pozițional. În școală se aplică sistemul de numerație zecimal. Calculatoarele, de regulă, utilizează sistemul de numerație binar. Algoritmii respectivi nu pot fi folosiți pentru numerația romană. Numerația romană folosește șapte simboluri grafice – **I, V, X, C, D, M** –, care reprezintă respectiv numerele 1, 5, 10, 100, 500, 1000. Această deosebire se explică prin faptul că numerația pozițională constituie un limbaj pur matematic de prezentare a numerelor, iar numerația romană este o reprezentare simbolică, în limbajul vorbit, a numerelor. Așadar, nivelul unui limbaj artificial depinde de „apropierea” lui de un limbaj natural, însă nu orice „apropiere” de un limbaj natural (vorbit) ridică potențialul limbajului artificial.

Metoda algoritmică și calculul computațional au un aport esențial la rezolvarea unor mari probleme matematice. Vom da unele exemple.

**Exemplul 2.** Teorema celor patru culori a fost prima teoremă demonstrată cu ajutorul calculatorului. Istoria problemei celor patru culori începe la 23 octombrie 1852, când Francis Guthrie a observat că pentru a colora harta Angliei erau suficiente doar patru culori. Această observație el a discutat-o cu fratele său Frederick, student la Universitatea din Londra, informându-l apoi pe profesorul Augustus de Morgan, care a și enunţat ipoteza: sunt suficiente patru culori pentru a colora o hartă ce reprezintă diverse ţări, cu condiția ca oricare două ţări vecine să fie colorate cu culori diferite. Asupra hărţii se impun următoarele condiţii:

- Harta trebuie să se afle pe o suprafață plană din spațiul euclidian;
- Orice țară este un domeniu al suprafeței și orice două puncte ce aparţin unei ţări pot fi unite cu o curbă la care toate celelalte puncte nu aparţin altor ţări (condiția de conexiune al interiorului ţării);
- Frontiera dintre două ţări este reuniunea unui număr finit de curbe. Punctele unde se întâlnesc mai multe ţări nu sunt considerate frontieră.

Cercetările problemei celor patru culori au fost impulsionate de aplicațiile ei directe în algebră, geometrie, topologie, mecanică etc. Percy John Heawood (1861–1955), care și-a dedicat viață cercetării acesteia, în 1890 a descoperit o lacună în rezolvarea problemei de către Alfred Kempe din 1879 și în 1891 a stabilit, folosind ideile demonstrării lui Kempe, că cinci culori sunt suficiente pentru colorarea oricărei hărți plane sau sferice. Ideile lui Kempe au un caracter algoritmic, folosit ulterior de mulți cercetători. În 1922 Philip Franklin demonstrează că patru culori sunt suficiente pentru hărți cu cel mult 25 de țări. Această problemă poate fi formulată și pentru alte tipuri de suprafețe din spațiile euclidiene cu mai multe dimensiuni. În 1910 Heinrich Tietze stabilește că pentru banda lui Mbius sunt suficiente șapte culori. În 1934 P. Franklin conchide că sunt suficiente șase culori pentru sticla lui Klein. Pentru diferite suprafețe mărginite numărul cromatic (numărul minim de culori) a fost stabilit de G. Ringel and J. W. T. Youngs în1968.

În anii 1960–1970, matematicianul german Heinrich Heesch a dezvoltat metode de utilizare a computerului în vederea rezolvării problemei celor patru culori. Împreună cu Ken Durre el a elaborat un test computerizat în acest scop. Din păcate, Heinrich Heesch nu a putut să-și procure timpul necesar la supercomputer pentru a-și continua activitatea. Problema celor patru culori a fost rezolvat**ă** pozitiv de Kenneth Appel și Wolfgang Haken în 1976 [30; 31]. Ei au aplicat ideile principale din lucrările lui Alfred Kempe și Heinrich Heesch pentru a elabora un program special de calcul computerizat. Inițial, demonstrația nu a fost acceptată de toți matematicienii, deoarece nu poate fi verificată manual. Pentru a risipi îndoielile, au fost propuse rezolvări mai simple, folosind idei similare realizate la computer.

**Exemplul 3.** Problemele cu caracter algoritmic au apărut în vremurile străvechi din anumite cauze. Faptul că numerele $x = y = 0$ sunt soluții ale ecuațiilor $x^2 - 2y^2 = 0$, $x^2 - 5y^2 = 0$, $x^2 - 11y^2 = 0$ în mulțimea numerelor întregi a adus la descoperirea numerelor iraționale, la introducerea mărimilor necomensurabile, la ecuații diofantice, la determinarea rezolvabilității ecuațiilor diofantice care constă în a elabora un algoritm capabil să determine soluțiile întregi, rezolvată negativ de Yu. V. Matiyasevich în 1970.

**Exemplul 4.** Remarcabilul savant francez Pierre de Fermat a formulat în anul 1637 pe câmpul cărții *Aritmetica* a lui Diofant următoarea ipoteză, care apoi a fost numită Marea Teoremă a lui Fermat: Ecuația diofantică $x^n + y^n = z^n$ nu are soluții, dacă $n \geq 3$ este număr natural, iar $x$, $y$, $z$ sunt numere întregi nenule. Încă din antichitate era cunoscut că pentru $n \in \{1,2\}$ ecuația $x^n + y^n = z^n$ are o infinitate de soluții în numere naturale. Cazul $n = 4$ a fost cercetat de însuși Fermat. Cazul $n = 3$ a fost demonstrat de Leonhard Euler în anul 1770, se cunoaște că acest caz a fost examinat anterior de Abu Mahmud Hamid ibn Khidr Khojandi (cunoscut și ca al-Hodjandi) încă în secolul al X-lea, dar lucrarea sa a fost pierdută. Cazul $n = 4$ a fost examinat de Johann Peter Gustav Lejeune Dirichlet și Adrien-Marie Legendre în 1825, iar cazul $n = 7$ de Gabriel Lamé în 1839. Ernst Eduard Kummer în anii 1837–1846, studiind Marea Teoremă a lui Fermat, introduce cu ajutorul teoriei idealilor corpurilor comutative noțiunea de *numere prime regulate* și reușește să o demonstreze pentru așa numere. Euler, cercetând cazul $n = 3$, a înaintat în 1769 următoarea ipoteză: pentru orice număr natural $n \geq 2$ nenul, suma puterilor de exponent $n+1$ a oricăror $n$ numere naturale nu poate fi un număr natural la puterea $n+1$.

Mulți matematicieni de seamă au încercat să demonstreze Marea Teoremă a lui Fermat [32]. Pe marginea ei, a fost înregistrat cel mai mare număr de demonstrări incorecte. David Hilbert, în raportul său *Probleme matematice*, prezentat la cel de-al II-lea Congres Internațional al Matematicienilor în anul 1900, a declarat că această teoremă este nesemnificativă, dar căutarea demonstrărilor s-a soldat cu rezultate profunde în teoria numerelor. Cercetări surprinzătoare au fost efectuate în a doua jumătate a secolului al XX-lea. În anul 1955 matematicienii japonezi Goro Shimura și Yutaka Taniyama au presupus că există o legătură între curbele eliptice și formele modulare (conjectura Taniyama-Shimura-Weil), două domenii complet diferite ale matematicii. Această ipoteză a fost redescoperită de André Weil în 1967. J. Lander și T. R. Parkin [33] în 1966, folosind cu succes primul supercalculator CDC 6600, determină că $27^5+84^5+110^5+133^5=144^5$ și prin aceasta stabilesc că ipoteza lui Euler nu este adevărată. În 1988 matematicianul american Noam Elkies oferă un șir de contra exemple cu primul din ele: $2682440^4$ +

$15365639^4 + 18796760^4 = 20615673^4$. La construirea acestor exemple se aplică teoria suprafețelor algebrice cu puncte raționale, calculatorul, calculul numeric și simbolic, programul de calcul MACSYMA [34]. Gerd Faltings în 1883 demonstrează ipoteza lui Mordell despre finitudinea punctelor raționale pe o curbă algebrică de genul $g \geq 2$. De aici rezultă că pentru $n \geq 5$ curba algebrică $x^n+y^n=1$ este de genul $g \geq 2$, iar ecuația $x^n + y^n = z^n$ are un număr finit de soluții $x$, $y$, $z$ reciproc prime. Gerhard Frey a sugerat în 1986 ideea că conjectura Taniyama-Shimura-Weil implică Marea Teoremă a lui Fermat, iar Kenneth Ribet confirmă această implicație în 1986. Deducția sa dovedește că Marea Teoremă a lui Fermat are legături esențiale cu valoroase domenii ale științei și contestă afirmația lui Hilbert că ea este o curiozitate nesemnificativă în matematică. Pe moment, bazându-se pe ideile lui Kummer și folosind calcule sofisticate la computer, matematicienii și informaticienii au putut extinde demonstrarea Marei Teoreme pentru toți exponenții primi până la patru milioane. Aceste rezultate continuau să alimenteze speranța că Marea Teoremă a lui Fermat este falsă. Dar în 1993 Andrew Wiles lansează o nouă demonstrare. Această demonstrare nu era completă și peste un an el, ajutat de Richard Taylor, propune demonstrarea părții esențiale a conjecturei Taniyama-Shimura-Weil care implică Marea Teoremă a lui Fermat [35; 36]. Demonstrarea dată conține rezultate spectaculoase ale teoriei analitice a numerelor și geometriei algebrice.

Istoria rezolvării multor probleme științifice conține elemente de aplicare a conceptului de algoritm însoțite de momente senzaționale.

**Exemplu 5.** Este bine cunoscut algoritmul lui Euclid de calculare a celui mai mare divizor comun a două numere naturale. Acest algoritm permite să determinăm multiplii comuni a două numere naturale $a$ și $b$ diferite de zero:

1. Calculăm cel mai mare divizor comun $c$.
2. Calculăm produsul $d = a \cdot b$.
3. Calculăm cel mai mic multiplu comun $m = d : c$.

$\{nm: n = 1, 2, 3, \ldots\}$ este totalitatea multiplilor comuni.

Pasul 4 nu poate fi realizat la calculator. Asemenea situații pot fi întâlnite la rezolvarea ecuațiilor trigonometrice.

## CONCLUZII

În procesul de dezvoltare a copilului, volumul cunoștințelor se extinde și se condensează. Dezvoltarea gândirii este rezultatul a două procese. Primul proces, de ordin biologic, ține de neurofiziologie – creșterea și dezvoltarea neuronilor, crearea legăturilor dintre ei. Al doilea proces, de ordin sociopsihologic, este determinat de relațiile cu societatea, de percepția altor oameni [25]. Acest proces include în sine studiul limbilor și procesul educațional de care depind comportamentul social, dezvoltarea abilităților, capacităților intelectuale și celor morale. Este evident că orice organizare a comportamentului unui sistem impune anumite restricții asupra sistemului și cu cât acesta este mai complex, iar condițiile privind gradul de libertate sunt mai rigide, cu atât restricțiile sunt mai riguroase. Din această cauză orice proces de prelucrare algoritmică a informației condiționează următoarele:

- să fie corect determinate condițiile inițiale și cele finale;
- descrierea datelor inițiale și a acțiunilor de la executarea fiecărui pas al algoritmului să fie clar și precis specificată, eliminându-se ambiguitățile în interpretare și în proceduri operaționale.

Aplicarea metodei algoritmice în activitatea de rezolvare a problemelor este determinată de modul de descriere a diferitor procese complexe. Rezolvarea problemelor prin orice metode necesită cunoștințe teoretice, capacități de sinteză și control, precum și abilități creative.

Pentru a forma o gândire rațională, constructivă și algoritmică este necesar de a elabora suplimentar diverse materiale didactice ce conțin demonstrarea unor teoreme și probleme cu conținut practic care pot fi rezolvate prin metode algoritmice. Multe procese naturale și multe activități umane se descriu într-o formă algoritmică. Pentru a învăța matematica este necesar de a învăța să rezolvi probleme, dar este necesar și de a reprezenta procesele reale în mod rațional, să dezvolți capacitățile de analiză și sinteză.

Unii consideră că metoda algoritmică nu dezvoltă creativitatea. Aceasta se poate întâmpla în cazul când cu algoritmul elaborat vom rezolva multe probleme din clasa de probleme dată. Orice algoritm este o secvență de algoritmi mai simpli. În acest caz creativitatea constă în arta și iscusința de a crea secvențe din algoritmi mai simpli. De exemplu, reducerea rezolvării unei probleme la o ecuație pătrată și aplicarea la rezolvarea ei a algoritmului lui al-Khwarizmi conține un element esențial de creativitate. Probleme de acest tip sunt multe. Gândire algoritmică înseamnă prezența abilităților și competențelor de a aplica algoritmi la rezolvarea problemelor și de a conștientiza secvențe de algoritmi în demonstrarea unor teoreme. Algoritmizarea este o cerință fundamentală în rezolvarea problemelor cu ajutorul calculatorului. Metoda de rezolvare a unei clase de probleme poate fi de tip algoritmic dacă problemele sunt descrise într-un limbaj matematic sau aproape de cel matematic. Aceasta reprezintă o condiție necesară, dar insuficientă, întrucât nu fiecare operație matematică elementară corespunde unor proceduri elementare când sunt executate într-un limbaj de programare. Orice algoritm cunoscut și aprobat la rezolvarea anumitor clase de probleme poate constitui un pas „elementar" al unui alt algoritm. Acest fapt poate fi realizat cu succes când se aplică diverse soft-uri (de exemplu, aplicațiile software Mathematica, GeoGebra, S-PLUS, MuPAD, Macsyma, MATLAB, Mathcad, Maple) la rezolvarea algoritmică sau parțial algoritmică a claselor de probleme.

**BIBLIOGRAFIE**


1. Negură I. Psihologia memoriei, gândirii și imaginației. Chișinău, 2016.
2. Lupu I., Zastînceanu L. Strategii algoritmice și euristice de rezolvare a problemelor. În: Studia Universitatis. Revista științifică a USM, 2009, nr. 9 (29), p. 248-254.
3. Marcus S. Gândirea algoritmică. București: Editura Tehnică, 1982. 131 p.
4. Vlada M. Gândirea algoritmică – o filosofie modernă a matematicii și informaticii. În: CNIV-2003, Conferința Naționala de Învățământ Virtual: Noi tehnologii de E-Learning, Software Educațional, Ediția I, 24–26 octombrie 2003, Editura Universității din București, 2003, p. 27-34.
5. Vlada M. Abordarea moderna a conceptului de algoritm. În: CNIV-2004, Conferința Naționala de Învățământ Virtual: Noi tehnologii de E-Learning, Software Educațional, Ediția II, 29–31 octombrie 2004, Editura Universității din București, 2004, p. 231-240.
6. Apostol C., Roșca I. Gh., Roșca V., Ghilic-Micu B. Introducere în programare. Teorie și aplicații. Universitatea din București, 1993.
7. Markov A. A. Theory of algorithms, Amer. Matem. Soc. Transl. 1960 (traducere din rusă: Trudî Matem. Inst. imeni Steklova, vol. 42, 1954), p. 3-375.
8. Turing A. On computable numbers, with an application to the Entscheidungs problem. În: Proceedings of the London Mathematics Society, ser. 2, 1937, vol. 42, p. 230-265.
9. Popovici C., Georgescu H., State L. Bazele Informaticii, vol. I, Universitatea din București, 1990.
10. Kleene S. Introduction to Metamathematics. North-Holland, 1952.
11. Kowalski R. Algorithm = Logic + Control. În: Communications of the ACM, 1979, vol. 22 (7), p. 424-436.
12. Adler A. Theorie der geometrischen Konstruktionen. Leipzig: G.J. Göschen, 1906 (traducere în rusă: Leningrad, 1940). 322 p.
13. Coșeriu E. Teoria limbajului și lingvistica generală. București: Editura Enciclopedică, 2004. 362 p.
14. Aristotel. Despre interpretare. București: HUMANITAS, 1998. 393 p.
15. Coșeriu E. Introducere în lingvistică. Editura Echinox, 1995. 143 p.
16. Coșeriu E. El hombre y su lenguaje. În: El hombre y su lenguaje. Madrid: Editorial Gredos, 1977, p. 13-33.
17. Scott D., C. Strachey C. Toward a mathematical semantics for computer languages. Oxford Programming Research Group. Technical Monograph. PRG-6, 1971. p. 43.
18. Vygotsky L. S. Thought and Language. The M.I.T. Press, 1962 (traducere din rusă, Moskva, 1934). 351 p.
19. Gardner H. Inteligențe multiple. Noi orizonturi. București: Sigma, 2006. 320 p.
20. Gașițoi N., Zastînceanu L. Tehnologii educaționale inovaționale la Universitatea de Stat „Alecu Russo" din Bălți: instruirea adaptivă. În: Akademos nr. 2(53), 2019, p. 99-105.
21. Zastînceanu L., Negară C., Moglan D. et al. Organizarea instruirii profesionale adaptive în instituțiile de învățământ superior. Bălți, 2019. p. 126.
22. Bărboianu C. Eficacitatea „irațională" a matematicii: abordarea fundațională a alternativelor teoretice. În: Rev. filos., București, vol. LXII, nr. 1, 2015, p. 58-71.
23. Eco U. O teorie a semioticii. București: Meridiane, 2003. 380 p.
24. Klinkenberg J.-M. Inițiere în semiotica generală. București: Institutul European, 2004. 448 p.
25. Manin Yu. I. Matematika kak Metafora. Moskva: MTsNMO, 2010.424 p.
26. Simpson J., Weiner E. (editori). Dicționar Oxford explicativ ilustrat al limbii engleze (tradusă din Engleză: Oxford English Dictionary, twenty volumes, Clarendon Press, 1989). Editura Litera, 2008. 1008 p.
27. Toma A. Lingvistică și matematică. De la terminologia lexicală la terminologia discursivă. Terminologie, limbaj, discurs. Interdisciplinaritate. Editura Universității din București, 2006 și 2008, 518 p.
28. Polya G. Cum rezolvăm o problemă? București: Editura Științifică și Enciclopedică, 1965. 320 p.
29. Pogorelov A. V. Geometria, manual pentru clasele a 7-11. Chișinău: Lumina, 1991, 367 p.
30. Appel K., Haken W. Every Planar Map is Four Colorable. I. Discharging. În: Illinois Journal of Mathematics, vol. 21 (3), 1977, p. 429-490.
31. Appel K., Haken W., Koch J. Every Planar Map is Four Colorable. II. Reducibility. În: Illinois Journal of Mathematics, vol. 21 (3), 1977, p. 491-567,
32. Edwards H. M. Fermat's Last Theorem: A Genetic Introduction to Algebraic Number Theory. New York: Springer-Verlag, 2000. 418 p.
33. Lander L. J., Parkin T.R. Counterexamples to Euler's conjecture on sums of like powers. În: Bull. Amer. Math. Soc, vol. 72, 1966, p. 1079.
34. Elkies N. D. On $A^4+ B^4+ C^4 = D^4$. În: Mathematics of Computation, vol. 51, nr. 184, 1988, p. 825-835.
35. Wiles A. Modular Elliptic Curves and Fermat's Last Theorem. În: Annals of Mathematics, vol. 141, nr. 3, 1995, p. 443-551.
36. Taylor R., Wiles A. Ring-Theoretic Properties of Certain Hecke Algebras. În: Annals of Mathematics, vol. 141, nr. 3, 1995, p. 553-572.