

Cuprins

Introducere	9
1 Structura grupurilor și cubul Rubik	21
1.1 Grupuri. Grupuri de permutări	22
1.1.1 Subgrupuri. Homomorfisme	30
1.1.2 Generatori. Relații	34
1.1.3 Ordinul unui element. Grupuri ciclice	41
1.1.4 Teorema lui Lagrange	44
1.1.5 Subgrup normal. Grup factor	45
1.1.6 Teoreme de izomorfism	50
1.1.7 Produs direct. Produs semidirect	58
1.1.8 Acțiunea unui grup pe o mulțime. Teorema lui Cayley	77
1.1.9 Teoreme de clasificare. Grupuri de ordin $2, 3, 4, 5, \dots$	87
1.1.10 Grupuri simple	101
1.1.11 Serii de compoziție	115
1.1.12 Grupuri rezolubile	126
1.2 Aplicații - Grupul cubului Rubik	130
1.2.1 Cubul $2 \times 2 \times 2$	130
1.2.2 Subgrupuri interesante	132
1.2.3 Rezolvarea cubului $2 \times 2 \times 2$ cu calculatorul	136
1.2.4 Cubul $3 \times 3 \times 3$	138
1.2.5 Subgrupuri remarcabile și structura grupului	140
1.2.6 Rezolvarea cubului Rubik cu calculatorul	143
1.3 Probleme propuse	146
2 Rezolvarea ecuațiilor polinomiale	151
2.1 Polinoame cu coeficienți într-un corp	151
2.1.1 Discriminantul unui polinom	160
2.2 Extinderi algebrice de corpuri	163
2.2.1 Corpul de descompunere al unui polinom	166

2.2.2	Polinoamele ciclotomice	169
2.2.3	Extinderi normale. Extinderi separabile	170
2.3	Grupul lui Galois	173
2.3.1	Teorema lui Galois	177
2.3.2	Calculul grupului Galois	179
2.3.3	Extinderi ciclice	181
2.3.4	Rezolvabilitatea ecuațiilor prin radicali	183
2.4	Aplicații – exemple în GAP	187
2.5	Probleme propuse	188
3	Topologie pe calculator	191
3.1	Elemente de topologie algebrică	191
3.1.1	Complexe simpliciale	193
3.1.2	Omologie simplicială	195
3.2	Aplicații. Exemple în GAP	198
3.3	Probleme propuse	200
	Bibliografie	201
	Listă de figuri	203
	Listă de tabele	204
	Glosar	206

Introducere



OLUMUL 1 PREMĂRGĂTOR ACESTUIA a avut un capitol introductiv mai deosebit: au fost prezentate direct, fără nici un scrupul și fără nici o pregătire prealabilă, câteva aplicații specifice algebrei computaționale, aplicații care ulterior urmau să fie fundamentate în câte un capitol al volumului respectiv. De ce nu am proceda la fel și de data aceasta? Să presupunem deci, și acum, că am parcurs deja câteva zeci de pagini de teorie și am ajuns la aplicații. Să vedem ce se va putea face cu teoria care se va prezenta în paginile care urmează, și cu realizarea algoritmică a acesteia încorporată în pachete de programe inteligente accesibile gratuit.

1. Să considerăm pentru început o problemă interesantă și dificilă: rezolvarea cubului lui Rubik. Cubul lui Rubik este un joc de inteligență cu totul remarcabil, care și-a câștigat un binemeritat loc simbolic printre invențiile secolului 20, prin urmare cu siguranță el nu mai trebuie prezentat în detaliu (vezi figura de mai jos).



Figura 1: Cubul lui Rubik

Cubul original pare să fie format din $3 \times 3 \times 3 = 27$ cubulețe, colorate în exterior cu 6 culori diferite, în poziția inițială fiecare față având o singură culoare. Fețele cubului însă se pot roti, astfel după numai câteva rotații culorile

cubulețelor se amestecă drastic. Problema este readucerea lor în poziția inițială.

Această poziție inițială se poate desfășura în plan. Obținem o reprezentare care arată cam așa (vezi figura 1).

Culorile fețelor după câteva rotații par să fie amestecate aleator, mai precis par să fie dispuse ca și cum am fi dezamblat cubul și am fi recompus aleator cele $20 (= 26 - 6)$ cubulețe ale sale (cubulețele din centrul fețelor rămân totuși pe loc, iar cubulețul din centrul cubului nu se vede). Această impresie este justificată deoarece aproape orice poziție, care s-ar putea obține în acest mod, se poate obține și prin rotațiile regulate ale fețelor. De

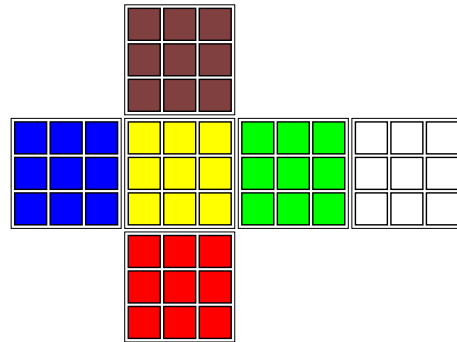


Figura 2: Poziția inițială a cubului, desfășurată în plan.

fapt am putea asambla aleator 7 din cele 8 cubulețe de la colțuri (respectiv 11 din cele 12 cubulețe de la mijlocul muchiilor), doar ultimul ar necesita o potrivire prin alegerea a numai unui din cele trei poziții posibile, rotite între ele cu 120° (respectiv una din cele două poziții posibile, rotite între ele cu 180° pentru ultimul cubuleț de pe muchie), pentru a asigura o configurație de culori, care apoi prin rotații regulate poate fi adusă în poziția inițială standard. Acest fapt arată că într-adevăr, prin rotații ale fețelor culorile se amestecă foarte aproape de maximum de dezordine posibilă. Demonstrația acestei afirmații va apărea oarecum indirect în capitolul următor.

Câte configurații – diferite între ele – de culori are cubul? Răspunsul este următorul:

Fapt. Numărul total al configurațiilor distincte de culori ale cubului este

$$43252003274489856000.$$

Numărul acesta este inimaginabil de mare, este peste 4×10^{19} . Dacă un calculator ar analiza una câte una, un miliard (adică $1000000000 = 10^9$) de configurații pe secundă, ar avea de lucru pentru peste 1000 de ani!

Merită în această situație măcar să încercăm analiza structurii cubului cu ajutorul calculatorului? La prima vedere pare că nu are nici un rost. Și totuși algebra computațională este în stare de performanță remarcabilă de a analiza această structură complicată. Mai mult, are loc următorul:

Fapt. Putem calcula rotațiile necesare de readucere a culorilor în poziția inițială, pentru orice poziție amestecată a culorilor.

Acest calcul este atât de eficient, încât un program de calculator îl poate rula practic în doar fracțiuni de secunde! Precizăm din start că nu este vorba de transpunerea pe calculator a vreunui din algoritmi cunoscuți de rezolvare, cu care se prezintă concurenții la concursurile contra cronometru de rezolvare a cubului!

Să vedem deci cum descriem cubul și rotațiile fețelor sale pentru calculator. Cele 6 cubulețe din centrul fețelor nu își schimbă poziția în urma rotației fețelor, ele constituie practic referința pentru culoarea feței respective. Prin urmare acestea își păstrează și poziția relativă între ele.

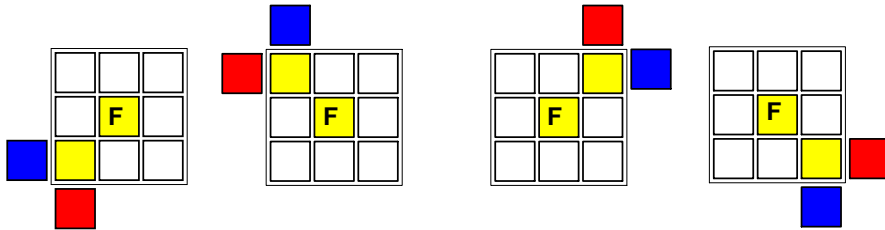


Figura 3: Rotațiile centrului unei fețe

Am spus poziția, nu însă și orientarea. Să menționăm aici o problemă interesantă: dacă am individualiza orientarea acestor 6 pătrățele din centrul fețelor cubului imprimând câte o săgeată pe ele (care ar putea simboliza starea lui de rotație din cele 4 stări de rotații cu câte 90° posibile), aceste pătrățele ar avea pe cub în total $4^6 = 4096$ de configurații diferite. Oare câte din acestea sunt accesibile din poziția inițială prin rotații regulate ale fețelor cubului? Mai explicit, plecând din poziția inițială cu una din cele 4096 de configurații ale pozițiilor săgeților pe pătrățele centrale, amestecând culorile și revenind din nou în poziția inițială pe altă cale, săgețile pot ajunge într-o altă configurație de rotire. Câte din cele 4096 de configurații pot apărea astfel efectiv?

Acest număr este de cel puțin 4. Într-adevăr, pentru oricare rezolvare a unui nivel de bază (o față și pătrățelele adiacente celor patru muchii) există o rezolvare a cubului, i.e. această rezolvare a primului nivel poate fi terminată. Pe de altă parte, în dreptul unui anumit vârf fixat al pătratului din mijlocul feței de bază poate fi adus oricare din cele 4 cubulețe de colț de pe fața respectivă (vezi figura 4 de la pagina 12, în care sunt prezentate 4 configurații fizic diferite. Acestea nu pot fi detectate-vizualizate pe un cub real, decât dacă imprimăm pe aceste pătrățele centrale și câte o săgeată, sau câte o literă!). Acum nivelul acesta de bază poate fi rezolvat plecând de la acest cubuleț de

colț fixat. Autorul nu cunoaște răspunsul la această întrebare. Pe baza unor considerații care vor fi făcute mai târziu în legătură cu structura grupului asociat cubului s-ar putea bănuși că se pierde doar un factor 2, deci numărul acestor configurații ar putea fi chiar 2048.

Să trecem acum la reformularea problemei rezolvării cubului Rubik în termeni matematici, și apoi la algoritmizarea acesteia. Ideea de bază este simplă, și se bazează pe următorul fapt.

Fapt. Dacă identificăm fețele colorate ale cubulețelor în poziția inițială (cu excepția centrelor fețelor cubului mare) cu câte un număr (natural, de la 1 la $48=54-6$), atunci rotațiile fețelor amestecă aceste numere între ele, adică produc câte o permutare a acestor numere. Fiecare poziție amestecată va fi descrisă deci de o permutare a mulțimii de numere $\{1, 2, 3, \dots, 48\}$.

Ordinea în care numerotăm pătratele nu are nici o importanță. Un exemplu posibil de numerotare inițială este arătat în figura de mai jos. Aceasta va corespunde permutării identice – și trebuie memorată.

Literele din mijloc identifică fețele cubului în poziția inițială: F, B, L, R, U, D sunt nume potrivite, fiind inițialele cuvintelor front, back, left, right, up, down. Tot aceste litere le putem utiliza și pentru a denumi o operație elementară de transformare, o rotație cu 90° a fețelor respective, prin convenție în sensul acelor ceasornicului. Dacă efectuăm pe rând aceste rotații din poziția inițială și urmărim pozițiile în care ajung numerele noastre, putem scrie următoarele 6 permutări.

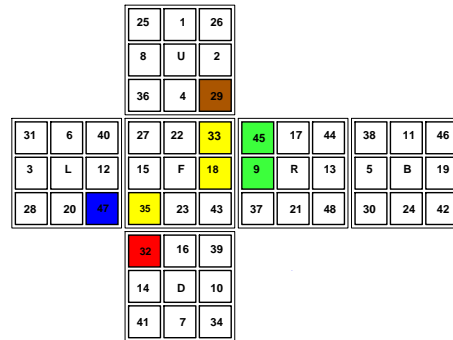


Figura 4: Numerotarea pătratelor
ajung numerele noastre, putem scrie următoarele 6 permutări.

$$F = (4, 9, 16, 12)(15, 22, 18, 23)(27, 33, 43, 35)(29, 37, 32, 40)(36, 45, 39, 47)$$

$$B = (1, 3, 7, 13)(5, 11, 19, 24)(25, 28, 34, 44)(26, 31, 41, 48)(30, 38, 46, 42)$$

$$L = (3, 6, 12, 20)(8, 15, 14, 19)(25, 27, 32, 42)(28, 31, 40, 47)(35, 41, 46, 36)$$

$$R = (2, 5, 10, 18)(9, 17, 13, 21)(26, 30, 39, 33)(29, 38, 34, 43)(37, 45, 44, 48)$$

$$U = (1, 2, 4, 8)(6, 11, 17, 22)(25, 26, 29, 36)(27, 31, 38, 45)(33, 40, 46, 44)$$

$$D = (7, 14, 16, 10)(20, 23, 21, 24)(28, 35, 37, 30)(32, 39, 34, 41)(42, 47, 43, 48)$$

Aici paranteza $(4, 9, 16, 12)$ înseamnă că 4 ajunge în poziția lui 9, 9 în poziția lui 16, 16 în poziția lui 12, iar 12 în poziția lui 4. Așa ceva se numește

un *ciclu*, și fiecare rotație are în componența sa câte 5 astfel de cicluri. Putem scrie $F(4) = 9$, și toate celelalte. Să reprezentăm grafic o astfel de rotație, cu scopul de a ajuta intuiția cititorului care nu dorește să graveze cu marker permanent aceste numere pe cubul său personal, pentru a nu-i strica astfel estetica. În figura următoare este reprezentată rotația F .

Să mai facem după F încă o rotație, să spunem R . Prin rotația F , 4 ajunge în poziția lui 9 (citit în expresia lui F), și prin R își continuă drumul în 17 (deoarece 9 merge în 17, conform formulei lui R). Așadar prin secvența de rotații FR , 4 ajunge în poziția 17. Dacă am folosi notația funcțională pentru a descrie aceste transformări, atunci ordinea corectă a celor două litere ar fi RF , și am putea scrie $RF(4) = R(F(4)) = R(9) = 17$. Așadar, în urma unei secvențe oarecare de rotații (i.e. permutări), numerele migrează pe cub, deci și pe graficul nostru, și astfel orice secvență de transformări va fi descrisă de o permutare, *compunerea* permutărilor corespunzătoare transformărilor respective. Figura următoare reproduce graficul corespunzător secvenței de transformări FR (în ordinea F , apoi R).

Pentru cei care au avut răbdarea să facă această numerotare în mod concret pe propriul lor cub, să menționăm că pentru a citi de pe cub permutarea finală după o secvență mai lungă de rotații, este necesară păstrarea numerotării poziției inițiale pe un cub de rezervă (sau cel puțin pe un grafic similar celui din figura 5). Poziția finală a numerelor (spre exemplu poziția finală a lui 4) se compară cu numerotarea inițială și astfel se identifică numerele în care acestea se transformă (17, întrucât 4 se află acum în poziția lui 17, în urma secvenței de transformări F, R).

Suntem acum pregătiți să rezolvăm cubul... din orice configurație a acestuia, oricât de amestecate ar fi culorile pe el. Începem deci cu amestecarea

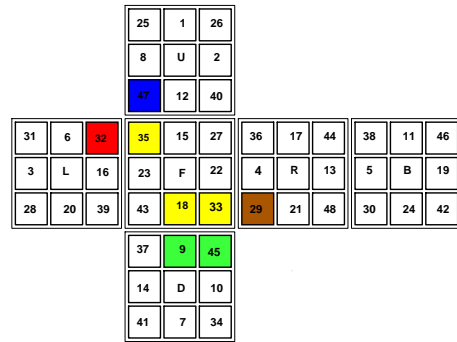


Figura 5: Rotația F

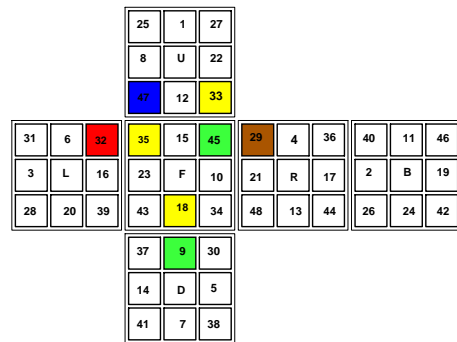


Figura 6: Secvența de rotații FR

Până aici am definit permutările corespunzătoare rotațiilor fețelor, grupul generat de acestea și permutarea poziției amestecate. Calculăm acum o expresie a permutării date cu ajutorul generatorilor grupului.

Secvența pe care am găsit-o (am denumit-o T) este secvența de transformări care aduce cubul din poziția inițială în poziția amestecată P . Calculatorul a reușit să exprime o permutare oarecare în forma unei secvențe de pași elementari, rotații ale fețelor cubului. Iată și verificarea acestui fapt, efectuat tot prin program.

Atât comparația directă, cât și verdictul calculatorului confirmă egalitatea. Concluzia? Da, am obținut rezolvarea cubului! Aceasta este secvența de transformări inversă.

Evident T^{-1} este secvența de transformări care face transformarea inversă, adică aduce cubul din poziția amestecată P în poziția inițială!

Atragem atenția din nou că, nu am folosit în acest calcul nimic specific cubului Rubik. Toate comenzile GAP folosite sunt de utilitate generală și sunt folosite la analiza structurii grupurilor finite comutative și mai ales necomutative.

Grupul de transformări ale cubului Rubik este un grup cu multe elemente, dar deși acest fapt nu înseamnă automat și complexitate ridicată, totuși acest grup are o structură destul de complicată și diversă pentru a putea exemplifica pe el o serie de noțiuni și construcții specifice teoriei grupurilor, cum ar fi: subgrup, subgrup normal, grup factor, centru,

```
gap> Fg:=FreeGroup("F","B","L","R","U","D");;
gap> hom:=GroupHomomorphismByImages(Fg,cube,
    GeneratorsOfGroup(Fg),
    GeneratorsOfGroup(cube));;
gap> T:=PreImagesRepresentative(hom,P);
B*R*D*U^-1*R*F*D*R^2*D^-1*F*B^-1*R^-1*B*L^-1*
U*L*B^-1*U*B*L*R*L*U*R^-1*F*U*L^-1*U^-1*F^2*
L*U*L^-1*U^-1*F*L^-1*U^-1*F^-1*U*F^2*L^-1*
U*L*U^-1*L*F*L^-1*U^-1*F^-1*R^-1*F*R*U*F*
U^-1*F^-1*U*L*F*D*F*D^-1*L^-1*F^-2*U^-1*
R^-1*F^-1*R*U*L*D*F*D^-1*L^-1*U^-1*R^-1*
F^-1*R*F*U*F*L*F*D*F^-1*D^-1*L^-1
```

```
gap> I:=Image(hom,T);
(1,2,16,8,13,19,11,17,23,6,5,3)
(4,15,10,14)
(7,24)
(12,21,20,22)
(26,29,36,44,33,40,38,45,27)
(28,39,48,42,37,34,41,43,30)
(32,35,47)
```

```
gap> P;
(1,2,16,8,13,19,11,17,23,6,5,3)
(4,15,10,14)
(7,24)
(12,21,20,22)
(26,29,36,44,33,40,38,45,27)
(28,39,48,42,37,34,41,43,30)
(32,35,47)
```

```
gap> I=P;
true
```

```
gap> T^-1;
U*R*D*R*L^-1*F^-1*R*F^-1*U^-1*R*U*B*D^-1*U^-2*
D*B^-1*L^-2*F^2*B*F^-1*L*B^-1*D*L^-2*D^-1*F*
L^-1*B*L*B*L*F*L^-1*F*U*B*U^-1*F^-1*B^-1*L*
F^-1*L^-1*B^-1*F^-1*B^2*L^-1*B^-1*L*B^-1*F^-1*
B*F*B^-1*F^-1*B^-1*U*B*U^-1*F*B*F^-3*B*F*B^-1*
F*B*F^-2*B^-1*F^-2*L*F*L^-1*B*F*B^-1*L*B*F^-1*
B^-1*F^-1*L^-1*F^-2*B*L*F*L^-1*F^-1*B^-1*L*R*
F^-1*R^-1*L^-1*B^-1*F*B
```

produs direct, produs semidirect, comutator, acțiune pe o mulțime, p-grup etc. Ca fapt divers și o curiozitate interesantă (legată de algebră, bineînțeles), amintim aici legătura descoperită de Solomon W. Golomb, și extinsă de Anthony E. Durham între cubul Rubik și anumite proprietăți ale particulelor elementare, în special ale quarkurilor.

2. Să prezentăm acum o altă problemă, rezolvarea ecuațiilor algebrice. Este vorba de ecuații polinomiale de o singură variabilă, pentru care căutăm soluții exacte (rădăcini ale polinomului), deci soluțiile numerice aproximative și metodele de a obține acestea nu intră în aria investigațiilor noastre prezente.

Evident rădăcinile unui polinom sunt funcții de coeficienții polinomului. Metodele de rezolvare ale acestor ecuații pot depinde în general și de mulțimea de numere din care se iau coeficienții polinomului. Pentru a ne situa în contextul cel mai simplu presupunem că mulțimea de numere din care alegem coeficienții polinomului este mulțimea numerelor reale, sau mai degrabă cea a numerelor complexe. Acest context este cel mai simplu deoarece în mulțimea numerelor complexe putem efectua neîngrădit nu numai cele (să spunem) patru operații – adunarea, scăderea, înmulțirea și împărțirea – dar și extragerea rădăcinii de orice ordin. În plus, pentru polinoame cu coeficienți complecși este valabilă teorema numită *Teorema fundamentală a algebrei*.

Theorem. *Fie polinomul de grad n*

$$f = a_0 z^n + a_1 z^{n-1} + a_2 z^{n-2} + \dots + a_{n-1} z + a_n,$$

unde a_0, a_1, \dots, a_n sunt numere complexe, $a_0 \neq 0$. Atunci polinomul are n rădăcini complexe (nu neapărat distincte între ele) z_1, z_2, \dots, z_n , cu ajutorul cărora f se descompune în factori de gradul întâi astfel,

$$f = a_0(z - z_1) \cdot (z - z_2) \cdot \dots \cdot (z - z_n).$$

Există numeroase demonstrații diferite ale acestei teoreme. Primele demonstrații, deși dovedite ulterior incomplete, se datorează lui d'Alembert (1746), Euler (1749), de Foncenex (1759), Lagrange (1772), Laplace (1795), Gauss (1799). Șirul demonstrațiilor complete începe cu Argand (1806). Gauss ulterior a mai dat două demonstrații în 1816 și în 1849.

Conform teoremei fundamentale a algebrei așadar, orice ecuație polinomială cu coeficienți complecși este rezolvabilă în mulțimea numerelor complexe și numărul rădăcinilor (socotite cu multiplicitate) este gradul polinomului. Care este atunci problema pe care o investigăm noi?

Problema este exprimarea rădăcinilor, cu ajutorul coeficienților ecuației printr-o formulă de rezolvare. Formula poate să conțină coeficienții polinomului și operațiile amintite, incluzând și extragerea rădăcinii.

Pentru ecuațiile de gradul 1 și 2 avem binecunoscutele formule, valabile pentru orice coeficienți complecși. Ecuațiile de gradul 3 și 4 au și ele formule de rezolvare. Pentru ecuația de gradul 3 Niccolo Fontana Tartaglia și Gerolamo Cardano au găsit formule de rezolvare, iar pentru ecuația de gradul 4 Lodovico Ferrari, formulele acestea fiind publicate în 1545 de către Cardano. Aceste formule sunt destul de complicate – mai ales pentru ecuația de gradul 4 – motiv din care folosirea lor efectivă este evitată pe cât posibil.

Reproducem aici doar formula de rezolvare pentru ecuația de gradul 3. Pentru ca formula să fie cât mai simplă, se împart coeficienții cu coeficientul termenului de rang maxim și se mai face o translație potrivită a variabilei pentru a anula primul coeficient. Formula de mai jos este forma în care pachetul *Mathematica* o afișează (repetăm și formula de rezolvare a ecuației de gradul 2 în acest spirit).

$$x^2 + 2px - q = 0, \quad \begin{aligned} x_1 &= -p - \sqrt{p^2 + q} \\ x_2 &= -p + \sqrt{p^2 + q} \end{aligned}$$

$$x^3 + 3px - 2q = 0, \quad \begin{aligned} x_1 &= -\frac{p}{\sqrt[3]{q + \sqrt{p^3 + q^2}}} + \sqrt[3]{q + \sqrt{p^3 + q^2}} \\ x_2 &= \frac{(1+i\sqrt{3})p}{2\sqrt[3]{q + \sqrt{p^3 + q^2}}} - \frac{1}{2}(1 - i\sqrt{3})\sqrt[3]{q + \sqrt{p^3 + q^2}} \\ x_3 &= \frac{(1-i\sqrt{3})p}{2\sqrt[3]{q + \sqrt{p^3 + q^2}}} - \frac{1}{2}(1 + i\sqrt{3})\sqrt[3]{q + \sqrt{p^3 + q^2}} \end{aligned}$$

Pentru a preveni o întrebare justificată legată de semnificația notației cu simbolul radicalului, comentăm pe scurt aceste formule. Notația cu simbolul radicalului este univocă doar atunci când sub radical se află un număr real pozitiv. În toate celelalte situații notația nu este univocă, mai precis notația se referă la două, trei, sau mai multe numere complexe, în funcție de indicele radicalului. Ei bine, în aceste formule se poate alege oricare din acele numere, dar cel ales trebuie folosit în toate formulele rădăcinilor ecuației date. Dacă facem o altă alegere, valorile furnizate de cele două, respectiv trei formule se pot permuta între ele.

Încercarea găsirii unei formule de rezolvare pentru ecuații de gradul 5 sau mai mare a fost un eșec continuu, până ce Paolo Ruffini în 1799 și Niels Henrik Abel în 1824 au demonstrat că ecuațiile de gradul 5, în general, nu au formule de rezolvare.

Rezolvarea completă a acestei probleme se datorează însă lui Évariste Galois (1811-1832), un tânăr matematician francez. El a găsit condiția ca o ecuație algebrică oarecare să aibă o formulă de rezolvare. Teoria lui Galois

stabilește o legătură între rezolvabilitatea ecuației cu ajutorul unei formule cu radicali și o proprietate a unui grup, numit astăzi *grupul lui Galois* al ecuației. Formula de rezolvare există dacă și numai dacă acest grup are o proprietate specială, anume este *rezolubil* (vom vedea mai târziu ce înseamnă aceasta). Teoria lui Galois a contribuit esențial chiar la cristalizarea conceptului de structură de grup, acesta nefiind conturat încă la vremea respectivă, așa cum îl știm astăzi.

Ce înseamnă toate acestea mai concret? Preferăm să mai rămână un mister încă puțin, în schimb dezvăluim rodul teoriei lui Galois. Iată aici un mic program scris în GAP pentru acest scop.

```
x:=Indeterminate(Rationals,"x");
f:=...;
G:=TransitiveGroup(Degree(f),GaloisType(f));
CompositionSeries(G);
DisplayCompositionSeries(G);
IsSolvable(G);
```

Problemă. Să se stabilească dacă ecuația $x^8 + 4x^7 + 10x^6 + 16x^5 + 19x^4 + 16x^3 + 10x^2 + 4x - 1 = 0$ are sau nu formulă de rezolvare prin radicali.

```
gap> f:=x^8+4*x^7+10*x^6+16*x^5+19*x^4+16*x^3+10*x^2+4*x-1
gap> G:=TransitiveGroup(Degree(f),GaloisType(f));
[2^4]D(4)
gap> CompositionSeries(G);
gap> DisplayCompositionSeries(G);
G (7 gens, size 128)
 | Z(2)
S (6 gens, size 64)
 | Z(2)
S (5 gens, size 32)
 | Z(2)
S (4 gens, size 16)
 | Z(2)
S (3 gens, size 8)
 | Z(2)
S (2 gens, size 4)
 | Z(2)
S (1 gens, size 2)
 | Z(2)
1 (0 gens, size 1)
gap> IsSolvable(G);
true
```

Concluzie: Ecuația $x^8 + 4x^7 + 10x^6 + 16x^5 + 19x^4 + 16x^3 + 10x^2 + 4x - 1 = 0$ este rezolvabilă prin radicali. Formula concretă de rezolvare am obținut-o folosind pachetul *Mathematica*.

$$\begin{aligned} x_{1,2,3,4} &= 1/2(-1 \pm \sqrt{-3 \pm 4i\sqrt[4]{2}}) \\ x_{5,6} &= 1/2(-1 \pm \sqrt{-3 + 4\sqrt[4]{2}}) \\ x_{7,8} &= 1/2(-1 \pm i\sqrt{3 + 4\sqrt[4]{2}}) \end{aligned}$$

Calculul grupului Galois nu este de loc simplu, și nici nu este un capitol încheiat al matematicii: algoritmi efectivi de stabilire a rezolvabilității prin radicali a ecuațiilor există astăzi (2011) doar până la ecuații de gradul 23 (cu 5 ani în urmă era gradul 15).

3. Problemele abordate până acum cu tehnicile algebrei computaționale aveau mai mult sau mai puțin un caracter algebric. Vom aborda acum o problemă cu totul diferită de cele de până acum, și anume o problemă de topologie.

În spiritul acestei introduceri – și anume să începem cu sfârșitul – trecem direct la aplicații ale algebrei computaționale în topologie algebrică.

Din punct de vedere topologic două obiecte geometrice distincte pot fi identice. Măsura identității lor se face cu ajutorul așa-numiților invarianti topologici.

Printre invariantii topologici cei mai simpli se numără grupurile de omologie simplicială.

Ideea de bază este de a descompune obiectul geometric – sau mai bine zis topologic – în cărămizi simple, numite simplexe. Aceste simplexe sunt: punctul (în dimensiunea 0), segmentul (în dimensiunea 1), triunghiul (în dimensiunea 2), tetraedrul (în dimensiunea 3) etc.

Să observăm că aceste obiecte trebuie înțelese în sens topologic, adică nu ca forme geometrice, ci mai degrabă ca imagini prin aplicații bicontinue ale acestora în diverse spații topologice.

Mulțimea simplexelor în care se descompune un spațiu topologic este baza unui modul liber peste un inel de coeficienți, care poate fi ales după necesități: noi ne vom limita la inelul întregilor, sau la inelul claselor de resturi modulo 2, și corpul numerelor raționale.

Să mai observăm că "frontiera" fiecărui simplex de dimensiune n este o colecție de simplexe de dimensiune $n-1$. Sensul precis al frontierei se definește astfel ca frontiera frontierei să fie nulă, în concordanță cu ideea intuitivă – spre exemplu – că frontiera unui disc este cercul care îl limitează, iar frontiera cercului... ei bine nu există, este nulă, deoarece segmentul din care este format cercul se închide la capetele sale, anulându-și propria frontieră.

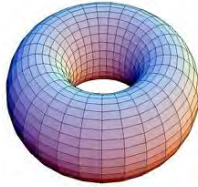
Numim o combinație liniară cu coeficienți din inelul fixat mai înainte *lanț*. Numim un lanț cu frontieră nulă *ciclu*. Numim *frontieră* un ciclu care este frontiera unui lanț.

Pentru un spațiu topologic dat lanțurile de dimensiune $n, n-1, \dots, 1, 0$ din care este format, formează un șir de grupuri abeliene (module peste inelele amintite). Între ele există operatorul de frontieră ∂_k , astfel ca $\partial_{k-1}\partial_k = 0$.

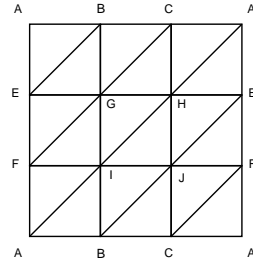
Grupul factor $H_k(K) = \ker \partial_{k-1} / \text{Im } \partial_k$ se numește *grupul de omologie* de dimensiune k al spațiului topologic respectiv.

Faptul crucial legat de acest grup este că el este un invariant topologic: spații topologice homeomorfe au aceleași grupuri de omologie.

Să trecem acum direct la o aplicație. Să calculăm cu ajutorul calculatorului grupurile de omologie a unei suprafețe. Mai întâi se face o descompunere simplicială a suprafeței.



Imagine



O descompunere simplicială

Figura 8: Torul.

Apelăm din nou la *GAP*, mai precis la pachetul auxiliar *simpcomp* care știe să reprezinte lanțuri și știe să calculeze grupuri de omologie.

Programul care calculează grupul de omologie cu coeficienți întregi pentru un complex simplicial arată astfel. Exemplul este descompunerea simplicială a unui tor triplu.

```
K:=SCFromFacets([
[1,2,4],[1,2,6],[1,3,7],[1,3,8],
[1,4,5],[1,5,7],[1,6,9],[1,8,10],
[1,9,10],[2,3,6],[2,3,7],[2,4,8],
[2,5,8],[2,5,9],[2,7,9],[3,4,8],
[3,4,9],[3,5,9],[3,5,10],[3,6,10],
[4,5,6],[4,6,10],[4,9,10],[5,6,8],
[5,7,10],[6,8,9],[7,8,9],[7,8,10]
]);
gap> T23.Homology;
[[ 0, [ ] ], [ 6, [ ] ], [ 1, [ ] ] ]
```

Comentariu:

- Prima instrucțiune definește complexul simplicial K , conform diviziunii pe care o definim în prealabil.
- A doua instrucțiune calculează omologia (redușă) peste \mathbf{Z} . Se citește astfel: $H_0(K) = 0$, $H_1(K) = \mathbf{Z}^6$, $H_2(K) = \mathbf{Z}$. Acestea sunt tocmai grupurile de omologie ale unui tor triplu.

Considerăm că toate acestea sunt suficient de incitante pentru a îndemna cititorul la parcurgerea paginilor ce urmează.