Jakub Howorus, Bartosz Polus, Kinga Konieczna, Jan Czechowski

MAT3 - Zliczanie orbit

4 czerwca 2025

Spis treści

1. Zadanie 1 (Teoretyczne) **2**
Krok 1: Liczenie $|A|$ przez sumowanie po g . . . 2
Krok 2: Liczenie $|A|$ przez sumowanie po x . . . 2
Krok 3: Grupowanie po orbitach . . . 2
Krok 4: Porównanie wyrażeń . . . 3

2. Zadanie 2 (Teoretyczne) **4**

3. Zadanie 3 (Teoretyczne) **5**

4. Zadanie 4 (Praktyczne) **6**
4.1. Implementacja w Mathematica . . . 6
4.2. Wyniki . . . 7

5. Zadanie 5 (Praktyczne) **9**

1. Zadanie 1 (Teoretyczne)

Polecenie: Dla każdego $g \in G$, niech $\operatorname{Fix} g = \{x \in X \mid \varphi_g(x) = x\}$ będzie zbiorem elementów stałych ze względu na permutację φ_g. Pokazać, że jeśli N jest liczbą orbit działania φ, to

$$N = \frac{1}{|G|} \sum_{g \in G} |\operatorname{Fix} g|.$$

Rozwiązanie: W tym zadaniu mamy udowodnić Lemat Burnside'a. Zacznijmy od przedstawienia twierdzenia o orbitach i stabilizatorach, które mówi nam, że dla dowolnego $x \in X$ zachodzi nam taka równość:

$$|\operatorname{Orb}(x)| \cdot |\operatorname{Stab}(x)| = |G|$$

Zdefiniujmy zbiór:

$$A = \{(g, x) \in G \times X \mid \varphi_g(x) = x\},$$

czyli zbiór wszystkich par, w których element $g \in G$ ustala punkt $x \in X$ na niego samego.

Krok 1: Liczenie $|A|$ przez sumowanie po g

Z jednej strony:

$$|A| = \sum_{g \in G} \operatorname{Fix}(g).$$

Krok 2: Liczenie $|A|$ przez sumowanie po x

Z drugiej strony:

$$|A| = \sum_{x \in X} |\operatorname{Stab}(x)|,$$

Z twierdzenia o orbicie-stabilizatorze:

$$|\operatorname{Stab}(x)| = \frac{|G|}{|\operatorname{Orb}(x)|}.$$

Zatem:

$$|A| = \sum_{x \in X} \frac{|G|}{|\operatorname{Orb}(x)|}.$$

Krok 3: Grupowanie po orbitach

Niech $\mathcal{O}_1, \ldots, \mathcal{O}_r$ będą orbitami działania G na X.
Dla każdej orbity $\mathcal{O}_i$, wszystkie punkty $x \in \mathcal{O}_i$ mają tę samą wartość $|\operatorname{Orb}(x)| = |\mathcal{O}_i|$, więc:

$$\sum_{x \in \mathcal{O}_i} \frac{|G|}{|\operatorname{Orb}(x)|} = |\mathcal{O}_i| \cdot \frac{|G|}{|\mathcal{O}_i|} = |G|.$$

Suma po wszystkich orbitach daje:

$$|A| = \sum_{i=1}^{N} |G| = N \cdot |G|.$$

Krok 4: Porównanie wyrażeń

Porównując dwie wyrażone wcześniej postacie $|A|$, mamy:

$$\sum_{g \in G} \mathrm{Fix}(g) = N \cdot |G|,$$

czyli:

$$N = \frac{1}{|G|} \sum_{g \in G} \mathrm{Fix}(g).$$

2. Zadanie 2 (Teoretyczne)

Polecenie: Obliczyć, na ile sposobów można pokolorować wierzchołki sześcianu n kolorami.
Rozwiązanie: Grupa symetrii sześcianu (izometrii zachowujących sześcian) jest znana jako grupa ośmiościenna i ma rząd |G|=24. Składa się z następujących elementów:

— 1 - identyczność,
— 9 obrotów wokół osi przechodzących przez środki przeciwległych ścian (kąty 90°, 180°, 270°),
— 6 obrotów wokół osi przechodzących przez przeciwległe wierzchołki (kąty 120°, 240°),
— 8 obrotów wokół osi przechodzących przez środki przeciwległych krawędzi (kąt 180°).

Liczba orbit, czyli nieidentycznych kolorowań jest równa (korzystamy z zadania 1):

$$N = \frac{1}{|G|} \sum_{g \in G} |Fixg|$$

gdzie Fix_g to zbiór kolorowań niezmienniczych względem działania g.
Teraz policzmy $|Fix_g|$ dla każdego typu elementów grupy:

— **Identyczność** (1 element) - wszystkie kolorowania są niezmiennicze, zatem:

$$|Fixg| = n^8$$

— **Obroty o 90° i 270°** (6 elementów) - aby kolorowanie było niezmiennicze, wszystkie 4 wierzchołki na ścianie muszą mieć ten sam kolor. Zatem są to dwie ściany, każda jednolita:

$$|Fixg| = n^2$$

— **Obroty o 180°** (3 elementy) - wierzchołki muszą być w parach przeciwległych.

$$|Fixg| = n^4$$

— **Obroty o 120° i 240°** (8 elementów) - wierzchołki muszą być w trójkach związanych obrotem wokół przekątnej sześcianu. Mamy 4 niezależne wybory: 2 kolory dla trójek + 2 kolory dla wierzchołków stałych), zatem:

$$|Fixg| = n^4$$

— **Obroty o 180° wokół osi przez środki krawędzi** (6 elementów) - wierzchołki muszą być w czterech parach związanych obrotem.

$$|Fixg| = n^4$$

Sumowanie punktów stałych:

$$\sum_{g \in G} |Fixg| = 1 \cdot n^8 + 6 \cdot n^2 + 3 \cdot n^4 + 8 \cdot n^4 + 6 \cdot n^4$$

$$\sum_{g \in G} |Fixg| = n^8 + 17n^4 + 6n^2$$

Obliczenie liczby orbit:

$$N = \frac{1}{24}(n^8 + 17n^4 + 6n^2)$$

Zatem ostateczna liczba nieidentycznych kolorowań wierzchołków sześcianu przy użyciu n kolorów wynosi:

$$N = \frac{n^8 + 17n^4 + 6n^2}{24}$$

3. Zadanie 3 (Teoretyczne)

Polecenie: Wyznaczyć liczbę obwodów z trzema przełącznikami, nierównoważnych względem permutacji sygnałów wejściowych.

Rozwiązanie: Dwa obwody logiczne są równoważne względem permutacji sygnałów wejściowych, jeśli po zamianie miejscami wejść (przełączników) jeden obwód zachowuje się identycznie jak drugi.

Dla trzech przełączników A, B, C, otrzymujemy $2^3 = 8$ możliwych kombinacji na wejściu obwodu. Obwód z trzema przełącznikami to dowolna funkcja logiczna

$$f : \{0,1\}^3 \to \{0,1\}$$

Ponieważ dla każdej z kombinacji możemy otrzymać na wyjściu 0 lub 1 istnieje liczba obwodów z trzema przełącznikami wynosi

$$|X| = 2^8 = 256$$

Zbiór G jest zbiorem wszystkich permutacji g zbioru trzyelementowego i wynosi

$$|G| = 6$$

Permutacje g działają na zbiorze $X = \{0,1\}^3$ (8-elementowym) i dzielą go na rozłączne orbity. Funkcja f należy do Fix_g wtedy i tylko wtedy, gdy ma taką samą wartość na wszystkich elementach każdej orbity.

Jeśli permutacja g dzieli X na m orbit, to:

$$|\mathrm{Fix}_g| = 2^m.$$

Dla wszystkich $g \in S_3$:

— id — permutacja tożsamościowa: $m = 8$, więc $|\mathrm{Fix}_{\mathrm{id}}| = 2^8 = 256$.
— Trzy transpozycje (np. (AB)): $m = 6$, więc każda ma $|\mathrm{Fix}_g| = 2^6 = 64$.
— Dwa cykle 3-elementowe ((ABC), (ACB)): $m = 4$, więc każda ma $|\mathrm{Fix}_g| = 2^4 = 16$.

Podstawiamy do wzoru z zadania 1:

$$N = \frac{1}{6}(256 + 3 \cdot 64 + 2 \cdot 16) = \frac{1}{6}(256 + 192 + 32) = \frac{480}{6} = 80$$

4. Zadanie 4 (Praktyczne)

Polecenie: Znaleźć wszystkie kolorowania wierzchołków sześcianu dwoma kolorami.

4.1. Implementacja w Mathematica

Poniżej przedstawiono implementację kolorowania wierzchołków sześcianu dwoma kolorami w języku Mathematica:

```
(*1. Generatory obrotów sześcianu o 90 stopni wokół osi Z,Y i X*)
(*Obrót o 90 stopni wokół osi Z:(x,y,z)->(-y,x,z)*)
obrótZ = Cycles[{{1, 2, 3, 4}, {5, 6, 7, 8}}];

(*Obrót o 90 stopni wokół osi Y:(x,y,z)->(z,y,-x)*)
obrótY = Cycles[{{1, 5, 6, 2}, {4, 8, 7, 3}}];

(*Obrót o 90 stopni wokół osi X:(x,y,z)->(x,-z,y)*)
obrótX = Cycles[{{1, 4, 8, 5}, {2, 3, 7, 6}}];

(*Tworzenie grupy generowanej przez obroty*)
grupa = PermutationGroup[{obrótZ, obrótY, obrótX}];
elementyGrupy = GroupElements[grupa];

(*2. Wszystkie kolorowania 8 wierzchołków dwoma barwami*)
wszystkieKolorowania = Tuples[{0, 1}, 8];

(*Słowniki do śledzenia odwiedzonych kolorowań i reprezentantów \
orbity*)
odwiedzoneWszystkie = <||>;
reprezentanci = {};

(*3. BFS wyznaczający reprezentantów orbity dla każdego kolorowania*)
Do[kolorowanie = wszystkieKolorowania[[i]];
  If[! KeyExistsQ[odwiedzoneWszystkie, kolorowanie], orbita = <||>;
   kolejka = {kolorowanie};
   While[kolejka =!= {}, obecne = First[kolejka];
    kolejka = Rest[kolejka];
    If[! KeyExistsQ[orbita, obecne],
     AssociateTo[orbita, obecne -> True];
     Do[nowe = Permute[obecne, InversePermutation[elementyGrupy[[j]]]];
      If[! KeyExistsQ[orbita, nowe], AppendTo[kolejka, nowe]];, {j,
       Length[elementyGrupy]}];];];
   (*Dodajemy znalezione reprezentanty i oznaczamy odwiedzone*)
   AppendTo[reprezentanci, kolorowanie];
   Do[AssociateTo[odwiedzoneWszystkie, c -> True], {c,
     Keys[orbita]}];];, {i, Length[wszystkieKolorowania]}];

(*4. Wyświetlenie liczby reprezentantów i samych kolorowań*)
Print["Liczba nieekwiwalentnych kolorowań: ", Length[reprezentanci]];
Print["Reprezentanci orbity (posortowani według liczby jedynek):"];
posortowaniReprezentanci = SortBy[reprezentanci, Total];
Do[Print[posortowaniReprezentanci[[k]]], {k,
   Length[posortowaniReprezentanci]}];

(*5. Funkcja rysująca sześcian z pokolorowanymi wierzchołkami*)
RysujSzescian[kolorowanie_] :=
  Module[{wierzcholki3D, sciany, krawedzieIndeksy, scianyPolygon,
    krawedzieLines, kolory},(*Współrzędne 3D ośmiu wierzchołków*)
   wierzcholki3D = {{-1, -1, -1}, {1, -1, -1}, {1, 1, -1}, {-1,
      1, -1}, {-1, -1, 1}, {1, -1, 1}, {1, 1, 1}, {-1, 1, 1}};
   (*Ściany sześcianu jako czwórki indeksów wierzchołków*)
```

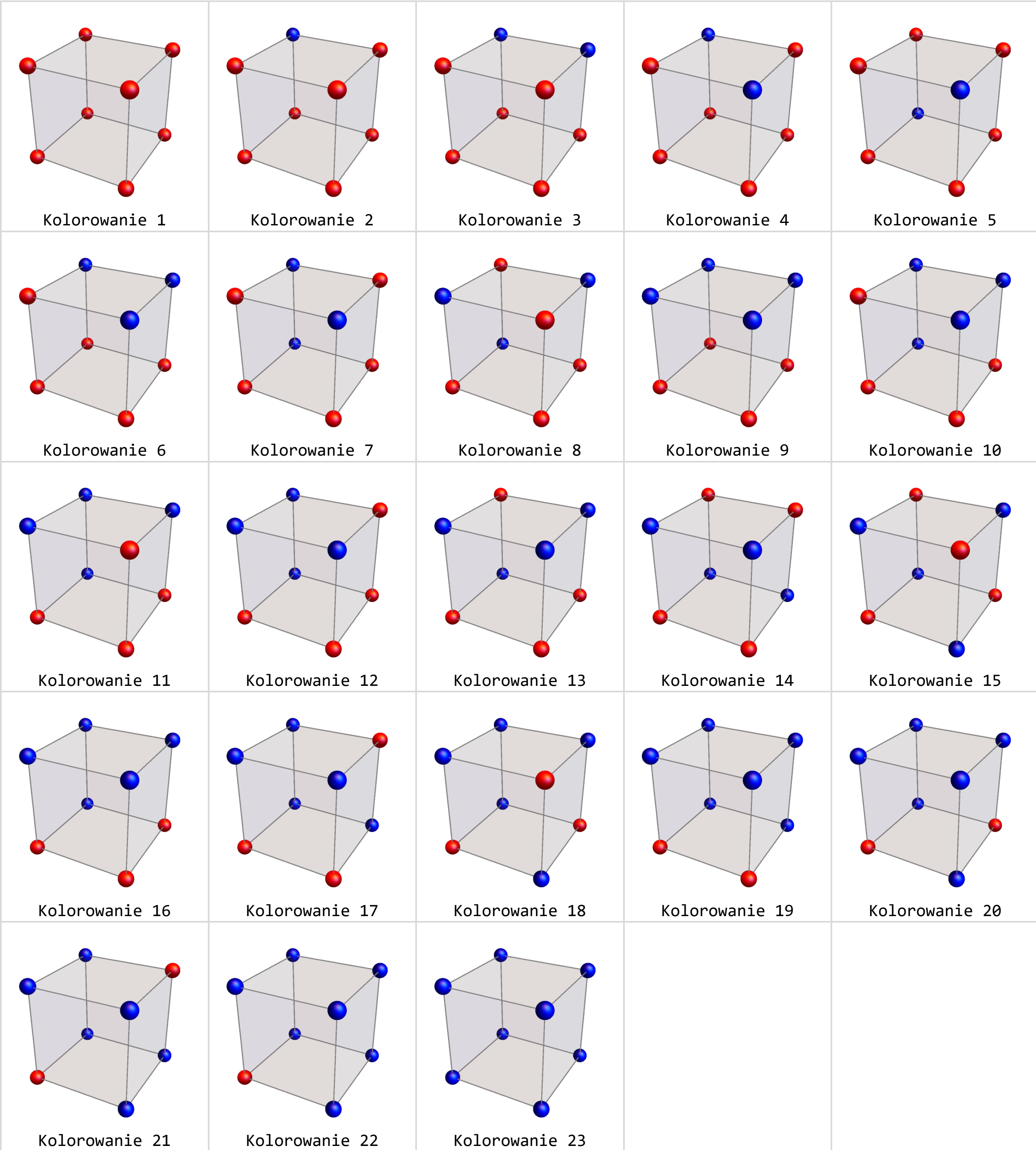
```
    sciany = {{1, 2, 3, 4},(*dolna*){5, 6, 7, 8},(*górna*){1, 2, 6,
       5},(*przód*){2, 3, 7, 6},(*prawa*){3, 4, 8, 7},(*tył*){4, 1, 5,
       8}    (*lewa*)};
    (*Krawędzie jako pary indeksów*)
    krawedzieIndeksy = {{1, 2}, {2, 3}, {3, 4}, {4, 1}, {5, 6}, {6,
       7}, {7, 8}, {8, 5}, {1, 5}, {2, 6}, {3, 7}, {4, 8}};
    scianyPolygon = Polygon /@ (wierzcholki3D[[#]] & /@ sciany);
    krawedzieLines = Line /@ (wierzcholki3D[[#]] & /@ krawedzieIndeksy);
    (*Kolory:0->Red,1->Blue*)
    kolory = kolorowanie /. {0 -> Red, 1 -> Blue};
    Graphics3D[{{Opacity[0.1], Gray, scianyPolygon}, {Gray,
       Thickness[0.005], krawedzieLines},
      MapThread[{#2, Specularity[White, 50],
         Sphere[#1, 0.15]} &, {wierzcholki3D, kolory}]},
     Boxed -> False, Lighting -> "Neutral",
     ViewPoint -> {1.3, -2.4, 1.5}, ImageSize -> 150]];

(*6. Generowanie i wyświetlanie wizualizacji reprezentantów*)
wizualizacje =
  Table[Labeled[RysujSzescian[posortowaniReprezentanci[[i]]],
    Row[{"Kolorowanie ", i}], Bottom], {i,
    Length[posortowaniReprezentanci]}];
siatka = Partition[wizualizacje, UpTo[5]];
Print[Grid[siatka, Frame -> All, FrameStyle -> LightGray]];
```

4.2. Wyniki

Dla powyższej implementacji otrzymano:

- **Liczba nieekwiwalentnych kolorowań:** 23
- **Reprezentanci orbity (posortowani według liczby jedynek):**

{0,0,0,0,0,0,0,0}
{0,0,0,0,0,0,0,1}
{0,0,0,0,0,0,1,1}
{0,0,0,0,0,1,0,1}
{0,0,0,1,0,1,0,0}
{0,0,0,0,0,1,1,1}
{0,0,0,1,0,1,0,1}
{0,0,0,1,1,0,1,0}
{0,0,0,0,1,1,1,1}
{0,0,0,1,0,1,1,1}
{0,0,0,1,1,0,1,1}
{0,0,0,1,1,1,0,1}
{0,0,0,1,1,1,1,0}
{0,0,1,1,1,1,0,0}
{0,1,0,1,1,0,1,0}
{0,0,0,1,1,1,1,1}
{0,0,1,1,1,1,0,1}
{0,1,0,1,1,0,1,1}
{0,0,1,1,1,1,1,1}
{0,1,0,1,1,1,1,1}
{0,1,1,1,1,1,0,1}
{0,1,1,1,1,1,1,1}
{1,1,1,1,1,1,1,1}



Rys. 1: Kolorowanie

5. Zadanie 5 (Praktyczne)

Polecenie: Znaleźć wszystkie, nierównoważne względem permutacji sygnałów wejściowych, obwody z trzema przełącznikami.

Poniżej przedstawiono implementację tego zadania w języku Mathematica:

```
(* 1. Wszystkie funkcje logiczne z 3 wejściami *)
allFuncs = Tuples[{0, 1}, 8]; (* 2^8 = 256 możliwych funkcji *)

(* 2. Permutacje zmiennych wejściowych *)
permutations = Permutations[{1, 2, 3}];

(* 3. Lista wszystkich wejść: kombinacje binarne trzech zmiennych *)
inputs = Tuples[{0, 1}, 3];

(* 4. Zdefiniuj funkcję, która permutuje wejścia i zmienia funkcję logiczną *)
permuteFunc[func_, perm_] := Module[{reorderedInputs, indices},
  reorderedInputs = inputs[[All, perm]]; (* permutujemy zmienne *)
  indices = FromDigits[#, 2] + 1 & /@ reorderedInputs;
  func[[#]] & /@ indices
];

(* 5. Zbiór unikalnych klas równoważności względem permutacji *)
equivalenceClasses = Union[
  Table[
    Sort[
      permuteFunc[f, #] & /@ permutations
    ],
    {f, allFuncs}
  ]
];

(* 6. Wynik końcowy: liczba klas równoważności *)
Length[equivalenceClasses]
```

Wynik to: 80.