

Michał Dobranowski

<https://dobranow.ski>

*Akademia Górniczo-Hutnicza w Krakowie
Wydział Informatyki*

Kraków, 18 listopada 2025

Formalizacja twierdzenia Balinskiego-Younga w Lean 4

Istnieje n partii politycznych, które startują w wyborach do sejmu, który w danym okręgu ma łącznie M miejsc do obsadzenia. Naszym zadaniem jest podział mandatów pomiędzy partie w sposób *proporcjonalny* do liczby głosów na nie oddanych.

Istnieje n partii politycznych, które startują w wyborach do sejmu, który w danym okręgu ma łącznie M miejsc do obsadzenia. Naszym zadaniem jest podział mandatów pomiędzy partie w sposób *proporcjonalny* do liczby głosów na nie oddanych.

Kwotę (czy też iloraz wyborczy) i -tej partii oznaczamy przez q_i i definiujemy jako

$$q_i = M \frac{p_i}{P},$$

gdzie p_i to liczba głosów oddanych na i -tą partię, a P to liczba głosów oddanych w okręgu.

Weźmy pewien okręg wyborczy; niech ma 8 mandatów do sejmu i 10 tysięcy wyborców.

Weźmy pewien okręg wyborczy; niech ma 8 mandatów do sejmu i 10 tysięcy wyborców.

	partia A	partia B	partia C
głosy	1250	3750	5000

Weźmy pewien okręg wyborczy; niech ma 8 mandatów do sejmu i 10 tysięcy wyborców.

	partia A	partia B	partia C
głosy	1250	3750	5000
q_i	1	3	4

Weźmy pewien okręg wyborczy; niech ma 8 mandatów do sejmu i 10 tysięcy wyborców.

	partia A	partia B	partia C
głosy	1250	3750	5000
q_i	1	3	4
mandaty	1	3	4

Weźmy pewien okręg wyborczy; niech ma 8 mandatów do sejmu i 10 tysięcy wyborców.

	partia A	partia B	partia C
głosy	1250	3750	5000
q_i	1	3	4
mandaty	1	3	4

	partia A	partia B	partia C
głosy	1220	3730	5050

Weźmy pewien okręg wyborczy; niech ma 8 mandatów do sejmu i 10 tysięcy wyborców.

	partia A	partia B	partia C
głosy	1250	3750	5000
q_i	1	3	4
mandaty	1	3	4

	partia A	partia B	partia C
głosy	1220	3730	5050
q_i	0.976	2.984	4.04

Weźmy pewien okręg wyborczy; niech ma 8 mandatów do sejmu i 10 tysięcy wyborców.

	partia A	partia B	partia C
głosy	1250	3750	5000
q_i	1	3	4
mandaty	1	3	4

	partia A	partia B	partia C
głosy	1220	3730	5050
q_i	0.976	2.984	4.04
mandaty	1	3	4

Zaokrąglamy kwotę w dół; następnie przydzielamy pozostałe mandaty względem największej reszty.

Zaokrąglamy kwotę w dół; następnie przydzielamy pozostałe mandaty względem największej reszty.

	partia A	partia B	partia C
głosy	1900	2100	6000

Zaokrąglamy kwotę w dół; następnie przydzielamy pozostałe mandaty względem największej reszty.

	partia A	partia B	partia C
głosy	1900	2100	6000
q_i	1.52	1.68	4.80

Zaokrąglamy kwotę w dół; następnie przydzielamy pozostałe mandaty względem największej reszty.

	partia A	partia B	partia C
głosy	1900	2100	6000
q_i	1.52	1.68	4.80
mandaty	1	1	4

Zaokrąglamy kwotę w dół; następnie przydzielamy pozostałe mandaty względem największej reszty.

	partia A	partia B	partia C
głosy	1900	2100	6000
q_i	1.52	1.68	4.80
mandaty	1	1	4 + 1

Zaokrąglamy kwotę w dół; następnie przydzielamy pozostałe mandaty względem największej reszty.

	partia A	partia B	partia C
głosy	1900	2100	6000
q_i	1.52	1.68	4.80
mandaty	1	1 + 1	4 + 1

Zaokrąglamy kwotę w dół; następnie przydzielamy pozostałe mandaty względem największej reszty.

	partia A	partia B	partia C
głosy	1900	2100	6000
q_i	1.52	1.68	4.80
mandaty	1	1 + 1	4 + 1

Dobra wiadomość

Ta metoda *zachowuje kwotę*, tzn. liczba mandatów przydzielonych i -tej partii jest równa $\lfloor q_i \rfloor$ lub $\lceil q_i \rceil$.



	partia A	partia B	partia C
głosy	1850	2600	5550

	partia A	partia B	partia C
głosy	1850	2600	5550
q_i	1.48	2.08	4.44

	partia A	partia B	partia C
głosy	1850	2600	5550
q_i	1.48	2.08	4.44
mandaty	2	2	4

	partia A	partia B	partia C
głosy	1850	2600	5550
q_i	1.48	2.08	4.44
mandaty	2	2	4

Zmieńmy liczby głosów oddanych na partie.

	partia A	partia B	partia C
głosy	1850	2600	5550
q_i	1.48	2.08	4.44
mandaty	2	2	4

Zmieńmy liczby głosów oddanych na partie.

	partia A	partia B	partia C
głosy	1890	2200	5520

	partia A	partia B	partia C
głosy	1850	2600	5550
q_i	1.48	2.08	4.44
mandaty	2	2	4

Zmieńmy liczby głosów oddanych na partie.

	partia A	partia B	partia C
głosy	1890	2200	5520
q_i	1.57	1.83	4.60

	partia A	partia B	partia C
głosy	1850	2600	5550
q_i	1.48	2.08	4.44
mandaty	2	2	4

Zmieńmy liczby głosów oddanych na partie.

	partia A	partia B	partia C
głosy	1890	2200	5520
q_i	1.57	1.83	4.60
mandaty	1	2	5

	partia A	partia B	partia C
głosy	1850	2600	5550
q_i	1.48	2.08	4.44
mandaty	2	2	4

Zmieńmy liczby głosów oddanych na partie.

	partia A	partia B	partia C
głosy	1890	2200	5520
q_i	1.57	1.83	4.60
mandaty	1	2	5

Definicja (Paradoks populacji)

Sytuacja, w której partia tracąca głosy otrzymuje mandat kosztem partii zyskującej głosy.

Określamy pewną liczbę głosów $D \in \mathbb{R}^+$, za którą partii przysługuje mandat. Wtedy każda partia dostanie $\lfloor p_i/D \rfloor$ mandatów.

Określamy pewną liczbę głosów $D \in \mathbb{R}^+$, za którą partii przysługuje mandat. Wtedy każda partia dostanie $\lfloor p_i/D \rfloor$ mandatów.

Przekształćmy:

i -ta partia dostaje k mandatów

$$\Leftrightarrow \left\lfloor \frac{p_i}{D} \right\rfloor \geq k$$

$$\Leftrightarrow \frac{p_i}{D} \geq k$$

$$\Leftrightarrow D \leq \frac{p_i}{k}$$

	partia A	partia B	partia C
głosy	2300	2800	4900
$k = 1$	2300	2800	4900
$k = 2$	1150	1400	2450
$k = 3$	767	933	1633
$k = 4$	575	700	1225
$k = 5$	460	560	980

	partia A	partia B	partia C
głosy	2300	2800	4900
$k = 1$	2300	2800	4900
$k = 2$	1150	1400	2450
$k = 3$	767	933	1633
$k = 4$	575	700	1225
$k = 5$	460	560	980

	partia A	partia B	partia C
głosy	2300	2800	4900
$k = 1$	2300	2800	4900
$k = 2$	1150	1400	2450
$k = 3$	767	933	1633
$k = 4$	575	700	1225
$k = 5$	460	560	980

	partia A	partia B	partia C
głosy	2300	2800	4900
$k = 1$	2300	2800	4900
$k = 2$	1150	1400	2450
$k = 3$	767	933	1633
$k = 4$	575	700	1225
$k = 5$	460	560	980

	partia A	partia B	partia C
głosy	2300	2800	4900
$k = 1$	2300	2800	4900
$k = 2$	1150	1400	2450
$k = 3$	767	933	1633
$k = 4$	575	700	1225
$k = 5$	460	560	980

	partia A	partia B	partia C
głosy	2300	2800	4900
$k = 1$	2300	2800	4900
$k = 2$	1150	1400	2450
$k = 3$	767	933	1633
$k = 4$	575	700	1225
$k = 5$	460	560	980

	partia A	partia B	partia C
głosy	2300	2800	4900
$k = 1$	2300	2800	4900
$k = 2$	1150	1400	2450
$k = 3$	767	933	1633
$k = 4$	575	700	1225
$k = 5$	460	560	980

	partia A	partia B	partia C
głosy	2300	2800	4900
$k = 1$	2300	2800	4900
$k = 2$	1150	1400	2450
$k = 3$	767	933	1633
$k = 4$	575	700	1225
$k = 5$	460	560	980

	partia A	partia B	partia C
głosy	2300	2800	4900
$k = 1$	2300	2800	4900
$k = 2$	1150	1400	2450
$k = 3$	767	933	1633
$k = 4$	575	700	1225
$k = 5$	460	560	980

	partia A	partia B	partia C
głosy	2300	2800	4900
$k = 1$	2300	2800	4900
$k = 2$	1150	1400	2450
$k = 3$	767	933	1633
$k = 4$	575	700	1225
$k = 5$	460	560	980
mandaty	2	2	4

	partia A	partia B	partia C
głosy	2300	2800	4900
$k = 1$	2300	2800	4900
$k = 2$	1150	1400	2450
$k = 3$	767	933	1633
$k = 4$	575	700	1225
$k = 5$	460	560	980
mandaty	2	2	4

Dobra wiadomość

Ta metoda (zwana metodą Jeffersona lub D'Hondta) jest *monotoniczna względem populacji*, to znaczy, że nie występuje tutaj paradoks populacji.

	partia A	partia B	partia C
głosy	2300	2800	4900
$k = 1$	2300	2800	4900
$k = 2$	1150	1400	2450
$k = 3$	767	933	1633
$k = 4$	575	700	1225
$k = 5$	460	560	980
mandaty	2	2	4

Dobra wiadomość

Ta metoda (zwana metodą Jeffersona lub D'Hondta) jest *monotoniczna względem populacji*, to znaczy, że nie występuje tutaj paradoks populacji.

Zła wiadomość

Ta metoda nie zachowuje (górnej) kwoty.

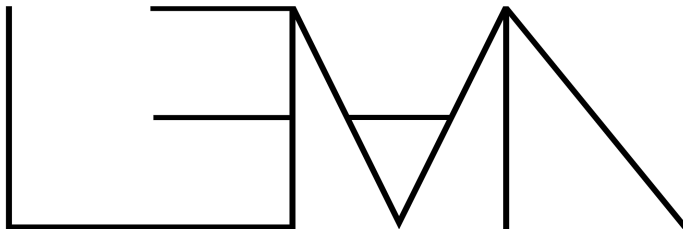
	partia A	partia B	partia C
głosy	1000	2760	6240

	partia A	partia B	partia C
głosy	1000	2760	6240
$k = 1$	1000	2760	6240
$k = 2$	500	1380	3120
$k = 3$	333	920	2080
$k = 4$	250	690	1560
$k = 5$	200	552	1248
$k = 6$	167	460	1040
mandaty	0	2	6

	partia A	partia B	partia C
głosy	1000	2760	6240
$k = 1$	1000	2760	6240
$k = 2$	500	1380	3120
$k = 3$	333	920	2080
$k = 4$	250	690	1560
$k = 5$	200	552	1248
$k = 6$	167	460	1040
mandaty	0	2	6
q_i	0.80	2.21	4.99

Twierdzenie (Balinski-Young)

Nie istnieje (anonimowa) reguła podziału mandatów, która jednocześnie zachowuje kwotę i jest monotoniczna względem populacji.



Programming Language and Theorem Prover

$$\frac{}{\Gamma, A \vdash A} (\text{Ax}) \quad \frac{\Gamma \vdash \perp}{\Gamma \vdash A} (\perp \text{E})$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} (\wedge \text{I}) \quad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} (\wedge \text{E1}) \quad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B} (\wedge \text{E2})$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} (\vee \text{I1}) \quad \frac{\Gamma \vdash A}{\Gamma \vdash B \vee A} (\vee \text{I2}) \quad \frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C} (\vee \text{E})$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} (\rightarrow \text{I}) \quad \frac{\Gamma \vdash A \rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B} (\rightarrow \text{E})$$

Rysunek: Reguły wnioskowania w intuicjonistycznym rachunku zdań (IRZ).

```
structure Party where
  name : String
  deriving DecidableEq
```

```
variable [Fintype Party]
```

```
structure Election where
  parties : Finset Party
  votes : Party → ℕ
  house_size : ℕ
```

```
abbrev Apportionment : Type := Party → ℕ
```

structure Rule **where**

res : Election → Finset Apportionment

non_emptiness (election : Election) : (res election).Nonempty

inheritance_of_zeros (election : Election) (p : Party) :

election.votes p = 0 → ∀ App ∈ res election, App p = 0

house_size_feasibility (election : Election) :

∀ App ∈ res election,

∑ p ∈ election.parties, App p = election.house_size

Reguła jest anonimowa, jeśli permutacja głosów partii powoduje permutację przydziału mandatów w ten sam sposób. Innymi słowy, nazwy partii nie mają znaczenia.

```
class IsAnonymous (rule : Rule) : Prop where
  anonymous (election : Election) ( $\sigma$  : Party  $\rightarrow$  Party) (p q : Party) :
    ( $\sigma$  p =  $\sigma$  q  $\rightarrow$  p = q)  $\rightarrow$ 
      let election' : Election := { parties := election.parties,
                                   votes := election.votes  $\circ$   $\sigma$ ,
                                   house_size := election.house_size
                                   }
      rule.res election' = (rule.res election).image ( $\cdot$   $\circ$   $\sigma$ )
```

Reguła jest konkordantna, jeśli partia z mniejszą liczbą głosów nie dostaje więcej mandatów niż partia z większą liczbą głosów.

```
class IsConcordant (rule : Rule) : Prop where
  concordant (election : Election) (p q : Party) :
    election.votes p < election.votes q →
      ∀ App ∈ rule.res election,
        App p ≤ App q
```

```
class IsQuotaRule (rule : Rule) : Prop where
  quota_rule (election : Election) (p : Party) :
    let quota := (election.votes p * election.house_size : ℚ)
                / (election.total_voters : ℚ)
  ∀ App ∈ rule.res election,
    App p = [quota] ∨ App p = [quota]
```

```
class IsPopulationMonotone (rule : Rule) : Prop where
  population_monotonone (e1 e2 : Election) (p q : Party) :
    e1.parties = e2.parties ∧ e1.house_size = e2.house_size →
      -- p' support grows faster than q's
      e2.votes p * e1.votes q > e2.votes q * e1.votes p →
        ∀ App1 ∈ rule.res e1, ∀ App2 ∈ rule.res e2,
          -- p gets less seats, q gets more seats
          ¬(App1 p > App2 p ∧ App1 q < App2 q)
```

Lemat

Jeśli anonimowa reguła podziału mandatów jest monotoniczna względem populacji, to jest konkordantna.

Lemat

Jeśli anonimowa reguła podziału mandatów jest monotoniczna względem populacji, to jest konkordantna.

```
lemma if_IsPopulationMonotone_then_IsConcordant (rule : Rule)
  [h_anon : IsAnonymous rule] [h_mono : IsPopulationMonotone rule] :
  IsConcordant rule := by
  sorry
```

Dowód.

	partia A	partia B	partia C	partia D
głosy	660	670	2450	6220
q_i	0.53	0.54	1.96	4.98

Dowód.

	partia A	partia B	partia C	partia D
głosy	660	670	2450	6220
q_i	0.53	0.54	1.96	4.98

Przy zachowaniu (górnej) kwoty, mamy $m_C \leq 2$ oraz $m_D \leq 5$. W takim razie $m_B = 1$ (korzystamy z lematu).

Dowód.

	partia A	partia B	partia C	partia D
głosy	660	670	2450	6220
q_i	0.53	0.54	1.96	4.98

Przy zachowaniu (górnej) kwoty, mamy $m_C \leq 2$ oraz $m_D \leq 5$. W takim razie $m_B = 1$ (korzystamy z lematu).

	partia A	partia B	partia C	partia D
głosy	680	675	700	6200
q_i	0.66	0.65	0.68	6.01

Dowód twierdzenia o niemożności

Dowód.

	partia A	partia B	partia C	partia D
głosy	660	670	2450	6220
q_i	0.53	0.54	1.96	4.98

Przy zachowaniu (górnjej) kwoty, mamy $m_C \leq 2$ oraz $m_D \leq 5$. W takim razie $m_B = 1$ (korzystamy z lematu).

	partia A	partia B	partia C	partia D
głosy	680	675	700	6200
q_i	0.66	0.65	0.68	6.01

Przy zachowaniu (dolnej) kwoty, mamy $m_D \geq 6$, więc $m_A + m_B + m_C \leq 2$, więc $m_B = 0$ (dwukrotnie korzystamy z lematu).



```
theorem balinski_young (rule : Rule)
  [IsAnonymous rule] [h_quota : IsQuotaRule rule] :
  ¬IsPopulationMonotone rule := by
  by_contra h_mono
have h_concord := if_IsPopulationMonotone_then_IsConcordant rule
sorry
```

Lean – dowód twierdzenia o niemożności

```

theorem balinski_young (rule : Rule) [IsAnonymous rule] [h_quota : IsQuotaRule rule] :
  ~IsPopulationMonotone rule := by
  by_contra h_mono
  have h_concord := if_IsPopulationMonotone_then_IsConcordant rule

  let e : Election := {
    parties := {⟨"A"⟩, ⟨"B"⟩, ⟨"C"⟩, ⟨"D"⟩}
    votes := fun
      | ⟨"A"⟩ => 660
      | ⟨"B"⟩ => 670
      | ⟨"C"⟩ => 2450
      | ⟨"D"⟩ => 6220
      | _ => 0
    house_size := 8
  }

  obtain ⟨App, h_App⟩ := rule.non_emptiness e

  have m_c_le_2 : App ⟨"C"⟩ ≤ 2 := by
    have h_c := h_quota.quota_rule e ⟨"C"⟩
    simp [e] at h_c
    norm_num at h_c
    grind

  have m_d_le_5 : App ⟨"D"⟩ ≤ 5 := by
    have h_d := h_quota.quota_rule e ⟨"D"⟩
    simp [e] at h_d
    norm_num at h_d
    grind

  have m_b_eq_1 : App ⟨"B"⟩ = 1 := by
    have h_b := h_quota.quota_rule e ⟨"B"⟩ App h_App
    simp [e] at h_b
    norm_num at h_b
    rcases h_b with (m_b_eq_0 | m_b_eq_1)
    · have m_a_eq_1 : App ⟨"A"⟩ = 0 := by
        have m_a_le_m_b := h_concord.concordant e ⟨"A"⟩ ⟨"B"⟩ (by decide) App h_App
        linarith
      have :  $\sum p \in e.\text{parties}, \text{App } p \leq 7 :=$  by
        simp [e]
        linarith [m_a_eq_1, m_b_eq_0, m_c_le_2, m_d_le_5]
      have :  $\sum p \in e.\text{parties}, \text{App } p = 8 :=$  by
        exact rule.house_size_feasibility e App h_App
        linarith
    · assumption

  have m_d_ge_6' : App' ⟨"D"⟩ ≥ 6 := by
    have h_d' := h_quota.quota_rule e' ⟨"D"⟩
    simp [e'] at h_d'
    norm_num at h_d'
    grind

  have m_b_eq_0' : App' ⟨"B"⟩ = 0 := by
    have h_b' := h_quota.quota_rule e' ⟨"B"⟩ App' h_App'
    simp [e'] at h_b'
    norm_num at h_b'
    rcases h_b' with (m_b_eq_0' | m_b_eq_1')
    · assumption
    · have m_a_ge_1' : App' ⟨"A"⟩ ≥ 1 := by
        have m_b_ge_m_a' := h_concord.concordant e' ⟨"B"⟩ ⟨"A"⟩ (by decide) App' h_App'
        linarith
      have m_c_ge_1' : App' ⟨"C"⟩ ≥ 1 := by
        have m_c_ge_m_b' := h_concord.concordant e' ⟨"B"⟩ ⟨"C"⟩ (by decide) App' h_App'
        linarith
      have :  $\sum p \in e'.\text{parties}, \text{App}' p \geq 9 :=$  by
        simp [e']
        linarith [m_a_ge_1', m_b_eq_1', m_c_ge_1', m_d_ge_6']
      have :  $\sum p \in e'.\text{parties}, \text{App}' p = 8 :=$  by
        exact rule.house_size_feasibility e' App' h_App'
        linarith

  replace h_mono := h_mono.population_monotone e e' ⟨"B"⟩ ⟨"D"⟩ (by trivial) (by decide)
  App h_App App' h_App'
  grind
  
```

```

let e' : Election := {
  parties := {⟨"A"⟩, ⟨"B"⟩, ⟨"C"⟩, ⟨"D"⟩}
  votes := fun
    | ⟨"A"⟩ => 680
    | ⟨"B"⟩ => 675
    | ⟨"C"⟩ => 700
    | ⟨"D"⟩ => 6200
    | _ => 0
  house_size := 8
}

obtain ⟨App', h_App'⟩ := rule.non_emptiness e'

have m_d_ge_6' : App' ⟨"D"⟩ ≥ 6 := by
  have h_d' := h_quota.quota_rule e' ⟨"D"⟩
  simp [e'] at h_d'
  norm_num at h_d'
  grind

have m_b_eq_0' : App' ⟨"B"⟩ = 0 := by
  have h_b' := h_quota.quota_rule e' ⟨"B"⟩ App' h_App'
  simp [e'] at h_b'
  norm_num at h_b'
  rcases h_b' with (m_b_eq_0' | m_b_eq_1')
  · assumption
  · have m_a_ge_1' : App' ⟨"A"⟩ ≥ 1 := by
      have m_b_ge_m_a' := h_concord.concordant e' ⟨"B"⟩ ⟨"A"⟩ (by decide) App' h_App'
      linarith
    have m_c_ge_1' : App' ⟨"C"⟩ ≥ 1 := by
      have m_c_ge_m_b' := h_concord.concordant e' ⟨"B"⟩ ⟨"C"⟩ (by decide) App' h_App'
      linarith
    have :  $\sum p \in e'.\text{parties}, \text{App}' p \geq 9 :=$  by
      simp [e']
      linarith [m_a_ge_1', m_b_eq_1', m_c_ge_1', m_d_ge_6']
    have :  $\sum p \in e'.\text{parties}, \text{App}' p = 8 :=$  by
      exact rule.house_size_feasibility e' App' h_App'
      linarith

replace h_mono := h_mono.population_monotone e e' ⟨"B"⟩ ⟨"D"⟩ (by trivial) (by decide)
App h_App App' h_App'
grind
  
```

- M. J. Balinski, H. P. Young. **Fair Representation: Meeting the Ideal of One Man One Vote.** *Yale University Press* (1982)
- Dominik Peters. **COMSOC Lectures 2025.** <https://dominik-peters.de/>
- Leonardo de Moura, Sebastian Ullrich. **The Lean 4 Theorem Prover and Programming Language.** *Springer-Verlag* (2021)
- Brandon Rorie. **Apportionment Calculator.**
<https://btror.github.io/apportionment.github.io/>

`https://github.com/mdbrnowski/apportionmentlib`